

# The NeceMoon Album

Technologies and Strategies to Keep Moving Forward

By  
**Necemon Yai**

## Featuring

Ahou The African Chick  
Jean-Patrick Ehouman  
Jean Luc Houédanou  
Nnenna Nwakanma  
Vanessa Lecosson  
Olivier Madiba  
Cyriac Gbogou  
Antoine Mian  
Israël Yoroba  
Darren Mart  
Karen Kakou  
Monty Oum  
Nanda Seye  
Yehni Djidji  
Edith Brou  
Holty Sow



First Edition

[TheAlbum.NeceMoon.com](http://TheAlbum.NeceMoon.com)

Evasium ®

# The NeceMoon Album

Technologies and Strategies to Keep Moving Forward

By Necemon Yai

First edition

Published by Evasium ®

April 2018

London, UK

[TheAlbum.NeceMoon.com](http://TheAlbum.NeceMoon.com)

The contents of this file are protected under the **UK Copyright, Designs and Patents Act 1988**.

## License

This file is free to distribute and give away to as many people as you would like.

I only ask that you do not sell it or publish the content onto any website.

If you use any quotes from this document, give me credit and link to the original file.

If you write a book and I quote you, I will give you credit and links too.

© Necemon Yai

[necemon@gmail.com](mailto:necemon@gmail.com)

[www.necemonyai.com](http://www.necemonyai.com)

All Rights Reserved.

Version 1.0.7.235

*To the ones I lost, to the ones I got back*

*For each one that gets lost, ten get laid back*

## Table of Contents

Introduction.....	8
For All Practical Purposes .....	11
<b>Volume 1: Moon Light (Softcore) .....</b>	<b>12</b>
<b>Chapter 1: Strategy and Tactics .....</b>	<b>13</b>
Words to Live By (by Monty Oum) .....	14
Monty Oum or the Bruce Lee of Digital Arts .....	15
The 5 Most Terrific Motivation Tips (From Monty Oum The Artistic Genius).....	20
Losing Your Arm (by Jean-Patrick Ehouman).....	22
On reading books and self-education: My Essentials.....	24
On reading books and self-education, My Essentials (Part 2).....	27
Becoming Your Own Boss: 7 Lessons Learned On The Spot (by Yehni Djidji) .....	35
[Interview] In The Bubble of Yehni Djidji: "I offer educational entertainment and life-changing experiences to children" .....	37
What will you do in life? .....	39
11 Personality Traits Of Successful Entrepreneurs (by Jean-Patrick Ehouman).....	40
5 Team Management Insights From Sebastian Marshall .....	43
5 Attitudes Without Which You Will Have Trouble Reaching Your Goals (by Jean-Patrick Ehouman) .....	44
Just think about it: 50 ideas to live by .....	46
Breaking Through Pebble : from Mediocrity to Excellence (by Karen Kakou) .....	51
[Interview] In The Bubble Of Karen Kakou: "my parents never knew their ways to successful careers, I had to figure everything out by myself" .....	55
10 Business And Management Skills To Own And Be Productive (by Vanessa Lecosson) .....	57
<b>Chapter 2: Digital Marketing and Web Visualisation.....</b>	<b>60</b>
A Quick Guide For New Bloggers.....	61
What (un)motivates us When We Run a Blog (by Jean-Patrick Ehouman).....	63
10 Marketing Tips For Promoting Your Next Blog Post (by Vanessa Lecosson) .....	64
Facebook Profile vs Facebook Page: What To Choose? (by Israel Yoroba).....	67
It's All About The Selfie (by Nanda Seye) .....	69
Confidential: Social Networks And Me (by Yehni Djidji) .....	73
You Have The Right To Publish Whatever You Want Online (by Jean Luc Houedanou) .....	75
Caution! Cultivate Your Digital Identity Now! (by Edith Brou).....	77
Job On The Web : How Not To Get Hired (by Israel Yoroba).....	79
[Interview] In The Bubble Of Israel Yoroba: "I am president, elite serial entrepreneur, veteran blog trotter, catalyst for change in our generation" .....	81
Did You Just Say Crowdfunding? (by Nanda Seye) .....	84

[Interview] In The Bubble Of Nanda Seye: "it was essential to be very strategic and organised in order to reach the defined objectives" .....	86
Exploring Some Advanced Visualisation Techniques .....	89
Data Visualisation : Thematic Maps VS Bubble Charts.....	98
A Practical Guide On Open Data From The World Bank (by Cyriac Gbogou).....	101
Educational Uses Of Internet Content Curation (by Antoine Mian).....	103
Marketing For Programmers (or what I learned from my marketing classes).....	104
Web Citizen And Media Coverage: Yes, But... (by Cyriac Gbogou) .....	111
<b>Chapter 3: Corporate Worlds and Emerging Markets.....</b>	<b>113</b>
Starting a New Job: 12 Tips for a Brilliant Outset.....	114
Your First Job is Like Your First Girlfriend.....	117
5 'Good Reasons' For Which You Will Not Be Hired (by Nnenna Nwakanma).....	121
GE FAQ : 3 months in the Cyber Pearl.....	123
5 lessons I learned from my internship at General Electric.....	125
12 Years Of A Dream Challenging Reality (by Olivier Madiba).....	126
[Interview] In The Bubble Of Olivier Madiba: "my goal is to reach my maximum human potential and to help others to do the same" .....	134
The Paradox Of Teachers In Africa : Rare And Yet Poorly Paid (by Antoine Mian) .....	136
Gamification: 5 Peculiar Concepts That African Schools Could Actually Use.....	137
Africa: ICT in Education for Entrepreneurship and Employability (by Antoine Mian).....	142
4 important revelations that could certainly accelerate the educational revolution in Africa.....	144
Africa, Cradle Of Humanity And Entrepreneurs (by Antoine Mian) .....	148
[Interview] In The Bubble Of Antoine Mian: "if one is not passionate, they miss out on life" .....	150
HELP! They Are Going To Steal My Project!!! (by Nnenna Nwakanma).....	152
<b>Chapter 4: Quick Wins, Tricks and Tips.....</b>	<b>153</b>
Where to Find Free Quality Audio Books? .....	154
Convert Video Files With VLC (by Jean Luc Houedanou) .....	157
[Interview] In The Bubble Of Jean Luc Houedanou: "a fleet of drones that prevent theft and abuse".....	160
FFmpeg, The Fastest Way To Edit a Video File (in 5 seconds or less) .....	162
Getting started with Metro Studio (or how to benefit from the largest collection of Metro-style icons for free) .....	164
Instant Messaging For Your Website : Have it the Giant Way .....	168
<b>Chapter 5: Transition - Extra Thoughts and Sharp Fantasy.....</b>	<b>169</b>
The Effects of Accelerated Evolution and Information Overloading: did you know? .....	170
Eponymous Laws For Tech Strategists .....	171
Success, A Reward To Enjoy In The Shadows (by Ahou The African Chick).....	173
[Interview] In The Bubble Of Ahou The African Chick: "I like to go beyond my limits, I like to be curious and try new things" .....	175

The General Assumption of Security .....	178
World Wide Worries .....	179
The Golf Ball Effect .....	180
Just 2 Things .....	181
Games for Everyone .....	182
<b>Volume 2: Full Moon (Hardcore) .....</b>	<b>183</b>
<b>Chapter 6: Software Development and Engineering .....</b>	<b>184</b>
8 reasons why you would enjoy being a programmer .....	185
Writing and Programming: Pretty Much The Same Thing (by Jean-Patrick Ehouman) .....	187
On Technical Orientation : 5 Basic Considerations When Starting To Code .....	188
4 Substantial Tactics Guaranteed To Boost Your Game Creation Skills .....	190
10 Years of Programming .....	192
Should you build it if nobody comes? (by Darren Mart) .....	196
Useful Job Search Websites For Programmers In The UK .....	200
Microsoft Professional Certifications for Developers .....	203
<b>Chapter 7: C# .NET Programming .....</b>	<b>205</b>
Why I like C# so much.....	206
C# Free Learning Resources For Beginners And Professionals.....	209
How to Track Mysterious Bugs with Visual Studio .....	211
How to Debug a Windows Service in Visual Studio (by Holty Sow) .....	213
My Essentials : Top 12 Tech Courses on Pluralsight for .NET Developers.....	216
Deploying a Web App to IIS / Windows Server In 7 Clever Steps .....	220
Launching Your Application At Windows Startup Without Hacking The Registry (by Holty Sow) .....	222
My 12 Favorite Entity Framework Tricks.....	224
ASP.NET MVC: Avoid Polluting The Model Of The View With Error Messages (by Holty Sow) .....	228
Limiting The Execution Of An Action To AJAX Requests Only (by Holty Sow).....	230
[Interview] In The Bubble Of Holty Sow: "for more efficiency, I prefer to work in an agile team" .....	232
The Xamarin Revolution .....	234
Top 8 Tricks with Xamarin Programming .....	236
<b>Chapter 8: Epic Prototypes, Classic Projects, Historic Genre .....</b>	<b>238</b>
Inside Bavardica - Part 1 : Discussion of the subject areas .....	239
Inside Bavardica - Part 2 : Progress and Achievement of the Project .....	242
Inside Bavardica - Part 3 : Design of the 2D characters.....	248
Inside Bavardica - Part 4 : Scope for future improvements .....	252
10 things I learned from building Bavardica.....	254
Babi Fraya: The Hot Winter Illusion.....	256
Relaunch : Kpakpatoya 2.0 as an Interactive Platform.....	259

Flux Project : Social News Web Applications.....	265
Stay up to date with your favorite topic on social platforms.....	267
Furtivue (Or How To Send Furtive Messages Like A Ninja) .....	269
<b>Chapter 9: Research and Case Studies.....</b>	<b>270</b>
Phidgets: First Steps In Robotics? .....	271
Paper Prototyping.....	278
Dual Shock 3 - Part 1: History of game controllers.....	280
Dual Shock 3 - Part 2 : The amazing story of the Dual Shock .....	284
Dual Shock 3 - Part 3: Device Analysis and Specification .....	286
Business at the speed of game production: OpenGL or XNA? .....	289
GOOSE (GOOgle Supply search Engine) .....	290
Final Reminder.....	294
Conclusion .....	295
Share The Album with Your Mates.....	296



## Introduction

### **About The NeceMoon Album: what is this all about?**

[The NeceMoon™](#) is a Blog about Technology and Strategy. [The Album](#) is The Best-Of, a compilation of the most popular articles on The NeceMoon™.

The main objective of The NeceMoon™ is to share tips and insights on a sensible range of topics, in order to let others learn from my mistakes and successes, and hopefully to make things easier for the next person. The main objective of The Album is to promote an optimal access to that content. The Blog format does not always do justice to techno-strategic content. Originally, Blogs were designed in a journalistic spirit and are more suitable to chronological events and (more or less trivial) discussions around daily news. Even if the usefulness and the importance of an analysis persist in time, it becomes almost impossible to find and difficult to consult, as more articles keep stacking.

That is why the best articles have been cherry-picked, reviewed and arranged in a logical order that better matches the layout of a book. The Album is free of charge.

The NeceMoon™ can be accessed from [NeceMoon.com](http://NeceMoon.com) (or [necemonyai.com/blog](http://necemonyai.com/blog))

The NeceMoon™ Album can be downloaded in various file formats, in full from [TheAlbum.NeceMoon.com](http://TheAlbum.NeceMoon.com) (or [necemonyai.com/Blog/page/The-Album.aspx](http://necemonyai.com/Blog/page/The-Album.aspx)). The available formats are PDF, EPUB, MOBI/AZW3/KF8 (Amazon Kindle) and MP3. Furthermore, the various chapters and volumes can be downloaded independently/separately, according to your interests.

### **About the Author: who is Necemon Yai?**



I am a Software Development Engineer extensively involved in Microsoft .NET technologies. Full time developer. Part time digital artist, strategist, essayist and entrepreneur. I majored in computer science from NIIT, Christ University and Swansea University (Master of Engineering, Computing).

At the time of publishing this, I have worked for a Europe leading E-Commerce company, a major UK financial group, the General Electric global corporation and a few tech start-ups that you probably never heard of.

Over the past decade or so, I have been running The NeceMoon blog, where I describe my experiences, my observations and my reflexions. I mostly talk about Technology and Strategy. Here I share my most popular articles.



### About the Contributors: who is in your War Council?

I invited the best writers in my network to include some contributions in this book, especially some of their most relevant insights in terms of Technology and Strategy. These top authors are, Ahou The African Chick, Antoine Mian, Cyriac Gbogou, Darren Mart, Edith Brou, Holty Sow, Israel Yoroba, Jean Luc Houedanou, Jean-Patrick Ehouman, Karen Kakou, Monty Oum, Nanda Seye, Nnenna Nwakanma, Olivier Madiba, Vanessa Lecosson and Yehni Djidji.

Along with their respective writings, you can find links to their own web pages. In addition, most of these contributors introduce themselves and provide you with some tactics in our exclusive interviews that you will also find in this book.

### About You, Dear Reader: who is this book for? What's in for you?

In The Album, there are 9 chapters organised in 2 volumes. Each chapter deals with a specific topic. You don't have to read everything. If you are interested (to one extend or another) in one or more of these topics, you would possibly appreciate the related chapter(s):

#### Volume 1: Moon Light (softcore)

- Chapter 1: Strategy and Tactics
- Chapter 2: Digital Marketing and Web Visualisation
- Chapter 3: Corporate Worlds and Emerging Markets
- Chapter 4: Quick Wins, Tricks and Tips
- Chapter 5: Transition - Extra Thoughts and Sharp Fantasy

#### Volume 2: Full Moon (hardcore)

- Chapter 6: Software Development and Engineering
- Chapter 7: C# .NET Programming
- Chapter 8: Epic Prototypes, Classic Projects, Historic Genre
- Chapter 9: Research and Case Studies

If you want to, you can download and read only the chapter(s) and volume(s) that you are interested in. Several file formats are available on [TheAlbum.NeceMoon.com](http://TheAlbum.NeceMoon.com) (or [necemonyai.com/blog/page/The-Album.aspx](http://necemonyai.com/blog/page/The-Album.aspx))

All the Web links in this document are working, feel very welcome to click on them.



## For All Practical Purposes

This document contains the complete Album, 9 chapters organised in 2 volumes. Each chapter deals with a specific topic. Depending on your interests, you may download, (re-)read or share any of the various chapters and volumes independently/separately. The PDF, EPUB, MOBI/AZW3/KF8 (Amazon Kindle) and MP3 formats are available.

To get them, just click on any of the links you like below or go to [TheAlbum.NeceMoon.com](http://TheAlbum.NeceMoon.com) (or [necemonyai.com/Blog/page/The-Album.aspx](http://necemonyai.com/Blog/page/The-Album.aspx))

## The NeceMoon Album (complete)

### Volume 1: Moon Light (softcore)

[Chapter 1: Strategy and Tactics](#)

[Chapter 2: Digital Marketing and Web Visualisation](#)

[Chapter 3: Corporate Worlds and Emerging Markets](#)

[Chapter 4: Quick Wins, Tricks and Tips](#)

[Chapter 5: Transition - Extra Thoughts and Sharp Fantasy](#)

### Volume 2: Full Moon (hardcore)

[Chapter 6: Software Development and Engineering](#)

[Chapter 7: C# .NET Programming](#)

[Chapter 8: Epic Prototypes, Classic Projects, Historic Genre](#)

[Chapter 9: Research and Case Studies](#)

The Album is available in French as well at [Album.NeceMoon.com](http://Album.NeceMoon.com) (or [necemonyai.com/Blog/page/L-Album.aspx](http://necemonyai.com/Blog/page/L-Album.aspx))

# Volume 1

# Moon Light

## (Softcore)

In this volume:

Chapter 1: Strategy and Tactics

Chapter 2: Digital Marketing and Web Visualisation

Chapter 3: Corporate Worlds and Emerging Markets

Chapter 4: Quick Wins, Tricks and Tips

Chapter 5: Transition - Extra Thoughts and Sharp Fantasy

Featuring Ahou The African Chick, Antoine Mian, Cyriac Gbogou, Edith Brou,  
Israel Yoroba, Jean Luc Houedanou, Jean-Patrick Ehouman, Karen Kakou,  
Monty Oum, Nanda Seye, Nnenna Nwakanma, Olivier Madiba,  
Vanessa Lecosson and Yehni Djidji

# Chapter 1

## Strategy and Tactics

Featuring Jean-Patrick Ehouman, Karen Kakou, Monty Oum,  
Vanessa Lecosson and Yehni Djidji



## Words to Live By

By [Monty Oum](#)

I continually hope that someday I'll get a chance to pass on everything I've learned. There hardly seems a moment considering [living by my words](#) only means having absolutely no time to say them. So at least I can take a moment in slight to utter in passing.

Never let anyone tell you that something is impossible.

I lived by that when it sparked in me the possibility of achieving something despite everyone telling me it being impossible otherwise. People have always told me it's impossible, that it's been tried. If you know what I'm talking about then you're already on your way. What I will say to you, ignore those who've been defeated by what they call "experience." Keep going and don't let anyone tell you otherwise.

I know this might sound familiar even, never give up etc. But it's how I live my life even currently as there still are more "experienced" people who will criticize that I'm doing it wrong. It's the human spirit that will keep us alive. This rings true to the events of my life, [when I started Haloid](#). I didn't do it because I wanted to get [a job in games](#), or that I wanted to become famous, or what have you. I did it because I knew it could be done, and that the road to finishing it was imperfect and difficult. What finishing it yielded me was only more steps on the path to pushing forward in what I believed in.

My life lacks much, I'm broke, my car is in horrible shape, my apartment is a mess, I spend 3 days in a row at the office regularly and sleep on the couch only when I need to. I'm helplessly antisocial. I see [my girlfriend](#) 3 times a year.

I come from an impoverished family of Cambodian civil war refugees. It's been a long journey between being so incredibly poor that I didn't have paper to draw on when I was little, to where I am now. It wouldn't have happened if I didn't dive headfirst and take a risk, and continually risk it all to keep going forward... Heh, success? They thought it was impossible.

What is the greatest reward for living in such difficulty?

I hope you see it someday too, because the world looks very different, [when you're pushing yourself](#) every second you've got.



## Monty Oum or the Bruce Lee of Digital Arts

By [Necemon](#)

There are some people who are willing to sacrifice almost everything, including their time, resources, sometimes even their social life, their sleep and some "normal" pleasures of life. All of that, in order to work hard and focus on their vocation, to improve things or try and make their world a slightly better place. Your grandparents had a word for that: Saints. Today we refer to them as Entrepreneers.



We lost [one of the top soldiers](#). An [Internet Legend](#) and amazing artist has left us. [Monty Oum](#) was an American digital artist known for his crossover fighting videos such as Dead Fantasy and RWBY. He was, among other things, a [self-taught 3D animator](#), whose education was mostly observational.

Most geeks first became aware of Oum after he released the crazy fighting mashup Haloid, using characters from the video games Halo and Metroid. A part time dreamer production:





The video caused a buzz in the gaming industry, and Monty joined Namco Bandai, where he worked on the video game [Afro Samurai](#). He was also responsible for some of the [most awesome Web Animation](#) around, most of which relied on [Rule of Cool](#). Millions of people enjoyed his web series "[RWBY](#)" ([Ruby - Red White Black Yellow](#)).

He chose being Awesome as a career and beyond his artistic skills, he was the man who never slowed down, who always wanted his life to be efficient, quick and as productive as possible. In the industry, Mister Oum's dedication was legendary. Monty Oum, also known as [the Bruce Lee of Digital Arts](#), was just one of those people that was [ferocious about the need to create](#), thus that great combination: an amazing creative ability and the drive to put it to use. [Monty](#) was the personification of motivation and determination.

Can you match this resolve? If so, then you will Succeed.



Personally, I have been following his work since [the Road](#). I admired him for working so hard and being so passionate about [what he created](#).

So much potential, but gone by age 33, following a sudden allergic reaction. That's the kind of stories that remind us how transient, fleeting and ephemeral life is. Sometimes we work extensively, getting into (large) projects... But everything may collapse all of sudden, due to a small accident, a simple medical error or some allergy coming out of nowhere...

Philosophically, it's a pretty important concept: to stay aware of the end which is inevitable. Living in the Borderlands at times, if not all the time. Here, "Borderlands" is a reference to the Bushido concept of meditating on your own death: realising that you will die at some point, and living your life accepting that. [Bushido](#) (the way of the warrior) is the code of moral principles that Japanese samurais had to observe and that inspired the code of honor of disciplines such as aikido, judo, jujitsu and karate. But in its purest form, Bushido requires the respect of danger and death from its practitioners, so that they can effectively assess and deal with the present moment in relation to their own death.

To this extent, we could have concluded with the cliché stating that life is short and that we must enjoy every moment. The problem is that there is some complexity in the proposition "*enjoying the moment*", given the widely divergent interpretations. What is a chore for some people can be a real pleasure for others. And vice versa, which is a distraction for some may be boring for others. The main thing is to [do what you love](#). Have you ever been so intensely absorbed in a creative activity, so that when you get disconnected, you feel that getting back to it would be more exciting than proceeding with "normal" distractions? Some of us execute like crazy because there is no alternative. Workaholic? Please. That's sounds like something lazy people would say. When you are emotionally engaged by passion or for a cause, you tend to explore ambitious perspectives without regret, no matter what might happen next.

[A lot of life lessons could be learned from this man](#), but the most important is probably how indomitable the human spirit is. If you endeavor to achieve, it will happen given enough resolve. It may not be immediate, and often your greater dreams is something you will not achieve within your own lifetime. The effort you put forth to anything transcends yourself, for there is no futility even in death.

Monty Oum may have left us, but he leaves behind many great works and inspiration.









## The 5 Most Terrific Motivation Tips (From Monty Oum The Artistic Genius)

By [Necemon](#)



Monty Oum was [the Bruce Lee of Digital Arts](#). That was the man who never slowed down, who always wanted his artistic life to be efficient, quick and as productive as possible. He was literally a personification of motivation and determination, here are some of his best lines:

***Time stops for no one and I have no time to stop.***

\*\*\*\*\*

***If you are looking to surpass your current self, then be prepared to sacrifice part of who you are and don't look back.***

\*\*\*\*\*

***Which software to use? It's not the tools you use, it's how you use them. I guess the best tip for beginners would be to really just start and do it. Keep moving forward...***

\*\*\*\*\*

**The World looks very different, when you push yourself every second you've got.**

\*\*\*\*\*

***Can you match my resolve? If so then you will Succeed. I believe that the human spirit is indomitable. If you endeavor to achieve, it will happen given enough resolve. It may not be immediate, and often your greater dreams is something you will not achieve within your own lifetime. The effort you put forth to anything transcends yourself, for there is no futility even in death.***

## Losing Your Arm

By [Jean-Patrick Ehouman](#)



One of the hidden differences between outstanding Success Stories and the average Joe stories, lies in [a huge sacrifice](#). This is also referred to as "losing an arm". This is indeed about some irreversible loss and the pain endured at the time of execution would be unbearable for most of us.

You will not find a single person who has accomplished a monumental work without giving up something generally considered very precious. In many cases, they decided to "lose an arm" in order to make their dreams come true. They ultimately accepted a physical or emotional pain that most of us would not have accepted. Sometimes they put up with it over several years.



When a teenager decides to become a sports professional or a music/movie star, he makes the implicit decision to ["set aside" certain teenage distractions](#) in order to reach his goal. You may have heard of Tiger Wood's long hours of training as a child. Encouraged by his parents, whatever the weather, he agreed to train on Sundays (days off for his competitors). Some may find it strange or insane to play golf in the rain, but that was not the case for the Wood family. Others may say that it is useless to rehearse Rap lyrics 15 hours a day without leaving home for several weeks. That was not the opinion of best-selling musician Jay-Z when he was still a teenager. For ordinary people, it would sound even more awkward to try the same chemical or mechanical experiment more than a thousand times, with only a minor modification in the process at each test iteration. That would be without taking into account the drive of Thomas Edison who indulged in it with pleasure during most of his life.

Achievement therefore has a price that only committed individuals can assess, understand and accept. At any moment in life, a woman can decide to do something remarkable. Open the Guinness Book of Records and you will see things as incredible as the people who accomplished them.

Only you can decide to "lose your arm" to achieve a dream. Whether it be notoriety, money, charisma, you will never reach the highest level without sacrificing something precious. A coach, a mentor, a teacher cannot do much for you in that respect, his contribution could not go beyond a permanent reminder of your commitment.

The good news is that very few people are willing to put aside what they view as precious. This gives you the opportunity to go further than others without worrying about real competition. Since at some point, it will be necessary to "lose an arm", and at that point, almost everyone drops out. If you agree to proceed, there is no reason why you would not [reach the top](#).

## On reading books and self-education: My Essentials

By [Necemon](#), 2012

There is a lot of great skills you get from school/university lectures, no doubt. But on the other hand, for some specific and practical "super powers" you want to acquire or enhance, you have no other choice but teach yourself. The potential reasons are legion. That could happen by mere interest or if you want to deepen your knowledge in one of your subjects. Or maybe for the sake of personal projects, say learning some essential life skill they don't teach you in class. So yes, when college doesn't prepare you for life battles, you have to coach yourself.

Anyway, I am not here to complain about any education system. I am writing this post because I would like to share the best and most useful books I went through, as I have been reading a lot lately. I read on a variety of fields and for each of them I am suggesting the top books (according to my humble opinion).

Here we go:

### **Productivity and Organisation**

Being productive implies being effective, deliver what you promise, go from talk to action: in short, being real. It's not always straightforward to be so concrete. It takes some will power and some organisation. It's a vital skill to have so that your dreams don't remain just dreams. So that your dreams come true:

[Getting Things Done](#) (by [David Allen](#)): Probably the best book on organisation. It teaches you how to free your mind, how to manage all your "stuff" and how to make efficient choices about your projects.

[How to Make a Complete Map of Every Thought You Think](#) (by [Lion kimbro](#)): As the title suggests, this book is about how to make a complete map of everything you think for as long as you like. If you do the things described in this book, your thoughts will be clearer to you than they have ever been before. You will see things you have never seen before. It may feel like, for the first time in your life, you really have a clear idea of what kind of thoughts are going through your mind.

### **Marketing and Business**

Learning about marketing is important, [even for software developers](#). Actually, it's important no matter [what you want to do in life](#), whether you want to run a business or work for an existing company. It helps you understand people and deal with them.

[Foundations of Marketing](#) (By [Jobber](#) and [Fahy](#)): That book would be a great way to start off, assuming you don't know anything about marketing. It's easy to understand and it teaches the basic concepts through concrete examples.

[Influence : The Psychology of Persuasion](#) (by [Robert Cialdini](#)): This book discusses the weapons of influence from a psychological point of view. Among other things, it explains the main factors that cause one person to say yes to another person, as well as real life examples.

[Breakthrough Advertising](#) (by [Eugene Schwartz](#)): Basically about copywriting (or how to sell with a few written words). What makes this one excellent is, it teaches how to identify the desires of the customers and write headlines and ads that trigger a reaction, as they present solutions to their needs.

[I Will Teach You To Be Rich](#) (by [Ramit Sethi](#)): Not just another boring finance book. Not only is this book informative, it is fun and includes practical tips that would help you master your finances and possibly make more money.

## User eXperience

UX is basically the way users feel about your product or service. It highlights valuable aspects of human-computer interaction. Whatever you build, UX is not a choice. This stuff matters, it affects the success (or failure) of your product. Bad design and bad experience happen.

[Mobile Interaction Design](#) (by [Matt Jones](#)): this book reveals how mobiles have the potential to be key building blocks in this new digitally augmented world, and how careful innovations in that field could change the world, bringing pleasure and helping businesses.

[Press On](#) (by [Harold Thimbleby](#)): Provides many ways of thinking clearly about the design of interactive devices and about interaction programming. The emphasis throughout is on making better interfaces.

## Digital Arts

Simply put, digital arts are computer-aided artistic works. For this selection, I focus on visual arts.

[How To Cheat In Photoshop](#) (by [Steve Caplin](#)): A goldmine of practical and useful tips, tricks and step-by-step walkthroughs introducing you to the art of creating photo-realistic montages.

[Digital Character Painting Using Photoshop](#) (by [Don Seegmiller](#)): a complete book on digital illustration and painting. It covers such theoretical topics as how to get ideas and how to get them out so that others may see them. Then it deals with artistic fundamentals that will help you take your ideas to the next step, drawing and painting. Finally, it provides a series of tutorials that show how to handle different issues when you draw and paint.

## Online Games

Yes, as a virtual worlds enthusiast, I had to come up with that selection too :-)

**Networking and Online Games : *Understanding and Engineering Multiplayer Internet Games* / *Multiplayer Game Programming*** : those 2 books would teach you the history of online gaming as well as the basic concepts and some technical aspects to be considered.

[\*Developing online games, an insider's guide\*](#) (by [Mulligan](#) and [Patrovsky](#)): This one provides insight into designing, developing and managing online games. This is where you learn from the top two online game developers through the real-world successes and mistakes not known to others. There are Case studies from 10+ industry leaders.

[\*Designing Virtual Worlds\*](#) (by [Richard Bartle](#)): while the previous book discusses broadly online games development, this one focus on virtual worlds. This is a book for people who design virtual worlds. Because of this, it's also a book for people who implement, operate, study, or play virtual worlds. Because of this, it's also a book for people interested in entertainment, education, creativity, art, society, culture, philosophy, space, architecture, psychology, identity, language, economics, government, theology, drama, literature, or cognition. Virtual worlds are of the future. If you want to create or understand that future, this is the book for you.

## C# Web Programming

We are getting very specific here. There are lots of technology/programming books out there, but I am going to focus here on what I know best: C#.NET Web development.

[\*ASP.NET 3.5 For Dummies\*](#) (by [Ken Cox](#)): if you want to start off quickly and just begin to create your own websites/web apps within days, this is the book to pick. For an immediate immersion in ASP.NET and the Visual Web Developer environment.

[\*Pro C# 2008 and the .NET 3.5 Platform\*](#) (by [Andrew Troelsen](#)): Well, now that you have made a few apps, as you move forward and it gets confusing at times, you realise that this .NET Framework is really wide. This is the book that give you an overall view on the technology from the philosophy of .NET through the core C# programming constructs, the advanced C# programming constructs and web and desktop user interfaces, among other things. It even goes through more obscure concepts such as shared assemblies, delegates, late binding, workflow foundation, COM interoperability, etc. In short, a very complete book.

[\*ASP.NET 3.5 Social Network\*](#) (by [Andrew Siemers](#)): after you grab the key concepts and play around with the technology through a few projects, maybe you would want to launch an actual product, something that people could use. The way to go about that is probably to see how experts do it, for instance, how they build discussion boards, membership systems or blogs. Better yet, this book will teach you how to build a complete ASP.NET social network from start to finish.

[\*Building a Web 2.0 Portal with ASP.NET 3.5\*](#) (by [Omar Zabir](#)): This one is definitely the pièce de résistance. This book not only teaches how to build a State-of-the-Art Ajax website using ASP.NET, LINQ, advanced ajax concepts, Windows WorkFlow and other cutting-edge Microsoft technologies; it also shows how to solve real life production challenges such as deployment, hosting, performance optimisation, scalability, caching, security, monitoring, etc.

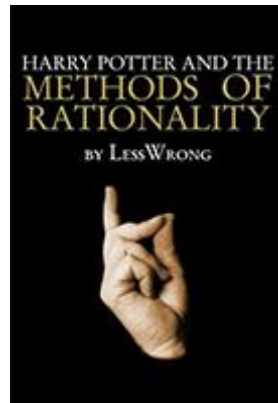
That's it for now...

## On reading books and self-education, My Essentials (Part 2)

By [Necemon](#), 2015

A couple of years back I [shared my reading list and wrote about the books I found most interesting](#). But then, after that, I have been reading some more over the past few months, here are my recent favorites:

### 1. [Harry Potter and the Methods of Rationality](#) - by [LessWrong](#)



A "parallel universe" Harry Potter fiction written by [Eliezer Yudkowsky](#) (A.K.A. LessWrong), an AI researcher, writer and decision theorist at the Singularity Institute for Artificial Intelligence. The fan fiction story illustrates topics in cognitive science and rationality, recasting the original story in an attempt to explain Harry's wizardry through the scientific method.

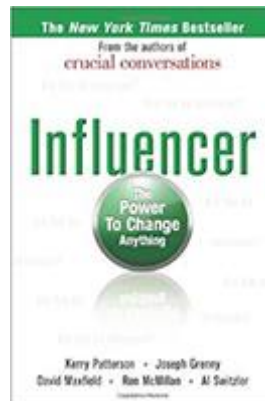
Petunia Evans has married an Oxford biochemistry professor and young genius Harry grows up fascinated by science and technology. When he finds out that he is a wizard, he tries to apply scientific principles to his study of magic, with sometimes surprising results. With magic now seemingly a proven reality, Harry now takes his place at Hogwarts, armed with [Enlightenment ideals](#) and the experimental spirit.

The text contains many clues: obvious clues, not-so-obvious clues and truly obscure hints. This is a rationalist story; its mysteries are solvable, and meant to be solved. All science mentioned is real science.

The fanfiction is known for possibly being the most reviewed Harry Potter fanfiction, with over 20,000 reviews on Fanfiction.net as of July, 2013.

Disclaimer: J. K. Rowling owns Harry Potter, and no one owns the methods of rationality.

2. *Influencer (The power to change anything)* - by Kerry Patterson, Joseph Grenny, David Maxfield, Ron McMillan and Al Switzler



Leading is hard. And the only thing harder than leading a team towards a common vision – provided you are able to unite it in the first place – is leading a team through change. Change is harder.

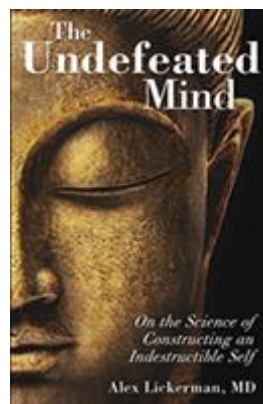
There is no magic formula for managing change, but the book "*Influencer: The Power to Change Anything*" provides a great framework.

The authors first point out what does not work: talking. You can't explain logically to people why they must change their behaviors and expect anything to happen. Peoples' behaviors are tuned to what they believe is in their best self interests. Unless you can change that belief, you have little chance of changing their behaviors.

Influence is about changing hearts, minds, and behavior to produce meaningful, sustainable results.

In short, don't waste time on how to create change until you've clarified what you want, why you want it, and when you want it. The book explains how to clarifying measurable results, find vital behaviors, and use a strategic, varied approach to influence individuals and groups.

3. *The Undefeated Mind: On the Science of Constructing an Indestructible Self* - by Alex Lickerman



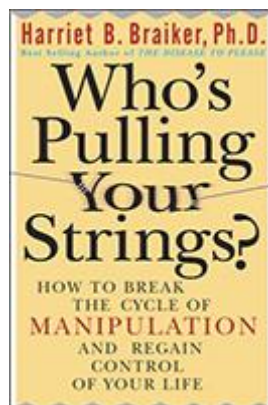
A book about how to surmount the challenges that life throws our way. It's a powerful book that helps you cultivate your inner strength to face hard times with a fearless heart. It's more than a book. It's a set of tools for life.

The tools in this book help you stand strong when tested. The key, according to [Dr. Lickerman](#), is turning adversity into wisdom. It's that wisdom that helps you develop your resilience and achieve indestructible peace.

It's also a beautiful blend of ancient wisdom and modern science, offering a set of guidelines not for solving problems but for establishing a life state that makes all problems solvable, and dramatically increase your ability to withstand adversity.

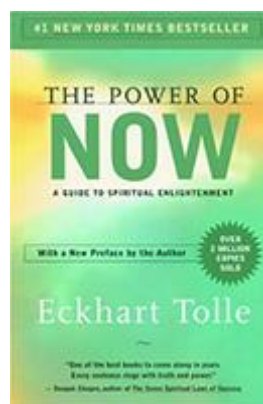
It doesn't just happen automatically, and it takes practice. Extraordinary people may be born, but they can also be made.

4. [\*Who is pulling your strings? \(how to break the cycle of manipulation and regain control\)\*](#) - by [Harriet Braiker](#)



In "*Who's Pulling Your Strings?*", Dr. Harriet B. Braiker exposes the most common methods of manipulators, and with the help of self-assessment quizzes, action plans, and how-to exercises, she helps you recognize, end and avoid the manipulative cycle (for good) through lots of straight-forward techniques.

5. [\*The Power of Now: A Guide to Spiritual Enlightenment\*](#) - by [Eckhart Tolle](#)



A guide stressing the importance of living in the present moment and avoiding thoughts of the past or future. The author writes that an individual should be aware of their "present moment" instead of losing themselves in worry and anxiety about the past or future.



According to the book, only the present moment is important, and both an individual's past and future is created by their thoughts. The author maintains that people's insistence that they have control of their life is an illusion that only brings pain. The book also describes methods of relaxation and meditation to aid readers in anchoring themselves in the present. These suggestions include slowing down life by avoiding multi-tasking, spending time in nature, and letting go of worries about the future.

Although the journey is challenging, Eckhart Tolle uses simple language and an easy question and answer format to guide you.

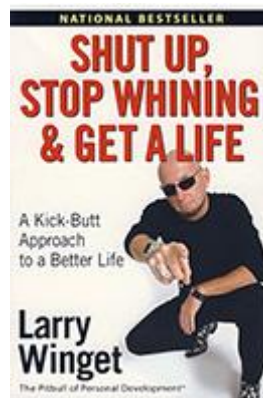
6. [\*I was blind but now I see\*](#) - by [James Altucher](#)



[James Altucher](#) is one of my favorite bloggers. In this book he attempts two things. First, he tries to create some awareness about a wide variety of issues that we tend to take for granted/as normal without actually considering why, using clear cut arguments and mentioning alternatives to make his points.

Secondly, the author describes how to deal with the awareness and to get the confidence/strength to do things in your own way. Funny, brutally honest and thought-provoking.

7. [\*Shut up, stop whining and get a life\*](#) - by [Larry Winget](#)



This is not your typical self-help book. This book won't make you feel good about yourself and that's a good thing. If you are hoping to improve yourself with a bunch of feel-good mantras, then this book is NOT for you. But if you are ready to get up and really change your life, then Larry Winget is your man. Larry (A.K.A. the Pitbull of Personal Development) is direct, caustic, and controversial. His tough-love approach boils success down to a simple formula: your life failures are your own freaking fault, and if you shut up, stop whining, and take action, your life will get better.

8. [\*The 50th Law\*](#) - by [50 Cent](#) and [Robert Greene](#)

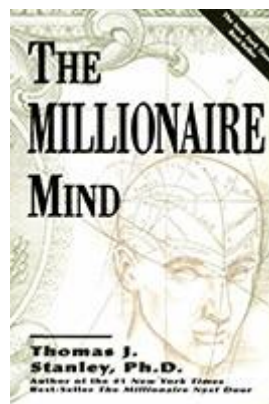


Pop culture and hip hop icon 50 Cent joins forces with Robert Greene, bestselling author, to write collaboratively a bible for success in life and work based on a single principle: fear nothing.

They came up with *The 50th Law*, a New York Times bestselling book on strategy and fearlessness.

The book is a semi-autobiographical account detailing 50 Cent's rise as both a young urban hustler and as an up-and-coming musician with lessons and anecdotes from historical figures such as Abraham Lincoln, Sun Tzu, Socrates, Napoleon, Malcolm X, and James Baldwin.

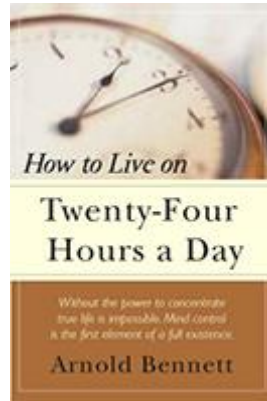
9. [\*The Millionaire Mind\*](#) - by [Thomas Stanley](#)



*The Millionaire Mind* draws from the author's research of America's affluent to examine the ideas, beliefs and practices of the segment of the financial elite that use little or no consumer credit.

The book analyzes the common environmental and lifestyle factors that preceded and resulted in this researched segment's ability to accumulate wealth, by exploring the attitudes and skills that enabled these millionaires to build and maintain their fortunes.

10. [\*How to live on 24 hours a day\*](#) - by [Arnold Bennett](#)

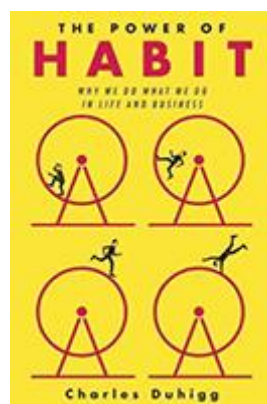


Time is a very funny thing; everyone gets the same amount per day. No one receives either more or less than you receive. It's not possible to go the store and buy time. But time is the inexplicable raw material of everything. With it, all is possible; without it, nothing is possible.

The author states that time is the most precious of commodities. He said that many books have been written on how to live on a certain amount of money each day, but he adds that the old adage "time is money" understates the matter, as time can often produce money, but money cannot produce more time. Time is extremely limited, and Bennett urged others to make the best of the time remaining in their lives.

*How to Live on 24 Hours a Day* offers practical advice on how one might live (as opposed to just existing) within the confines of 24 hours a day. A classic piece on self improvement teaching you to live to the fullest.

11. [\*The Power of Habit \(Why We Do What We Do in Life and Business\)\*](#) - by [Charles Duhigg](#)

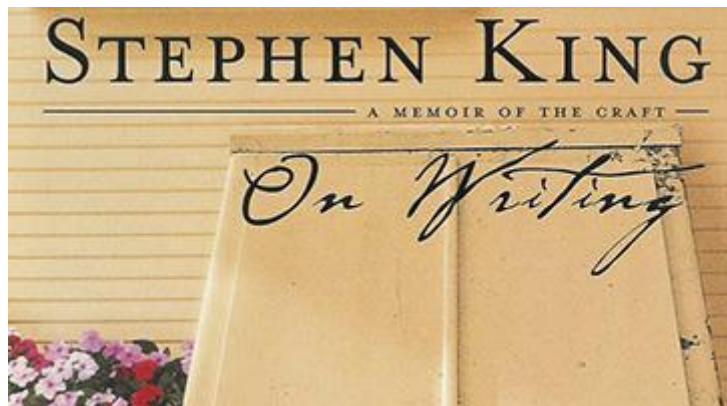


Habits can save us or ruin us in a thousand ways; in fact, these "automatic responses" affect every arena of our lives.

In *The Power of Habit*, Pulitzer Prize–winning business reporter Charles Duhigg explores the science behind habit creation and reformation. He takes us to the thrilling edge of scientific discoveries that explain why habits exist and how they can be changed.

At its core, *The Power of Habit* contains a sensible argument: The key is understanding how habits work.

12. [\*On Writing, a Memoir of the Craft\*](#) - by [Stephen King](#)



*On Writing* is part biography and part how-to. [Stephen King](#) (world best-selling author of contemporary horror, supernatural fiction, suspense, science fiction, and fantasy) writes about his childhood and young adulthood, relating stories that made him the writer he has become. Stephen King then moves into the mechanics of writing, offering advice and insight into a successful career that has worked so well for him but remains distant for thousands of others.

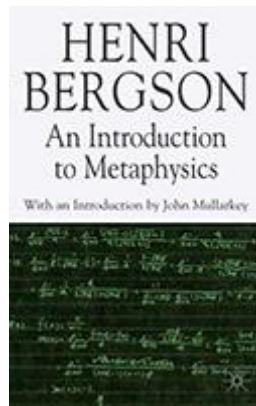
13. [\*Principles\*](#) - by [Ray Dalio](#)



*Principles* is like a modern day meditation guide aimed at personal, professional and career development.

[Ray Dalio](#), self-made multi-billionaire, details the purpose and importance of having principles in general, his most fundamental life principles that apply to everything he does, and his management principles.

14. [\*Introduction to Metaphysics\*](#) - by [Henri Bergson](#)



A thought provoking essay about the concept of reality by Henri Bergson. For Bergson, reality occurs not in a series of discrete states but as a process similar to that described by process philosophy or the Greek philosopher Heraclitus. Reality is fluid and cannot be completely understood through reductionistic analysis, which he said "implies that we go around an object", gaining knowledge from various perspectives which are relative. Instead, reality can be grasped absolutely only through intuition, which Bergson expressed as "entering into" the object.

This is a difficult book but his ideas about time and the value of immediate experience are definitely worthwhile.

That's it.



## **Becoming Your Own Boss: 7 Lessons Learned On The Spot**

By [Yehni Djidji](#)



A few months ago, I left my job in order to [become my own boss](#). Well, that's what I thought. But I could learn a few lessons from this experience that I would like to share.

### **1. Fun working**

When you do a job you love, it turns out that you do not really work. You have [fun while learning and making money](#).

### **2. Varied activities, balanced intellect**

I no longer need to mechanically repeat the same gestures, or to work on the same project all day long. I can help someone start a magazine while working on a movie script, [writing articles](#) for a radio and tweaking a short story. Of course, it's important to be very organized. I'm still looking for the secret.

### **3. Guaranteed savings**

Some people have a salary that seems huge but they spend most of it on transportation, [communication](#) and food in the workplace. If you work at home, you won't spend any money on transport except for value-added appointments and you will simply eat the homemade meal of

the day... if you are disciplined. Additionally, the informal side of the start-up activity saves you a lot of taxes. You have to make the most of this phase, and save as much as possible, and formalize the activity quickly because there are opportunities that you cannot seize when you are not legally constituted.

#### **4. Master of your time and space**

You can arrange your schedules to be available for your family while carrying out your activities. You do not have to worry about rush-hour traffic anymore and waste precious time wondering if the slowdown is due to an accident or to the movements of an official convoy. At home, in a restaurant, at the beach, [you can work everywhere](#). I do not advise people to work exclusively from home and I will explain why a little further.

#### **5. Did you say "No More Boss"?**

If you thought you could actually become your own boss, think again. [Your customers are your new bosses](#). Some will even make you regret leaving your job. Each of them has their demands, their constraints, their inappropriate language sometimes. But this will allow you to discover some [other sides of entrepreneurial life](#).

#### **6. Be aware and beware of weight gain**

In order to work from home you have to [be disciplined](#). Otherwise, you will be tempted to gulp the fridge contents by snacking excessively. True story.

#### **7. Knowing how to set limits**

You must be able to explain to the family that staying in front of a computer does not mean that you are sitting idle. They must comprehend that you may be in the house without being available. Otherwise you will be involved in a parade of servants coming to request various things that they normally handle well when you are not around. And yet, that does not even [take the children into account](#). I cannot work with my son around. He will knock on the door of my room until I open it. For those with children who are still too young to understand that you may need some quiet time, it may be better to go out.



**[Interview] In The Bubble of Yehni Djidji: "I offer educational entertainment and life-changing experiences to children"**



**1. Quick (but effective) introduction of the personage and her achievements?**

I am Rosine Kakou Ano. Graduated in Communication, Marketing and Management. I have been operating under the pseudonym "[Yehni Djidji](#)" since 2008, which was the year when I launched [my blog](#). I am the founder of a literary agency called [Livresque](#) that helps those who wish to publish a book throughout the process, organises events about literature and offers educational entertainment to children.

**2. What are the values that better define you?**

Piety: the fear of God is the hallmark of men of value. This gives them integrity, loyalty and honesty.

Responsibility: It comes along with maturity and allows to respect and completely fulfill the commitments made to others and to oneself. It is a quality that lots of people are sorely lacking.

Empathy and liberality: To love one's neighbor, to be able to put oneself in their shoes, to understand their burden and help them to relieve themselves.

**3. What are your main goals in life?**

[Books helped me to strengthen myself and to launch my career](#). Therefore, I am actively striving to put them back at the heart of habitudes so that the maximum of people can enjoy their benefits. Additionally, our continent has great stories and I like to participate in relating them to the world through increasingly varied formats.

**4. What tools and techniques do you use to accomplish things efficiently?**

I am an Internet addict who tries to moderate her indulgence, even though it is difficult, since it's also my work environment. My favorite gadgets are my computer and my phone.

**5. Any recommendations for your juniors?**

I recommend my [coloring book with traditional outfits drawings](#) for children aged 3 to 6, and why not more if there's a fit. There's no maximum age for coloring. I also encourage them to commit to appreciating books, not just as learning tools, but as daily companions too. That would be a life-changing experience.

**6. What is the best way to contact you?**

The best way to contact me is by [email \(yehnidjidji@gmail.com\)](mailto:yehnidjidji@gmail.com), and then through [social networks](#).

## What will you do in life?

By [Necemon](#)

This goes to those who haven't decided yet what they are going to do in life.

It's important to start thinking about it now, no matter how old you are. You better start early because it takes time, effort and dedication to [improve one's skills and to reach goals](#). The earlier you start, the more time you have to mess up (and you will probably mess up on the way).

To decide about what to do, there are basically 3 things you need to consider:

- what you love, what [you are passionate](#) about.
- what your talents are, what you are naturally good at, but [the skills you acquired](#) on the way matter too.
- what value you can create, what you can do that [people will pay you](#) for.



Now, it's generally easy to come up with what you love most. However, I noticed that many people have issues combining that with their strengths and the value they can create. If that's also your case, you can try the following.

Let's consider your phone book. In that, you got a friend for pretty much anything. When your computer is screwed up, you know which friend to call. When you feel depressed, there is that one friend, you call to cheer you up. When you want to hang out in town, you also know whom to contact. Well, we can come up with many similar examples, but the point is, you are also a contact to your friends. So ask yourself, what is it that they contact you to help them for? That may give you a clue of what you are good at and how [you can help others](#).

Wishing you all the best,

Necemon

## **11 Personality Traits Of Successful Entrepreneurs**

By [Jean-Patrick Ehouman](#)



I spend a good deal of my time trying to understand what is unusual about successful start-ups founders and product creators, that is, what is special about their way of doing things. Hence we can infer some attitudes and "design patterns" that are often found among these people. Here is a (non-exhaustive) list of traits that I have noticed in these people. Of course, some entrepreneurs succeed without these traits.

### **Curiosity**

They read a lot and do not hesitate to ask questions for things that may seem trivial at first. They have a thirst for knowledge. They want to understand what could have led to a given success or failure. They are also constantly looking for things that will motivate and retain their best employees.

### **The ability to collaborate**

Those who know how to delegate and build strong relationships with their employees are more likely to come up with solutions or products that meet the real needs of their customers.

### **Spending as little as possible**

When it comes to successful entrepreneurs, as the saying goes: "an entrepreneurial adventure should tends to create money and not to lose it". That is why they always look for ways to avoid digging into their pockets whenever possible. By choosing goods exchanges over purchases for example, they always find ingenious means to acquire the resources they need without having to reduce their funds.

### **Fulfilment**

Doing something that they love and that makes them happy is essential for these entrepreneurs. They like to have total control of their income. They also derive enormous satisfaction from the fact that they create value-added products, services and/or concepts.

### **Planning maniacs**

From their perspective, everything must be planned. Whether it be short-term actions or those to be implemented in the medium and long term, they take the time to write a plan with a fairly high level of detail. This allows them to have an effective guideline and to be able to adjust their strategy at any time through daily monitoring.

### **Power users (for those who have access to technology)**

The best entrepreneurs do not hesitate to invest in high quality websites while taking the time to set up internal management and communication tools. They are keen to spend on boosting those core assets.

### **Giving back to the community**

In addition to their commercial activities, successful entrepreneurs often act in favor of the community. They are involved in charity work or they simply initiate projects to help others.

### **Man of Action**

They try to differentiate themselves from their competitors at all times by being proactive and taking the lead. They take risks that the average Joe would consider insane. Also, they do not view mistakes as a fatality. On the contrary, they use those lessons to set up new strategies and therefore to increase their chances of succeeding at their next attempts.

**Hyper motivated**

Finally, successful entrepreneurs are people with extraordinary motivation. When it looks like they are hitting rock bottom, they surprisingly bounce back and come up with new ideas that they implement with tremendous determination and unshakeable confidence.

**A constant and objective self-evaluation**

They remain honest with themselves even when it comes to self-assessment. So they make every effort to correct their imperfections. They are always willing to challenge themselves and their unproven tactics. For example, if an employee fails to complete a task, they would assume that it is mainly because they picked the wrong man for the job.

They manage to **make the most of the competencies in their network**, they keep learning relevant "best practices" and "design patterns" from their competitors and colleagues.

## **5 Team Management Insights From Sebastian Marshall**

By [Necemon](#)



There are already tons of good insights available on [Sebastian's website](#). However, having a one-to-one discussion with him was very useful to discuss specific challenges relevant to some ongoing projects. Through that conversation, Sebastian gave concrete, actionable pieces of advice on project management, recruitment and teamwork:

1. It's better, faster and cheaper to work with people who are already motivated instead of investing in motivating people who are not. The key is to find people that are passionate about the project field(s).
2. For a given field, a good way to find passion-driven individuals is to connect with relevant lecturers, tutors and other educators. Those field experts are likely to mentor interested juniors looking to earn experience through projects/internships.
3. With respect to the people working with you, don't give them an abstract description of the project. Be specific in your explanation with a detailed description and ideally have something concrete already working to show.
4. Don't give them an abstract definition of their mission either. Have very specific tasks for them to do.
5. Time estimates, milestones and regular catch-ups help to keep the project on a good track.



## **5 Attitudes Without Which You Will Have Trouble Reaching Your Goals**

By [Jean-Patrick Ehouman](#)

You probably made resolutions at the beginning of the year. Whether you have achieved your goals or not, there is no need to feel sorry for yourself. You still have time to comply with these resolutions. But you would still have trouble getting there if you don't adopt the appropriate mindsets. In my view, these are 5 attitudes that remain paramount when it comes to raising to the next level:

### **1. Staying honest with yourself**

To acquire a know-how, a skill, a recognition, etc., it takes time and a lot of practice. At first, you will have to acknowledge that you do not know much about how things work. As long as you don't accept the fact that you do not have the required abilities (yet), you will keep an invalid perception of reality. Don't be fooled by the flattering comments or the admiring amateurs. Get closer to knowledgeable people who are way ahead of you and who are willing to mentor you.

### **2. Starting with the basics**

Competition generally tends to highlight different levels of proficiency, and each level has its specific features. Those of the lowest level remain the most important ones, as they constitute the basis of your skills. You will need to take the time to master these basics.

### **3. Asking questions**

There is no such thing as a dumb question. The best approach is to record the answers on the medium of your choice. However, I can assure you that your brain is not the place where you should record them. You must have a (paper or electronic) notebook where you will mark everything that will be useful for your evolution. This will ensure that you won't need to ask the same questions several times. Asking the same question twice to the same person is like giving them a motive to stop answering your questions. It would be unfortunate to lose people who have agreed to mentor you.

### **4. Accepting criticism**

This will always bring you back to point 1. Only by accepting criticism will you be able to identify your weak points in order to overcome them.

## **5. Working hard, paying attention to details**

If you only believe in innate ability, you are doomed. I don't know much about academic studies relating to human abilities, however my field experience strengthens my conviction that the most important factor for success is hard work. And in order to move forward through work, you need to pay close attention to details.

## Just think about it: 50 ideas to live by

By [Necemon](#)



1. Just Keep Moving Forward.
2. K.I.S.S. (Keep It Simple & Smart) and H.U.G. (Have Utilitarian Goals)
3. The best way to meet opportunities is probably to start by being an opportunity yourself.
4. The best way to reach your goals is to help those you need to reach theirs.
5. Motivation is what gets you started, habit is what keeps you going.
6. Yesterday you said tomorrow. Just do it.
7. The main thing is to keep the main thing the main thing.

8. There's a difference between knowing the path and walking the path.

9. This is 10% luck, 20% skill, 15% concentrated power of will, 5% pleasure, 50% pain.

10. The more you sweat in training, the less you bleed in battle. The harder you are on yourself, the easier life will be on you.

11. Fake it until you make it.

12. It is our choices that show what we truly are, far more than our abilities.

13. The World looks very different, when you push yourself every second you've got.

14. If you are looking to surpass your current self, then be prepared to sacrifice part of who you are and don't look back.

15. That which is important is rarely urgent, and that which is urgent is rarely important.

16. Parkinson's Law: Work expands to fill the time available for its completion.

17. A journey of a thousand miles just starts with one step.

18. The difference between theory and practice is smaller in theory than in practice.

19. Those who try to do something and fail are infinitely better than those who try to do nothing and succeed.

20. No matter how many mistakes you make or how slow you progress. You're still way ahead of everyone who isn't trying.

21. A ship in harbor is safe, but that is not what ships are built for

22. Your Dream is not what you see in sleep, it's the thing which doesn't let you sleep.

23. There's no traffic jam when going the extra mile

24. Vague Plans Provoke Vague Results. You have to name it before you can claim it.

25. When nothing is sure, everything is possible.

26. If you're going through hell, keep going.

27. People don't care how much you know until they know how much you care about them.

28. To be realistic is to find rational ways for wonderful achievements

29. If you want to look successful, maybe you should actually become successful. Nothing looks more successful than success.

30. You're basically unstoppable if you've got the right model and you execute like crazy.

31. Life remains the same until the pain of remaining the same becomes greater than the pain of change.

32. Extreme Programming is based on 5 core values : Communication, Simplicity, Feedback, Courage and Respect.

33. Success is a lousy teacher. It seduces smart people into thinking they can't lose. When everything is alright, you feel invincible, that's when things get dangerous. Everything is alright... Until it's not.

34. You are not lost, you are locationally challenged.

35. Plan your work and work your plan.

36. The way out is through.

37. Murphy's Law

Anything that can go wrong, will go wrong (at the worst possible time)

Most ideas look great on paper. But by the time events go from A all the way through the alphabet to Z, you see who rules when it comes to getting something from the theoretical stage to the practical stage: it's Murphy's Law.

No battle plan survives first contact with the enemy. Hope for the best, be ready for the worst.

38. Pareto's Law : for many events, about 80% of the effects come from 20% of the causes.

39. Information is difference that makes a difference.

40. The guy who is controlling his emotions is going to win.

41. If you don't write your own rules, someone else will. And the results won't be pleasant.



42. Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

43. Hesitation is always easy, rarely useful.

44. First, just start. Start somewhere. Start anywhere. Start with SomeOne. Start with AnyOne. Start with no one. Start with Something. Start with Anything. Start with Nothing. Just start.

45. Worrying is a waste of time; it doesn't change anything, it just messes with your mind and steals your happiness.

46. Time stops for no one and I have no time to stop.

47. Knowing others is intelligence; knowing yourself is true wisdom. Mastering others is strength; mastering yourself is true power.

48. The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge.

49. When you compete against everyone else, no one wants to help you. When you compete against yourself, everyone wants to help you.

50. All of us can certainly afford to know more than we already do. Information has never been more abundant or more easily accessible.

## **Breaking Through Pebble: from Mediocrity to Excellence**

By [Karen Kakou](#)



### **How Do You Shift from a Restricted Mindset to Thinking Excellence for Greater Success?**

In this briefing, I share a [few tactics](#) from my personal experience of exiting the realm of mediocrity to a mindset of excellence that leads me toward success. I also discuss some observations and principles that have been helpful in [acquiring knowledge and acting on it](#).

#### **1. Being aware that greater opportunity is available to you and you are permitted to own it.**

In my last year of high school, I remember being invited to "Laureat", a government merit event where the political leaders were rewarding the students who earned the highest grades during a sort of award reception. I received my award in "facultative subjects", subjects like music and secondary languages. I was never aware of such program. But when receiving my award, I just felt that, if only I knew, I would have studied with a specific mission and who knows what the results might have been? Maybe I would have gotten an award each year in [some other relevant subjects](#)! Or maybe not... I was a Queen of Overthinking. I used to think too much to the point of getting overwhelmed by the little things because I also wanted

things to be so perfect. I quickly learned that no one will know what I have in mind, what I can offer if I do not present it at all, because no one can read my mind or see through me for my potential. Others can only observe my actions and my results, so I need to get out and do it, then refine it as I go with the input I get.

## **2. Once aware of your chance, it takes discipline to seize it.**

Many have ideas, know the direction to get to where they dream of being, but very few act on it. It is not easy. It takes discipline. The best way to get it done is [to get started with it](#). I love to tell myself these two things, I have them written everywhere in my house: "Do it!" and "Do it anyways". "Do it!" forces me out of bed, go to the gym, get things started especially when I hate the idea of doing it. "Do it anyways" gets me to get things started and follow up until completion, even when I am in doubt.

So, for me, the number one thing that contributed to my success in reaching the best version of myself has definitely been discipline. That is, doing the right thing at the right time, remaining consistent and persistent through it, paying attention to detail, planning and doing anything that keeps you in an orderly, preparative state of mind to recognize the opportunities that will lead you to success. I came from a rough background in which the best future people could have dreamed for me was the hope I someday find a good husband who will treat me right. But the little girl I was already had bigger dreams, bigger ideas for her future than the usual expectations. I wanted to be successful, I wanted my work to have an impact on society, I wanted to feel fulfilled and feel like I leveraged every bit of my potential. But I had to figure out how to get there for myself as I did not have a model to look up to, someone to guide me, nor the financial means to facilitate the process. Discipline has been the key attribute that took me from an impoverished environment and mindset, where the best future that was laid out for me was to potentially become just a good wife and to be grateful to have enough to eat, to now be self-fulfilled. Today with discipline, I was able to leverage my intellectual talent and have the career I knew I could, I was also able to improve my perspective on life heading toward being the person I know I can be, and have it all which I set my mind to in a way that is balanced.

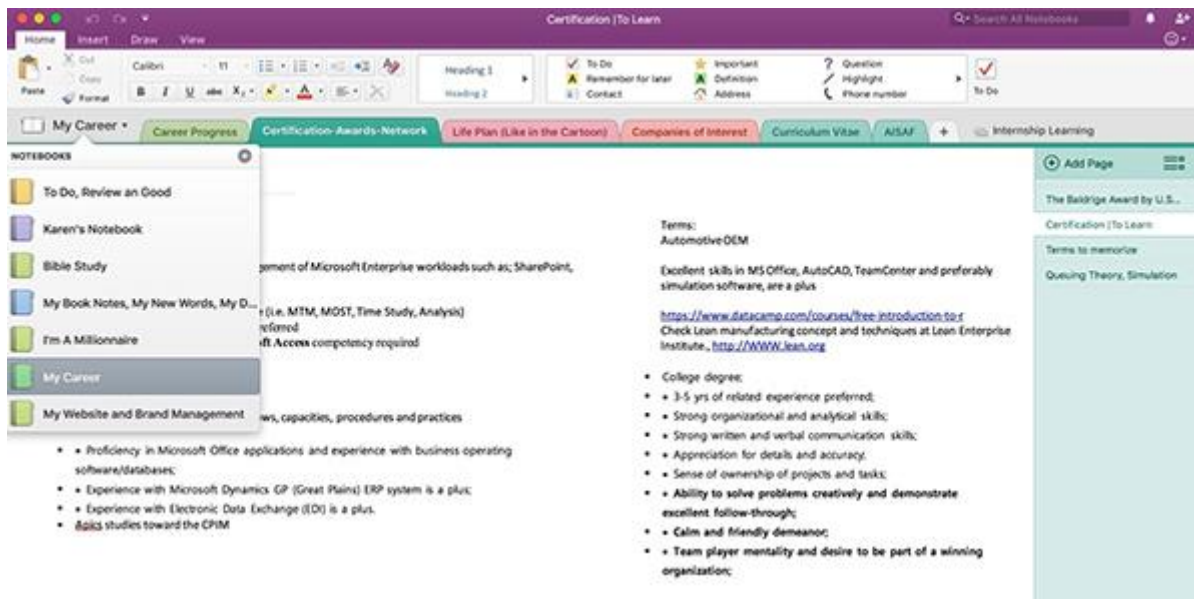
One main discipline skill for me has been planning. I would write out my objectives, what I felt I needed to get accomplished that would lead me to success someday, and then break it down to the day-to-day tasks to do. Living on a schedule, pushed me to get into action because I wanted to check the boxes, it brought clarity into what opportunity is available to me, and helped me seek for more. If it wasn't for the discipline of planning, I would not have been able to go to school, earn good grades, while working many jobs to sustain, and staying alert to what was happening around me to advance my career, given all the work restrictions applied to a foreign student in the USA. I planned for everything, today planning has become like a part of me and it helped me multitask/swich-task and be prepared, which led me to the management role I currently hold.

Two tools I use to remain organized and disciplined are a planner and Microsoft OneNote:

- I use the Franklin Covey planner but I would recommend you use any planner that works for you. It's important to write down what you would like to achieve, then break it down to smaller objectives and plan your daily activities accordingly. Writing down your objectives is the best way to measure your productivity level, improve and possibly address anything that may be holding you back.

- Then I use OneNote to stay more organized, to classify and retain everything that is important to me. And we are talking about the list of books I am writing, my career planning, my economic budget, my recipes etc. I see it as a very big planner that has unlimited pages, you can write down and plan whatever you

would like. As an example, I inserted below a screenshot of my OneNote Files to give you an overview of what could be done with that tool but you should explore it for yourself, it is pretty awesome.



### 3. Continuous Improvement: getting in Kaizen Mode

Getting things done is not a one-off task. You have to keep doing things and doing them better; thus you need to continually improve.

Kaizen is the Japanese word for continuous improvement. In business, Kaizen refers to activities that continuously improve all functions and processes. To me, the Kaizen process applies in every aspect of life. When you can define what you want, put your plans into action and are disciplined through the process, you can analyze your progress and keep on learning new and better ways to achieve your objectives. Adopting this mentality will give you a greater sense of control on what happens to you.

My biggest tools to continually improve has been reading, [listening to podcasts and audio books while driving](#), interacting with individuals who seek growth as well and keep on educating myself. For instance, soon after finishing my Industrial Engineering degree, I started a data analysis course online. But another big aspect of continuous improvement is that your only competition is the measure of your own capacity. When you are focused on getting better and being the greatest you can be, you do not have the time to compare yourself to others if not only but to learn from their ways. In continuous improvement, the focus is your growth and your attention should be on the process that will eventually lead to result. This eliminates unnecessary pressure that could stop you from getting into action. I remember feeling at times pressured by the external 'noises' to settle for less and not try to achieve as much ambition as I intended to because they seemed unrealistic. When I wanted to be an Engineer, they advised to be grateful if I can get an assistant job somewhere which can enable me to pay for my bills. When I wanted to pursue a higher education abroad, they laughed. But because I was fixated on my growth and not the immediate result, I kept on pushing through which allowed for my increased in knowledge which resulted in increase in opportunities.

#### **4. The Road to Success may be a lonely road. Get ready to cope.**

You may hear a lot about the inventors and geniuses that were very depressive individuals, and you may wonder why when they seem to have figured out everything there is to know. That is because continually focusing solely on yours books, your invention, your computer could isolate yourself from the rest of the world. It's also worth keeping in mind that few people act on their ideas, work on growth and achieve self-actualization. Therefore the crowd whom a committed person could relate to and connect with is generally quite limited. That state of seclusion can be hard on the mind. In order to succeed while maintaining a healthy balance, it helps to join or interact with (groups of) [like-minded individuals](#), and to assist them occasionally with your skills or knowledge. While in school, I was involved in a lot of professional organizations and clubs, I became the president of some of them, which helped me build good networks. Later, these networks opened opportunities such as being a commencement speaker at the university, becoming a recruitment partner with a school, being the face of many campaigns and publications, etc. Networking doesn't have to be a waste of time or a distraction.

I hope this note comes as a reinforcement towards the path you envision for yourself.



**[Interview] In The Bubble Of Karen Kakou: "my parents never knew their ways to successful careers, I had to figure everything out by myself"**



**1. Quick (but effective) introduction of the personage and her achievements?**

I am [Karen A. Kakou](#). I am an Industrial Engineering graduate from the University of Wisconsin in the USA. Currently I work as a Manager in a renowned food manufacturing company, where I get to apply my analytical, continuous improvement and [people management skills](#) on a daily basis. My strength throughout my career has been to always seek the next thing to learn. I am a hungry for [anything books and new knowledge](#). With my background - my mom never gone to college, my father barely knowing the ways around for a successful career, and no one to guide me in general, I had to figure everything out for myself. This may be the reason why I have always been thirsty for more to learn. This helped me in defying what the expected future for me was - a passable future, to strive for a more excellent future in which I live up to my potential. Today I could not be any happier because I am living the path of what was meant for me, and I never settled.



## 2. What are your main goals in life?

To be the best of me. Although it's a very abstract response, it is what one must seek. Saying I want to become the Manager of so and so as a main goal is limiting. Once you reach that goal, what's next?! Hence, my [main goal in life](#) is to live up to my greatest potential. In doing so, I know I have the potential [to become a CEO](#) of a company someday, I also have the potential to live happy and healthy, so I am going to work toward being in that position and living that life. It's a holistic approach and a vision that incite continuous growth.

## 3. What tools and techniques do you use to accomplish things efficiently?

My favorite tool is Microsoft OneNote. It saved my life! Everything for me is organization. I know I can get a lot done but if I do not [plan well and prioritize](#), nothing will be done and I could get overwhelmed. OneNote helps me be very organized. It is my brain poured out on a screen page- I have written down my books draft, my objectives, my investment planning, my [entrepreneurial ideas](#), even my grocery list. All is well classified and [locked with a password](#).

## 4. Any recommendations for your juniors?

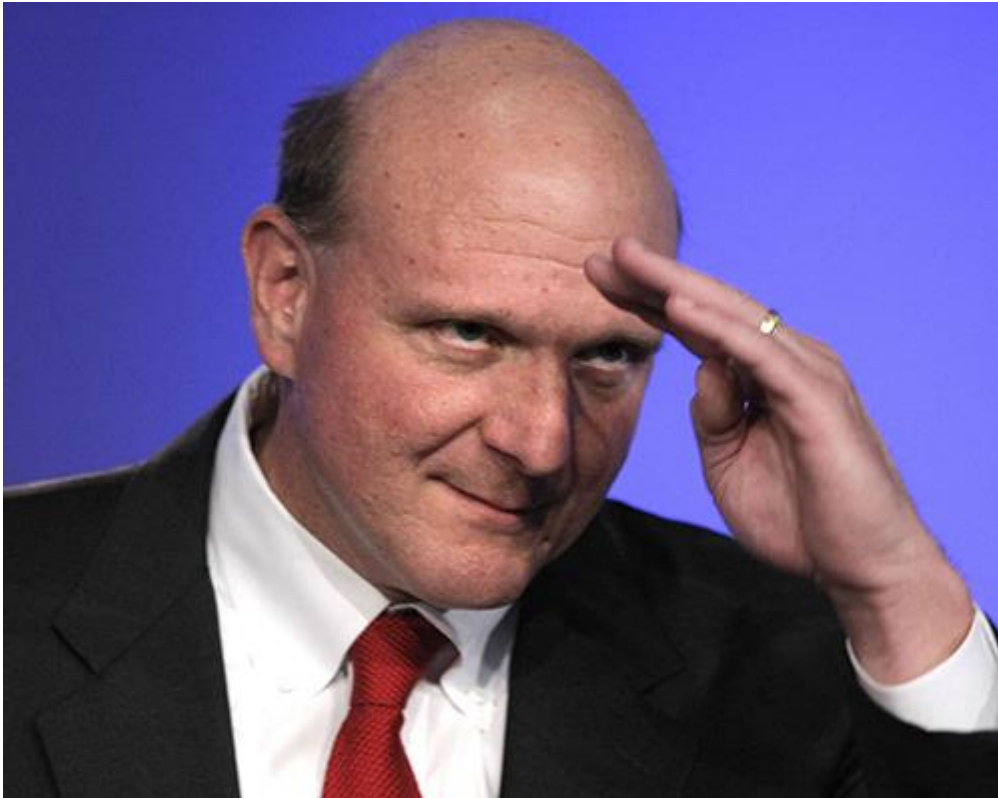
You got to check my website [karenkakou.com](http://karenkakou.com), of course (lol). I share everything I have learned throughout my life's journey, which I think would be a great source of [inspiration and motivation](#) for anyone who strive to be great. Then you got to keep reading and learning from the greatest. A few reads I would recommend are as follow: Malcom Gladwell's books, any books of Napoleon Hill, any major CEOs biography such as Elon Musk, Jeffrey Bezos, Steve Jobs'. "Rich Dad, Poor Dad" and "In, But Not Of" are some other great ones. I also like to listen to the Tim Ferriss podcast as well as watch videos from Mindvalley Academy.

## 5. What is the best way to contact you?

You can contact me through my website: [karenkakou.com](http://karenkakou.com)  
Or by email: [kkakouwebsite@gmail.com](mailto:kkakouwebsite@gmail.com)

## **10 Business And Management Skills To Own And Be Productive**

By [Vanessa Lecosson](#)



Management skills tend to play a very important part in business.

It is thanks to these skills that company directors can assign, assess and stay aware of the work completed by their teams.

They decide, implement actions and take control of each situation. Their skills and roles should necessarily match their career ambitions.

In this article, I suggest 10 skills in Business and Management to own in order to be productive and to be considered as a good Manager.

### **1. Define your top priorities**

Listing your daily tasks can be a pretty effective way to improve your productivity. Focus on what's important by regularly marking and sorting the tasks that are more valuable than others.

It is essential to assess which tasks should be completed first.

## **2. Manage time as required**

Starting early is one of the secrets that business leaders use to accomplish multiple tasks in the day. They focus on the "now" and avoid postponing.

Another thing about good managers is that they understand the importance of short, effective breaks.

## **3. Know when to say "no" and when to say "yes"**

What makes people say "yes" instead of "no" is the fear of losing an opportunity. In some cases, they say "yes" just to avoid seeming rude, so that they could be viewed as caring individuals.

Everyone has priorities in life. There is no harm in saying "no" to things you do not want to do or things that you are not interested in.

## **4. Always start with the end in mind**

Get into the habit of having a clear picture of the goals you want to achieve within a day, a week, a month, etc. You must have a plan before you commit to any project or task.

This will keep you motivated and focused on the most important parts.

Effective and timely strategies can lead to successful achievement of goals. It is important that you keep in mind that nothing will come to you unless you start making things happen. One action always leads to another. And it is the consistent succession of actions that lead to results.

## **5. Keep learning and improving**

Knowledge is drawn from everywhere. It is not only learned in the classroom, but is acquired through personal experiences as well.

One of the secrets used by many business leaders and business men and women is that they are constantly looking for hobbies that will allow them to learn/discover something new.

They are open to exploration and to those things that will allow them to grow mentally, physically and intellectually.

## **6. The efficient allocation of resources**

In business vocabulary, the allocation of resources is an economics concept which relates to the use of scarce resources, particularly the factors of production (labor, capital, raw materials) to satisfy the population's consumption needs in the short and long term. Resource management is used to get timely results and to optimize day-to-day productivity.

## **7. Use the right tools to stay productive**

Appropriate tools will allow you to improve your productivity level and maximize your efficiency in the job. Mobile phones, although often criticized, can obviously be very important tools for communicating with team members, co-workers, etc.

Some tools such as emails, conversation groups on some social networks (public/private) can help to fill the gaps related to communication within a company.

With the rise of new communication technologies, productivity applications are very useful and a powerful help for businesses. Mastering them will not only help you keep track of your agenda, but also increase your level of productivity.

## **8. Living in The “Now”**

Business leaders renowned for their productivity focus on things they have yet to achieve. But they always take breaks to enjoy life and the present moment.

## **9. Dealing with the unexpected**

Unforeseen events may spring from nowhere, at any moment and surprise anyone. Business leaders and managers are always prepared for the worst. And that's why they are always ready to respond quickly to any situation.

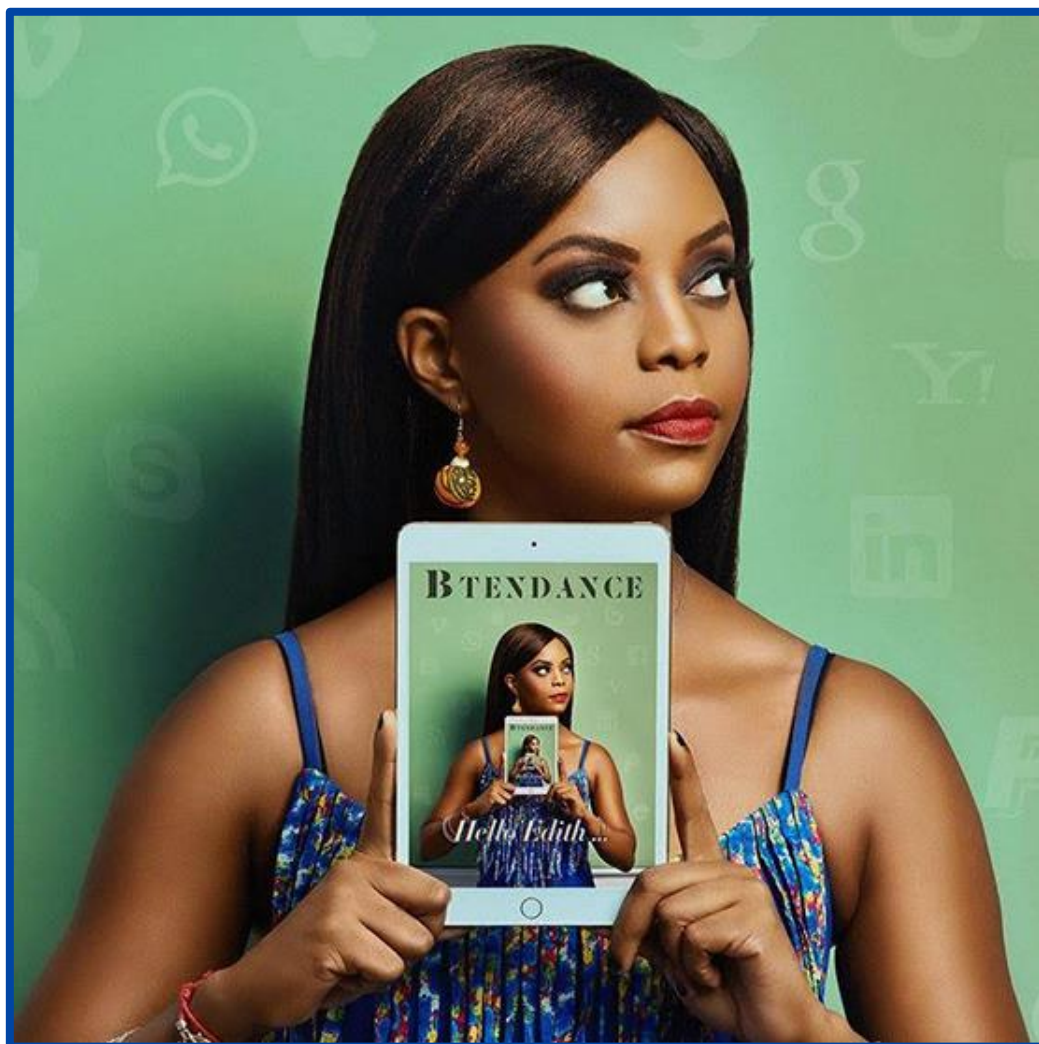
## **10. Staying organised**

Just like beauty, organization skills come from within. They will take root in you before outsiders may notice. Organized people know how to keep things in order. They are often in the habit of taking notes and making lists.

# Chapter 2

## Digital Marketing and Web Visualisation

Featuring Antoine Mian, Cyriac Gbogou, Edith Brou,  
Israel Yoroba, Jean Luc Houedanou, Jean-Patrick Ehouman,  
Nanda Seye, Vanessa Lecosson and Yehni Djidji



## A Quick Guide For New Bloggers

By [Necemon](#)



A new online "friend" was recently asking me: "What should be done to get a more lively blog or social media page? I mean, more visits, more likes; how to have some attractive blog and online profiles."

That question comes back from times to times. I don't particularly see myself as an expert in blogging or in social media marketing, but if I were to briefly sum up my humble opinion on the matter, that would be in 3 words: education, production and promotion.

### 1. Education

I couldn't detail everything you actually needs to know within 5 minutes, because there are lots and lots of elements that come into play. You must therefore continually be learning, checking out the best blogs that discuss [blogging, social media, copy writing, online marketing](#). Subscribe to them, read them as much as you can. Also reach out to those who are doing well in those areas, ask questions. They may not be able to teach you everything in a minute, so you could keep in touch and keep learning, observe their methods, take their feedback on what you're doing.

### 2. Production

Write often, regularly write, [write on topics that interest your readers and yourself](#). Writing regularly is a matter of discipline, writing interesting content is more about experience (it may take several tries). There again, you can do some research on what people like to read and get inspired by experts. But be careful not to copy them verbatim, you are trying to come up with something new, rather than publishing again what already exists.



### 3. Promotion

Keep your social media pages [updated, complete and professional, preferably with a graphic identity](#).

Find some people who are interested in what you have to say or write. Every time you write a new article, let them know through e-mail or their favorite online platform, take their feedback into account, to improve. Also, keep networking with other bloggers and writers in your niche. Establishing some good relationships could help when it comes to sharing content and other exchanges of services.

## What (un)motivates us When We Run a Blog

By [Jean-Patrick Ehouman](#)

Let's start with the dark part of this post. When you decide to [start running a blog](#), you are likely to fail before the end of the first three months. It is not easy to write, and moreover, to write things that interest others. Professional life and personal constraints (family, schedule, etc.) are often the first causes of slacking. In the same vein, one can also note the useless pressures and deadlines that we impose upon ourselves. For example, some may start by committing to writing at least one article every day at a given time. This is not really necessary because you need a minimum degree of inspiration to write. And when you don't have it, you should have a way to get it.

In 2005, when I was in my last year of study, I had created a personal site (to present what I do) and profile pages on sites dealing with [technical topics related to the development of web applications](#). Since then, I have tried to keep several blogs (often in collaboration with other people). But it's only since starting [this latest blog](#) that I've been able to find the rhythm, the motivation, the desire to publish despite the lack of inspiration.

This article you are reading right now derives from the fact that I did not have a specific topic but I wanted to see a new post on this blog. Hence, it's only natural that to start talking about what prevents or motivates us to write. As a result, I find myself looking on the web for elements that can help me fill-up this ticket. That's how I came across a 7-points list written by Jamie Wallace of "Live to Write - Write to Live". This list is intended to help cultivate real habits to keep a blog:

1. Find the time to write, otherwise create this time
2. Have an end goal
3. Avoid saying "I should..."
4. Start small. For example with notes of a few dozen words at a reasonable pace. or big tickets fairly spaced in time
5. Stay consistent
6. Measure progress and welcome it
7. Have fun

If you are lacking inspiration for [your next article](#), you can simply complete the list above with two or three sentences per point. As for me, I would say "talk about what you love" because there are always people who like the same thing and it is therefore a great way to make good contacts.

## 10 Marketing Tips For Promoting Your Next Blog Post

By [Vanessa Lecosson](#)



You have [created your blog](#) a few months ago. You regularly produce content to supply the "few readers you have" but that doesn't seem to be enough.

Your numbers are hardly increasing and you don't know how to boost your web traffic. The few Facebook and Twitter ads that you buy to promote your publications don't really help and you do not know what to do.

Today I would like to suggest 10 tips that will help you to better promote [your next blog post](#) and develop your audience.

### **1. Share your article to your contacts via email**

This should be one of the essential practices that you must take into consideration right away.

Once your article is ready, do not hesitate to share it with your contacts by email.

The goal is not to annoy them with long boring messages, but to stay brief and close to them. Summarize your article, add a few links towards your blog and the article in question, insert an illustrative image and you're done.

Do not forget also that the goal is to grow your traffic, and it also means proximity to your readers. So make sure you include custom formulas in your email, this could serve you well.

Some tools to help you manage your email addresses and send your different articles:

- MailChimp
- MailJet

## **2. Share your article on your social networks**

Sending emails is good, but your article will have even more visibility if you also share it on your social networks (Facebook, Twitter, etc.).

To that end, tools like Hootsuite, Buffer, Tweetdeck can be useful. You just have to schedule the publication of your article on these tools and let them share your content on your social pages.

## **3. Social bookmarking**

Use social bookmarking sites like Digg, Diigo, Reddit, Delicious etc to allow your readers to record and share the link of your article with a more or less wide/relevant audience.

## **4. Reviewing discussion forums and communities**

Do not hesitate to join online discussion groups, including Facebook, Google plus and LinkedIn and share your article. This will allow you to have more readers and to see the traffic of your blog grow as you take part in group activities.

## **5. Using curation sites**

Sites such as Scoop.It, Storify, are good places to search for ideas but also to share content. Utilize it you from time to time, they can get very useful.

## **6. Identify/Tag the influencers**

How would you feel if a blogger mentioned you in an article? Good?

Well, it's the same thing that happens if you mention influencers in your posts from time to time.

Every time you produce an article and quote an influencer inside, share your production with them. Send them an email, mention them on Facebook or Google+ or mention them in a tweet. Some of them will return the favor!

## **7. Find some fellow bloggers who can help you**

There are some bloggers out there that, from time to time, reach out to their peers in order to gather a wider audience. Once a week or once a month, they are willing to share content from other blogs with their readers so that other sources of information benefit their audience. You can try to get in touch with them and offer to publish your recent article on their blog. There will certainly be a counterpart, because everyone is trying to be more visible but it's worth a try.

## **8. Reuse your old publications**

Old articles can still be used to produce new content. Turn them into video or infographics, why not? A video can have more impact than an infographic, that can itself have more impact than an article to your readers. Do not hesitate to play the game by transforming your old publications into new productions.

## **9. A bit of advertising never hurt**

Some paid ads published over a period of time will always strengthen your actions. Set a monthly budget for your items.

## **10. Guest-posts**

This kind of campaign is not always easy, but when a blog gives you this opportunity, it is better to seize it. It is clear that they will (certainly) enforce a topic, or ask you to produce an article taking into account their given criteria, but once the article shared on his blog, you will be benefiting from exposure to their audience.

Most of us bloggers do not give more time to the promotion of our productions than to their creation. This is one of the reasons why our articles are not very visible.

I hope these tips will help you, as they have helped me, to spend as much time marketing your articles as writing them. Over time, more promotion techniques will become available, so stay up-to-date.

Knowledge cannot all be acquired in one day... Good luck!

## **Facebook Profile vs Facebook Page: What To Choose?**

By [Israel Yoroba](#)

Exactly one year ago, I tried an experiment. I disabled my personal Facebook account and kept only my Page to mark my presence on this network. The objective was to analyse the "pros" and "cons" of choosing one or the other. Here I give you some of those points.

Actually, I created my Facebook page because I had reached the 5000 friends limit. But, over time, I wanted to position "Israel Yoroba" as a brand. It was therefore necessary to communicate on a "professional page". There was then a distinction in my communication approaches. My personal information was on my personal profile and my professional activities were on my page.

### **Saving time**

Closing my profile allowed me to win, on average, 4 to 5 hours a day. When you have 5000 friends, you navigate between the information that appears in your newsfeed, the notifications of the groups to which you belong, or the update notifications of posts that you commented on, the chats, etc.

With my Page, none of this happens anymore. I do not have a newsfeed, I rarely comment on other people's profiles (except when I'm mentioned) and chat is more moderate. So I have a lot more time to stay focused on my work. Because, let's face it, Facebook can easily become a "weapon of mass distraction".

### **News**

For a journalist and someone who is watching the news, I miss out on scoops or interesting information (fortunately, my communication staff brief me on the most relevant parts). To have only one page is to lose sight of what is being said elsewhere. It is a kind of self-sufficiency that makes you "blind" and even a little "narcissistic" not to say "selfish". Because your communication is one-sided. You can publish, but you cannot see what others are doing on their wall or in groups. I admit that it's a bit frustrating, especially when you belong to many communities on Facebook.

### **Visibility**

The Facebook policy for pages is designed so that your posts are not instantly visible to your subscribers (43,000 of them, in my case). The conversion rate in terms of commitment remains relatively low (less than 3% on average in my case). Their goal is to encourage you to pay for advertisement in order to highlight the publication (and by extension, your page too).

On a profile, however, a publication is immediately shared on the news feed of all your friends. You benefit from a faster and much more consistent commitment (between 15 and 20, or even up to 40% on average).



## Positioning

Being on a page is like being a "public figure" (in my case). This requires much more rigor in the publications. Typing mistakes that may go unnoticed into a personal account are less tolerated on a page. I often get poked every time I let one out.

Communicating on a Page requires planning. Every weekend, I update my editorial calendar. I know what I will publish from the beginning of the week and what could be put on weekends. By following the local/international news and according to my activities.

I am surrounded by a team that works with me to succeed. Because despite all the experience I have, it's almost impossible to manage your own page. Rest assured, I am the only one who accesses it. I personally respond to messages and comments.

Upstream, I have my calendar validated. Each post is re-read by at least one person before being published. Unlike a profile where I can publish any way depending on my mood, my page respects a publication quota. No more than 3 publications per day (with few exceptions).

## What's the key point to keep in mind?

From this 12-month experience, I conclude that for an individual, a Page is only necessary when there is something to sell. The question does not arise for products, services, companies, institutions, associations...

Because on a personal account, you can also decide to make more professional publications. By setting rules, limits, and adopting a strategy.

## **It's All About The Selfie**

By [Nanda Seye](#)



The Selfie. Everyone knows everything about it... Or do they? How was it born? Where does it come from? Here I decided to dedicate an article to the Selfie story. I, a good disciple of the Web Lord, took a geeky pilgrim to write this article and preach the Good News of Technology to you, my dear readers...

### **So what is a Selfie?**

Have you ever [photographed yourself with your smartphone](#)? Your webcam? In front of a mirror? If so, you have already done a selfie. A Selfie (or self-portrait) is a photograph that one has taken of oneself, typically one taken with a smartphone and shared on social networks. Nowadays there is a growing trend towards group selfies, often effectively supported by selfie sticks.

### **Some of the milestones of the Selfie**

Everyone thinks they know everything about the Selfie, but how many know its actual history? Here are some important dates in the Selfie evolution:

**1839: first appearance of the selfie technique**

Robert Cornelius (1809-1893) took the first selfie in history. This chemist (and fan of photography) took a self-portrait in front of the family shop in Philadelphia using a daguerreotype (a photographic process involving silver plate created by the French photographer Louis Daguerre).



**1920: first group selfie**

Five photographers (Uncle Joe Byron, Pirie MacDonald, Colonel Marceau, Pop Core, Ben Falk) of the Byron Company (a photo studio in Manhattan) snap a picture of themselves on the roof of Studio Marceau in December 1920.



**1964-1965: best-selling record:** A self-portrait by Andy Warhol, which sold in New York at the May Post-War and Contemporary evening sale in Christie's, fetched \$38.4 million.



**2002:** the word "selfie" is used for the first time

The word Selfie appeared for the first time in September 2002 on an Australian online forum; a young man explained that he took a picture of himself to show the consequences of a fall down the stairs while he was drunk.

**2004:** The first #Selfie hashtag appeared on the Flickr site in 2004

**2012:** Rise of the selfie

**2013:** The word "Selfie" was added in the Oxford English Dictionary right between selfhood and selfish.

**2014:** "Selfie" was selected as Word of the Year at the 10th Word Festival in Nièvres where nearly 98,000 Internet users voted

**2015:** The word "Selfie" was added in Le Petit Robert 2015, along with "Hashtag"

## **Confidential: Social Networks And Me**

By [Yehni Djidji](#)



A few years ago, I could easily spend hours sitting in front of the computer, watching photos on Facebook or on Hi5. That's right, if people publish their photos, the whole point is that we can watch those indeed.

Sometimes I went on [profiles of people I didn't even know](#)! When I was asked what I was doing, I used to answer: I look at people's prosperity! Actually, I was a little envious!

I have so many projects running through my mind, [so many dreams](#) still unrealised, that sometimes I have these "down" periods, when I feel that everyone is happier than me and that I am running late! Not specifically late with regard to others, but mostly late on the vision that I have for myself!



I envied those who were able [to reach some of my dreams](#) and who don't always realise that they are privileged! I also envied these people who enjoyed their youth [in the fast lane](#) because, stuck in my perpetual "school-house-sleep" routine, I might not have had any interesting story to share with my children!

But at least, it's fortunate that I happen to be the prime confidant among my connexions. Therefore, for lack of better insight, I would still have the experience of others as a reference point!

And then one day I realised that no one ever photographs their bad times!

Who's ever taken [a picture of themselves](#) crying and posted it on Facebook? During a funeral, upon hearing sad news: break-up, failure, dismissal? Who's ever created a Facebook album with a title such as "Pictures of my Failure", where we would see their despondency upon the announcement of the unfortunate event, their parents in the process of scolding them, and why not, marks cards showing their poor grades? If you've ever seen anything like this on a social networking site, please send me the address!

I understood that the pictures I saw were just tiny moments in a lifetime! And I have plenty of happy moments too! We all have those!

More importantly, I understood that people immortalize these moments because they would like those to last for eternity even if they are fully aware that it is impossible !! Also, there is no point in moping around !!! If lamentations were a way of achieving dreams, we would know about it!

Hence, now when I have my down times, I wear make-up, I dress up chic and [I shoot very beautiful photographs](#)! Lol!

Stay blessed!

## You Have The Right To Publish Whatever You Want Online

By [Jean Luc Houedanou](#)

Dear teenagers and young adults who have been shocked by [this article](#), and who did not fail to let me know, **I owe you an apology**.

Indeed, I wrote without considering the two absolute truths of social networks, namely:

1. In front of a (webcam, smartphone, video) camera connected to the Internet (or not), you may [expose yourself](#) (literally or figuratively).

The results can range from the topless picture (with a duck face, optionally) to the sextape - for the boldest among them, not to mention the full disclosure of the finest details of their personal life.

But hey, the most important thing is to get "likes", "retweets" and "+1s".

I know that some of you have fallen into the illusion that their [degree of self-exposure](#) (in every sense of the word) is proportional to their acceptance in one or more social circles.

Anyway, I cannot understand it, but it's your life and you may do what you want. Yes, I am among those who think that, in theory, everyone is free to do what they want in their spare time.

It goes without saying that you can publish whatever you want on the [social networks associated with your name](#).

As a Web Consultant, I could explain to you that before hiring or recruiting you, employers will track your online activities on Google. But I guess you would reply: "Anyway, I would be [a \(web/social\) entrepreneur](#). To hell with the bosses". Then you would be forgetting that working under the orders of a boss is a good way to acquire personal discipline – i.e. getting up in the morning, arriving on time, learning management by objectives and results, learning to save money - which separates [professional entrepreneurs](#) from the rest, but I'd digress.

I could tell you that common sense would compel any sensible person to [avoid leaving compromising traces on the Internet](#). But it seems that the trend these days is to thoroughly reveal every nook and cranny of one's life on the Internet.

I therefore prefer to let you live in your wonderful world, where all actions are inconsequential. Especially when you take into account the second absolute truth of the web:

2. You know all your virtual friends and you can easily distinguish between your real friends, all of the jealous folks, and the perverts, or other individuals in need of thrills and sensational actions, as well as the bloggers/youtubers/scavengers looking for a buzz to promote their web platforms.

Well, I know it may be surprising, but all those people you have never met, and who subscribe to content updates that you publish here and there on the web might not be all trustworthy.

Indeed, many of the people described above are particularly inclined to share the [privacy of others](#) via Facebook, Bluetooth, unauthorized DVDs and any other means available.

But, you're not worried about that, since you personally know every single one of [your 1,200 followers on Twitter and 3,400 friends on Facebook](#), right? It's not like any of them could publish your twerk video on some shady websites, obviously...

So, dear teens and young adults living in a wonderful world where you can put everything and absolutely anything online for any reason, such as:

To draw attention to oneself,

To please a boyfriend/girlfriend,

Or, and I quote, "It's my body, my private life and I do whatever I want."

**Do as you wish.**

Do not listen to this grumpy old blogger... He does not even have any friends on Facebook.

And then it's not like the reality will eventually come and bite you once you turn 25 and you would spend your days erasing any detrimental content that might affect you, while you would then [be interested in LinkedIn and Viadeo](#).

Until then, I wish you all a good time on the Internet.

## **Caution! Cultivate Your Digital Identity Now!**

By [Edith Brou](#)



Did you know that you have a [digital identity](#)? Well, yes, you do. You don't have to request it from any legal authority (for now, at least). It already exists and it can either elevate you or harm you. How? Because from the day when you connected to the Internet, and filled out a profile by registering your first and last names, posted photos or opened an electronic mailbox, your Identity was [self-built without your knowledge](#). And it gets updated with each of your actions on your keyboard.

You can no longer go unnoticed because [you are certainly "referenced"](#), that is, by typing your name and surname, I may find you on the first page of the [Google Search](#) results. You therefore benefit from the natural referencing of this search engine (see [SEO : Search Engine Optimisation](#)).

However, there are [tools much more sophisticated](#) that allow you, by simply right clicking the mouse, to know in seconds, all the [public and private information](#) you have [dropped over the years](#) on "the Web".

In the worst cases, those we have taken their digital identities lightly (this amounts to misplacing their physical identity card), found themselves in very uncomfortable situations. For instance, it happened to that employee who got fired within a few months because his boss caught him happily smoking a big "joint" on his [Facebook profile](#).

In the coming years, be aware that, to access some sites, your digital identity will be scrutinized, and [for recruiters](#), it will be a very good way to have access to the profile of a [potential candidate](#) within a few seconds.

So now is the time to cultivate [your digital identity](#), and take care of it to avoid a very bad harvest.

## **Job On The Web : How Not To Get Hired**

By [Israel Yoroba](#)

Yesterday I posted an urgent advertisement to recruit a multitasking and bilingual intern. The required skills were:

- To know how to write and speak English.
- To know the Web environment and social networks.
- To be curious and to be available immediately.

Within 10 hours, I received 65 applications by e-mail or via my [Facebook page and profile](#). I received very good profiles. I will not talk about those.

I will focus on those who have been systematically rejected, since many of them were disqualified because they haven't realised that social networks are now a place where recruiters are "stalking" too. And if you dream of getting [hired for an Internet job](#), here is what you must avoid doing:

### **- Not respecting the instructions of the announcement**

Reply in comment while the instructions clearly state: "Contact me Inbox". The candidate who responds in the comments shows that, either he did not understand the instructions, or he does not care! The employer may want to discuss the finer details directly with the relevant candidates. He wishes to answer their specific questions, hence why the offer itself deliberately lacks precision.

### **- Replying to the offer with abbreviations**

I have seen all kinds of things. I read it all. Here is a good selection: "Im up 4 da internship" or "Mornin its fr the job". But wait, there is worse: "hey gimme more info re: ur stuff".

Editorial quality is required. By sending SMS messages, you indicate that, even without pressure, you are not able to write a sentence correctly.

I also browsed Facebook profiles. What a catastrophe. Abbreviations, posts littered with spelling mistakes. One or two typing mistakes are okay! But when it's all over the place, the case gets hopeless.

### **- Being away from the Internet**

Working on the Web obviously involves being on the Web. Some of the Facebook profiles (or Twitter accounts) haven't presented any activity for 6 months or more.



### - Leaving inappropriate traces

On some of the profiles, I saw some awkward links and/or comments (girls with suggestive poses/clothes, people (almost) naked, videos of people being beheaded, etc..., others unknowingly (?) click on pornographic links and this is displayed on their profiles.

### - The latecomers

Several candidates asked me inbox how to apply. I asked them to write a few lines of (their) motivation in English. Some of them replied later on the same night. As for the others... I'm still waiting.

I said it at the beginning. It should not be forgotten that our [activity on the Internet](#) (and specifically [social networks](#)) is scrutinized by employers. The CV and the cover letter are not enough anymore. The recruiter will check who you are, what you read, what you write. How you react. How soon. If you want to work, especially on the Web, do not make these mistakes.

**[Interview] In The Bubble Of Israel Yoroba: "I am president, elite serial entrepreneur, veteran blog trotter, catalyst for change in our generation"**



**1. Quick (but effective) introduction of the personage and his achievements?**

My name is [Israel GUEBO](#). Better known as [Israel Yoroba](#). I am a Media Expert and Coach, and specialist in social networks. I am also President of the Génération Innovante association. Furthermore, I am a member of the Bureau of the Network of Online Press Professionals of Ivory Coast, I am in charge of training and self-regulation.

I am a journalist by training and profession, graduated from the Superior School of Journalism of Lille, France.

From my experiences, I keep the most important ones. I trained the press service of the office of the president of the National Assembly, the communication department of the Presidency of the Republic and worked as coordinator of the digital communication for the Ministry of Youth.

From Print to Radio and Web, I have just over 10 years of experience with International Media. For 6 years (2008-2013), I had been a correspondent in Abidjan for the Web editorial of the French channel TV5. As such, I covered the 2008 US elections, the 2010 presidential elections and the subsequent military-political crisis in The Ivory Coast.

Between 2014 and 2017, I was chief editor of the "Studio Mozaik" Radio School, and later Head of the Hémisphère Media Production Office. That is the company that provides correspondence for TF1 in Africa and for France 24 in some countries of West and Central Africa.

For a decade, I have traveled around Francophone Africa to train state institutions. In Togo, Benin, Burundi, Gabon, Guinea... I train Senators and Members of Parliaments for the use of new media in institutional communication. This is taking place as part of the Noria program, initiated by the Francophone Parliamentary Assembly (APF), an institution for which I am a consultant.

I was also a web editor for the France 24 blog. In 2011, I did a brief stint at Radio France Internationale (RFI) on the program "L'atelier des médias". What I was most reknown for was the "Avenues", these local information sites with a presence in several African countries. Besides the Avenues, I launched, in 2013, the first Web Radio of information in Ivory Coast: Class FM ([www.classe.fm](http://www.classe.fm)).

All this earned me several distinctions: Best French Speaking Blogger in 2008 in Bonn, Germany. Best Blog of West African Journalist in 2009 in Dakar, Senegal. Harubuntu Awards Winner in 2011 in Brussels, Belgium. [Francophonie 35<35 Prizewinner in the Blog category](#) and Media Innovation Laureate in 2016 in Abidjan, these are the many distinctions obtained by the renowned Ivorian "Blog Trotter".

I saw my name in the Top 4 Most Promising Young People of Ivory Coast in 2016. This ranking was carried out by the Association of Lobbying and Public Affairs Councils of Ivory Coast (ACLAP-CI). In 2017, I created and supervised a media enterprise that won the second prize for the best Ivorian digital press company.

I am also the author of two books: "La Vie Comme Elle Va" (Life As It Goes, Balafons Edition, 2013), a chronicle that describes day-to-day life in the Ivory Coast, and "Compagnie Caïman" (Caiman Company), a novel published by Livresque Edition in October 2016.

## 2. What are your main goals in life?

My main goal is to be a catalyst for change in our generation. It sounds very idealistic, and maybe even utopian when we look at how our countries are evolving. But I remain convinced that each of us, one way or another, can bring his brick to build a different world, a generation with new values. And therefore, I do my best to teach and raise awareness around me. I believe in the power of sharing. It is the best weapon of development.

## 3. What tools and techniques do you use to accomplish things efficiently?

My first tools are my diary and my notepad. These are essential to better organise myself. Plus, I don't have to rely on electricity while using them. Every morning, I list everything I have to do on a page of my notebook. E-mails to send, phone calls to make, appointments (as scheduled on my diary), tasks to complete. Then I cross them out as they are done. At the end of the day, whatever has not been done is postponed to the next page. It's an effective monitoring tool.

Then there are all the Google Drive Apps, to [work effectively remotely with my team](#).

Last but not least, to communicate, there's nothing like Social Media. First my blog (which is being enhanced), then Facebook, Twitter and LinkedIn are my main communication tools.

## 4. Any recommendations for your juniors?

First advice, keep reading. Sometimes it may sound like a trivial expression. But nope. [It's important to read, read a lot](#). Not just on the internet, but read books, magazines, newspapers. This develops our intellectual faculties, and our culture.

Further, it's great to use search engines thoroughly. Google, in particular. This is an incredible opportunity for our generation. With this tool, you can learn anything.

Finally, my last advice is an attitude in life, it's about remaining humble. Because life is unpredictable, and today's victories must not be considered as final, but they should be a source of motivation to achieve a little more.

## **5. What is the best way to contact you?**

The best way is [by mail \(isyoroba@gmail.com\)](mailto:isyoroba@gmail.com). I read them all the time. Then, the comments sections on [my blog](#). On [social networks](#), I respond to specific messages. However messages starting like "yo", "sup", "baby", etc. will remain unanswered (lol)!

## Did You Just Say Crowdfunding?

By [Nanda Seye](#)

During the month of August, I was helping out the [TechOfAfrica team](#) with the organization of an [afropreneur afterwork](#) event in Abidjan. So I was talking with a friend about that session, I was explaining that the purpose was to present the Afropreneur project and to promote the related crowdfunding campaign. But, as soon as I mentioned the word "Crowdfunding", my friend immediately shifted to a shocked face.

"Huh? What's that, Nanda? Some of your geeky stuff again?", she asked.

And then, I realised that what may seem obvious for some of us is not always that clear for others. This situation is amplified by the fact that many are not particularly comfortable with the language of Shakespeare. So I wrote this little article just to clear things up for everyone. Happy reading.

### **So what is crowdfunding?**

Crowdfunding is a collective financing practice that is executed via the Internet to help start-up projects. It connects project leaders and Internet users who want to invest in these projects.

### **Oh, is it? But how does it work?**

It's simple, the operation can be summarized in three entities:

Project Managers <-> Web Platforms <-> Internet Users (Investors)

#### **A. Project leader or startup:**

You may have a project that is important to you, that you want to achieve. Instead of turning to the banks for funding, you can opt for a platform dedicated to this matter, where you would submit all the project details. This allows you to:

- Give visibility to your project
- Receive donations/loans
- Find potential partners/sponsors

#### **B. Web Platforms:**

These are websites dedicated to this purpose, on which one can run a fundraising campaign for their project. There are 4 types of platforms:

Donation platforms: the user (i.e. the investor) provides financial support to the project without receiving any reward.

Platforms with counterpart in kind: the user receives a consideration in kind

Loan platforms: Internet users grant a loan to project leaders

Direct investment platforms: Internet users invest directly in the capital of a company by acquiring a stake in the project or by becoming co-producers. In exchange, they earn the right to acquire financial counterparties in case of commercial success of the project.

In order to sustain themselves, the platforms keep a commission of 5 to 10% on the amounts collected.

### **Hey, that's interesting, but what are the platforms?**

There are quite a few platforms, some of the most popular are:

- KissKissBankBank
- Kickstarter
- Ulule
- Indiegogo

### **C. The Web user:**

This is the one who is interested in the project and decides to offer you some donations or loans, you may refer to him as your "Angel" investor (no pun intended lol).

That's all folks, I hope the concept of crowdfunding is now clear for you, and that you too can impress your friends lol and use it for your own projects.

One last piece of advice: a successful crowdfunding campaign involves choosing a platform that is in line with your project, a good presentation of the project and, above all, an excellent marketing strategy. Good luck.



**[Interview] In The Bubble Of Nanda Seye: "it was essential to be very strategic and organised in order to reach the defined objectives"**



**1. Quick (but effective) introduction of the personage and her achievements?**

I am [Nanda SEYE](#), Ivorian, in charge of private sector programs in a training firm. I am completing a training in Computer Engineering Option Networks and Systems in a school in Ivory Coast. I have experience in Digital Marketing, Project Management and Computer Network Administration. Also, I have [a blog on ICT](#). What sets this blog apart is the fact that it educates and informs on ICT topics through humour. Hired as a Community Manager by a coworking space from 2013 to 2014, I managed to bring together the different ICT communities around this space and to organize several events that helped in establishing the reputation of the company. This was my greatest accomplishment because JokkoLabs was the first collaborative space in that country; the concept was therefore unknown to the public. It was essential to be very strategic and organised in order to reach the defined objectives.

**2. What are your main goals currently?**

I have always had a particular interest in project management. So my goal is, God willing, to pass several certifications in this area, including the PMP (Project Management Professional).

Besides that, I plan to produce a Comic Strip because I really like the world of Comics. So I'm currently working on a comic book project that will later be derived into animated videos and video games adaptations. For this purpose, I currently work with a caricaturist and a graphic designer. After releasing this comic book, I will be keen to collaborate with folks specialising in [video game development](#), in 2D/3D videos and other videos to carry out this project. Specifically, people who somewhat share values such as respect, surpassing oneself, perseverance, humility, simplicity, rigor, creativity, professionalism and, most importantly, humor.

### 3. What tools and techniques do you use to accomplish things efficiently?

My favorite software:

- Project management : Ms Project, GanttProject:
- Digital: Hootsuite, Photoshop, LikeAlyser, Giphy, Canva, Bitly, Social Bakers, MailChimp, Social Shakers, Google Adwords, Google Analytics

### 4. Any recommendations for your juniors?

Websites :

- [gestiondeprojet.pm](http://gestiondeprojet.pm)
- [abilways-digital.com](http://abilways-digital.com)
- [openclassrooms.com](http://openclassrooms.com)
- [1min30.com](http://1min30.com)
- [webmarketing-com.com](http://webmarketing-com.com)

Blogs :

- [jeysmacademie.net](http://jeysmacademie.net)
- [blogdumoderateur.com](http://blogdumoderateur.com)

Books :

Religious:

- Good morning, Holy Spirit: Benny Hinn
- Understanding the Divine Direction: David Oyedepo
- The hippopotamus in the garden: how to hear the voice of God in the 21st century: Alain Bouffartigues

Personal development :

- The alchemist: Paulo Coelho
- Think and Grow Rich: Napoleon Hill
- Attitude Is Everything: 10 Life-Changing Steps to Turning Attitude Into Action: Keith Harrell
- The Power of Positive Thinking: Norman Pearl
- The Leader In You - How to Win Friends, Influence People and Succeed in a Changing World: Dale Carnegie

Digital Marketing :

- The marketing plan - digital communication: Denis Pommeray
- The art of the digital war: Caroline Faillet

#### Project management

- Project planning technique: Gilles Valet
- The project manager's toolbox: Jérôme Maes and François Debois

#### Life tips (quotes):

##### David (The Bible)

Psalm 146: Do not rely on influential people: they are only men, they are unable to save. As soon as they blow their last breath, as soon as they return to the ground, their projects are buried with them. Blessed is the one whose help is the God of Jacob, who puts his hope in the Eternal God.

##### Ralph Waldo Emerson

- Do not go where the path may lead. Go where there is no path and leave a trace

##### Abraham Lincoln

- The best way to predict your future is to create it
- Always keep in mind that your decision to succeed is more important than anything else
- You can fool a few people all the time; You can fool everyone for a while. But you can not deceive everyone all the time.

#### **5. What is the best way to contact you?**

I would say, via my Facebook profile: [Nanda SEYE](#)

## Exploring Some Advanced Visualisation Techniques

By [Necemon](#)

This study is about the production and the description of some advanced visualization charts related to the higher education in South West Wales. Some [actual data](#) had been made available to complete this task and the objective is to get some meaningful and interesting insight about the data.

The [data files](#) are from a Management Information Officer in the Statistics Unit in conjunction with the Planning and Strategic Projects Unit (PSPU), both at Swansea University.

The tool used is [IBM Many Eyes](#). It's a web service that lets user upload data sets and can turn them into various visualizations. There were a lot of choices but only advanced visualization types (especially maps) had been highly recommended rather than basic charts. Further, graphs that show rises and falls over time had to be avoided as the provided data was not time oriented. So the five visualization types demonstrated here are block histograms, bubble charts, maps, matrix charts and tree maps.

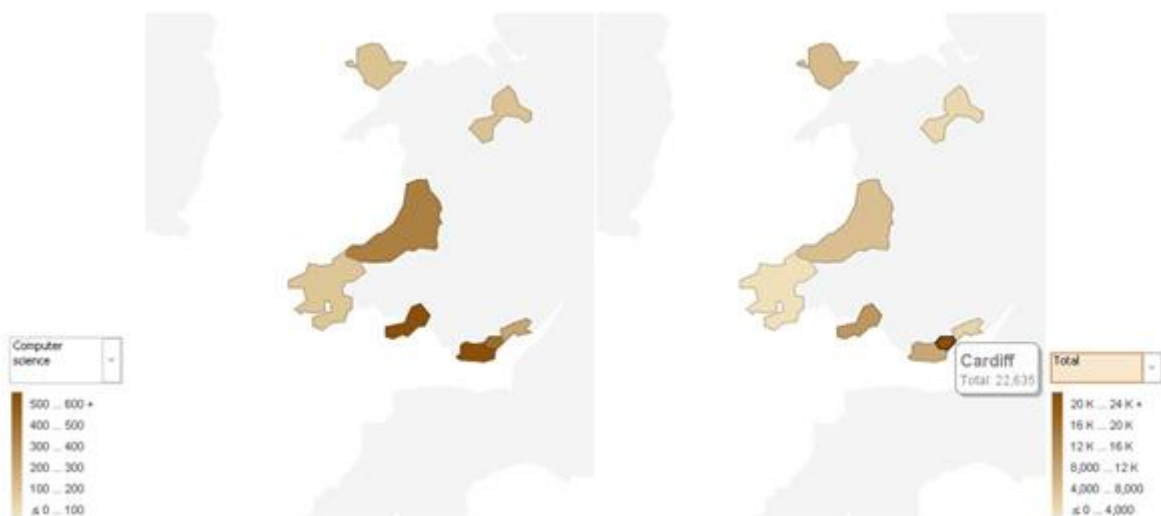
### **1. Wales Map**

Maps are really intuitive when it comes to show a comparison based on the location. In Many Eyes, the Wales map is slightly inaccurate but it allows us to get an interesting overview of the most popular regions where people go to study.

The following maps show the most frequented areas for each field of studies for the full time undergraduate students in Wales. The original data was dividing students according to their institutions. However, for the sake of mapping, those institutions had to be aggregated according to the respective cities they belong to. The darker an area is, the more it's frequented for that field of study and inversely, the lighter a region is, the less it's frequented. This doesn't apply to the gray area, which represent the zone for which we don't have any data.



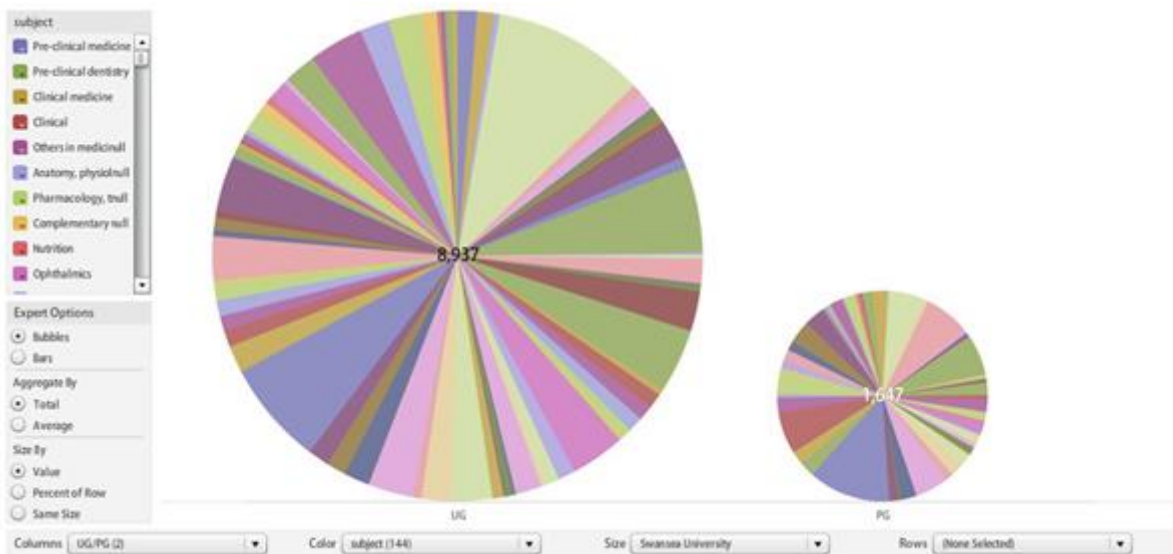
Now let's consider a particular case, the field of computer science. We can clearly see that Swansea is one of the most popular destinations to study computer sciences in Wales. However, when we consider all the subjects together, Cardiff appears to be the most popular choice overall.



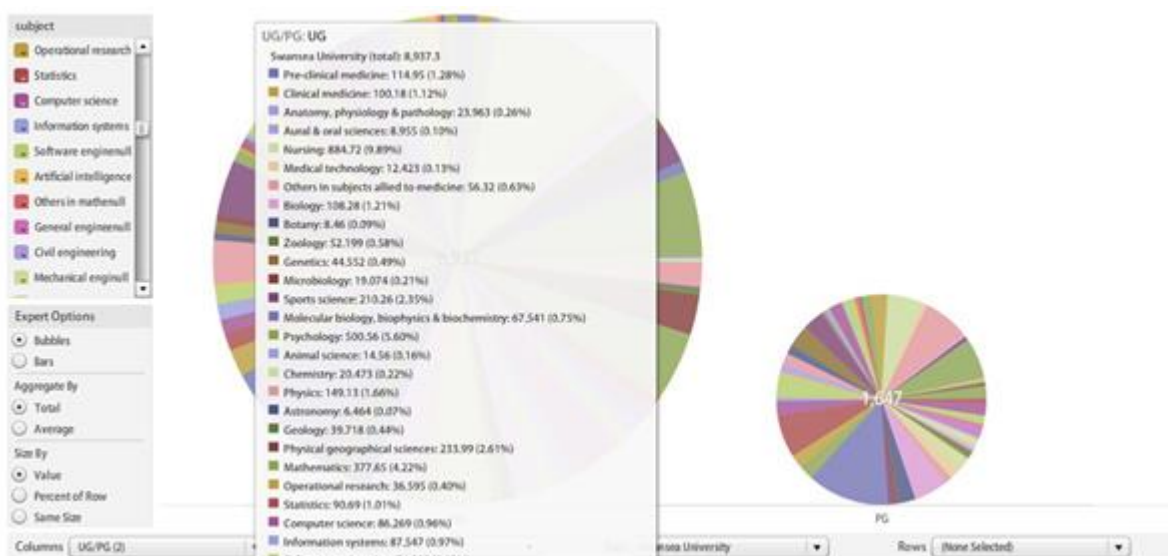
## 2. Matrix Charts

Matrix Charts are typically used to display multidimensional data in a grid. It's very useful when we have many criteria to describe the same data, for example we use matrix charts to describe students based on the subjects they are studying, their institutions and whether they are undergraduates or postgraduates. Our grid down here offers a single row with 2 discs: UG and PG and within each disc, we can see the proportion of each subject relatively to others. The first result that strikes is that there are much more undergraduate students than postgraduates students.

Another obvious result is that there is a wide range of subjects studied in Wales.

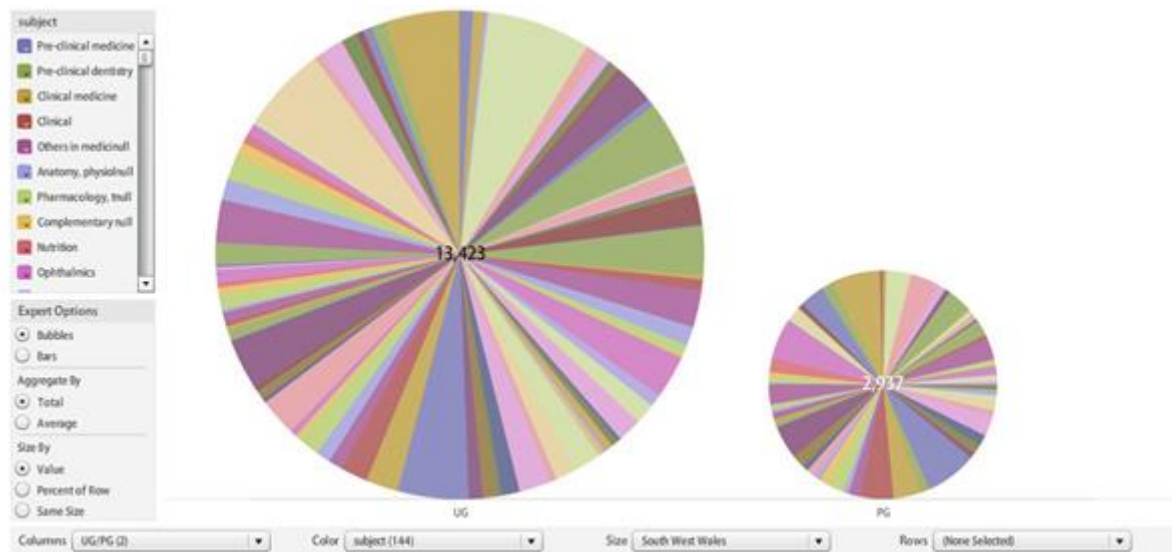


Below we see that Nursing (9.89%) and Psychology (5.6%) seem to be quite popular among undergraduates in Swansea university whereas Computer Science only represents 0.96%.





The matrix chart below represent almost the same data but only at the South West Wales level.



### 3. Treemaps

Treemapping is a method for displaying tree-structured data by using nested rectangles. Treemaps display hierarchical (tree-structured) data as a set of nested rectangles. Each branch of the tree is given a rectangle, which is then tiled with smaller rectangles representing sub-branches (Wikipedia definition). For our purposes, we use treemaps to visually compare the differences in number of students among areas of studies at the full time postgraduate taught level. We subdivide the students in the fields of studies according to their origin (UK, EU, Non-EU). The diagram obtained gives us some interesting insight.

For example, we notice that a lot of non- Europeans comes to the UK for business and administrative studies. Further, for most of the subjects, the number of non – Europeans exceeds the number of other EU students.



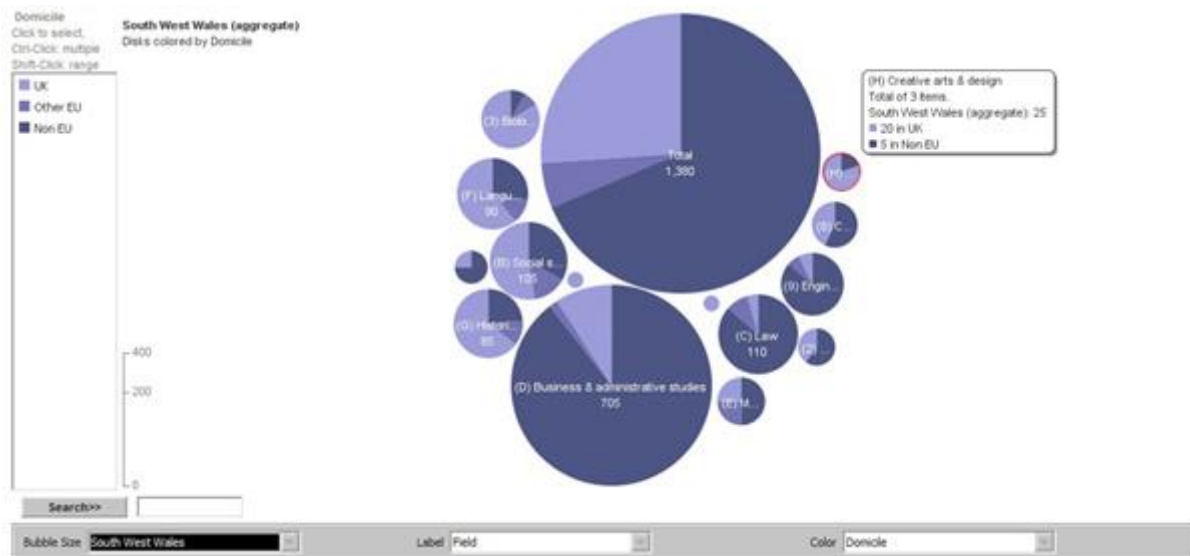
What we observed at the UK level is quite similar to what we see when we consider the Wales only. However the colours tend to imply that the percentage of international students is less in Wales.



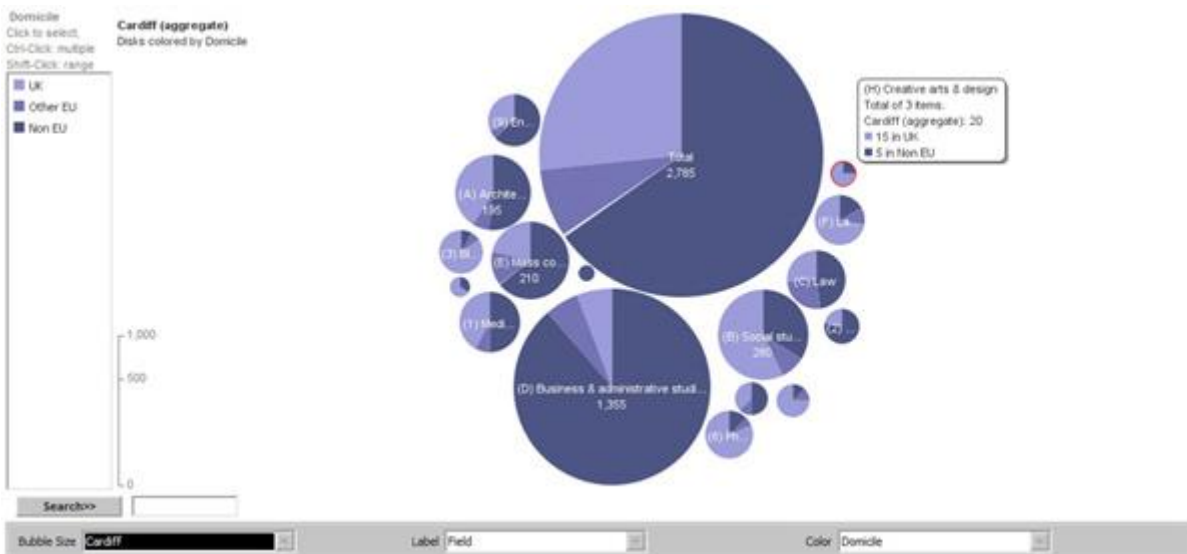
#### 4. Bubble Charts

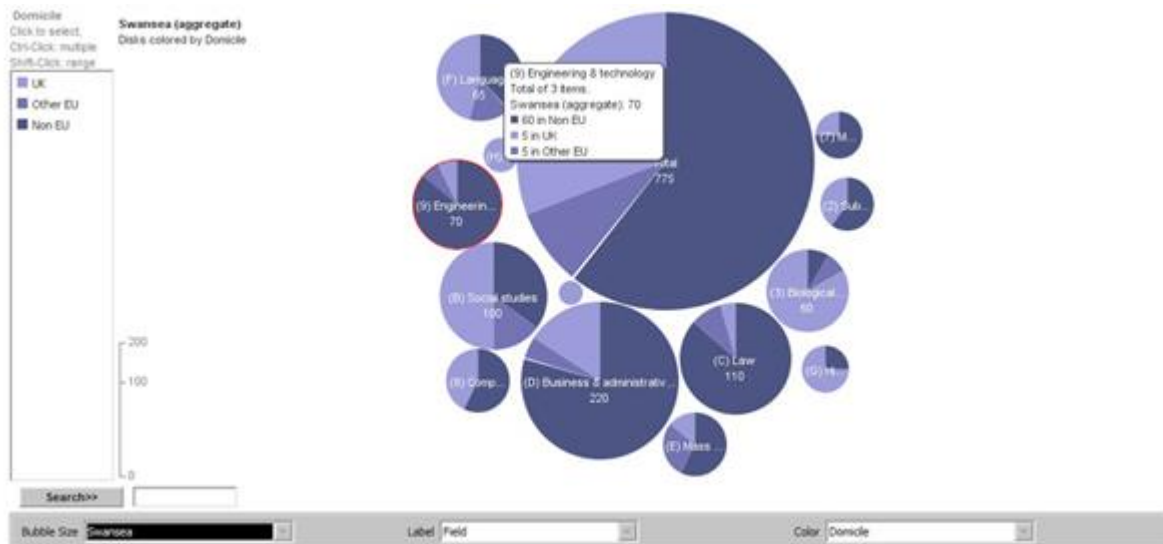
Bubble Charts are another intuitive and aesthetically pleasing to visualize “who is studying what”. Here again we are studying the full time master students.

The figure below confirms the wide proportion of Non EU students in South West Wales. And here again we see they are highly represented in Business and Administrative studies. Anyways, Business and Administrative studies apparently represent the most popular field of study.



The two following Bubble Charts display the cases of Cardiff and Swansea. The trend is quite the same with the rest of the South West Wales but we can notice that compared to Cardiff, Swansea presents a smaller proportion of Business and Administrative Studies students. However, in Swansea, Engineering and Computer Science are quite more popular than in Cardiff.

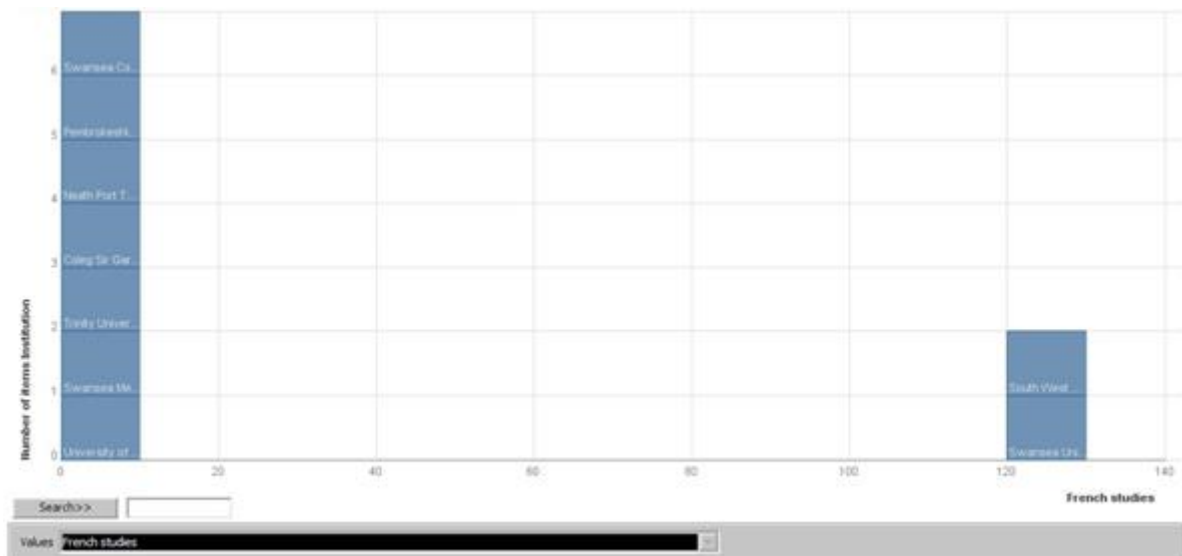




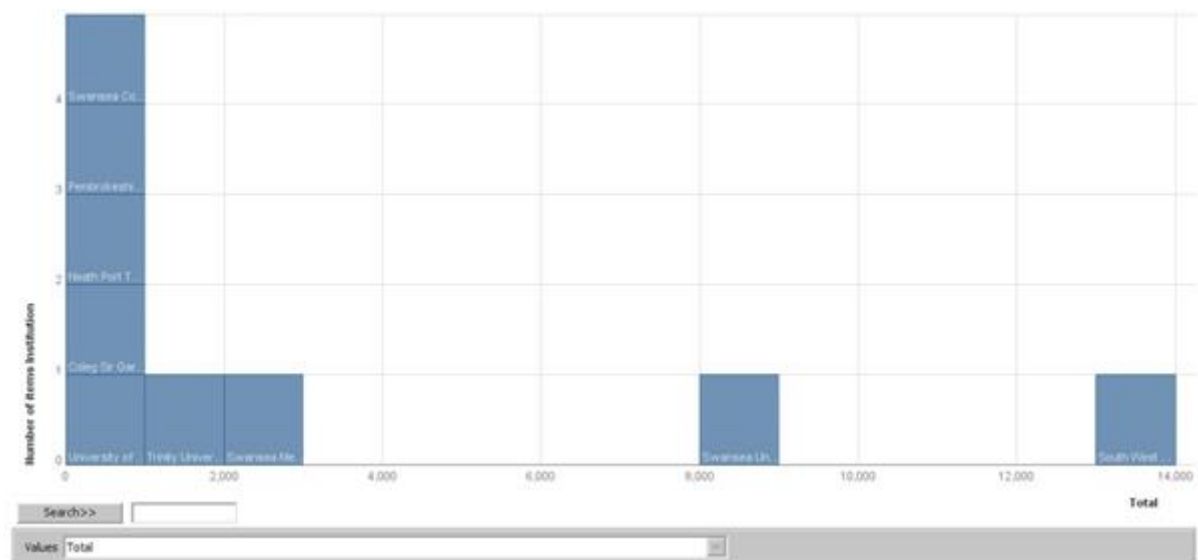
## 5. Block Histograms

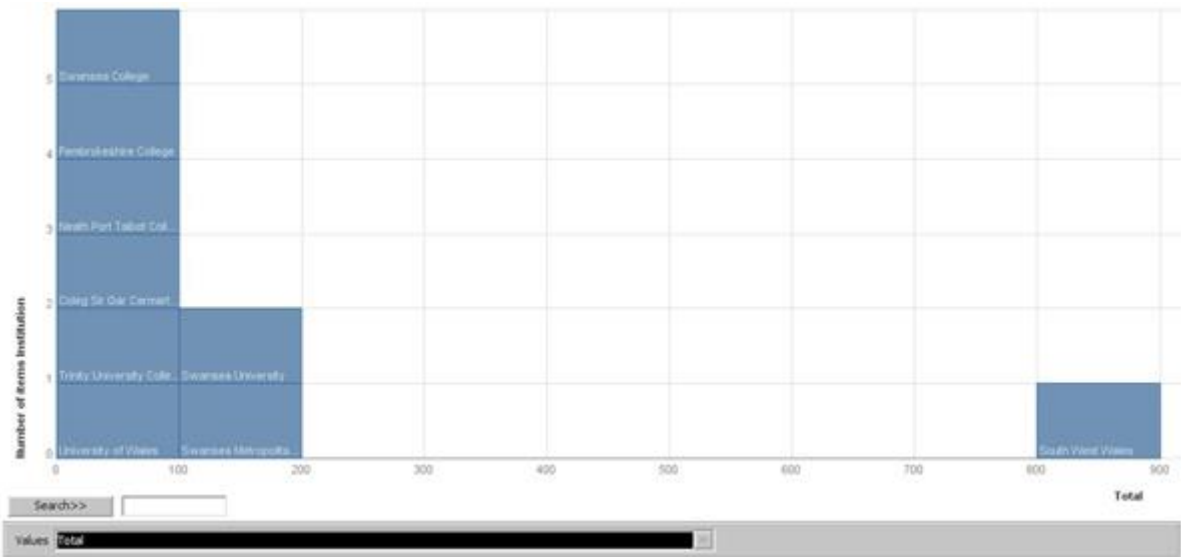
A block histogram lets you see the distribution of numeric values in a data set. The x-axis is divided into "bins" that correspond to value ranges. Each item in the data set is drawn as a rectangular block, and the blocks are piled into the bins to show how many values in each range. (Many Eyes definition)

The figure down here shows the populations of full time undergraduate students based on their institutions. It clearly comes out that Swansea University is by far the favourite destination for studying French with more than 120 students. All the other institutions only have less than 10 students for the French Language.



We then observe a comparison between full time UG and part time UG students (again based on their institutions). The first thing that comes out of those visualizations is that there are much more full time students than part time students (which makes sense). Further, while Swansea University seems to be quite popular among full time students, the population of part time students is pretty much equally spread among the institutions.







## Data Visualisation : Thematic Maps VS Bubble Charts

By [Necemon](#)

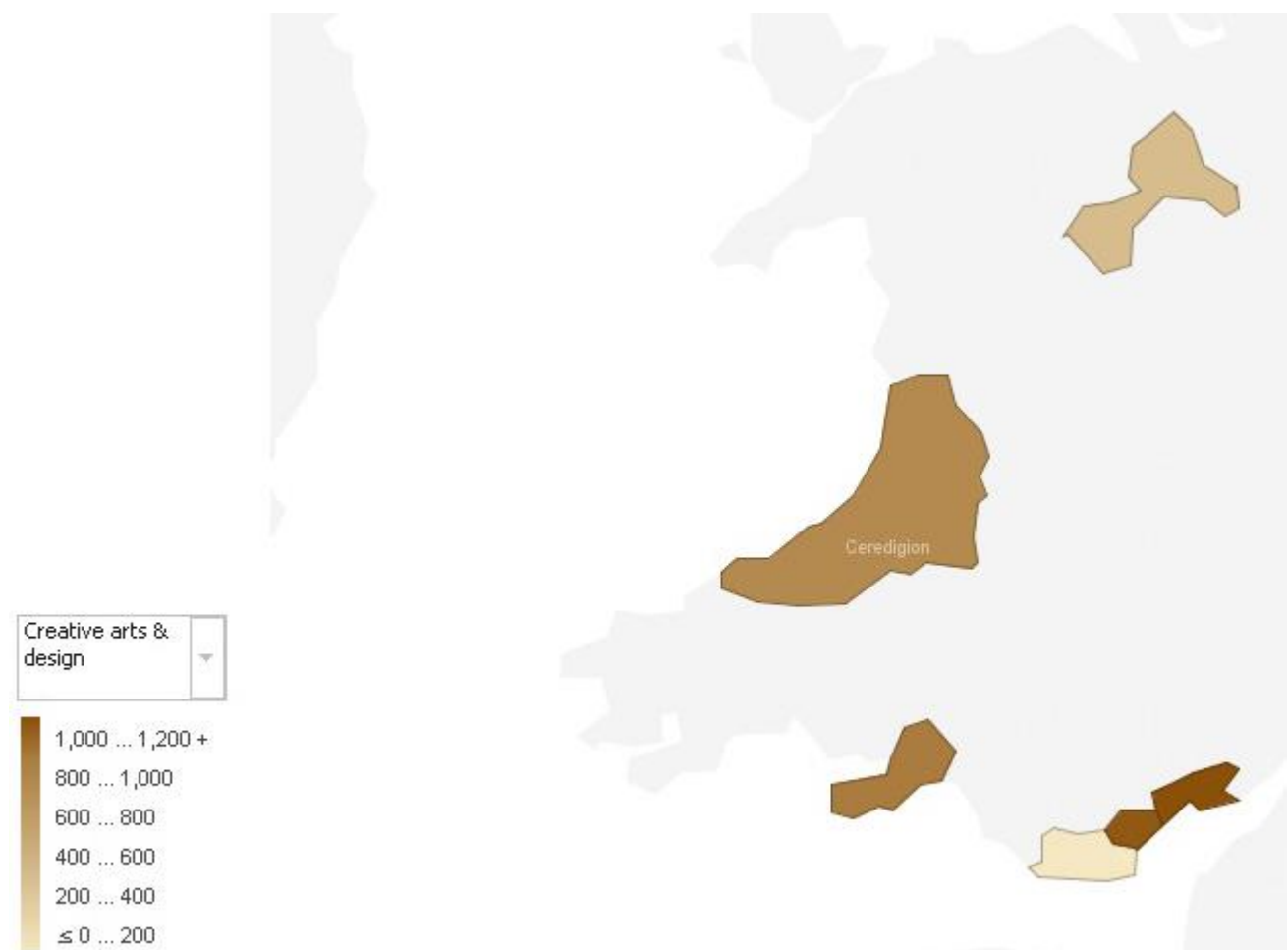
This is a comparative study of two particular types of visualization techniques in detail.

We chose to study the map-based visualization and the bubble charts because they had been the most fascinating visualizations [throughout this study](#). We are going to compare those two types on many factors.

### Defining the visualizations

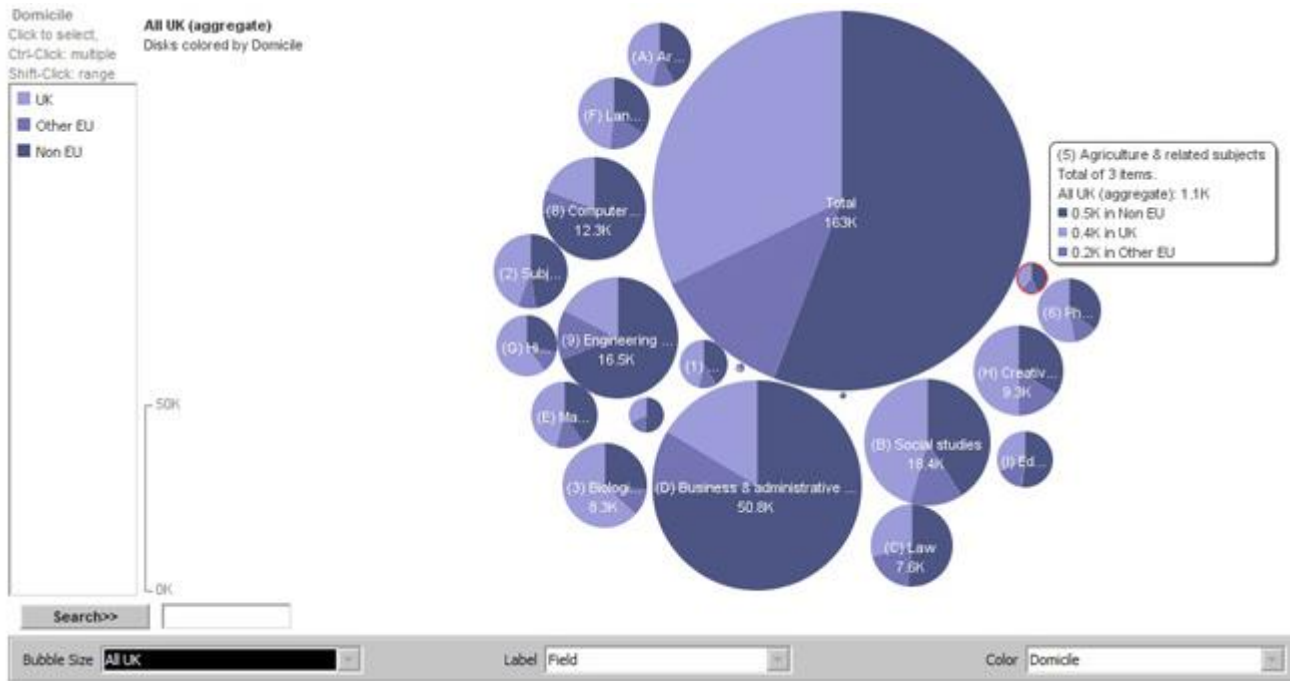
A thematic map is a type of map or chart especially designed to show a particular theme connected with a specific geographic area. These maps can portray physical, social, political, cultural, economic, sociological, agricultural, or any other aspects of a city, state, region, nation or continent.

An example of map is shown below. It shows the repartition of undergraduate students studying creative arts and design in Wales.



A bubble chart is a type of chart where each plotted entity is defined in terms of three distinct numeric parameters. Bubble charts can facilitate the understanding of the social, economical, medical, and other scientific relationships.

An example of bubble chart is shown below. It shows the repartition of master students, in the whole UK, according to their fields of study and origins.



### Expression of the information

Both visualizations can convey the information pretty well to the viewer but the map based visualization is limited to spatial patterns. As mentioned above, it can be used to describe data related to places (cities, states, regions, nations or continents). On the other side, bubble charts can convey information of any nature. Further, they can convey many different kind of information at a time.

### Discovery of deviation, hierarchy, relationship and association

The maps show well the relationships between places and the association among regions. They are also good for comparing 2 places based on a specific criterion.

The bubble charts are good for comparison too since they allow to show many different things at a time. Further, the circles can contain inside them some more data. This nesting can help understanding hierarchy when it occurs.

### **Depiction of data at different scales**

Bubble charts are especially useful for data sets with dozens to hundreds of values, or with values that differ by several orders of magnitude. They can legibly show data values that differ by a ratio of 100,000, and can display hundreds of individual values at once (according to many eyes).

However the maps are not always efficient to display huge and/or complex data. The problem is that some regions on the map can be very small and they just don't fit to add some complexity or some advanced charts. It's true that we can zoom, but if regions are too far apart from others, zooming somewhere make us miss another important part of the data. We can then say that maps don't really provide much level of details.

### **Ease of use and understanding**

Maps are really intuitive, easy to learn, to use and to understand, even to lay persons. Actually we had been using maps for ages, at primary school during the geography lectures for example.

Bubbles charts on the other hand are quite unusual to lay persons. Further, they can quite confusing and difficult to understand when they contain a lot of data.

### **Look and feel**

Both the visualizations are aesthetically pleasing to visualize, especially when we use colours to demonstrate the differences between two categories. We can also use bubbles on maps, and the sizes of the bubbles make obvious which part of the map is the bigger, greater, etc.

### **Conclusion**

Thematic maps are great tools that communicate the information well to everyone. However they only apply in a context of space. Further, they hardly deal with complex data.

On the other hand, bubble charts are a bit more difficult to handle but they are really powerful, they can handle any kind of data whatever the scales.

And also some data may not be displayed properly if not spelled as in the map.

So maps are good to represent simple and spatial data whereas bubble charts are better for representing the most compound data.

## A Practical Guide On Open Data From The World Bank

By [Cyriac Gbogou](#)



If you have ever checked the weather online, used the navigation function of your smartphone to find an on-duty pharmacy, or calculated the road expenses of your municipality, they you have already used "Open" Public Data. But maybe you didn't know it... For a long time, it was difficult (or even impossible) to access this information from the public authorities.

This concept was born from the belief that the huge amount of information systematically collected by public administrations should be accessible to all citizens. In the late 2000s, states and their various services began to grant access to these resources to a greater number of users. The first government open data policies were introduced in 2009. Currently, more than 250 national and local governments, nearly 50 developed/developing countries, and institutions such as the World Bank and the United Nations have launched initiatives to release the data, and these are growing every year.

Data is considered "open" if everyone can use, reuse and redistribute it freely, free of charge, for any purpose and without restrictions. Many data sets are published on government websites, but most of them are only meant to be viewed in isolation, and are not reusable for other purposes. Open data must be reusable, that is, downloadable in an open, software-readable format, and users must be legally permitted to reuse it.

The availability of large-scale, easy-to-use data offers significant benefits: it can help streamline public services, foster economic opportunities, encourage innovation, improve public safety, and reduce poverty. As the benefits of open data reach an ever-increasing audience and more effective uses are being invented, governments and institutions around the world are eager to launch new Open Data programs or develop existing initiatives. It will take time to fully understand the complexity and enormous potential of open access to data, which is inseparable from the free license environment. But Open Data is still in its infancy, and best practices and communities are just beginning to emerge.

## About this Handbook

As developing countries, as well as a growing number of other entities and organizations, embrace the Open Data technology revolution, the World Bank has designed this handbook to support their efforts. Based on online tools and knowledge, this guide gives them the advice they need to create their own strategies and platforms for accessing public data.

One way to look at Open Data initiatives is to view them from the economic point of view of supply and demand: data sets are goods; the data users are consumers, they are the ones who generate the demand for data; and government agencies and other data sources produce the offer. This vision is the focus of this guide and of the approach it proposes to structure an open data initiative.

In addition to this introduction, the practical guide includes seven sections:

1. [What you need to know](#): A brief presentation of the key aspects of the openness of public data and its uses, illustrated by examples of initiatives at different levels; This section is for everyone, especially inexperienced users or those discovering Open Data.
2. [First Milestones](#): Overview of the tools governments need to undertake an open data initiative; this section will be particularly useful for managers, coordinators and teams of administrations in charge of Open Data.
3. [Technology Options](#): Synthesis of key technical questions regarding the development of an open data catalog; this section is specifically intended for IT specialists.
4. [Demand and Public Engagement](#): How to create communities of data users and promote culture and data literacy among local, national and regional actors? How to advance Open Data concepts by soliciting partnerships and the media in particular? These issues will be of particular interest to public bodies, civil society organizations, technologists and journalists.
5. [Data supply and quality](#): How to manage datasets, locate data and ensure their quality, relevance and accuracy? This section is specifically for those who produce these data.
6. [Readiness Assessment Tool](#): Presentation of the methodological tool developed by the World Bank to diagnose the actions that governments must take in a given country or community to launch an Open Data Initiative. This section will be particularly useful for public officials who are committed to opening the data but do not know where to start.
7. [Technical Assistance and Funding](#): An overview of the main technical assistance and financing resources provided by the World Bank and other organizations. This section is primarily intended for the public authorities of developing countries.

## **Educational Uses Of Internet Content Curation**

By [Antoine Mian](#)

As a marketing tool, how can "Content Marketing" be used as teaching tool in higher education? We tried to provide an answer to this concern by exploring [François Arnal](#) works.

"Curator" is a term used in museums. It refers to the person who chooses artworks for an exhibition, the curator is called upon to make a choice, a comment and an introduction of some existing works. He is the link between the artist (or the cultural heritage) and the public. He is an expert. The curator on the Internet, after defining the topics of his technological watch, may choose to work from a curation platform allowing him to automatically or manually harvest, filter, editorialise and share some online content.

Web 2.0, which now amounts to an audience of more than one billion people, influences our didactic practices. What is changing today in didactic postures is the virtual presence of a student on the web who, in addition to having a teacher, can interact with millions of people for free. This situation forces us to move from a one-way pedagogy to a participatory and contributory pedagogy. Indeed, through social networks, we even end up into situations where a student get in touch with people having a high level of expertise. The teacher should therefore recycle "Web-marketing" strategies in order to build a digital didactic strategy that allows the acquisition, and above all, the sharing of information and knowledge.

Thus, if curation tools are originally intended for businesses and the general public, digital monitoring tools can nevertheless be used by teachers and learners.

The dual objective for the teacher is, on one hand, to thoroughly understand social networks, access to information and digital identity design, and on the other hand, to gain access to information, and therefore to knowledge.

A first step may be for the teacher to acquire these tools and to make them available to his students through an effective and operational digital environment.

A second milestone is reached when students access the digital tool not as mere consumers, but as producers of knowledge. The student becomes an actor. And curation allows them, among other things, to learn and control the flow of information, to classify them, to package them and to offer them to others in a sharing community and in a context of mutual exchanges. His interest in this theme makes him an 'expert' in the field of information sharing or in a specific area of knowledge.

Through an elaborate content strategy that allows the student to search for information, to grasp it, to modulate it and to relay it while issuing a critical opinion; the age of vertical knowledge (from the teacher above, to the student below) is replaced by another mode of cognitive transmission: horizontal or reticular. It's a horizontal transmission because the hierarchy is shaken up. The student may then teach the teacher; or he may lecture, learn from, and interact with other students.

## Marketing For Programmers (or what I learned from my marketing classes)

By [Necemon](#)

I had to take a marketing class in my last year. At first, I didn't see the point for a software developer to learn those things, but then I realised that it's very important. I am sharing here, the key points I am keeping, regarding how marketing applies to the software business. I hope this note would be of some interest to both (aspiring) programmers and marketers.

Now there is a lot that could be said on that topic, here I just try to stick to the key points, and as it's still a lot of information, I would rather not send it all at once. Instead I would spread that as small blog articles.

Let's first define "Marketing". It can be defined in many ways but it's basically the achievement of corporate goals through meeting and exceeding customer needs better than the competition. Indeed, the marketing concept is a philosophy of business that puts the customer and customer satisfaction at the center of things.

Why should developers care about it? Because they also want to make money! Software is also a business and if you don't have customers, you don't have a business. Further, the marketer and the developer have surprisingly a lot in common. For example, one of the main challenges faced by the marketer is to be able to anticipate the next need, and to deliver a response to it. That's also a key challenge in the software industry, turning innovation into business. That's what made Microsoft, Apple, Google and Facebook so successful : coming up with the right product at the right time in response to people demand (an huge demand that competitors didn't clearly see yet).

It has been observed that companies which perform best are the ones that are marketing oriented: they aim to create a superior CUSTOMER VALUE in order to attract and retain customers. Customer Value: it's critical here. Customer value depends on how the customer perceives the benefits of an offering and the sacrifice that is associated with its purchase:

Customer value = perceived benefits - perceived sacrifice

Perceived benefit can be derived from the product (simple, fast, easy to use), the associated service(customer support, documentation) and the image of the company (Apple, Microsoft, Google) Perceived sacrifice is the total cost associated with buying the product. This consists not just of the monetary cost, but also the time and energy involved in the purchase (download time, quality of the payment system).

Again, the key to marketing success is to exceed the value offered by competitors. But marketing is not just a concept. In the coming part, we will see how it can be applied through concrete actions. The ideas we are going to discuss include the following:

- [1. Reaction to the global and local marketing environment](#)
- [2. Understanding the customer behaviour](#)
- [3. Research and Information systems](#)
- [4. Market segmentation and targeting](#)
- [5. Branding](#)
- [6. Communication techniques](#)



## 1. React to the global and local marketing environment (on marketing for programmers)

By [Necemon](#)

Whether you are an independent developer or a company, you can't just do your own thing and make programs that you find cool. You have to consider the microenvironment and the macroenvironment in which you operate, to adapt and take advantages of emerging opportunities and to minimize potential threats.

Your microenvironment consists of actors that affect your ability to produce effectively in your chosen markets. Those actors are the suppliers, who build and maintain software, products and technologies that you use yourself (For example, Microsoft makes and maintain .NET and Visual Studio). There are also the distributors, and these days, the software distribution happens mostly online, on your website, on your partners websites or on marketplaces (like the Android market place). Some other important actors in your microenvironment are your customers (users and companies that buy or use your applications) and your competitors (other developers, products and websites that target the same customers).

Your macroenvironment consists of broader forces that affect not only your company but also the other actors in your microenvironment. These forces are largely uncontrollable. If we consider the economic forces for example, due to the effect of supply and demand, things such as national economic growth, inflation and unemployment can affect your sales. When money people make less money, they are less likely to buy your apps. There are also the social forces such as world population, the change in the age distribution and the change in household composition. Moreover, cultural forces may have implications for the way in which business is conducted. For example, it has been observed that some types of products and services are considered offensive or banned in some eastern countries, like China where Facebook and Google face some censorship issues.

There are some other factors in the macroeconomic environment such as technological forces and political and legal forces, which, in our context, ensure compliance with local and global IT laws, regarding patents and copyrights for example.

Last but not least, the physical forces: it is mainly environmental issues and natural disasters, such as if some random hurricane suddenly came up in the area where your servers are located, blowing them up and incidently making you lose your data and your customers. Unfortunately, as I mentioned before, the macroenvironment factors are mostly uncontrollable. However, it's important to monitor the risks and be prepared to react.

## 2. Understanding the customer behaviour (on marketing for programmers)

By [Necemon](#)

It's mostly about how people spend their money, that is to say: who buys, how they buy and what the selection criteria are.

**Who buys?** And more importantly, **who is important in the buying decision?** That's not always the same, the person making the purchase is often different from the person who will use the product. Both may be different from the person with the power and the financial authority to make the ultimate choice of the product purchase (parents?). And of course, there may be some other persons influencing the outcome of the decision or even trying to impose their choice criteria on the decision. As you see, there may be many actors that you would need to convince that your solution is the right one.

**How do they buy?** The way organisations buy things is radically different from the way users (as individuals) purchase. Allow me to focus on consumers. It generally starts with the identification of a need or a problem. Then the user will do some information search and try to find a solution. They might of course come across many solutions that they would evaluate and select the one they judge as best.

But how do they know, **which one is the best? What are the choice criteria?** Well, the criteria are various:

Technical criteria are related to the performance of the product or service (reliability, durability, performance, convenience, etc.)

Economic criteria concern the cost aspects of purchase (price, running costs, value for money, etc.)

Social criteria concern the impact that the purchase makes on the person's perceived relationships with others, and the influence of social norms on the person.

Personal criteria concern how the product relates to the individual psychologically, given that emotions are an important part of decision-making for the customer (for example, some prefer a product just for love for a brand).

Wow, that's quite a lot of information to process, isn't it? Wait, to process it, you need to have that information in the first place. Before even understanding what they want, you need to know what they want, so you need to know your users, how do you do that? Through some more or less advanced research, hence the next point : [Research and Information systems](#).

### 3. Research and Information Systems (on marketing for programmers)

By [Necemon](#)

Marketing research is extremely important. Companies that are really marketing oriented recognize that they need to always be in touch with what's happening in the market (and so should you). Customer needs are constantly changing, often in ways that are very subtle. If we consider the history of computing, where we went from huge machines, to smaller ones, reaching the home personal computer; and from disconnect machines to the need to connect them as a network and then to the Internet. There were then so many resources online that there was a need to sort them (as Yahoo did), and then to search through them (as Google did). More recently, there were a need to take the physical life social experience online (as Facebook did) and now we can feel that the people need is to take the internet everywhere they go, and we are slowly going mobile.

Many technological revolutions happen accidentally but most of them are the results of relevant research and observation. On the Internet, there are many (cheap) ways a programmer can find out what people want:

**Internet surveys:** usually managed by email or by a specialised web service like Survey Monkey.

**Website traffic analysis:** Measurements of the areas of the site most frequently visited, which products are purchased and the payment method used.

**Marketing databases:** Inviting users to create accounts on your service, deriving profiles by gathering relevant information about those users. Therefore knowing your users and their needs better.

**Prototype testing:** simply to make some users test the basic version of the application before it is completed, to see if they get the idea and if they like it, what they like about it, what they don't.

## 4. Market segmentation and targeting (on marketing for programmers)

By [Necemon](#)

Marketing segmentation is the identification of individuals and organizations with similar characteristics that have important implications for determining marketing strategy. Therefore, market segmentation involves the division of a diversified market into a number of smaller submarkets that have common features.

Consumer segmentation can be based on many factors:

**Behavioural:** benefit sought, purchase occasion, etc. for example: almost everyone use Microsoft Windows but people don't make the same use of it. Therefore, Microsoft came up with many flavors and the customer can pick one according to their need : Home Edition, Professional Edition, Enterprise Edition, etc.

**Psychographic:** lifestyle, personality, age, etc., for example: Children have become a very important market as they are spending an increasing amount of time on the internet. As we can observe, Disney which is one of the biggest sellers for children, has created a web-based video game called Virtual Magic Kingdom, in an effort to stay relevant in the market.

**Profile:** demographic, geographic, etc. At a very basic level, markets can be segmented on the basis of country, region or city. Also, it's obvious that the language you use may change according to the country you target.

Once the market segments have been identified, the next important activity is the selection of target markets. Target marketing refers to the choice of specific segments to serve. Suppose you make a social network site, you should first target a specific niche. An example could be to create a site for French speakers, aged 25-35, who like to draw.

## 5. Branding (on marketing for programmers)

By [Necemon](#)

What's the difference between products and brands? A product may be anything that has the ability to meet customer needs whereas branding is the process in which companies distinguish their product offerings from competition.

The benefits of brands:

**Company value**, because the financial value of companies can be greatly enhanced by the possession of strong brands.

**Consumer preference and loyalty**: strong brand names can surely have positive effects on consumer perceptions and preferences. If we consider fan boys, like the Apple fan boys, who purchase nothing but what comes from Apple (Mac, iPod, iPhone, iPad, etc.). Most of the time, their choice depends more on the brand than on the actual devices quality.

**Barrier to competition**: the impact of the strong, positive perceptions held by consumers about top brands means it's difficult for new brands to compete. For example, Google has become notorious as the best search engine on the Internet. Even if a competitor like DuckDuckGo, Microsoft Bing or Yahoo (or you) come up with better algorithms, most of the online users would probably still use Google, as they got the Google brand on their mind.

**High profits**: Strong, marketing-leading brands are rarely the cheapest (Here again, think different, I mean, think Apple).

A good brand name should evoke positive associations and be easy to pronounce and remember (Yahoo, Bing, Color...). Also, it should not infringe an existing brand name and it should be distinctive (Google, Nintendo...).

Another piece of advice about internet branding, make sure the ".com" is gettable (as in [www.yourbrand.com](http://www.yourbrand.com)). When I say gettable, I mean that it's either available for you to get it first, or someone else owns it and is ready to sell it to you for a reasonable price.

## 6. Communication Techniques (on marketing for programmers)

By [Necemon](#)

The promotional mix, which is the overall range of techniques available to the marketer, comprises many elements but I am going to focus here on Internet marketing: products distribution, information and promotional benefits to consumers and businesses through internet technologies. Internet promotion is very convenient for a developer on a low budget (global reach at relatively low cost, metrics, dialogue with users, avoiding the necessity for intermediary salespeople...). The main ways to go about it include:

**Online advertising:** advertising your products by posting links and banners across the web, on your partners websites or through ad networks like Google Adwords.

**Search engine optimization:** maximizing your chances of being found by search engines (most importantly Google) and be showed among the very first results. There are many methods that helps, like mentioning relevant keywords in your page text and meta data. Also, getting popular by getting people to mention your product online and link to it.

**Social media:** the use of social network sites and blogs to get in direct contact and interact with your customers and potential customers

This ends our [series of articles on "marketing for programmers"](#). There is a lot to say on this topic, but I wanted to stick to some key points related to the software industry. If you are already a black belt ninja marketer type, you would probably find most of those notes boring, sorry about that, but please let me know the important things that I missed here. Anyway, those notes were mostly a summary of things I wish I would know earlier. If you are a white belt, new to marketing and want to know more, you should check out ["Foundations of Marketing" by David Jobber and John Fahy](#).

Thanks for reading.

## Web Citizen And Media Coverage: Yes, But...

By [Cyriac Gbogou](#)



These days, with the advent of social networks and blogs, every citizen yearns for information as mentioned in **Article 19 of the Universal Declaration of Human Rights** which says:

**"Everyone has the right to freedom of opinion and expression; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas through any media and regardless of frontiers."**

By virtue of this right, some citizens may sometimes tend to disseminate information without respecting certain rules of journalism that we will recall later.

It's worth indicating that elections are a time:

- to assess the state of the country at all levels (health, education, safety, etc.)
- to evaluate what changed and what didn't since the last election
- to monitor public opinion
- to review the promises that candidates had been making

In the run-up to elections, it goes without saying that we citizens, like the journalists whose job it is, must be very responsible in our dissemination of information.



**Here are some rules to follow:**

- To inform
- To explain
- To be reasonable and, above all, neutral
- To be independent and honest
- To be accurate and to verify any information before releasing it

To this we can add : to act as **watchmen** (watchdogs, lol)

During that period a lot of rumors, propaganda and false information circulate.

So how do you know whether what you see online is true?

Here are some tools to debunk the myths and manipulations.

[share.america.gov/is-what-youre-seeing-online-true](https://share.america.gov/is-what-youre-seeing-online-true)

[verificationhandbook.com/book](https://verificationhandbook.com/book)

In summary, freedom of expression has some rules that every web citizen must respect.

# Chapter 3

## Corporate Worlds and Emerging Markets

Featuring Antoine Mian, Nnenna Nwakanma and Olivier Madiba



## **Starting a New Job: 12 Tips for a Brilliant Outset**

By [Necemon](#)



**1. Right from the start, look for quick wins,** and do your best to achieve them. You have to find something you can do as soon as possible to show some quick positive results. This will lower the pressure you feel and help you get more confident. Also, it is important to secure these early wins because they help to build your momentum and establish your credibility. But make sure you know how your new company defines a win. Whenever possible, it is also important to choose a victory that will be important to your boss.

### **2. Behave as if you are still being interviewed**

Different workplaces have different practices when it comes to probationary periods. Some use this time to provide you with extra training and support from a mentor, and, of course, to ensure you're up to the job. Think of your first few weeks as an extended interview. Show up every day thinking you need to prove you deserved to be hired. You'll work harder, work smarter, won't take anything for granted, and in a short time you will confirm your level.

### **3. Start demonstrating and documenting what you promised on your resume**

Whatever you sold them during the interview, your mission is to demonstrate that you are really going to do it. For example, if you said you were a computer security expert, first check that their website has no basic vulnerabilities. And take notes on your accomplishments, your main contributions, and the positive comments you receive. This information will be useful during future performance reviews and salary negotiations.

#### **4. Understand your goals**

In the first few weeks, make sure you understand why you were hired and what your goals are for the coming months. This can help guide your efforts in the coming weeks.

Work with your boss. Schedule a meeting with your boss as soon as possible and tell him that you would like to produce an exceptional job. Ask him: "How can I help you? What can I do to exceed your expectations?" Use this initial meeting to establish what success would look like in the first period. It is possible to start a new job without knowing precisely the objectives that directors expect, but it is wise to enquire, while taking into account one's strengths and limitations.

#### **5. Write your knowledge base**

Setting good habits and getting organized from the start will make your life easier down the line, especially since a lot of new information is coming your way. Do not rely on your memory to retain everything in the long run. Create a new file from the software you would feel most comfortable with from the first day and use it as a journal for the large amount of information that you will certainly receive at the beginning, which will increase over time, and which will certainly become useful to gather your observations, your ideas and questions. Adds keywords to facilitate future searches. Organizes information by sections, subject, areas of activity, think about "how to use this information in the future" or "why would I look for it?". Describe what each division does, which is responsible for what, the acronyms used within the company. Draw processes and hierarchical diagrams. Soon it will be your working documentation, a bit like a wiki of everything you know about your work and the related subjects.

#### **6. Get to know your team**

After surviving the first day in your new position, your next task is to make sure you become a valuable member of the team. Try to find out how long each team member has been in the business, what they do, what positions they have occupied, where they have worked in the past and what is the hierarchy. You should approach these questions subtly, trying to slip them into a general conversation rather than throwing a ton of questions at them at once.

Take the initiative to introduce yourself. If the pace of work is accelerated, they probably won't have time to come to you. Start with your team, the people you work with directly. It is also in their interest that you take a good start, since your work will directly affect theirs.

#### **7. Manage expectations**

Essentially, what to avoid is to create too many expectations. Be realistic about what you could achieve, especially in the first few months. Do not expect constructive comments to come to you spontaneously. Sometimes people do not communicate expectations clearly (you have to ask directly, clarify, ask again).

**8. Trust yourself to execute and develop the new skills you need**

It's easy to feel overwhelmed when you start a new job, and you might be worried about not functioning as expected. But remember this: those who hired you knew what they were doing. They want you and only you: remember that they have specifically picked you among many other candidates. Your approach may be different, but the company could change and grow richer, it's ok to bring in your special skills and attributes.

To help you start to acquire more skills in your new role, create a learning plan by listing the skills you will need for this position and identifying any improvements that may be required. Learning these new skills should be your goal during your first few months.

Find people to help, volunteer to help in an area where you would like to improve. Do not hesitate to ask for help. If you do not know how or where find the information you need, you might lose time if you look for it yourself. Get in touch with the colleague or team that specializes in that area.

**9. Study the strategy behind your efforts**

Consider your current tasks as a piece in a puzzle you have to solve and build, with your team. Think about how you could take advantage of your current responsibilities. Think about the skills you could learn, the visibility you could acquire, the contacts you could create. Every task, every project and every job can lead to other opportunities.

**10. Find a way to differentiate yourself**

Works hard to be known for something specific (specialist of a technology or domain). Could you be known as the one who answers the fastest or who is always ready to offer help?

Choose a valuable mission that truly benefits the company and other employees, and strive to excel at this mission.

It is important to be seen doing these things. Do not hide your talents and contributions, and remember that business is not a place for false modesty.

**11. Reinforce your connections on social networks**

Once you're officially in the role, it's important to update your title on your own social media platforms and start following your new company and colleagues. When you meet new people, reinforce relationships by finding them on Twitter or LinkedIn.

Stay in touch with your former colleagues. This may seem contradictory, but the first week of a new job is the perfect time to contact colleagues from your previous companies. Get back to them and ask for recommendations on LinkedIn. The best time to get referrals is when you are not looking for work.

**12. Never forget why you were hired : to help advance the goals and mission of the company.**

What is your job, really? Your job is to make your boss's job easier, to help your boss succeed. Always keep this in mind, even if it is not written in the job description. It is possible to achieve both your goals and those of the company, that's what you should do.

See your boss as a person you help, not as someone who tells you what to do. The more you help them achieve their goals, the more you will be valued. Moreover, you will see that it is easier to work hard when you know that you are helping someone (rather than obeying him). And you'll enjoy your work even more. It's much more fun and rewarding to help than to comply with the instructions.

## Your First Job is Like Your First Girlfriend

By [Necemon](#)



Several of my readers have expressed some anxiety when it comes to getting their first chance. Many think they are hopeless, as they have never done it before. By not having certain socio-emotional skills and languages on their "psychological resume", they feel like they have an obvious flaw, as if applying for a job were like asking someone out.

Given their lack of experience, others worry that their first opportunity will be a complete failure because they wouldn't know how to handle it.

So let's talk about that first time.

### **Your first one**

The one in which you dive deep without really knowing what to expect, the one that blends happiness and apprehension at once. The only problem is, the odds that your first connection is also going to be your last one are pretty slim. Right, nobody likes to think about their project being temporary. Beginning by telling yourself from the start that you will end up splitting up, that's not very romantic. But again, pretending that it's not a possibility doesn't do you any good either.

Don't take it the wrong way, I'm not saying you should start each project with the idea that it is doomed to failure. Quite the opposite, you should enter it with the idea that you will appreciate every moment. You will enjoy a lot, you will learn a lot and it will make you stronger. So yes, your first experience may not be the last one, but it's not something you have to fear. Appreciate it for what it is, enjoy every instant.



### **Your first break up**

Okay, the first experience probably will probably not be the last one, but how do you know when it's time to move out?

If you ask this question to 10 people, you will get 10 different answers. For some people, it's after 3 to 5 years. There is no single answer that applies to all cases, but there is a great deal of research showing that there are a few key signs that signal it's time to move.

One of the hardest parts of life is discerning the pointless attempts from the decisive struggles, deciding whether to walk away (adapting to reality) or to try harder (and when insisting, to what extent that would be reasonable). It is a silent battle. No one understands how you really feel.

But if it ends, the separation does not cancel out the good things that have happened. On the contrary, everything you learned during that first experiment could help you prepare the next one. Growing up is sometimes a painful process but you become more confident and more experienced.

### **Getting ready for changes**

Meeting and spending days with a person or team for weeks, months, years.

Seeing them and exchanging with her almost every day, sharing, and evolving together on common projects.

Seeing them more often than your own friends, more often than your own family (especially if you live far from your hometown).

And then, just stop seeing them. Overnight. This is the change.

You will need to get out of your comfort zone and out of your routines. Adaptation, the ability to become accustomed to new environments, and to new conditions with speed and ease, it is not just an important skill, it is essential.

### **Moving on to something better**

You would not be able to reproduce this first experiment in the same way, you should not expect to find the same thing elsewhere. You have to be open to new experiences and new people. Do not be so stuck in the past to the point of not appreciating the present. You were lucky to have met that interesting opportunity, but it does not mean that there would not be other opportunities equally interesting, and why not, even more interesting, if you open yourself to the idea of studying them.

You cannot recreate that first connection exactly, but you can find much better.



Begin by acknowledging that there is nothing special about the first one. It is true that it was a wonderful time and you discovered many things, but people tend to romanticize, to idealize, even to glorify the first time they experience something. There is probably nothing special about your first experience beyond the fact that it was the first one. Remember that you are biologically wired to place your first experiences on a pedestal. You certainly do not want to let this mentality to get in the way of enjoying the present. There is a good chance that you will remember this so-called first experience in a somewhat exaggerated manner. As a result, this may lead you to some inappropriate comparisons. Comparison causes conflict. Conflict causes anxiety. Anxiety causes fear. Fear prevents you from thinking. There is no intelligence where there is fear.

### **Learning experience : getting stronger**

Instead of seeing the first times as perfect situations, view them in terms of experience, it was practice to some extent. Analyze and value what you have learned about yourself, about others, about your abilities, your affinities and your ambitions.

### **Your first job is like your first girlfriend**

The similarities of the contexts are indisputable. On the other hand, there are also some differences. Let us explore these differences before separating. Let us emphasize the case of employment, or it is easy to imagine everything that can go wrong during the transition. There are 2 fundamental questions that you want to ask yourself before you leave.

#### Did you learn all that you could ?

Before you leave a job, consider the set of skills you would like to present to a future employer. Works long enough to show that you have evolved through concrete achievements. Depending on the industry and the type of work you are doing, it may be possible to standout, to demonstrate your skills and the successful execution of a few projects. In any case, it is always better if you can present or describe a strong track record to those who are likely to recruit you.

#### Did you sign your next contract ?

The key recommendation is not to leave a position until you have the next one lined up. It is important to have the safety net of a new position because you never know how long it will take to find something that really suits you. Moreover, it is harder to find a job when you are unemployed than when you are employed.

### **Getting far away from here...**

Everyone ends up leaving their first job, whether it happens after 2 months or after 40 years. However, certain steps must be taken in order to leave properly and professionally, in order not to burn any bridges.

When checking out:

- Schedule, negotiate your handover and final dates
- Organize your projects, documents, and other files for other people to use easily.

Keep positive relationships with them until you leave. You'll probably need their recommendations.

Staying in touch. Your former employer and your co-workers are now members of your professional network, so make sure you stay connected with them, for example on LinkedIn or Twitter.

To conclude, we can't stop students and yuppies from thinking of themselves as the stars of history, but we can remind them that the early contacts constitute the first chapters, not the final chapters. Your first love, the one you must protect at all costs, is yourself. You do what you think is best for yourself, and if it doesn't work, you say "never mind, at least I would have tried."

## **5 "Good Reasons" For Which You Will Not Be Hired**

By [Nnenna Nwakanma](#)

I have been wanting to put these thoughts down since end of 2011. Quick story, I am running a training for West Africa and also doing some recruitments. For one post, I was interviewing candidates in the morning, and for the other position, interviews were in the afternoon. So I had a full day of people trying to convince me and the panel I was heading that they were the right persons for the right tasks!! Then 7 PM rolls round and I am back in my hotel room, trying to ease off. And someone knocks.

"Madam, I see you have been interviewing people, please can you also interview me"? First, I am shocked, then insulted, and really offended. But my "other brain" says: "Cool down and get to the bottom of this".

So I feign interest and I ask: "Please what is your name, what is your academic qualification, what are your skills, and what kind of job are you looking for?"

That was the beginning of a long story. Let's cut it short and make it simple. ALL the reasons the gentleman gave me for which he should have been considered are "good reasons" for which nobody ever gets hired.

### **Most common reason: I need a job.**

Okay, so YOU need a job. The NEED is yours. Here is the summary. Your problem is not my problem. That you need a job is your issue not mine. There are millions of people saying the same. I get such lines at least 5 times a week. It may seem a bit callous, but your need is yours, not mine. That you need a job is no reason to hire you.

### **Reason 2: I need more money.**

Quick answer. "Same here". I need more money. In fact, as an employer, I most probably need more money than you do. My social, economic, fiscal, and life responsibilities certainly outweigh yours. So in terms of cash needs, mine is greater than yours. Give me a break.

### **Reason 3: I am not enjoying my job, I need excitement, enjoyment, adventure.**

Seriously? Go start your own stuff then! Do what you like. Choose when to wake up, only do stuff that are exciting. Never accept an assignment unless there is adventure attached to it! Nobody will hire you to make you happy. They will hire you to work. Sorry, but your employer's interest comes first.

**Reason 4: There is no future in this country. I need an escape.**

And you think I am stupid? If I started off like you, do you think you will be here asking for me to hire you? It was because I believed in a future, and I committed to making the then-present better, that you are coming up to me and thinking I will be the one to offer you a better future. You are not looking for work, you are looking for escape. Odds are, when I invest in you, you will either jump ship at the first sign of difficulty or disappear once you find another escape. I am looking for someone to solve problems with, build, construct and ensure a better present and promising future. I do not provide platforms for escapists.

**Reason 5: I like you. You are such a wonderful person. I want you to be my boss.**

Over the bar!! Because I change planes, gave a tip, I speak with respect and I keep to time. Or maybe because of the dressing or the "fashionable" things you see around me. You most probably saw me after or off work. It is a mirage. You need to work by me to see the "entrepreneur" me. I work long hours, do reports that run into 100s of pages. I lose sleep. I stretch myself!! You better wake up!!

\*\*\*\*\*

So you are asking, "what are the good reasons" for which I can be hired. I will be brief.

Every employer is looking to solve a problem. I will hire you if you can provide a solution or can work towards providing the solution

If you, by your own skills, will bring me more benefits than I am currently getting. If you can make me richer, you are welcome

If you are resourceful person. Someone who is willing to research into solutions, make trials, accept failures and help colleagues do better

If you have a skillset or knowledge or capacity that I do not have

If your network is larger, more promising and more interesting than mine.

\*\*\*\*\*

Conclusion, nobody will hire you because of you. I hire because of me. So next time you come knocking...

Think of me, not you.

## GE FAQ : 3 months in the Cyber Pearl

By [Necemon](#)



### What is GE?

[General Electric](#) is a USA based multi-national company, where I worked as an intern over this summer. This [100+ years old corporation](#) operates in more than 180 countries with more than 300,000 employees. GE's broad range of products are from jet engines to power generation equipment, financial services to media content. The unit I have been working for was GE Energy.

### How did that happen?

I heard about this internship through the [International Development Office of my University](#). Among other things, they help students to find international academic and job experience. I got selected based on [my resume](#) and [an interview](#) which went pretty well. I briefly explained the [technologies I was into](#) and [the kind of projects](#) I had been working on.

The office was located in a state-of-the-art IT Park, known as the Cyber Pearl. I was about to [start my first corporate project](#). It felt like I was getting in the IT industry [for the first time](#). Kind of. After [long hours of lonely coding from my student room](#), I was now in an actual work environment. It was somewhat different [but I was ready](#). Technically there was not much else I could do because I didn't really know what my task would be. I just knew it would be exciting.

### What exactly were we doing over there?

I can't disclose actual details about the project I had been working on but I would say that it has been a chance to improve my problem solving skills. I could sum it up in two words: "[Fun Learning](#)". Even though [the technology context may have been an advantage](#) from the start, it's quite clear that this internship involved [a much wider range of skills throughout the project](#), including effective project management, time management, [corporate culture](#) and [team collaboration](#), to name a few. I was learning and improving through my assigned tasks, and (perhaps more importantly) through my team feedback. Plus it was a pleasant experience because my teammates kept encouraging me to tackle ambitious tasks. it was challenging in a good way.

So... To make a long story short, what was the point?

As [a wise man](#) used to say, it's quite hard to learn swimming just from reading a book about it, there are some Truths that one doesn't easily perceive in a classroom setting, or [while coding alone](#). It's been convenient to interact with [brilliant professionals](#) from the engineering industry, to [learn from their experience and advice](#), and definitely, to [explore the reality of large scale projects](#).





## 5 lessons I learned from my internship at General Electric

By [Necemon](#)



This article is a follow up to my previous article on my [internship at General Electric](#). It's just a short note to sum up some important things I learnt.

1. **Theory learnt in class is actually applied in the industry:** there are some things that you learn in university you will probably never use again but other things you will use every day of your life. At the beginning I was like, who cares if the 7 layers of the OSI model are the application, the presentation, etc.? And who care about normalising the databases until it is in Boyce-Codd normal form? Actually a lot of people take those things very seriously.
2. **Documentation is important.** I used to see it as a time consuming process that was barely worth doing. Now I understand how it increases teamwork efficiency.
3. **Technical skills are not the only thing that matter.** Actually a lot of other things matter in the workplace like the business focus, the interpersonal skills, etc. But this experience made me realize the need to emphasise on personal skills like self-confidence and emotional resilience (coping with pressure and quickly bouncing back from setbacks).
4. **Follow the instructions.** The discipline of doing what is useful and required, instead what sounds cool.
5. **Secrecy is the first law of magic.** That one is pretty straight forward. The team work remains in the team. The corporate information should not be disclosed outside of the organisation.



## 12 Years Of A Dream Challenging Reality

By [Olivier Madiba](#)

To be realistic is to find a rational way to achieve a magnificent goal.

PS: This is neither a biography, nor a success story. This is just the testimony of an African entrepreneur for those who can be inspired by it, here or elsewhere.



### The Triggers

We are in the **1997 - 2001** era, the golden years of PlayStation 1, and I play two games that will change my life.

Yes, Final Fantasy 7 and Metal Gear Solid 1. Two outlets of a shy and insecure adolescent's life (like most geeks of that generation). Two games that made me dream, wonder, travel (in my head). And more importantly, two games that made me say this sentence "One day, I want people to feel that way because of me. I want to see people holding game controllers, shouting, laughing, feeling strong, frustrated and happy because of something I would have created".

We are in **2003**, I am 17 and I just graduated from high school. I join the University of Yaoundé 1, and cybercafés are fashionable in Cameroon. I want to get into computer science, hoping that one day I could go and work for Ubisoft or Square Enix in order to achieve the perfect RPG. I start to read on the Internet about how to make a video game and I frenziedly start scribbling my first game design notes.

**February 2004:** I post on JeuxVideo.com that I will build the perfect RPG someday:

[jeuxvideo.com/forums/1-31-8247500-1-0-1-0-0.htm](http://jeuxvideo.com/forums/1-31-8247500-1-0-1-0-0.htm)

Obviously, people laugh at me, not out of malice, but out of "realism". As you can see on the discussion forum, on that day, I challenged the whole world.

### **Playing with the cards we have, instead of crying the ones we don't have**

I join Oniro, an online French community of amateur developers and I relentlessly read all the tutorials. Cybercafés consume around 2,000 CFA francs (\$3/\$4) out of the 5,000 CFA francs (\$9/\$10) that I am sent weekly as pocket money. I have never eaten so much bread with chocolate before (100FCFA bread, 50FCFA chocolate).

**2005**, I fail (many of) my classes in college and I have less course material to study during the repeated years. I work on Aurion so as not to get depressed and not to think about my life of non-stop school and personal failures.

I post adverts around the Yaoundé city centre stating that I'm looking for people who want to make video games. People think that I must be under the influence of some very powerful hallucinogens, but let me proceed in my delusion.

From **2004 to 2008**, I live the worst years of my life. Non-stop school failures in a mind-numbing educational system. I realize that my parents do not have enough money for me to study one day out of the country (death of the naive Ubisoft dream). Besides, they have so much trouble paying for these college classes that I keep failing. Every day, I can feel the specter of poverty that is narrowing around me.

This is also the period when I leave my bubble and I enter the sphere of collective negativity. I spend my days complaining (and listening to people complaining) about the lack of job opportunities for new graduates, about the ethnic conspiracies, about the social injustice, about the neocolonization that nip any hope in the bud, etc. In short... There's nothing else to say, my life was ruined before it even started.

**2008**, I owe my wake-up call to two things:

- (1) When you are a student in Cameroon, you certainly have a "back-up" uncle who helps you out with 10,000 FCFA or 15,000 FCFA when you are at your worst. One day when I went to see him he kept me waiting from 8AM to 5PM in his office (I had not eaten since the previous day) and gave me 2000 FCFA. Instead of getting upset, I instantly understood the symbolism, he was showing me how he struggled to earn money and how his life was not easy. That day I told myself, "I no longer want to be just a consumer"
- (2) I spend one week of holidays with a friend whose father "succeeded". And while I used to tell myself that everything is about luck, connections, good schools (in other words, the national mantra of laziness and defeatism), I observe how this man lives. He gets up at 6am, works until 10pm even on Saturdays. On Sundays, he rests but keeps reading about his line of business. This man trains continuously to stay motivated and ambitious. I then have another epiphany: I have to train to become like that."

I dust off my AURION project. It will be my motivation to learn to be excellent even if I can't live from it, it will keep my mind active. This time I fix my beginner's mistakes by starting to write a complete scenario and design specifications before getting into "programming and other details". I also decided to delve into how to program your game from scratch (or almost) in order to no longer face the limitations of amateurish engines.

**From that period, 2008 to 2010,** I had created a local tech startup : MADIA (starting from nothing, with college friends who were stuck like me). I have gained experience in people management, business management, accounting and computer graphics. 2010-2011, MADIA is running out of gas. We are struggling to keep the company afloat. No initial investment (therefore, no working capital), some customers paying with a minimum of 6 months delay (since they are being paid with some 3 years delay themselves), public contracts randomly cancelled, and a micro-finance agency that closed down while still holding our savings in early 2010. I also went through some great family and social pressures that urged me to try and find a job as a civil officer, or some similarly stable condition. I wouldn't mind, but I don't have the credentials required by big corporations.

Psychologically, I could only survive this period of huge frustration because I was working on Aurion from 9:00 pm to 1:00 am every day when I am back to my small room on a hill in Biyemassi.

**Mid-2012:** I read an article about video game budgets. A studio which blew \$4million into their "employee entertainment department". The budget of "small" games which requires anything between 4 and 5 million, just to be completed in basic conditions. The achievements of independent games, download sales, Kickstarter success stories. I tell myself that I can build a prototype with Aurion, and then ask people on Kickstarter to support me in making a semi-professional game.

I do a simple calculation on Excel to visualise what it would cost me to open a studio with 20 people in this country with ideal conditions. And I realize that, even if we hike all the costs, we would hardly need \$1.5million to operate for 2 years, even if we include a \$500,000 advertising budget.

At this time, I'm so broke, still living off my parents' money, that is 10,000 FCFA (15 €/20\$) that my mother send me every week. I just want to be funded for 1 year to make a game demo, that would not be market-ready, but it would show a huge narrative and technical potential. I want 2 million FCFA (€3000 / \$3650).

I have been neglecting the management of my company, which is stuck in Cameroon's heavy and slow system. I start to leave the office earlier to go home and work on Aurion. Obviously, no one understands me, and the pressures increase. I want to put forward a playable prototype ready to go on kickstarter by December 2012. I registered on the platform since March and I always saw the "Post your project" button, without ever clicking on it.

The prototype that demonstrates my game is pretty much ready. So I get on Kickstarter on December 19, 2012 to start testing the form interface in order to submit my project, and then... Existential punch in my face...

Because I am connected from Cameroon, I am not allowed to post my project on Kickstarter... Huge disillusionment, extreme disappointment, once again the world shows me that I was born in a country that is destined to non-stop failure. However I will spend Christmas with peace of mind, I did my very best, I do not regret anything.

From that moment, I start requesting funds all over the place. Online, I message Diaspora friends, aunts, uncles and project supporters, asking them to help as they can, whether it be with €15, €50, etc. I write the business plan (version 0) of Kiro'o Games within a week of frenetic writing. All my dreaming years blend on paper. I'm considering a budget for a small amateur team of 6 people with computers and survival wages for 1 year. I am looking for 12 million FCFA (21000\$, 18000€).

Nobody really brings any concrete contribution. There were just micro-donations (which were very much appreciated indeed, they didn't have to) but no real investments. Some people didn't want to simply donate, they would have preferred to invest and own shares of the "real" game later.

A night of inspiration and intense reflection on how to go beyond this stage. How to escape from this feeling of labyrinth involving a scheduled and inevitable failure. We have foreseen that we need to evolve, without being afraid of responsibility and higher risk in order to succeed. After all, we wanted to play in the international league, so we had to think like international conquerors. I reviewed all my calculations and redesigned the entire investment system.

The Crowdfunding Equity of Kiro'o Games was born, a unique financial model in the business world, unmatched in Africa so far. Almost an economic "accident" according to some observers. It was a fusion of the Wall Street principles and the Bamileke tontine system of Cameroon. We added a legal process with certified e-mail validation.

Investors were still skeptical. People want to see visual prototypes, period. The usual Ouroboros, the snake that bites its tail. We want investments in order to do a job and we have to do that same job in order to have the investments. How to make professional graphics without paying a team? Without even the right equipment to draw correctly? In a country where there are not even "decent part-time jobs" that could help us survive (most of us were still living off parental funding)

**August 2013:** we are coming out of a few months of silence. Everyone was already saying that it was yet another dead idea in the bud. National legends even began to say that we might have planned to rip people off.

The next phase sounded like a melody of interactions, from August to September every week, every day we fed the Web with our visual models. We have written to the press around the world.

We also set up a standard e-mail for investment request, a video that tells our story and explains : "here is what we did with inadequate resources, imagine how far we will go with normal means".

**Mid-August 2013,** we confidently launch a funding campaign on Indiegogo. We are energized by less impressive games that have raised huge sums. We rely on our history, our potential, our journey. After all, video games and computers were born in garages. Players of our time will want to be part of this story.

The magic will not work. As far as most gamers are concerned, we are too amateurish and are probably a gang of African crooks on the internet (akin to the infamous Nigerian Prince scams). Epic fail with just 2% fund raising on Indiegogo.

I experience some humiliating moments on gamers forums, where we are accused of spamming communities, not to mention a mailing list bug, when we accidentally emailed 15,000 American and European folks and got insulted continuously for a week. We had to disable the mailbox for this to stop (lol).

This experience makes us realize that the "la la land" video game "era is over. The Internet begins to slide on its dirtiest side: trolling. Our morale takes a hit, but we are already used to smashing walls. We will come back again when we are ready.

**September 2013:** An article in the "Le Monde" newspaper will baffle the most skeptical bystanders. This follows a first article in the video game section of their online edition. Concrete transactions and funding increase a bit.

The newspaper wants me to explain our business model beyond the game and they seem to appreciate our ingenuity. We earn a second publication in their Economics and Enterprise section. Stock purchases are increasing, reaching new heights, but the milestone of 100 shares sold is still unreachable.

**Kiro'o Games office opens on December 2, 2013.** That event drives an Internet Buzz. A 10-year epic journey has just ended. A success showing how we pushed the limits of our society to earn the right to be what we chose.

This was the beginning of another challenge: to actually complete the game. But we had our first happy Christmas for 10 years.

**June 2014 - October 2014:** We managed to set up our own game engine, the challenge is more complex than expected but it's all arranged and we see the game run for real. We publish pictures of the game and we have positive feedback, however the project's popularity has raised expectations. The "small" game that was planned initially must now fulfill these expectations, without falling into the traps of infinite improvements. We reassess the game specifications to increase the development time and improve the engine.

**November 2014 - December 2014:** The videos of the game are published online with concrete content. We show what has been done through a video that generates a buzz, the reactions are beyond our expectations. This also brings exposure to the international specialized press. We get our profiles on the biggest video games sites.



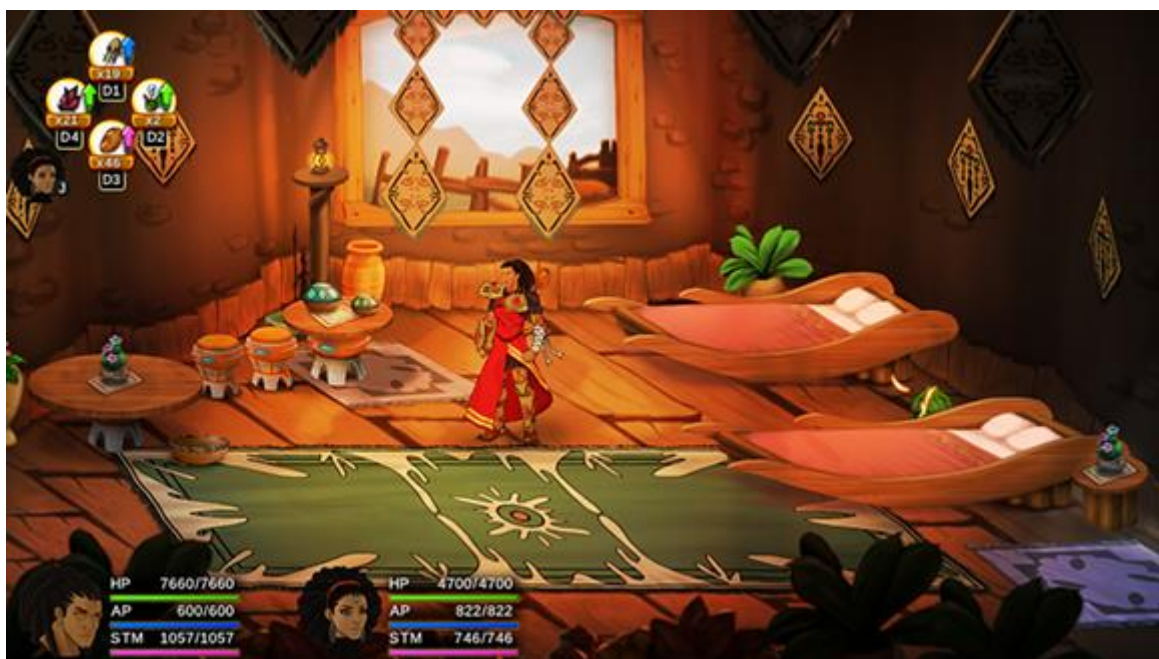
**January 2015 - February 2015:** For the Video Game press to be interested in the game and for gamers to support us, we can't just rely on videos. There is still this doubt that this startup living its first game does not know what it is doing. There is always this suspicion floating in the comments below the news. The game must pass an ultimate test that will seal or sweep our fate: it must be played.

The demo is distributed to the specialized press and it's an intense moment. The live sessions are raining, the test previews are 80% positive. They acknowledge that, even though we are not perfect, we do provide the essential factors: those who play have fun and are intrigued by history.

**August 2015 - September 2015:** The realisation of the game continues, the story progresses well, we do internal tests and the testers are "shaken up" by the scenarios. The gameplay is full of surprises, even for us. We discover new possibilities every day.

This memo ends here for now. Our entrepreneur life goes on, our minds are bursting with great plans and projects. Some exciting, frustrating, exhilarating battles to be fought in the future. Defeats, victories, even betrayals, new alliances, in one word: Life. Whatever happens, I just know we'll do as usual: finding our way.

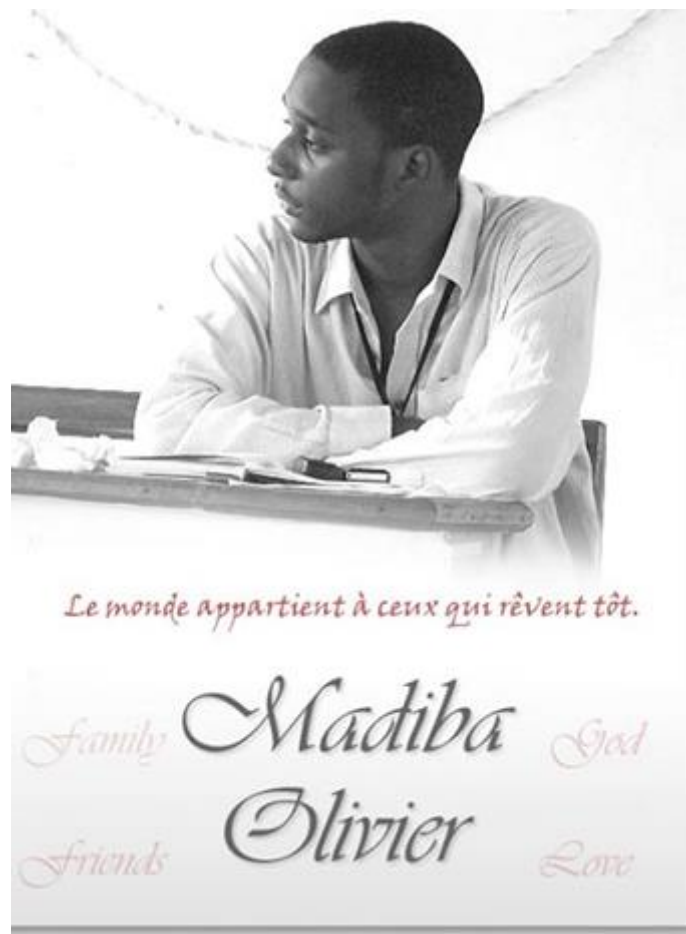








**[Interview] In The Bubble Of Olivier Madiba: "my goal is to reach my maximum human potential and to help others to do the same"**



**1. Quick (but effective) introduction of the personage and his achievements?**

I am [MADIBA Olivier](#), 32 years old Cameroonian. I have a Bachelor Degree in Computer Science obtained at the University of Yaoundé 1. I launched 2 startups (MADIA and KIROO).

I am currently CEO / Game Designer / Scriptwriter / Programmer at [Kiroo](#) which is the first video game studio in Cameroon.

**2. What are your main goals in life?**

My goal is to reach my [maximum human potential in this life](#) and to help others to do the same. I recruit those who want to improve things and who love the trip more than the destination.

### **3. What tools and techniques do you use to accomplish things efficiently?**

I use a lot of software like Word / Excel actually, I spend my time writing and doing simulations. I also like to use paper and my pen to constantly draw what I think, it makes understanding easier.

### **4. Any recommendations for your juniors?**

I would recommend going to [www.kiroorebuntu.com](http://www.kiroorebuntu.com), where I have compiled all my project management experience.

In terms of good reads, here are my favorite books:

- Warrior's Manual of Light
- Conversation with God (3 volumes)
- Shoe Dog (Biography of the founder of Nike)
- Creativity (Memory of the founder of Pixar)

### **5. What is the best way to contact you?**

The best thing is to tweet something on my Twitter [@madibaOlivier](https://twitter.com/madibaOlivier). I can't reply to everyone, but I read everything.

## **The Paradox Of Teachers In Africa : Rare And Yet Poorly Paid**

By [Antoine Mian](#)

There is a quote that says "Everything that is rare is expensive". In sub-Saharan Africa, in that "Everything", it seems that we must include Everything but the teaching profession.

Indeed, the shortage of teachers that affects the whole world in general is particularly more pronounced in sub-Saharan Africa. If we stick to the quote, this situation should have had the effect of raising the salary of teachers to attract as many people as possible in the profession. But no! On the contrary, it appears to be driving teachers' salaries down. In The Ivory Coast for example, in the schools, you can find benevolent teachers and volunteer teachers alongside state civil service teachers.

Often recruited by local communities and/or by NGOs, benevolent teachers with extremely low salaries (often less than 15000fcfa [\$30/€23] per month according to a Regional Director of Education) hold classes, usually in rural areas. Alongside benevolent teachers, there are those who are called volunteers. To cope with the serious shortage of secondary school teachers, the Ministry of National Education and Technical Education recruited 3,000 of those in 2012. Paid at 100,000 CFA (\$200/€150) per month, these volunteer teachers have held classes alongside colleagues often paid three or four times more.

Based on discussions we had with members of the Cabinet and the General Inspection of the Ministry of Education, it turns out that the good success rates on Ivory Coast examinations this year (33.58% in High School, 40.17% in Middle School and 67.03% in Primary School) are largely due to the presence of these benevolent and volunteer teachers. It is therefore clear that these two categories of teachers are important for the quality of education. However, while all of them strongly agree on their importance in the education system, they are less confident about their future. Indeed, when we start addressing the question of their future, everyone goes quiet, especially volunteer teachers.

With such a low salary and an uncertain future, how can we keep these teachers in their positions, whether they are benevolent or volunteers?

The situation in The Ivory Coast reappears identically throughout the Subregion. Thus, the main cause of teacher shortage in Saharan Africa seems to be related to the inability of education systems to retain them in the post. Indeed, quality education requires quality teachers while retaining quality teachers is another challenge in the education system. In sub-Saharan Africa, the job of teaching "on the field" is often neglected for the sake of juicy positions in ministerial offices, central government or private corporations.

If we want strong educational systems in our countries that strive to emergence, it becomes imperative to look into the motivation of teachers through a revaluation of the profession. Indeed, if Africa wants to build a good education system, it must invest in teachers by way of quality training, respecting their professionalism, rewarding and recognizing those who are effective and efficient and improving their morale and motivation.

## Gamification: 5 Peculiar Concepts That African Schools Could Actually Use

By [Necemon](#)



[Gamification](#) is the transfer of game mechanics to other fields, especially websites, learning situations, work contexts or social networks. The objective is to increase the acceptability and the use of these applications by relying on our human predisposition to entertainment.

This design technique allows people to adopt certain behaviors, usually some behaviors that could be considered irrelevant or things they would not normally do: filling out a questionnaire, buying a product, watching advertisements or, as in our current instance: **assimilating knowledge**.

Therefore, by applying these [video-game-inspired](#) techniques to a classroom context, most courses can become much more interesting:

### 1. Progressive evaluation system

#### **What's that?**

Let's admit that the education system in Africa is currently rather demotivational.

From the beginning of the school year, and starting the first homework, the ambitious student sees himself with a perfect score of 20 out of 20, but from there, the only possible direction is the reduction of his points, depending on his mistakes. If he is really brilliant, he could stay close to the average of 20, otherwise in general his average score drops with each of his mistakes.

There is a sort of feedback loop that encourages failure: when you have a bad grade, you feel less motivated to study, so you study less, so you have more bad grades, and so on. Basically, the more you fail, the more you fail.

However, in games, we learn that progress encourages progress and that the desire to be effective is a [much stronger motivator](#) than the fear of failure.

Hence we could assess the students performance just as in video games, so that any given assignment would have an XP (Experience Points) value, and if all the students start with 0xp, and they earn points as and when they find answers, complete homework/quizzes, and progress from level to level.

In this case, each assignment or exam appears to be a reward, rather than a discouraging punishment.

### **Why is it cool?**

It's a lot more fun to win things than to lose things. Plus, this methodology never leads students to a point where they feel that they have to give up.

The good news is that you do not even have to change how average grades are calculated. It is still the same total number of points, to be divided by the same number of tests.

For example, if there are 5 homework out of 20, that's 100 points. Instead of starting from 100 points and losing points with mistakes, we move to a perspective where each starter has 0 points, and depending on his performance at each test, he earns a few points: +10, +5, + 15, etc. Thus the pupil only goes up and not down.

## **2. Global bonus**

### **What's that?**

To extend the evaluation system described above, one can even add global bonuses, for example if one of the students reaches 200 points, [the whole class gets a bonus](#) of 10 points. Or if 5 students reach 100 points, the whole class gets + 2xp or a chocolate cake.



**Why is it cool?**

This kind of reward system encourages solidarity among students and encourages the best students to help their classmates.

**3. Power to Impact****What's that?**

One of the big challenges in the education system is about taking care of children who do not feel that their actions are relevant to their own lives. To feel that you have an impact, means to feel that you control your own destiny, and that your choices make a difference.

Students who do not feel this impact tend to believe that they have no control over their parents' choices about their education. Without this feeling of impact, it is almost impossible to feel motivated. Instead of making decisions about the future, one merely undergoes the events of life without having long-term goals.

The more you feel that you have the power to impact in your life, the harder you work, the more you are keen to engage in more ambitious tasks.

**Why is it cool?**

In fact, people with this sense of power are more resilient in the face of external forces outside their control that are disturbing their plans. They adapt more easily, overcome obstacles and proceed towards their goals.

So how to amplify this sense of power? Through games, obviously. Games that instill the idea that [you control the future](#). In games, the cycle between choice and results is usually much shorter than in real life (and also much better indicated).

In games, the player tries something, he fails; he tries something else, he fails again, he continues [to make new decisions until he succeeds](#).

Games teach that different choices have different outcomes and that everyone controls the choices they make. The philosophical implication is that life is not simply a series of things that just happen to you.





#### **4. External motivators**

##### **What's that?**

In Africa, in general, there are not enough teachers and not enough hours of classes a day to teach children everything they need to know. That's why we need children to stay interested and continue [to learn voluntarily](#), even after leaving the classroom.

There is a solution that costs virtually nothing: [alternate reality games](#).

An Alternate Reality Game (ARG) is a multi-media game that deliberately blurs experiences perceived within and out of the game.

Immersive fictions that blur the boundaries between the real and imaginary world, they unfold both online and in real life, they spread through all channels: treasure hunts in the city, texting, posters, e-mails, websites etc.

(Marie Lechner - Libération)

More than a riddle game, ARGs are essentially a new way of telling stories. These are fictions that are played in real life, instead of being confined to the pages of a book or the images of a screen. Players can see the plot hatch and expand in real time. It's a much richer and fun interactive experience.

(Adrian Hon - Puppetmaster)

Basically, it is the encounter between a large-scale treasure hunt, implying the [interactive unfolding of a story](#), and a community of players (or students, in this case).

**Why is it cool?**

We can use the information we want students to learn as keys to unlock more and more games. And while looking for this information, they will necessarily learn many other things at the same time. And if it's really interesting, they could even meet after school to continue the treasure hunt.

**5. Unexpected connections****What's that?**

The previous game is good but it requires a lot of preparation time for anyone who wants to arrange it properly... For those who want something simpler, there is a research game which is about choosing 2 unrelated interesting concepts, and find the connection between them.

For example: Africa and Deep Learning.

Each player finds a series of links that lead from one theme to another.

The game works well with Wikipedia. It's not just about scanning links, but they should also describe the connection between these links. All successful students earn experience points but the winner is the one who makes the connection using the fewest possible links, he obviously gets bonus points.

**Why is it cool?**

This exercise encourages curiosity and allows students to come across information that they would not have explored otherwise.

Also, encouraging them to take the shortest path is giving them the motivation to go back to information, to think about it, to explore new directions that derive from this information.

But more importantly, this exercise leads them to study the connection between information, that is, how ideas are related, which is a vital skill, even for adults, it is important to contextualize knowledge and to understand [the importance of certain topics](#) that do not interest them directly.

So for our example, if we want to find the relationship between Africa and Deep Learning:

[Africa](#) => [Evolution](#) => [Artificial Intelligence](#) => [Deep Learning](#)

Try to beat that...

## Africa: ICT in Education for Entrepreneurship and Employability

By [Antoine Mian](#)



While facing the problem of [youth employability](#), African countries today strive to improve the [quality in their education systems by using ICTs](#). One of the lessons learned from the case studies in relevant countries shows that the integration of ICT in Education can be beneficial, not only in terms of Education, but also in terms of Entrepreneurship and Employability.

In Lebanon, the use of digital tablets in the educational system has enabled the creation of a digital ecosystem for the production of mobile terminals and educational content.

In Portugal too, the implementation of an ICT plan for education has led to the creation of local companies in the field of computer manufacturing, as well as in the creation and production of digital educational content. This state of affairs has created jobs and economic growth in the country.

The examples of Lebanon and Portugal are not isolated. In most Latin American countries, such as Argentina, Brazil, Venezuela, Uruguay and Paraguay, computers and digital educational content used in the education system are produced locally. In some African countries such as Morocco, Tunisia and Kenya, ICT integration in education has led to job creation around content production.

The main advantage of such a policy is that it stimulates an absolute enthusiasm of all social groups around the integration of ICT in education. As a matter of fact, this policy encourages local entrepreneurship, which leads to the development of a digital ecosystem and the creation of jobs. It also makes it possible to constitute a critical mass of local human resources for the maintenance of the equipment, which plays a fundamental role in ensuring sustainability.

It is therefore indisputable that the policy of [integrating ICTs into education in Africa](#), if it is well conducted, will not only have an impact on education systems. It will also have impacts on the economy through business creation and on the social life through job creation. Therefore, as the vice president of Intel said, a good policy of integration of ICT in education, beyond the aim on Education should have a focus on Entrepreneurship and Employability.

## 4 important revelations that could certainly accelerate the educational revolution in Africa

By [Necemon](#)



When we talk about academic training, a fashionable expression is to say that the established education system is defective - the current system is not defective, it is built in a great way. It's just that we do not need it anymore, it's obsolete. - [Sugata Mitra](#)

Ok, I cannot speak for everyone, but based on what I could see in West Africa and South Asia, here are some hypotheses to consider:

## 1. The accumulation of data in the head is often unnecessary

It is often about attending classes all day, every day and memorizing a lot of data. Students are often forced to absorb information, on the assumption that they **might** need it some day. The truth is, they may never need most of it.

There are some people who can quite easily stay seated still and take notes for 8 hours each day, and then 2 weeks later, they can recite everything they have written and memorized. But there is also this huge population of very talented people who cannot learn that way, especially since any relevant information is easily accessible if they ever need it...

As [Sugata Mitra](#) explains, you do not need to memorize anything. You can find the information when you need it. The role of the teacher is especially to guide the young minds to the right questions, not to give the answers, since the answers are everywhere.

## 2. Information is everywhere, just be connected

You just have to be connected, your brain can accept/give remote contributions. It is a fact that the value of your ideas is limited when you are isolated but there is one great hope: wherever I go, even if it is in a remote village or in the middle of the savannah, the very first thing that I ask or check is, do I have that little bandwidth that could give me some GPRS connection or something equivalent? And in the middle of jungles, I see that sometimes it says "connected". And then I know that everything I say can go everywhere and work exactly the same way. It's just a matter of time...

[Lois Mbugua](#), Customer Project Manager at Ericsson Kenya, says connectivity is about opening up to the world. By connecting a village, students can communicate with other students and teachers around the world.

Her colleague [Margaret Kositany](#) adds that education used to be limited to what the teacher could tell the students, and the teacher was relying on a few text books. But now students have access to a lot more information.

In fact there is a very big difference between access to information and school. They used to be the same thing, now the information is available online for everyone.

A great example of this would be [Coursera](#), a digital company offering open and massive online courses. According to [Daphne Koller](#), co-founder, the idea is to enable the best universities in the world to put their courses online, so that anyone with an internet connection can benefit from having free access to a great education.

[Coursera](#) provides courses in computer science, social sciences, mathematics, statistics, economics, finance and many more. More than 200 courses are currently available.

Since the information is available to anyone connected, it means that if we give access to 10-12 years old children, they will acquire the knowledge **if they want** - as [Seth Godin](#) could deduce.

It would not work so well for topics that they are not interested in because...



### **3. It is much easier to learn when one is involved in activities that they enjoy**

What I liked most about education in the West is freedom, students being free to choose the subjects they prefer and to leave out the options they do not like. There are very few restrictions, as long as one can earn enough credits through their selected courses.

The biggest advantage is that students are motivated to work, since they do what they really want to, as opposed to some systems where students are obligated to take on all the subjects on the curriculum, whether they like those or not. Being forced to learn things that do not interest them, they will probably forget those after the exams anyway.

Furthermore, researches show that those who find the answer themselves retain it better than those who are told the answer. What schools should do ideally is [to give children the motivation to get involved](#), to create an environment where children, driven by their curiosity, solve interesting problems, instead of simply memorizing answers to the problems that have already been solved.

This may seem unrealistic with regard to some traditional systems but...



**4. It's not because certain things were done in a certain way in the past that we should allow this to continue.**

The origins of traditional education come from the military to a large extent. They needed identical people, soldiers, administrators, etc. So they produced such a system. When the industrial revolution happened, they wanted identical people too in order to meet certain standards in factories and assembly lines.

So if we look at school that way, we note the fact that we process 30 or even 40 kids at a time, in a batch just like in a factory, while trying to maintain a common evolution for the whole group, often with the effect half of the class finds the classes too difficult while the other half gets bored.

And if you fail an exam, what do you do? We hold you back and we re-process you, it all matches the way a factory works. This methodology was really useful in the traditional context, but we don't have a shortage of factory workers anymore (on the contrary, unemployment rates have increased in some industries).

We take all these children and we force them to try to adapt to this really complex bureaucracy, but actually it is the system that should adapt to them!

When a kid doesn't conform to the system, he is pressured to fit into the system, as if there was something wrong with him. But in reality, the system is here for the kids, not the other way around...

Maybe things will change in the near future, maybe not. In any case, [nobody says that it will be easy](#). There will be challenges, but revolutions destroy the perfect to enable the impossible.

## **Africa, Cradle Of Humanity And Entrepreneurs**

By [Antoine Mian](#)



In addition to being the cradle of humanity, Africa is becoming the cradle of entrepreneurs. As a matter of fact, when it comes to the world rankings of the countries with the largest number of entrepreneurs, Africa plays the leading role!

While entrepreneurs in the United Kingdom and the United States are associated with start-ups, in some other countries, especially in Africa, entrepreneurship for freelancers is more of a necessity than a dream. Indeed, when there are not enough jobs available like in many African countries, you have to create their own opportunities.



[Approved Index](#), a UK-based research network has just ranked countries with the largest number of entrepreneurs. This number is calculated as the percentage of the adult population that owns or co-owns a new business and has paid wages for three months or more. This study shows that countries with the largest number of entrepreneurs are going through a commercial revolution.

Approved's ranAfrica: ICT in Education for Entrepreneurship and Employabilityh 13.7% of the population, Angola 6th with 12.4% of its population and Botswana 8th with 11.1% of its population.

Note that in this ranking, Great Britain is 37th with 4.6% and the USA 41th with 4.3% of the population that are entrepreneurs.

[Click here for the ranking of the 9 countries with the most entrepreneurs in the world.](#)

**[Interview] In The Bubble Of Antoine Mian: "if one is not passionate, they miss out on life"**



**1. Quick (but effective) introduction of the personage and his achievements?**

Bi Sehi Antoine MIAN, I am a teacher-researcher in ICT. Currently Assistant Director of Techno Pedagogy at the Virtual University of Ivory Coast. From my experiences as Ivorian Web actor and blogger, I was able to integrate social media as Teaching Units into the educational model of the Virtual University of Ivory Coast.

**2. What are the values that better define you?**

My main value is the consideration that I must always have for the human person. Man is at the heart of all my actions. Whether it be educational or social actions, I place the Human Being at the center.

**3. What tools and techniques do you use to accomplish things efficiently?**

I work extensively on free software. I do it because I view myself as a demonstrator. Through me, teachers and actors in the education system must realise that, with open solutions, many problems can be solved in the context of ICT integration.

**4. Any recommendations for your juniors?**

Life advice: to be passionate about what one does. Because if one is not passionate, they miss out on life. Never stop learning and opening up to others, it costs nothing but it can provide a lot. Everything that one does must aim at the development of the human person. This must be put at the center of everyday concerns.

**5. What is the best way to contact you?**

[Facebook](#), I would say.

## **HELP! They Are Going To Steal My Project!!!**

By [Nnenna Nwakanma](#)

Lately, I have been interacting with some major project leaders.

1. These are people who are confident about their projects.
2. They generally avoid discussing these projects with others, for fear of having their ideas stolen.
3. But they still needed a trustworthy advisor... And finally they decided that it would be Nnenna.

After some "maximum security clearance" considerations and "highly classified" discussions, I noticed:

1. Most of these project leaders are amateurs.
2. Most of these projects have not yet been properly detailed/documented.
3. The leaders in question are not able to pitch their projects clearly within 5 to 10 minutes.
4. These project leaders want a face-to-face meeting with me. They think I can boost their chances...
5. It seems that the environment in which they want to operate is unknown to them, for the most part.
6. Those who know a bit about the environment have less than 2 years experience.
7. 80% of these projects tend to view their country as the whole world.
8. I noticed a glaring lack of feasibility study. They rarely take into account the parties that are already on the field. I have not seen any competition strategy.
9. These project leaders lack relevant connections.
10. Half of these projects are illusions.



# Chapter 4

## Quick Wins, Tricks and Tips

Featuring Jean Luc Houedanou





## Where to Find Free Quality Audio Books?

By [Necemon](#)



An audiobook (or talking book) is a recording of a text being read.

Your relationship with "reading" is modified by the use of hearing, instead of sight which is traditionally attached to printing. Technological advances (mostly in terms of recording and sound reproduction), availability of new features and many other advantages (saving time, more mobility, less physical storage space, experience sharing, listening pleasure, literacy learning, etc.) contribute to the considerable growth of the audio book in recent years.

Simple to use, the audiobook attracts different age and socio-professional categories, often for different purposes, sometimes for opposite reasons: only solution for some categories of disabled people (visually impaired for example), reading comfort for the elderly , mobility and new training experience for younger people, saving time for the productivity geeks (can be used while commuting or doing other physical tasks), listening pleasure and user-friendliness...

If you are looking for free audiobooks to download, here are some websites I can suggest for your listening pleasure.

### [Librivox](#)

Free public domain audiobooks. More than 10 000 audiobooks available, read by volunteers from all over the world.

[Project Gutenberg](#)

The oldest digital library (established in 1971).

[Open Culture](#)

900 free audio books from one the best cultural and educational media on the web.

[Thought Audio](#)

Free audio books for a selection of classic literature and philosophy titles.

[Learn Out Loud](#)

Your destination for audio learning. Thousands of the best free audiobooks online.

[Digital Book](#)

Browse thousands of free audiobooks. Either download entire audiobooks to listen offline or use their simple online audio player to listen directly in the browser.

[Loyal Books](#)

Free Public Domain Audiobooks Downloads. About 40 genres.

[Free Classic Audio Books](#)

Digital Narration for the 21st Century. The downloads are available in mp3 and in m4b formats.

[Educational Technology Clearinghouse \(Lit2Go\)](#)

A free online collection of audiobooks in mp3 format. An abstract, citation, playing time, and word count are given for each of the passages. Many of the passages also have a related reading strategy identified. Each reading passage can also be downloaded as a PDF file.

[Many books](#)

About 2000 audiobooks. Browse through the most popular titles, recommendations, or recent reviews.

[Youtube](#)

Great content, both in terms of quantity and quality.

If you understand French (or are trying to learn it), you may also want to check [these recommendations](#).

Wishing you a good listening time.

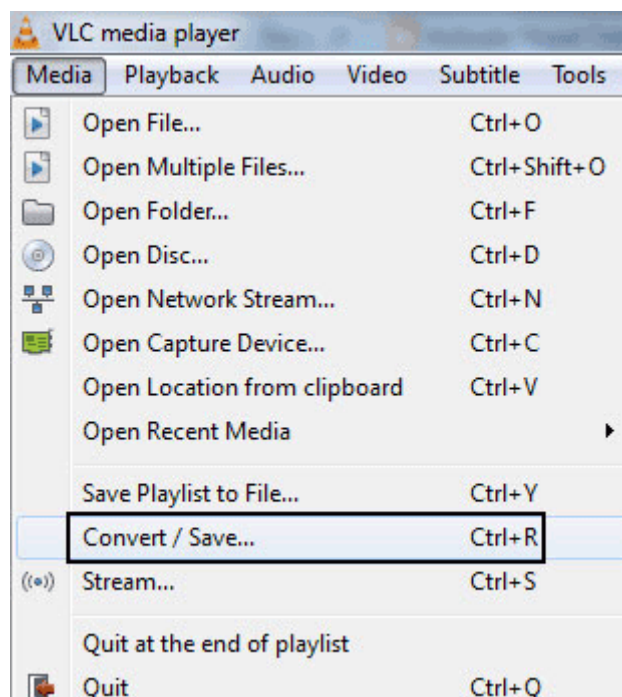
## Convert Video Files With VLC

By [Jean Luc Houedanou](#)

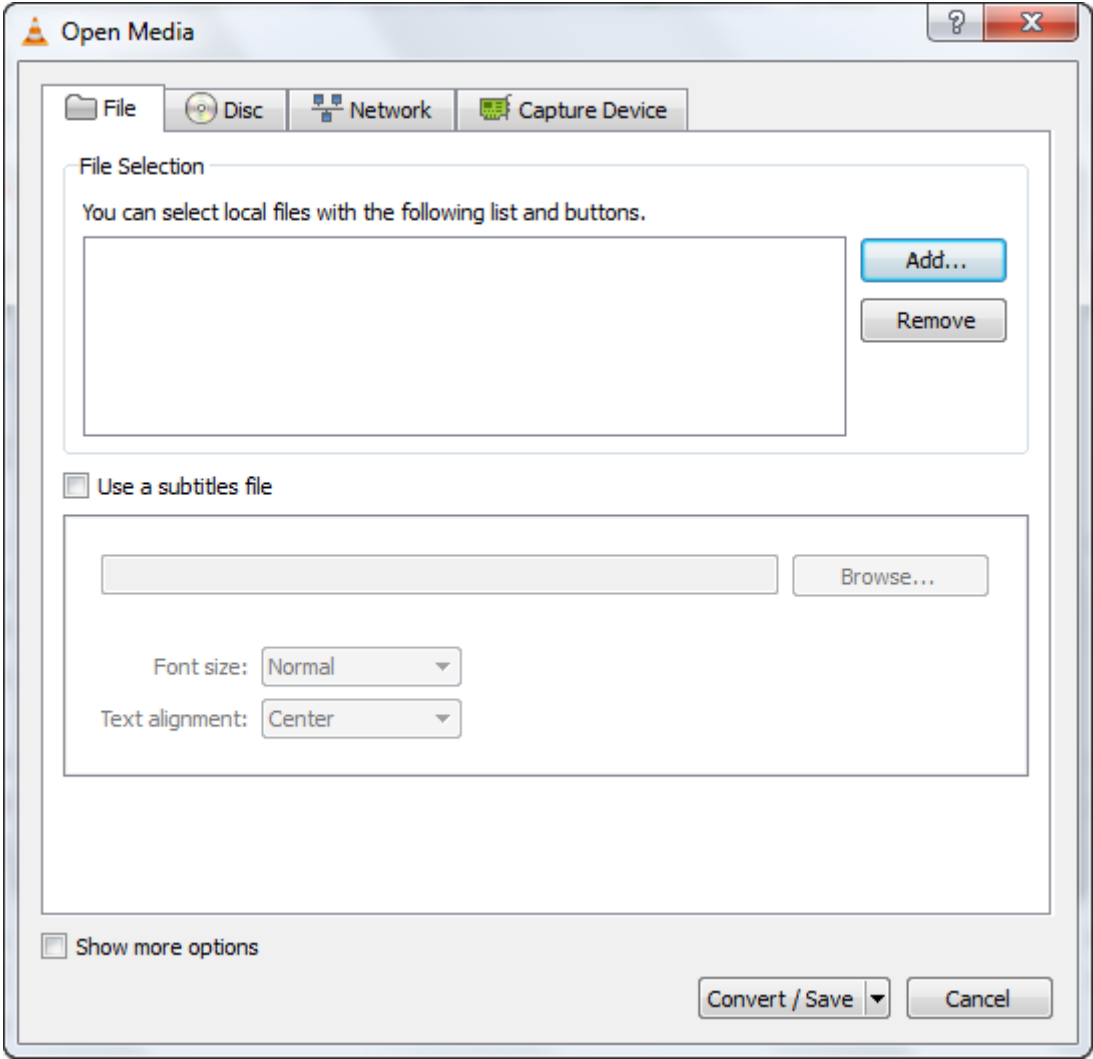
How to use the best free video converter/encoder available on all platforms, also known as...

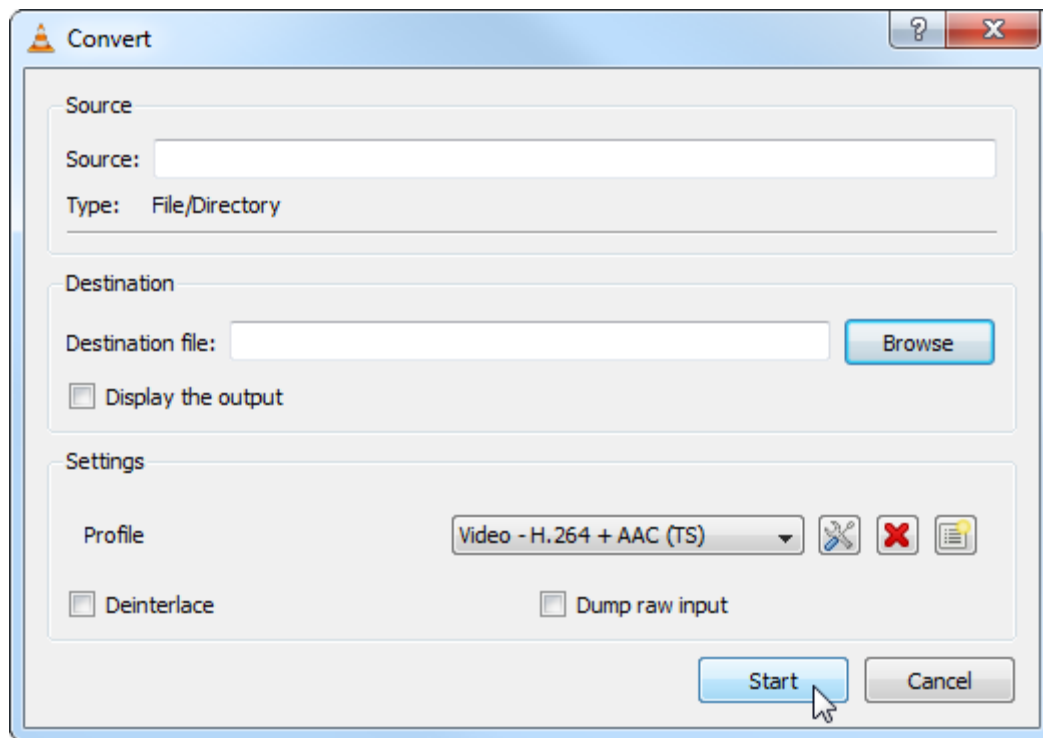
VLC. That's right, in addition to being a true Swiss army knife for video playback, VLC has a video-to-video (and audio) conversion functionality that is little known to the general public. Here is the procedure for using it:

- Go to the Media/File menu and select Convert/Stream (on MacOS) or Convert/Save (on Windows)



- In the first field, select the file to convert, then the format/profile of the converted file and finally click Convert/Save/Start.





That's all.

Admittedly, VLC does not allow batch files to be processed and it does not provide conversion presets (Youtube format, iPod format, etc.).

But considering its price, it would be outrageous to ask for more.

## [Interview] In The Bubble Of Jean Luc Houedanou: "a fleet of drones that prevent theft and abuse"



### 1. Quick (but effective) introduction of the personage and his achievements?

Jean Luc Houedanou, currently Head Of Design in a startup called WeDev Group and having 7 years and several months of experience as a web consultant (including CERAP Inades, as well as the Electoral Reform International Services). I have a Masters degree in E-commerce and a diploma in [Information and Systems Management](#), both obtained from the University of Sherbrooke in Canada.

In terms of skills, I am a web integrator and a web design/strategy consultant. I am also involved to some extent in SEO and Training.

My achievements that I am most proud of are related to the field of Training: I have had the opportunity several times to train journalists or media professionals in the use of social networks, a web radio deployment, content creation and [blog management](#). It's a real pleasure to see every person negatively biased towards these technologies who ends up fully embracing them afterwards and even becoming advocates of journalism 2.0.

### 2. What are your main goals in life?

My goal by the end of 2018 is to write an e-book on WordPress template design and also to launch a podcast on information technology and (my second passion) computer-aided music. I started writing the e-book and regarding the second objective, I am in partnership with a Web TV with adequate equipment. So I am well on the way.



The idea behind these two projects is the same one that motivated my return to The Ivory Coast: to promote a more reasonable and more productive [use of information technologies in Africa](#). There are several problems in Ivory Coast, in Benin (my home country) or in Senegal (my country of birth) which would find a quick solution if young minds of the continent mastered the information technologies better: for example, I have been working for several months with young people who developed WeFly Agri, a plantation management software coupled with a fleet of drones that prevent theft and abuse on plantations and increase productivity.

Africa is a fertile ground for innovation, so I try modestly to plant my seed in that area.

### **3. What tools and techniques do you use to accomplish things efficiently?**

Software: Gimp, Inkscape Sketch, Affinity Designer, Sublime Text, AMPPS, IA Writer, OmmWriter, Firefox Developer Edition, Emmet LiveStyle, FileZilla (a must), iMovie (yes, iMovie, it's a perfectly capable editor) as well as GarageBand

Gadgets: Macbook Pro (for ease of use) and related devices (magic mouse and keyboard). I have an entry-level Android smartphone (Tecno W5 lite) and supplementary, a MyKronoz Zefit3 that I treated myself to for Christmas, in order to remind myself to be less sedentary and do a little more physical exercise.

Technologies: ReactJs, html, jQuery, Gulp and Compass

### **4. Any recommendations for your juniors?**

In terms of podcast - the [Daily Tech News Show - www.dailytechnewsshow.com](http://www.dailytechnewsshow.com).

### **5. What is the best way to contact you?**

[@jlhouedanou](#), on all social networks, except on Facebook. On that latter social network, my address is [facebook.com/bloghouedanou](https://facebook.com/bloghouedanou)

## FFmpeg, The Fastest Way To Edit a Video File (in 5 seconds or less)

By [Necemon](#)



What's your problem?

You want to remove or extract a part of a video.

You could download, install and/or buy some advanced editing software which has such feature (like Adobe Premiere Pro), but it takes a lot of time and energy, especially if you are working with several, long videos and the software has to decode and re-encode the files every time.

The solution: FFmpeg.

[FFmpeg](#) is a complete, cross-platform solution to record, convert and stream audio and video.

Ffmpeg has a simple command that lets you edit the file as you will, almost instantaneously. It also works for audio files.

1. Download the package

=> [Windows](#)

=> [Mac](#)

=> [Linux](#)

2. How to use it ?

Just run the following line from your command prompt

```
ffmpeg -i InputPath.mp4 -ss 00:10:00 -t 00:00:05 -acodec copy -vcodec copy OutputPath.mp4
```

Where -i is the path to your file.

-ss is the start position in Hours:Minutes:Seconds

-t is how long a clip you want - again in HH:MM:SS

The acodec and vcodec tells ffmpeg just to copy the file without changing its other audio and video properties.

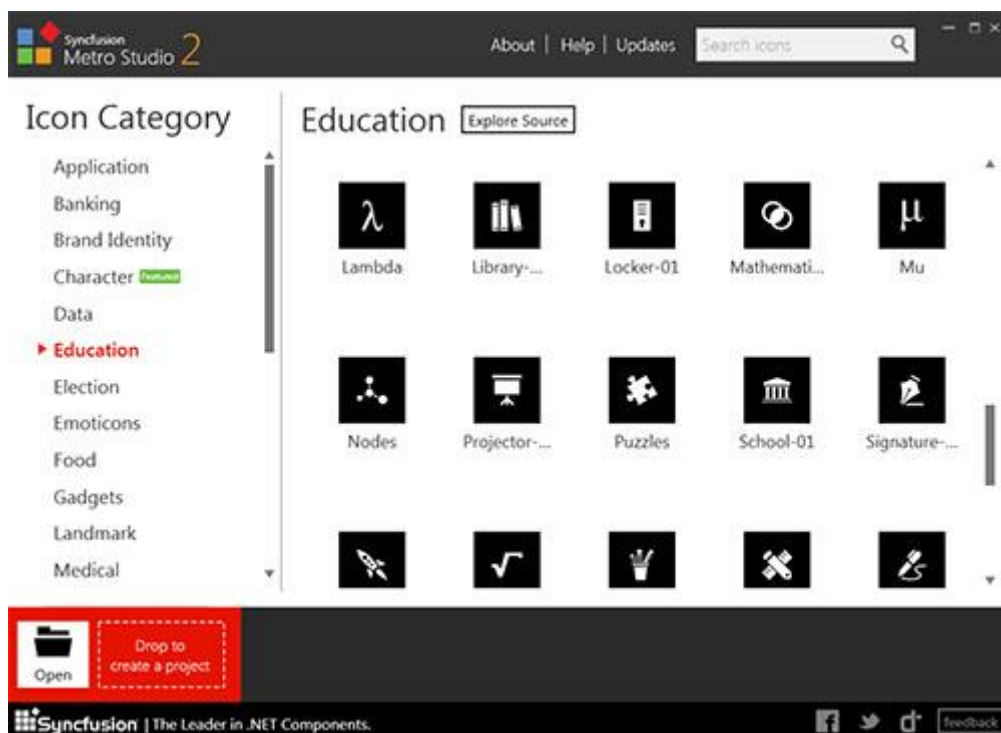
That's about the fastest you can get it done.

## Getting started with Metro Studio (or how to benefit from the largest collection of Metro-style icons for free)

By [Necemon](#)

[Syncfusion Metro Studio](#) is a tool which provides a collection of over 2500 [Metro-style icon templates](#) that can be easily customized to create thousands of unique Metro icons. The icons are available royalty-free and can be used in [open source](#) projects as well as [commercial](#) applications.

To get started, just [download and install MetroStudio from the Syncfusion website](#). As soon as you do this and run the program, you can see a wide range of icon categories from (mobile) application to data, to education to emoticons, etc.:



You can directly browse the icons files (.svg) through your file browser by clicking on the "Explore Source" button next to the category title. But MetroStudio is not just a browser for a collection of static files, it's also an editor. Actually, an Edit button appears while hovering any icon. By clicking on it, that icon can be customised at will and many properties are editable : dimensions, background shape, background color and icon color to name a few.



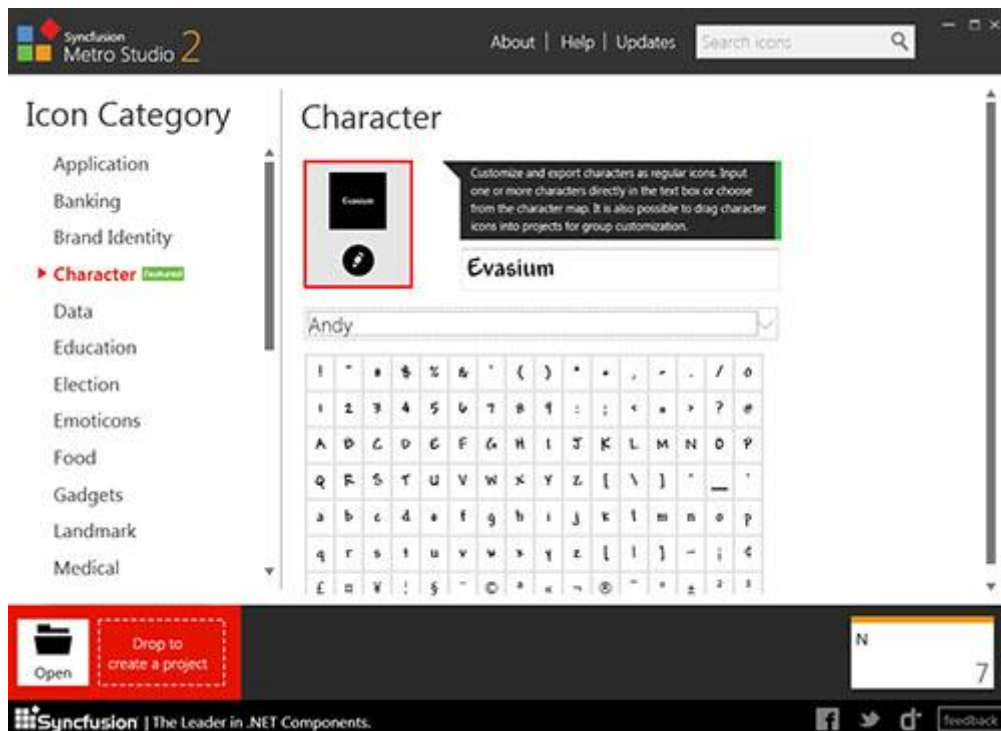
Your selected (and customised) icon can be copied to other applications, saved as xaml data or exported as picture (.ico, .jpg, .png, etc.). The collection is therefore suitable for many types of projects, from PowerPoint presentation to software products including [WPF apps](#), [Web apps](#), [games](#), [Mobile apps](#) and [websites](#).



With a simple drag-and-drop move toward the bottom-left corner, you can also edit multiple icons at the same time by organising them as a project.



Finally, MetroStudio deals with text styling. Easily customize and export thousands of font characters as regular icons. Input one or more characters directly in the text box or choose from the character map. It is also possible to drag character icons into projects for group customization and export them to the desired formats.



It's a very complete tool and here is where it gets even cooler: the product is worth **\$499**, but you can [download it for FREE today at syncfusion.com/downloads/metrostudio](https://syncfusion.com/downloads/metrostudio)



## Instant Messaging For Your Website : Have it the Giant Way

By [Necemon](#)



You have probably used the in-browser Instant Messaging service of one of the Internet "giants", whether it's Gmail, Msn or Facebook. You know, that small window with your online contacts at the bottom corner, which can be extended into conversations. Wouldn't it be cool if you could have a similar system right into your own website? Well, now you can, that's what the new Mango Chat is all about.

Mango Chat is a small, fast, cross-browser and full-featured chat component which allows you to add chat functionality to your ASP.NET website or web application. It includes features such as high load support, font/color customization, emoticons, and many more! The best thing about Mango Chat is that it's a web control which would integrate in your website as easily as a simple drag and drop.

Mango Chat uses JQuery/Ajax and makes sure that minimum bandwidth and server usage is utilized. It can easily support high volume of users without any degradation in response time. Further it supports multiple options for data store (in memory, XML, SQL Server, etc.). So Mango Chat can help you connect your website members regardless of the data store you use.

And as I mentioned before, it's a component that runs in the bottom right end corner of your website screen (as on Gmail). An obvious advantage is that it doesn't take too much space and let the user make some other operations on the same webpage.

Mango Chat comes with a [premium unlimited license with full support](#) for a price of \$999. But wait, there is also a [basic license](#) that costs only \$99.

Go get Mango Chat now at [www.mangochat.net](http://www.mangochat.net)

PS: Don't take my word for it. See Mango Chat in action at [demo.mangochat.net/testoptions.aspx](http://demo.mangochat.net/testoptions.aspx). You can also download and try a demo version with no time limit at [www.mangochat.net/Download.aspx](http://www.mangochat.net/Download.aspx).

# Chapter 5

## Transition: Extra Thoughts and Sharp Fantasy

Featuring Ahou The African Chick



## The Effects of Accelerated Evolution and Information Overloading: did you know?

By [Necemon](#), 2009



As I am writing this:

There are 255 million websites and 152 million blogs online, 35 hours of video are uploaded to Youtube every single minute, 119 millions tweets are posted every day...

It is estimated that a week's worth of the New York Times contains more information than a person was likely to come across **in a life time** in the 18th century.

3000 scientific articles are published every day, that's about one every 10 seconds.

My point? There is much more information that is getting produced than one person can possibly absorb. And it just gets bigger and bigger. We are living in exponential times...

## Eponymous Laws For Tech Strategists

By [Necemon](#)



### **Parkinson's Law**

Work expands so as to fill the time available for its completion

In other words, if you have a certain amount of time to complete something, that's how long it will generally take.

### **Murphy's Law**

Anything that can go wrong, will go wrong (at the worst possible time)

Most ideas look great on paper. But by the time events go from A all the way through the alphabet to Z, you see who rules when it comes to getting something from the theoretical stage to the practical stage : it's Murphy's Law.

No battle plan survives first contact with the enemy. Hope for the best, be ready for the worst.

### **Murphy's Law for DevOps**

If thing can go wrong, it means it already went wrong but you didn't get a notification from the monitoring tool yet.

### **Moore's Law**

The power of computers per unit cost doubles every 24 months.

Because of the nature of this law, anything that an extremely clever programmer can do at one point can be replicated by a merely competent programmer some number of years later.

### **Hofstadter's Law**

It always takes longer than you expect, even when you take into account Hofstadter's Law.

**Brook's Law**

Adding manpower to a late tech project makes it later.

The bearing of a child takes nine months, no matter how many women are assigned.

**Kranzberg's law**

Technology is neither good nor bad, nor is it neutral.

It's powerful. And it's complicated.

**Pareto's Law**

For many events, about 80% of the effects come from 20% of the causes.

## Success, A Reward To Enjoy In The Shadows

By [Ahou The African Chick](#)



Success has several meanings in the dictionary. It may be about a happy outcome, a job, a sporting event. For a public figure, it may be a favor, a public hearing. It is also about pleasing someone, seducing someone or achieving a desired goal. In all these definitions, one thing is certain anyway, one must work hard to achieve a marked success. But today, success is regarded as the result of practices which are unorthodox, shady or even occult. Anyone who gets successful is systematically accused of having made a deal with the devil.

Nowadays, in many people's mind: "Once someone has been dirty, they can never get the right to become clean". Some (mostly public) people have seen their past resurface and be exposed to public scrutiny. We don't have anything better to do than to draw upon people past history and mistakes, hence giving rise to avalanches of accusations that are all equally wacky.

I wonder if we are no longer allowed to succeed on this earth; if we are bound to die poor and unhappy; if we should remain in a state of failure instead of trying to improve ourselves.

As of today, I, Strella, am neither the daughter nor the protégée of anyone. I have been knocking on every door while hoping for help, but those doors usually remain closed. Nevertheless I persevere and I continue to move forward. I struggle as much as I can. Sometimes, I work without being paid. While I am in the face of adversity, no one shows up, there are no rumors about me, there are no nude pictures or videos of me anywhere.

But tomorrow, when I will start taking off, just a little bit, even my mother (may she rest in peace) will come back to life and will suddenly get some "twin" sisters. I will be everyone's protégée. And they will not stop there at all... Some will go as far as suddenly finding photos or videos exposing my anatomy. This is the part where I am told that I am dating all men in the whole world and that I am the mistress of the president

himself. And that my success is due to my village's mascot or to a magic cauldron that I laid at some mystical crossroads. Even though I worked hard to get there. Of such fantasy are ignorant people capable.

When you are going through a tough life, nobody hates on you. They do not even pay any attention to you. You are neither the daughter, nor the protégée, nor the friend of anyone. But look at how some people react towards the success of their peers. They would start hating on you for any reason or for no reason. It's like you started a world war. They will dig into your history, bring up your past mistakes, tarnish your reputation and wage war against you. It's as if people would like to see you remain in poverty, debauchery and idleness.

If a young girl had been sleeping around with the whole neighborhood, and then decides to settle down, that would provoke a big "international outcry", especially if some guy decides to marry her despite her controversial past.

If a person deprives himself of distractions, works hard and ends up succeeding, some will still remind him that he comes from a modest background or whatnot.

There are people who commit to expressing their arts. In the event that they produce some popular hits, their victory is attributed to their presumed pact with the devil. But when their projects fail to take off, no one would assume that the devil intervened. If a producer or a singer works hard, it makes sense that his music becomes fine. It's that simple. It's possible that some people may resort to the occult sciences, but let's not generalize.

Enumerating people's past mistakes leads nowhere. You better make a proper use of time by working, investing or thinking about a business, so that you will achieve success too.

Not all successes come from the devil, God is good and generous. If a person keeps learning, it's quite normal that they become better. So let's stop worrying about others and let's try to move forward in our own projects, because while you spend your time tracking their flaws, other people evolve, get married, have children, they are happy. While they become independent, you still live with and depend on your parents. Let's appreciate the success of others, because success also comes from God.

At the sight of all this hurly-burly around the success of people, I humbly conclude that it's better to stay in the shadows with your success in order to avoid seeing your past come back and being called names. Therefore, before proclaiming your success and advertising your victories, think of all the collateral damages you could be creating for yourself. That may imply having your origins and your past exposed to the entire world. Choose carefully with whom you want to share your success, or else...



**[Interview] In The Bubble Of Ahou The African Chick: "I like to go beyond my limits, I like to be curious and try new things"**



**1. Quick (but effective) introduction of the personage and her achievements?**

My name is Bénédicte Kouakou or, if you wish, Ahou l'Africaine (Ahou The African Chick, my blogger name).

Holding a Master's degree in Judicial Law obtained from the Catholic University of West Africa (UCAO), Ahou is passionate about digital communication and content writing. She worked as a Social Media Manager at YooMee, Bogolan Distribution (for the "Living Together" project) and RCG West Africa where she also worked as Communication and Event Assistant, on content writing and digital strategy development. Hence she has accrued 4 years of professional and corporate experience.

Unsatisfied with her "brief" internship in a law firm, and her [4 years of business experience](#), Ahou will combine her two passions, which are the Law and Digital Communication, by creating [The Blog of Ahou The African Chick](#) in May 2015... I have been fighting for what I wanted. I have not yet fully realized my ambitions so far, but for me, these have been great experiences.

## 2. What are your main goals in life?

My main goal is to see every African woman getting interested in their Rights, because very often they don't fight enough to claim those and prefer to settle for very little. My other goal is to see more women at school, and getting involved in areas such as law, medicine, politics and whatnot; i.e. more women who do not restrict themselves. In my view, women are potentially able to do much more, beyond being a wife and a mother.

My project is to organise Masterclasses in order to educate women and teach them about their rights, and also to survey/address their specific legal needs.

I like to go beyond my limits, I like to be curious and try new things. I like loyalty and to stay true to myself. I like to believe in others because we all have talents. I plan to be a recognized blogger, and to participate in debates where Women Rights are a priority, and to make my blog a reference in women's empowerment.

## 3. What tools and techniques do you use to accomplish things efficiently?

My tool is my computer (I'm thinking of getting another one, most certainly a Mac) because it allows me to write most of my articles, my phone (I have to change it too lool), my mobile Wifi for connection, Canva (for my occasional visuals), a camera (that I would like to have lool)

Google helps me a lot (Google Drive, Gmail, Google Translate, Youtube, etc.)

Social networks (Facebook, Twitter, LinkedIn, Instagram) to be [closer to my subscribers](#).

As for my methodology, I have a notebook where I list my pending tasks (along with their related deadlines), which I eliminate as and when they are completed.

## 4. Any recommendations for your juniors?

My recommendation is to believe in yourself and in what you do. To give oneself the means to realize their dreams.

There is a quote that I like: if life gives you lemons, make fresh lemonade!

My favorite book is the Bible. It is filled with messages of life that lift up the spirits for anyone who needs to refresh themselves. Also, "We are all feminists" of Chimamanda Ngozi Adichie.

I also read websites like Femme d'Influence Magazine, elle.fr, Aufeminin.com, and some other female motivation websites. I follow the blogger Fatou Ndiaye who is a beautiful inspiration, The chronicles of Sapitou, etc. I also listen to Zack Mwekassa. I follow many others, but well, I could not name them all.

I read various legal codes: criminal law, civil law and anything related to Law!

### 5. What is the best way to contact you?

You can reach me on most of my platforms

Website: [www.ahoulafricaine.com](http://www.ahoulafricaine.com)

Email: [ahoulafricaine@gmail.com](mailto:ahoulafricaine@gmail.com)

Social networks: [Twitter](#), [Facebook](#) and [LinkedIn](#) : [Ahou L'Africaine](#)

## The General Assumption of Security

By [Necemon](#)

According to some [Hacker News](#) folks, the general assumption of computer security is:

- The attacker is smarter than you
- He has a bigger computer
- He knows your own software better than you
- He is after you, specifically.

Well...

## World Wide Worries

By [Necemon](#), 2010

Wherever you move on Internet, you are often forced to leave [some information about yourself](#). To access a service, you may be asked to give [your real name, phone number, address](#) and so on. They call that required fields. Either you fill the form or you just leave it. But a lot of optional information is also intentionally released by some users.

The current trend is to [share private information and photo in an intimate way](#), with family and friends via social networks. Even some people seem to enjoy displaying this personal data over the public internet. Nowadays, you can tell your own story to the world, describing your day-to-day activities as it happens (e.g. via Twitter). Some say that it's fine. It's also ok if some people get an access (somehow) to your [Facebook wall](#) and check your pictures. It seems that if you didn't do anything wrong, you don't have anything to hide so it's ok if people see it.

I don't know how much of that is true but I observed that on the other side, [some people are very careful](#) about what they display over the internet. They never connect to you if they don't know you in real life or they simply avoid social networks. If they let you access their profile or resume, don't even think about seeing their phone number or address. You can never find a picture of them anywhere online and when you get a chance to chat with them, they always use some nickname.

And in the middle of those two philosophies, there is the kind of average guy, who is a bit worrying about his privacy but still publish some photos from times to times; he blogs using his real name, accepts friendships from strangers on Facebook or Hi5, probably hoping to become popular someday, or whatever. I guess you see the kind of person...

So who is right? What is the right attitude toward [the public Internet](#)? Should we be worrying that a stranger can see our pictures (if so, why)? What is the actual risk of using a real name when interacting online?

UPDATE: a few months after writing this (and after a bit of research), I am starting to figure this one out. There is certainly a risk, [exposure is a sort of sacrifice](#), as long as you know exactly what you lose and what you get, maybe it's ok.

## The Golf Ball Effect

By [Necemon](#)



What is the Golf Ball Effect?

First, note that small causes can have huge consequences. [Butterfly Effect?](#)

When hitting the ball, a difference of a few angles will result in a big difference in the position where the ball will land.

As in a triangle ABC, if a point X is on [AB] and a point Y is on [AC], the further x and y are from A, the bigger is the distance [XY].

Therefore, a minor shift of 2 degrees could drive the ball yet another 10 meters away from the target (or closer to the said target).

That's just a metaphor for the small changes, small factors and small decisions in life that can make a big difference on the long term.

## Just 2 Things

By [Necemon](#)



When meeting someone, anyone, especially for the very first time or for the few first times, whether it's online or in person, I think there are just 2 essential things to consider:

- The **harmony** factor : whether the person is friendly or hostile (and to what extent)
- The **skill** factor : whether the person is strong or weak, that is, in the areas which are relevant to you (and to what extent)

Those 2 critical factors (or the way they are perceived) would determine the direction of the relationship; depending on the context, other parameters may affect the next steps, but the general idea is the following.

If the person is:

- **Friendly** and **strong** in your areas of interest: your next move could be to try and figure out how you could collaborate, what you could do together, etc.
- **Friendly** and **weak** in those respects: you could then try to figure out how to help them improve and evolve.
- **Hostile** and **strong**: this may trigger your defensive system or make you consider being relatively careful, watching out for opportunities while avoiding any crisis.
- **Hostile** and **weak**: in many cases, it may be ok to simply ignore them.



## Games for Everyone

By [Necemon](#), 2010

We are living a terrific OS wars with the rise of Linux over the last two decades. The big Google is also launching its Google Chrome OS. Microsoft and Apple are far from throwing in the towel with the Windows 7 and Mac OS X respectively. Well, as a [wannabe game programmer](#), I feel it is getting difficult to choose a side. The will to reach the maximum audience could lead to code mainly for windows gamers. But then, there is still a trouble. Let's talk about a very exciting field of gaming: Multiplayer online gaming. Suppose you build your game client using any Microsoft .net language, say C#. Are you compelled to code your server application in C# as well? I am asking because a C# server app will normally need to be hosted on a C# enabled web server (IIS is all I can't think of at the moment).

See, the thing is in the game industry, I know they use a lot of Linux servers. But worldwide most of the gamers are running Windows. So I think either they are using the same language on the server and on the client (like C or Java) or there is a sort of conversion. I wonder how that conversion happens (for now, I would bet on web services).

Well, I don't know how much [C# is used in the game industry](#) but how can it be used in collaboration with another language that would make the server app? Anyways which language would be suitable for a game server? Last doubt on this post: is it feasible to build a large scale game server (like a MMORPG) using C#? (I assume this would mean running it on IIS)

UPDATE: It's funny how, about 2 years after writing this post, I naturally got answers to my doubts. Yes, I think I am getting it now. I should probably write about what I found some time. More on that later (maybe).

# Volume 2

# Full Moon

(Hardcore)

In this volume:

Chapter 6: Software Development and Engineering

Chapter 7: C# .NET Programming

Chapter 8: Epic Prototypes, Classic Projects, Historic Genre

Chapter 9: Research and Case Studies

Featuring Darren Mart, Holty Sow and Jean-Patrick Ehouman

# Chapter 6

## Software Development and Engineering

Featuring Darren Mart and Jean-Patrick Ehouman



## 8 reasons why you would enjoy being a programmer

By [Necemon](#)



If you are a programmer, there are probably a lot of reasons that motivate you to do what you do. I hope you find here some more motivations.

If you [want to be a programmer](#), you may find here some more reasons to go for that awesome path.

However, I guess I am writing this mainly for [those who don't know what to do with their life](#). Here are some clues about a job that is fun, useful and satisfying to many extents.

From the top of my mind, here are 8 reasons why you would enjoy being a programmer. I hope this inspire you.

1. Programming makes your dreams come true. When you understand programming, you can [give life to your thoughts](#) by applying them to real life. You can literally create things.
2. Programming is the [ultimate form of interactive art](#). You can make software, websites and games that others can play with. So you can talk to them indirectly and they can talk back. No other art form is quite this interactive. While drawing, painting, movies and music go to the audience (in one direction), code goes both ways.
3. That's the kind of job you can do from anywhere. From your couch, from home, from office, on travel, whichever country... The only things you need are a computer and your brain.
4. It's easy to learn. There are [tons of ressources available online, many of them are free](#). Also, there are many online communities that can support you through forums, chats, emails, etc.

5. You don't have to rely on anyone to do programming. You may have a chance to do it [for a company or for a research programme](#). Even if it's not in a corporate job, you can work in a team. And even if you don't find a team that fits you, [you can work by yourself](#). Also, you don't need much money to get started.

6. You can't have enough of it. You can't get bored. [Requirements and technology](#) are moving up so fast that you always face new challenges.

7. It's a field of meritocracy. They know you don't fake your skills. You [know what you know](#). You do what you can do and you get a fair recognition for it.

8. It's fun !

N.

## Writing and Programming: Pretty Much The Same Thing

By [Jean-Patrick Ehouman](#)

5 years ago, if I was told that I could [run a blog](#), I would not have believed it. As a [software engineer](#), I was spending more time [designing and writing programs](#). However, a priori, nothing suggested that I [could write good articles](#) too.

In hindsight, I can say that these two activities have more similarities than differentiation points. When you write, you create, you fill a blank, you give life. Similarly, when you design a program, you create a system that will be used on a daily basis. So in both cases you need to deeply understand the reader or the user of your creation.

Putting yourself in their shoes leads you to be imaginative and creative like a painter or a pianist. So, when I am asked about the [fundamentals required](#) to learn how to make good computer programs or software, I ask the interested party if he has ever written a letter, a short story, an article, etc. If the answer is "yes", then the person already has prerequisites to learn how to write programs. If not, they can still try to write [a short story](#) and assess their abilities to create or innovate.

[Coding is just writing.](#)

## On Technical Orientation : 5 Basic Considerations When Starting To Code

By [Necemon](#)



A [fresher](#) sent me the following query:

*Hi Necemon, thank you for accepting my invite. I am new to the IT field and I would appreciate if you could give me some guidance. Based on your work experience, can you tell me what companies look for in a computer guy ?*

*I heard about you from senior students at [Christ University](#), I am in Bangalore and I like computer science but I do not know what to learn and how to begin.*

*Actually, I think [I like programming](#) but I am told that the C language is no longer relevant, and [I am also told about Ruby, C#, Python, etc.](#) I'm confused.*

Please don't get confused. Technology is simply a way to solve a problem or to achieve a goal. What is your goal?

Create apps? What apps do you want to create and why?

It's a bit as if you come to me to ask me what vehicle you should be driving. If I ask you what you want to do with that vehicle, you wouldn't just tell me that you just want to move away, right? I know you want to move... My question is, where are you going?

What companies are looking for ? Ok, I fully understand what you are asking here. You want to make sure your education will guaranty [an interesting job](#) later in the [IT development Industry](#). Obviously, I could tell you that a certain technology T is in high demand right now, but it's not that simple. There are a few other things to consider:

1. The requirements may vary with location (country or region). The hottest jobs in the US are not necessarily that popular in India. So unless you know already where you are going to work, it's not that easy to target on a trend basis.
2. The demand changes with time. What is relevant today may not be (as) prevalent tomorrow. The technologies evolve and replace each other. So what's fashionable now might be different from what will be popular by the time you get your degree.



3. You might not like the technology in vogue or the uses of that technology. If I tell you that a given technology T is in high demand, that it allows you to locate and fix bugs / errors in a super boring / huge / complicated banking system, plus there are plenty of calculations... What if you don't like calculations? Are you still going to embrace this technology and to accept this path for the rest of your career?

4. Even if we consider only one city and a given time, various companies are seeking different things, depending on what they do. There is no perfect technology that is better than all others in all areas. Each technology has its own sets of advantages and disadvantages. C and C++ may be better than Ruby at a few things (and vice versa), Python is better than C# in some respect (and vice versa), etc.

5. As I said above, the technology is just a mean to get somewhere. When you consider [Facebook](#) for example, most users don't care if it was built with [PHP, C, Java, Perl, C# or Python](#). What is important for people is, how the site or application can help them in their lives.

I think that's where you should start. What strengths and assets do you already have? (don't tell me you don't have any). What contribution do you intend to come up with for your family, your friends, your community, your country, and for the world? And what do you expect in return?

If you do not know what to do with your life, some time ago I wrote an article that might inspire you : [What will you do in life?](#) Take time to reflect on your ambitions and we can talk about the resources you will need.

If you know WHAT you want to do, it would be easier for me to tell you HOW to do it.

Let's speak soon,

N.

**UPDATE - [Shabbir Kahodawala](#) shared a few clever insights on this matter :**

*I agree very much with your response to the fresher - about the need to realise and work on his talents and concentrate on writing good code and immersive UI.*

*I would like to add a few points as well taking the perspective that every Indian student goes through the same dilemma due to lack of job oriented education, because institutions focus on technical oriented education.*

*IT is not only about writing code. Think of it as a factory where there is Marketing, Client Requirement gathering, Planning, Product development, Product Testing, Infrastructure planning, Product Deployment, Product Maintenance, Customer Support, Issue Resolution and Product Improvements.*

*Each of these creates many job opportunities for IT students and one needs to understand where his strengths lie. He can do that by talking to IT professionals who can introduce to other professionals in each department and willing to share what skills they require on today's world. That will make his goals more clear.*

*Secondly, a competitive IT professional should always have his basics right. To be able to write good and neat code. He should be always able to visualize a requirement into an algorithm and then the language syntax that he uses can always change. That's why colleges teach C & Java (object oriented) because these help a coder develop his basics about Data Handling, Functions, Objects, Classes and runtime environment.*

## 4 Substantial Tactics Guaranteed To Boost Your Game Creation Skills

By [Necemon](#)

A user of [one of my web applications](#) sent in the following question:

Hi my name is T. and I am 15, when I am older I want to be a part of the gaming industry and I am just wondering can you guys help with that, can you give me any tips or any experience?

Would posting to your website help me out in this? I would love to hear your reply, it would really help me out.



Here was my reply:

There is a lot to say on the topic but for today I will try to keep it simple and give you some practical advice you can start using right away.

**1. Read a lot on the topics you are interested in.** Read every day if you can. There are tons of free resources available online, some of my favorite sites and blogs for game creation are:

[gdne.ws](http://gdne.ws)

[lostgarden.com](http://lostgarden.com)

[procworld.blogspot.co.uk](http://procworld.blogspot.co.uk)

[whatgamesare.com](http://whatgamesare.com)

[gamestudies.org](http://gamestudies.org)

[designer-notes.com](http://designer-notes.com)

[higherorderfun.com](http://higherorderfun.com)

[gamecareerguide.com](http://gamecareerguide.com)

[webwargaming.org](http://webwargaming.org)

[raphkoster.com/gaming](http://raphkoster.com/gaming)

[gamedevelopment.tutsplus.com](http://gamedevelopment.tutsplus.com)

[nwn.blogs.com](http://nwn.blogs.com)

2. **Establish specific goals.** You may have noticed that in point 1, I only gave you a small portion of what's out there, but it's quite an extensive read already. [We are living exponential times](#), you can't do and learn everything by yourself at once. You have to be specific about what you want. Vague plans provoke vague results. What is it that you want to do in the video game industry? There are so many options, you may want to do some research on your options.

3. **Pick your stars.** To bounce back from the previous point, after you make a list of the skills you want to master, do some research on people who are already really, really good at that. Find some role models, see how they started and how they made it. Combine their tricks and throw in your own style to develop your own skills. It may take months but if you practice consistently, you will get really good too. Some of the people you can check out:

Game development:

- [Shigeru Miyamoto](#)
- [Hideo Kojima](#)
- [Notch \(Markus Persson\)](#)

Digital artists:

- [Akira Toriyama](#)
- [Masashi Kishimoto](#)
- [Monty Oum](#)

Scenario writers:

- [J.K. Rowling](#)
- [Stephen King](#)
- [Tom Clancy](#)

4. **Make a game.** No, [you are not too young](#) to start. It doesn't have to be anything big and right now, you can at least start learning. The earlier you start, the more time you will have to practice and the sooner you will become excellent. You can start with something very basic or at least start reading about it. If you want advice on how to start, how to find tutorials and resources, I can help.

On a related note, I made a game myself : [babifraya.com](#) It's not great and I am working on an improved version. But the important thing is [to start and keep practicing](#) :-)

In short, educate yourself consistently.

As to whether posting to [degammage.com](#) would help, my honest answer is: maybe.

You see, it's a new project with a small community and it's too early to say how successful it will be.

However, there will be some fresh content everyday, so there will be things for you to learn, and you may also visit [flux.evasium.com](#) which is even more into virtual worlds, education and technology.

Your [Evasium](#) account will work on there too.

Additionally, you may want to join other game forums and communities, like the ones I listed above. The more you learn, the better!

## 10 Years of Programming

By [Necemon](#)



I [started programming](#) when I was in high school. I was taught Pascal and HTML among other things. But I was also taking some summer programming classes in an IT institute. That's where I learned Visual Basic and software/database design (during those days, we were using MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise)). I eventually got Visual Basic installed on my home computer and a book from which I could practice alone, with books.

But, it's only after I graduated from high school, that I started [learning and applying C#](#) almost full time during higher studies, and that's what I considered to be my real start into programming and software engineering.

I learned a few lessons over the past decade, and I thought I would take a moment to gather my thoughts on these things. It took me about ten years and a lot of experimentation to figure out some of this.

### 1. Learning a programming language is the easy part: be aware and beware of the platforms

Take C# for example. Learning the C# language is not difficult. If you already have a good understanding of computer language fundamentals, and if you have some experience in other object-oriented languages, [you can become a competent C# programmer within a few days](#), at least as far as the language itself is concerned. However, the real price to pay is not about the language, it's about the platform. To develop with C# on .NET, you need to know:

- the .NET framework
- [one or more .NET technologies](#) such as ASP.NET or WPF
- and the Visual Studio development environment.

The time required to become proficient in .NET development is usually measured in months, even for an experienced developer. [Learning a platform](#) is always more expensive than learning a specific language, therefore choosing the platform is the most crucial decision.

Learning always has a cost and this cost is one of the key factors to consider when choosing the technology you want to learn. The real cost of learning is in time, learning always takes time. Since you do not have time to learn everything, it is important to think strategically about what you want to learn. And since the languages are easy, it's the platforms that you have to be careful about: the technologies associated with the language, the development and deployment tools, the operating systems, and other infrastructures.

## 2. I repeat, learning a programming language is the easy part: meet the underlying concepts of software engineering

The syntax itself, the words you use when applying the language are relatively simple and you can easily pick them up as you go. However, that's far from being enough [for producing quality code](#), which often involve OOP and SOLID principles, unit testing, design patterns, TDD, BDD, and other technical concepts which are beyond the scope of this article. Anyway...

## 3. Actually writing code is only one (small) part of the job

A software engineer is often expected to be involved into technology research, tools and projects configurations, DevOps and admin tasks, debugging and testing procedures, documentation, and technical debt (fixing and refactoring existing code). Also, they have to be thinking about solutions and designing systems : [sometimes the most important work is done away from a keyboard](#).

## 4. Tried and true : old and boring is sometimes the best.

It's not so much old vs new, or cool vs boring, but rather the thing you are most experienced with. As they say, I trust not the developer who practiced 1000 technologies once, but I trust the developer who practiced the relevant technology 1000 times. If the goal is to "just build the damn thing", go with a stack you would be most productive in.

For example, one of my contacts is making \$25,000 per month with a SaaS that was built on boring ASP.NET+SQL Server+Angular 1 because that's what he knew. he hosts it on Windows because he knows how to make it fast and secure. He succeeded by focusing all his time on building the features that clients were asking for, instead of learning fancy tech.

It's important to realise that [the technological treadmill never stops](#). Yet another JavaScript framework could have been launched while you are reading this. Today's cutting edge tech didn't even exist when I was getting started ([EF Code First](#), [Xamarin](#), ASP.NET Core, Razor), and this leads us to the 2 next points.

## 5. Focus on sustainable technologies

The only constant in the world is change. Actions and time management is an important skill for developers, particularly because we are on a technological treadmill that keeps moving or even accelerating.

For example, Web technologies that were popular around the year 2000 (Flash, ASP Classic and Java Applets) are becoming almost obsolete and decreasingly marketable. Today, we are talking about ASP.NET Core, SignalR, Angular2, React and VueJS. None of its technologies existed in the year 2000, and these new technologies are likely to be obsolete within 10 years.

What hasn't really changed? The fundamentals of languages such as C++/C#, their implementations of algorithms and their principles are still relevant after several decades. [If you master the basics of a stable system, you could adapt to change better](#), you could appreciate it and use it to evolve.

## 6. Balance between exploration and exploitation

Exploration is about [learning new things, studying new techniques, reading books, watching video tutorials, practicing and improving skills](#). Exploitation however, is to take advantage of what we already know [to solve real life issues](#). It's about thinking creatively about ways to use the knowledge we already have to create value for others.

So yes, both of these tasks are both necessary and important. The risk is to be too focused on either activity.

Too much exploration, and you will never achieve a useful level of expertise in the chosen technology.

There is a huge opportunity cost with this kind of light learning, because, although [it expands your mind](#), the time it takes implies that you don't really improve on the skills that you have already acquired.

On the other hand, too much exploitation can keep you from evolving in new technologies, and can limit your employment opportunities.

## 7. It's easy to be great... It's hard to be consistent

It's easy to be great for 2 minutes. It's hard to stay great constantly, every day.

When you have a good idea for a new project, you feel a great desire to start researching, designing and programming. You feel a rush to turn your idea into something real and you become super productive. But the problem is that this motivation fades over time.

Yes, it's fun and it's easy to have new ideas and start working on them. But then there are the efforts to be made, the adjustments, the launch, the maintenance, the corrections, the improvements, and so on. Over several months. This is where it gets hard. It is hard to stay focused on the same idea, on the same project for months and years. [It takes a lot of discipline](#).

It's easy to be great. It's hard to be consistent.

## 8. Diversify your skills

Don't be just a programmer, become an Expert Who Programs, an expert in [another relevant field that you are passionate about](#). You can be an entrepreneur, a project manager, a Big Data scientist, a researcher, a security specialist, etc. If you are an Expert Who Program, in addition to being able to program (maybe full-time), you also have [an additional credibility](#) that is related to [something other than software engineering](#). Hence the importance of getting an education. If you go to university and you already know how to program, you probably won't learn much about programming. That does not mean you should not go to these schools. You will need some culture, and universities are great places to get that. You acquire culture by [studying and understanding the world](#) that humans have created, from different angles. It would be difficult to acquire this kind of knowledge if you do nothing but study programming.

## 9. Pick your niches and standout

The smaller the niche you choose, the greater your chance of being viewed as a standout in your field. It's very hard for a developer to become a standout in "PHP Web development". They're great, versatile, useful, but not noteworthy. One developer who knows how to work with these technologies, feels they are easily replaceable because there are so many out there with a comparable skill set. These areas are too broad for you to easily standout from the pack. If, on the other hand, you become known for a niche, like [Xamarin.Forms](#) or JavaScript Visualisations you're much more likely to be valuable to those looking specifically for that skills set.

## 10. Age of Skills

Information is the specific knowledge that you need to solve problems. Skills represent the ability to implement solutions using your knowledge.

In a world where [most of the knowledge and tools are virtually free](#), what makes the difference? The skills, of course. We are no longer a knowledge-based society, we are a skills-based society. There was a time when almost all university degrees guaranteed a good job. Now this is no longer the case. Nobody cares about what you know. People care about what you can do. [They pay you to do things, not to know things.](#)

### Teach yourself programming in 10 years.

Researchers have shown it takes about ten years (or 10 000 hours) to develop expertise in a field.

The solution is reflective practice: it's not enough to repeat the same things over and over again, but challenging yourself with a task that exceeds your current ability, to try it, to analyze its performances during and after, and to correct all error. Then repeat. And repeat again. It seems there are no real shortcuts. Learning through reading is good. But [getting your hands dirty in practice](#) is better. [The best type of learning is learning by doing.](#)

Personally, [small projects and prototypes actually helped me improve](#). But there are still interesting things to master, therefore let's keep learning.

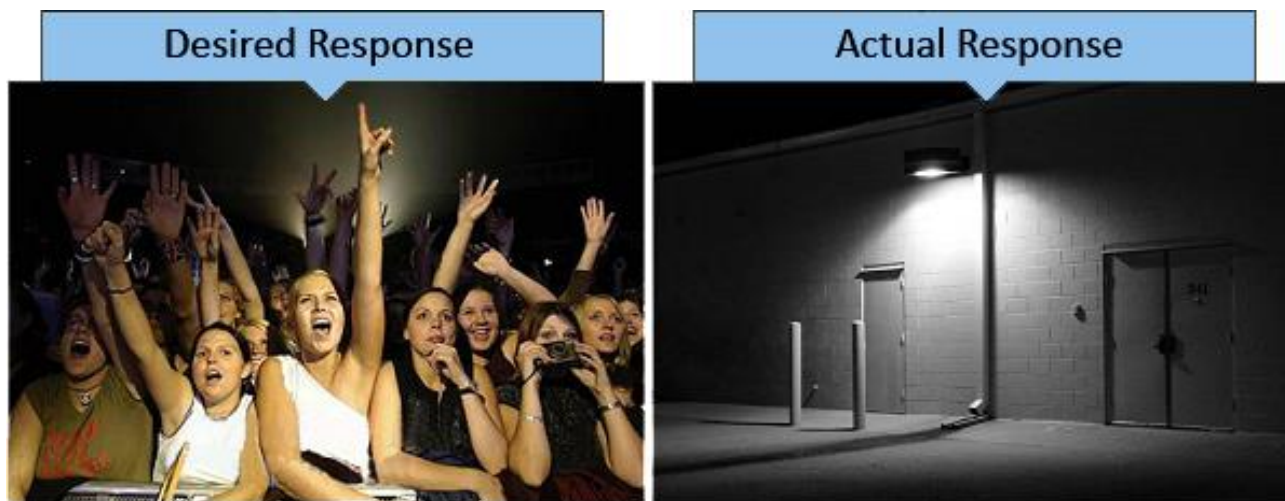


## Should you build it if nobody comes?

By [Darren Mart](#)

Not long ago I received an E-mail from an energetic and talented developer who was tackling an ambitious software project. He asked if I had any advice to pass along.

It all came flooding back. The daydreams. Springing out of bed in the dead of night because an idea can't wait. Months of painstaking work, coaxing that baby from rolling to wobbling to walking on its own. All of this for an eager audience. Except...



### **The cycle after the cycle**

I must have drafted a half-dozen responses to his mail. I admired his enthusiasm and understood where his head was. But I also knew what likely awaited him after the arduous development cycle: another cycle of turbulence, a series of clashes between expectations and unfortunate realities.

### **Where did it all go wrong?**

This is a dangerous question we ask ourselves as developers. "Dangerous" because it carries the assumption that we did something wrong. It's entirely possible that what you've crafted is quite good, maybe even brilliant. The issue may have nothing to do with the quality of your work, rather your criteria for success.

So how does a developer accurately gauge the success of a solo project? I wish I had a concrete answer to that. Instead I can offer a few tips on how not to gauge it:

**Tip 1: Beware the social platform sirens**

Ah, but they lure us in with the promise of mass consumption and acceptance. We seldom consider their ability to trample our spirits. It's possible that you'll spend months on a project, share it on Facebook with the vigor and excitement of a dog with a chew toy, and then watch the "Like" count soar to... 3. One of those came from your mother, another from a page you made to promote your project, and the other from an errant tap on a cell phone.

It's the same chilling effect you get from photos of abandoned amusement parks. You strain to imagine giggles of delight and unbridled excitement, and you struggle to accept what's staring you right in the face.

Meanwhile, your friend's profound status of "I like ravioli!" earns 23 likes and 16 comments. Popularity and substance are two different things, and we'll just have to live with that.

**Tip 2: The apathy isn't personal (it might not even be apathy)**

You're already aware of this but it helps to be reminded. Most people have no concept of what it takes to pull off what you've accomplished. They don't realize you single-handedly built something that'd rival the efforts of an entire team. You can't, and won't, make them genuinely care.

Apps that set the world on fire generally fall under two categories: those that help users promote themselves, and those that require little thought. If your project requires a mental investment greater than that of Candy Crush Saga, here's the harsh reality: you're gonna end up with a lot of leftovers at your launch party.

**Tip 3: Think twice before soliciting feedback from your peers**

In the film *Midnight In Paris*, aspiring writer Gil asks Ernest Hemingway to read his novel and offer an opinion.

HEMINGWAY: My opinion is I hate it.

GIL: You do? But you haven't even read it.

HEMINGWAY: If it's bad I'll hate it because I hate bad writing and if it's good I'll be envious and hate it all the more. You don't want the opinion of another writer. Writers are competitive.

Don't overlook the profundity of this exchange. It applies to developers, too.

**Tip 4: Like it or not, you're an artist**

If you pour your heart and soul into solo projects because you enjoy the creative challenge, you're an artist. If you do it because you're attempting to solve a problem, you're probably an artist-engineer hybrid. If you do it strictly for the money, you bailed on this article a long time ago so it doesn't matter what I call you.

Just like the painters and writers and decorators and gourmet chefs of the world, we secretly hope the public will be elevated and inspired by our creations.

But there's a key difference. If someone isn't inspired by a painting, they'll still be able to appreciate the individual effort. With software there's no such luxury. It doesn't matter if you're the Rembrandt of the coding world; your app is a dud compared to Office 365 or Google Maps or Skyrim, never mind the fleet of resources required by the latter.

**So... should you build it if nobody comes?**

Yes. Because as artists, we are enriched by the overall process regardless of the end result. Trite as it sounds, we have the sense of accomplishment and the pride of knowing how much discipline it took to see it all the way through. We stimulated our minds, we learned what does and doesn't work, we gained applicable experience. The next project, whether it's personal or professional, will reap the rewards.

## Useful Job Search Websites For Programmers In The UK

By [Necemon](#)



Job boards are websites that facilitate job hunting.

There are many career websites designed to allow employers to post job requirements for a position to be filled; prospective employees can locate and fill out job applications and/or submit digital resumes for the advertised positions. Those sites may also offer employer reviews, career and job-search advice, and describe different job descriptions or employers.

As a technology specialist in the UK, here are some of the websites I found helpful in the past few years.

### **Graduates**

#### [targetjobs.co.uk](https://targetjobs.co.uk)

Graduate jobs and schemes. Internships and placements. Great advice to help you get hired.

#### [eurograduate.com](https://eurograduate.com)

Featuring thousands of graduate careers and job opportunities across Europe.

#### [insidecareers.co.uk](https://insidecareers.co.uk)

Graduate jobs, internships, placements and school leaver schemes as well as career advice by sector.

#### [milkround.com](https://milkround.com)

The UK's most widely used student and graduate job website.

#### [prospects.ac.uk](https://prospects.ac.uk)

Prospects for graduate jobs, postgraduate study, advice about work experience, internship opportunities and graduate careers.



## **IT World**

[uk.dice.com](http://uk.dice.com)

Formerly [The IT Job Board](#). UK Contract and Permanent IT Jobs

[purelyit.co.uk](http://purelyit.co.uk)

Lots of contract and permanent IT Jobs including Software Engineer, IT Director, Senior Web Developer and many more.

[computerjobs.com](http://computerjobs.com)

Specialising in Permanent and Contract IT jobs in the UK with thousands advertised daily.

[cwjobs.co.uk](http://cwjobs.co.uk)

One of the leading UK IT job board. Search information technology jobs and apply online.

[currentitjobs.co.uk](http://currentitjobs.co.uk)

Formerly [findingitjobs.co.uk](http://findingitjobs.co.uk). IT job board focused on providing the best IT Jobs in London.

[stackjobs.co.uk](http://stackjobs.co.uk)

A trending IT job board in the UK with effective IT recruitment solutions.

## **Science and Engineering**

[newscientistjobs.com](http://newscientistjobs.com)

Technology and science jobs, courses and career advice.

[naturejobs.com](http://naturejobs.com)

Featuring access to job listings, editorial content about scientific careers and other information.

[justengineers.net](http://justengineers.net)

A wide range of Engineering Jobs in the UK and Worldwide.

[engineeringjobs.co.uk](http://engineeringjobs.co.uk)

Latest Engineering Jobs from across London and the UK.

## **General Job Search Sites**

[reed.co.uk](http://reed.co.uk)

One of the leading UK job sites.

[monster.co.uk](http://monster.co.uk)

Possibly the most popular job site worldwide. Resources to create a killer CV, search for jobs, prepare for interviews, and launch your career.

[jobserve.com](http://jobserve.com)

Powerful, quick job search. Specialising in permanent and contract jobs in the UK with thousands advertised daily.

[jobs.ac.uk](https://jobs.ac.uk)

UK & international job search for academic jobs, research jobs, science jobs and managerial jobs

[totaljobs.com](https://totaljobs.com)

Instant job matches, alerts and more from UK companies and recruiters.

[topjobs.co.uk](https://topjobs.co.uk)

Searchable database of vacancies by type and region.

[neuvoo.co.uk](https://neuvoo.co.uk)

Presumably, your job search starts here.

[jobs.trovit.co.uk](https://jobs.trovit.co.uk)

Job ads from thousands of websites in just one search.

[jobbydoo.co.uk](https://jobbydoo.co.uk)

Aggregating, analyzing and listing job openings from more than 1,000 sources, including UK career sites, job boards and recruitment agencies.



## Microsoft Professional Certifications for Developers

By [Necemon](#)



Microsoft offers a [wide range of online certification programs](#) designed to help you grow your skills and your career.

This article will focus on developer certifications. Microsoft Certified Solutions Developer (MCSD) is a certification intended for IT professionals seeking to demonstrate their ability to build innovative solutions across multiple technologies. For example, the [MCSD App Builder certification](#) validates that you have the skills needed to build modern mobile and/or web applications and services.

I got my first Microsoft Certification back in 2008. I have been upgrading over the years and I am now a Certified Solution Developer. Was it worth it? Sometimes it didn't matter, sometimes it was pretty useful. In my experience, here are 5 advantages of getting certified:

- Getting a Microsoft Professional Certification doesn't guarantee anything about getting a job, a raise or a promotion but it does increase the odds.
- When hiring someone new, some companies check out his certifications as well as experience. Many hiring managers verify certifications among job candidates and consider those as part of their hiring criteria. Some consider IT certifications a priority when hiring for IT positions.
- Getting a certification can boost your self-confidence, your confidence about your skills.
- It shows seriousness and passion (you did it all because you wanted to, not because you had to).
- The process of preparing for the certification increases your theoretical and practical skills.

### Exam and preparation tips

My key recommendation is to dedicate a specific period to prepare right before the exam. Get some books, attend a course (virtual courses as a personal preference, but classroom courses available) and practice the technologies involved.

- Books: there are often preparation books related to the given exam. Example: [Exam Ref 70-480: Programming in HTML5 with JavaScript and CSS3](#)
- Classes : The most popular exams have short courses available online. Example: [Course 20480B: Programming in HTML5 with JavaScript and CSS3](#)
- Beyond the training material: researching the topics involved
- Actually practicing the technologies
- Answering to multi choice questions: Proceed by elimination. In case of doubt about the right answer, exclude the answers that don't make (any) sense, (or that makes less sense).

### Career paths for developers

Make sure you double check that part. They keep updating the offers based on the latest technology but at the time of writing this, there are 5 options for MCSD (Microsoft Certified Solutions Developer)

- MCSD: Web Applications. Expertise in creating and deploying modern web applications and services.
- MCSD: Windows Store Apps. Expertise at designing and developing fast and fluid Windows 8 apps. There are two paths to achieving this certification-using HTML5 or C#.
- MCSD: SharePoint Applications. Expertise at designing and developing collaboration applications with Microsoft SharePoint.
- MCSD: Azure Solutions Architect. Expertise covering the full breadth of designing, developing, and administering Azure solutions.
- MCSD: Application Lifecycle Management. Expertise in managing the entire lifespan of application development.

### Certification Planner

The [Certification Planner](#) is a tool to help you know what requirements are necessary to achieve your next certification. It's about planning your next steps (your options, which exams to take, in which order).

Other links of interest

- [MCSD Certification](#)
- [Web Apps certification](#)
- [Wikipedia Page](#)

That's it for now. Now get back to work.

# Chapter 7

## C# .NET Programming

Featuring Holty Sow



## Why I like C# so much

By [Necemon](#)

If you are not [into programming](#), I need to start by telling you that a programming language is basically an artificial language (system of communication) designed to communicate instructions to a machine, typically a computer.

Now there are a lot of programming languages out there. Some are more popular than others, some are more recent, some are more powerful to some extent.

In an ideal world, each programming language serves a specific purpose. So an engineer should be able to adapt to the on going project and choose the optimal technologies. But the truth is, we very often tend to feel comfortable with some languages and find some others kind of painful. It depends on the features of the language and how long we have been using it. It's a bit like natural languages (human languages). You may learn many languages (French, Italian, Spanish, German, Japanese, etc.) and wherever you go, there will be one language that would be more relevant and that you would have to use, but as a native English speaker, you would mostly feel more comfortable speaking English than anything else. If you get in a non English speaking country, you may sure have to adapt to the local language but you would feel some relief when you meet people you can speak English with, as this is what comes naturally to you.

The difference is that you don't choose your main human language. It's generally the language they speak at the place you were born and grew up, the language your parents speak, the language your friends and teachers speak at your school, etc.

Programming languages are a different matter. Programmers [do choose to learn](#) a language, even though their motivations may be different. What I want to discuss here is, [why and how people choose their programming languages](#) ? What's the best way to go about it ?

From what I observed, there are 2 main reasons people go for a specific language:

- social proof: I guess this concern mostly the beginners, when you want to learn programming for the first time, you don't know much about the languages but you got to start somewhere. So you just go for the language that's popular among your friends, your lecturers/mentors or at your school/college. In short, you go for the languages you are most exposed to or you take advice from people around you. The upside is that you would definitely be surrounded with people who are into the same technologies so they will be able to guide and support you, work along with you on same projects. It's a bit like choosing to buy a specific video game console because all your friends have the same. Suppose you get stuck at some level, there is probably one of your friends that can tell you what to do. Plus you can exchange games with them, discuss game news & cheat codes, enjoy playing together. In short, you become part of the community and it goes pretty much the same when it comes to choosing a language.

- project requirement: [throughout their career](#), developers come across many projects they are compelled to do. The requirements may lead them to learn other languages and technologies, for example, if you work for some company and they assign you to work on that new Python project, you would need to learn Python. Even if you are an independent developer, you may learn another language as it becomes popular with your customers or as it better meets the needs of your users (in terms of speed or user experience for example).

Personally, I think both reasons are valid points. Regarding the first reason, I would just add that [it's not about following the trend just for the sake of it](#). You should make some research to find out which language will help you better do what you are trying to do. Regarding the second reason, I realise it is required to do things that you don't really like but, as often as possible, I would advise you to use the technologies and languages that you like better. If programming is your job, you better enjoy it. [Do what you love](#), really.

[Coming to my own experience](#), I guess I went for C#.NET firstly because it is very popular in the institution where I started getting serious about programming (NIIT). But this is only the reason why I got introduced to C#. There are many other reasons that keep me going.

The main reason being that I find it comfortable but not just because I had been using the most that time. Anyway it wasn't my first programming language. I did program in VB and Pascal before, so it's not that I got stuck to the first language I liked to use. By comfortable, I mean that I enjoy writing C# code. Personally, I would rather write code if I have a lot of fun doing it than write code in some obscure language that's just painful. It's true that we need to worry about things like performance and user experience but I believe that one should enjoy what they are doing.

Now what's so cool about C# ? I won't be discussing how C# compare to other programming languages and whether it's technically better or not. That's not the point of this article and as I said before there is no perfect language, there is just a right language for the right circumstances. I will just say what make C# so awesome to me (and probably to you as soon as you give it a try):

I can't help it, I need to mention that Visual Studio, the work environment for C#, is probably the world best IDE. Automated features like IntelliSense and controls drag and drop save a lot of time and effort. It's not being lazy, it's really about productivity. Whether you work solo or as a team, you always have some important tasks that are not necessarily programming oriented. Automation helps you avoid spending time and effort on the obvious, frequent, basic tasks and focus on the things that are most important.

[The language itself is easy to grab](#), it follows the same kind of syntax that C, C++, Java, etc. So anyone with a similar background can quickly get going. Also, it's known to be simple and elegant.

C# is powerful. It helps me make ANYTHING I want, whether it's a website, a web application, a web service, a smart client application, a game, a windows application, a windows service or a in-browser (Silverlight) application, etc. While most of the languages are used just for a specific purpose, either for the

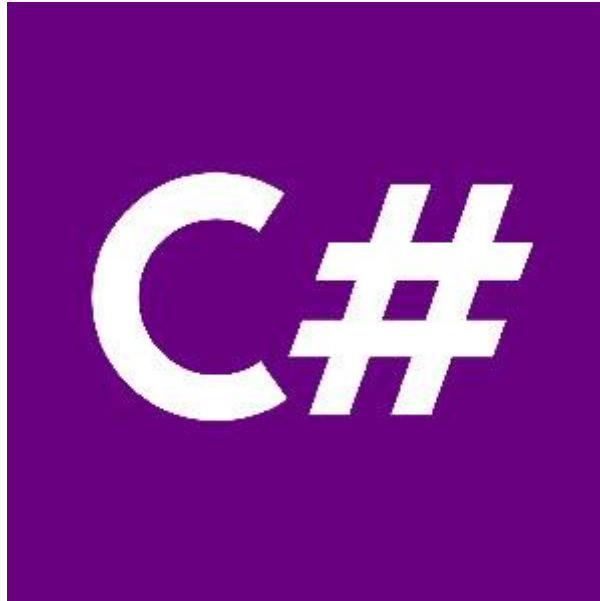
web server, for the client or for the browser, [C# does it all](#). The obvious advantage is that you don't have to learn a new language when you want to start a new kind of application.

Interoperability and language integration: C# applications and services can talk to each other. In fact, they can communicate with any other .NET applications. For example, you can easily make a WPF C# application get data from a WCF VB.NET service and pass it to a Silverlight C# application.

Well, now you know why I like C# so much...

## C# Free Learning Resources For Beginners And Professionals

By [Necemon](#)



[C# \(C Sharp\)](#) is Microsoft [flagship programming language](#), hence its [popularity in the tech industry](#). Used by a large and growing number of professionals, it helps when building all kinds of applications.

A frequently asked question is, where do I start with C#, what are the best free courses?

Here are a few suggestions.

### [Microsoft Virtual Academy](#)

C# fundamentals for absolute beginners. Step through 24 practical and easy-to-understand C# training episodes. Tune in to learn the basics of the C# language, and learn to apply them in your programming endeavors, like video games, mobile environments, and client apps.

### [learncs.org](#)

Free interactive C# tutorial. Whether you are an experienced programmer or not, this website is intended for everyone who wishes to learn the C# programming language. There is no need to download anything, just click on the chapter you wish to begin from, and follow the instructions.



### [Solo Learn](#)

The best way to learn to code is to code. Gain an understanding of C# concepts by going through short interactive texts and follow-up fun quizzes. Their beautifully designed code editor lets you make changes to existing code or write and run your own custom code and see the output on your mobile device. You can code while going through the core lessons or as a stand-alone learning activity. The more you play, the better you get!

### [Codeasy](#)

A free interactive online course for learning to program C# language. It is designed for absolute beginners and does not require any prior knowledge to start. It is really fun to learn with Codeasy just by reading an adventure story about fighting machines in the future. While reading, the user meets challenges, which require real coding to solve. User can write code directly on the website. The course consists of chapters, like a real book. Each chapter has several tasks to solve by coding. The final goal is to become a programmer and to save the world.

### [Visual Studio Dev Essentials](#)

Free tools and free training. Free access to technical training from industry leaders such as [Pluralsight](#), Wintellect, and [Xamarin University](#).

### [Channel 9](#)

A Microsoft community site for Microsoft customers. It hosts video channels, discussions, podcasts, screencasts and interviews. This includes courses like C# Fundamentals for Absolute Beginners.

Bonus : Top Youtube playlists and tutorials on C#

[Youtube 1](#)

[Youtube 2](#)

[Youtube 3](#)

[Youtube 4](#)

[Youtube 5](#)

## How to Track Mysterious Bugs with Visual Studio

By [Necemon](#)

[Debugging](#) is the process of finding and resolving defects or problems within the program that prevent correct operation of computer software or a system. Debugging tactics can involve interactive debugging, control flow analysis, unit testing, integration testing, log file analysis, monitoring at the application or system level, memory dumps, and profiling.

Some bugs may be easy to track. I don't want to talk about those. Let's discuss the really mysterious ones.

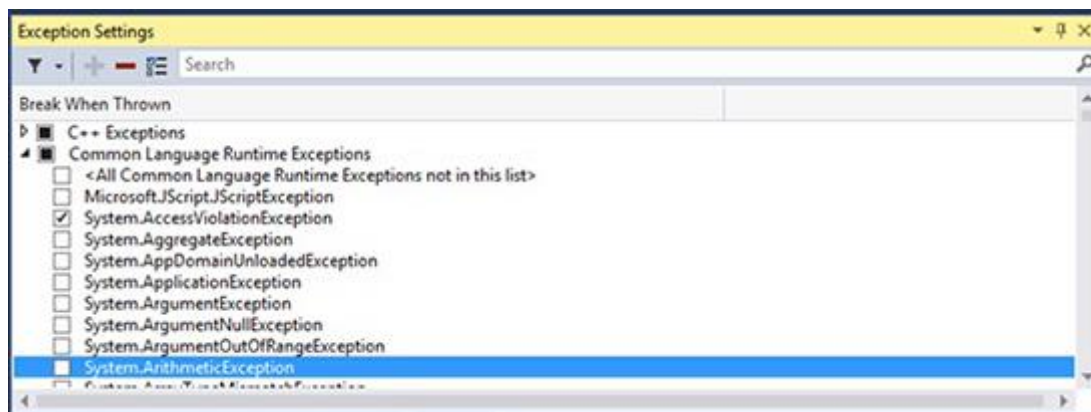
Like the saying goes, Theory is when one knows everything but nothing works; Practice is when everything works but nobody knows why. Sometimes Theory and Practice go hand in hand: nothing works and nobody knows why. In such cases, it's worth remembering about some [effective techniques](#) that often allow us to reach the root of the problem and to display its finer details. [Over time](#), here are the 3 techniques that have been most useful when debugging with Visual Studio:

### 1. Managing relevant exceptions with the Exception Settings Window

An [exception](#) is an indication of an error state that occurs while a program is being executed. You can and should provide handlers that respond to the most important exceptions, but it's important to know how to set up the debugger to break for the exceptions you want to see. You can use the Exception Settings window to specify which exceptions (or sets of exceptions) will cause the debugger to break, and at which point you want it to break. You can add or delete exceptions, or specify exceptions to break on. Open this window when a solution is open by clicking **Debug / Windows / Exception Settings**.

You can find specific exceptions by using the Search window in the Exception Settings toolbar, or use search to filter for specific namespaces (for example System.IO). The debugger can break execution at the point where an exception is thrown, giving you a chance to examine the exception before a handler is invoked.

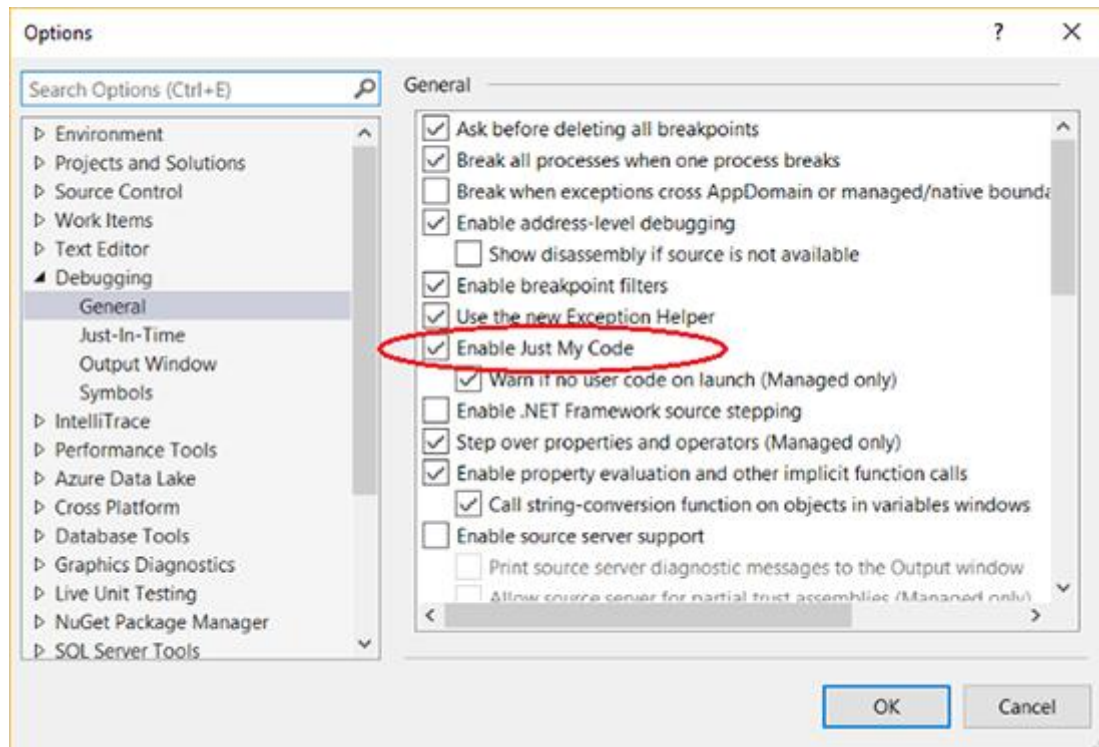
In the Exception Settings window, expand the node for a category of exceptions (for example, Common Language Runtime Exceptions, meaning .NET exceptions), and select the check box for a specific exception within that category (for example System.AccessViolationException). You can also select an entire category of exceptions.



## 2. Not "Just My Code"

By default, the Visual Studio debugger only breaks on exceptions generated from your own (user) code, hence skipping other system, framework, and other non-user calls. The feature that enables or disables this behavior is called "Just My Code". Depending on what you are debugging, you may want to disable it, because the source or description of the issue might well be outside of "your" code.

To disable (or enable) Just My Code, choose the **Tools > Options** menu in Visual Studio. In the **Debugging > General** node, clear (or choose) **Enable Just My Code**.



## 3. Recommended tools for Tracing and Error Logging

Sometimes you need to record and analyse the full details of the errors, the events and the inner exceptions: that **tracing** involves a specialized use of logging to record information about a program's execution, typically for debugging purposes. Here are my favorite logging and tracing tools:

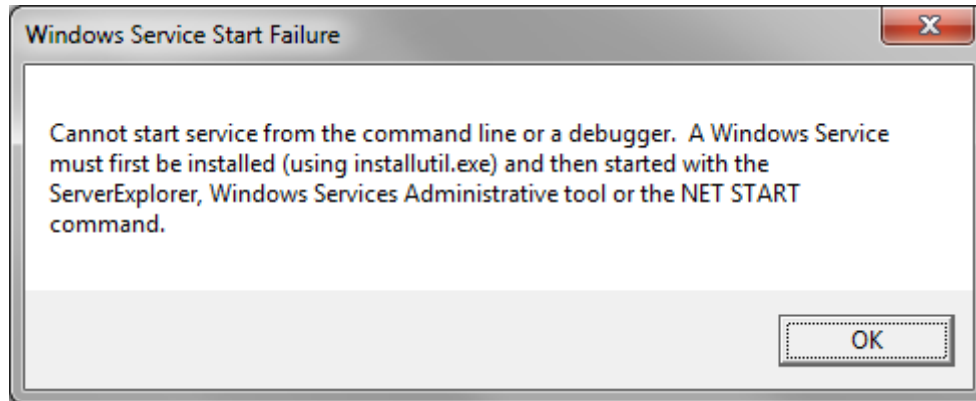
- [Systems.Diagnostics](#)
- [Microsoft Enterprise Library](#)
- [NLog](#)
- [Elmah](#)
- [Log4net](#)

Voilà.

## How to Debug a Windows Service in Visual Studio

By [Holty Sow](#)

When you create a Windows Service project under Visual Studio, you may have noticed that the following dialog box appears when you try to run the service:



In summary, it is simply impossible to run a **Windows Service** in Visual Studio, you must necessarily go through the **NET START** command to start the service after having previously installed it with the command **INSTALLUTIL**. Except that this method prevents us from easily debugging the Windows Service. As a matter of fact, to debug our code, we must install the service, start it, then link the debugger to the relevant service process from Visual Studio. And by the way, let's keep in mind that we will also have to stop, recompile and restart the windows service in order to load up any change we make to the code. In short, it's kinda annoying :D

There is a simpler solution though. We can formulate different compilation directives to detect which mode we are running: **RELEASE** or **DEBUG**. If we are in:

- **DEBUG** mode: we will treat our windows service as a simple client application by displaying a dialog box indicating that the service has started
- **RELEASE** mode: the operation will remain the same as when we tried to run our Windows Service under Visual Studio. In other words, this mode should be used in production, after the debugging session is complete

Our use case will be very simple, we will just be running a Windows service and debugging the WCF service that it hosts. Here are the steps you need to follow:

1. Tweaking the code of the windows service: we will add two methods, **StartWCFService** and **StopWCFService**, which should start and stop listening to WCF incoming requests respectively. The first method will be called in the **OnStart** method definition, and the second one will come from the **OnStop** method. Below is the code:

```

private ServiceHost host;

public MyWindowsService()
{
    InitializeComponent();
}

protected override void OnStart(string[] args)
{
    StartWCFService();
}

protected override void OnStop()
{
    StopWCFService();
}

public void StartWCFService()
{
    host = new ServiceHost(typeof(IWCFService));
    host.Open();
}

public void StopWCFService()
{
    if (host != null && host.State == CommunicationState.Opened)
        host.Close();
}

```

2. Tweaking the **Program.cs** file: it's in the **Main** method that we would detect our selected compilation mode. If we are in **RELEASE** mode, then the windows service is running as usual and if we try to run it in Visual Studio within this mode, we will obviously get the so-called dialog box. If, on the other hand, we are in **DEBUG** mode, then it gets easier, we directly call the **StartWCFService** method of the instance of our windows service (so the **OnStart** method will not be called) and then we display a dialog box to notify the user. When they close the dialog box, the **StopWCFService** method is called to stop the WCF service (therefore, the actual **OnStop** method of the Windows service will not be called).

Here is the code of the **Main** method:

```
static void Main()
{
    MyWindowsService service = new MyWindowsService();
    #if DEBUG
        service.StartWCFSservice();
        MessageBox.Show("Le service a démarré...");
        service.StopWCFSservice();
    #else
        ServiceBase[] ServicesToRun;
        ServicesToRun = new ServiceBase[]
        {
            service
        };
        ServiceBase.Run(ServicesToRun);
    #endif
}
```

I hope this post has been helpful ;-)

## My Essentials : Top 12 Tech Courses on Pluralsight for .NET Developers

By [Necemon](#)



Pluralsight is the largest online tech and creative library on the planet: an online education platform that offers a variety of video training courses for software developers, IT administrators, and creative professionals through its website.

I have been learning quite a bit from Pluralsight, and here are my favorites so far.

### [Learning Technology in the Information Age](#)

So much to learn, so little time... This course will show both beginners and experts how to learn more in less time.

Would you write code without a design? Build hardware without a schematic? Configure a server without a plan? Of course not. Yet, how many of us learn technology without any planning, choosing resources at random in the hope that one of them will be worthwhile? This is horribly inefficient, and worse, can leave critical holes in your knowledge and skills. In this course you'll learn how to design a plan for learning any technology effectively and efficiently based on your own needs and goals.

### [Reprogramming the Developer Mind](#)

Explore the habits and career tactics that create remarkable developers. Take control of your career, and set yourself apart from the pack. Become an Outlier.

This course is about making a paradigm shift in how you manage your career. We'll discuss concrete activities and skills that transform average developers into outliers. You'll learn why developers can't afford cable, ways to improve your "luck surface area", and techniques to compress your career through accelerated development. You'll learn the foundational skills for becoming an outlier: command your time, hack your image, and own your trajectory. Prepare to think about your development career in a whole new way.



### [Encapsulation and SOLID](#)

This course teaches how to write maintainable and flexible object-oriented code. Learn how to write maintainable software that can easily respond to changing requirements using object-oriented design principles. First, you'll learn about the fundamental object-oriented design principle of Encapsulation, and then you'll learn about the five SOLID principles, also known as 'the principles of object-oriented design.' While this course is aimed at beginner to intermediate developers, it's based on decades of experience, so even advanced programmers can learn a thing or two. There are plenty of code examples along the way; while they're written in C#, they should be easily understandable to readers of C, C++ or other curly-brace-based languages.

### [Date and Time Fundamentals](#)

This course will help you to understand dates and times, and how they should be used in software development.

Managing dates and times properly is one of the most difficult things to get right in software. This is mostly due to how humans have introduced nuance into our calendars and clocks. In this course, you can straighten it all out. You will learn about UTC, daylight saving time, time zones, and calendar systems. You will also learn how date and time values are represented and manipulated in various programming languages. We will look closely at the different kinds of time zone data, and discuss various fallacies and gotchas that are commonly encountered. We will deep dive into how date and time are handled in the .NET Framework, and in JavaScript. We will also look at various libraries that make things slightly more bearable. Throughout the course, you will learn about real-world situations that require deeper thought about how date and time are handled in your applications, and I will give you practical advice on how to solve them.

### [Web Security and the OWASP Top 10: The Big Picture](#)

Security on the web is becoming an increasingly important topic for organisations to grasp. This course takes you through a very well-structured, evidence-based prioritisation of risks and most importantly, how organisations building software for the web can protect against them.

### [Visual Studio : Essentials to the Power-User](#)

This course introduces Visual Studio and includes productivity boosters for everyone to make writing and reading code easier and more fun. Visual Studio is an integrated development environment you can use to create applications and libraries with many different frameworks and languages. It has a rich feature set, including an intelligent editor, built in compiler (and related tools), and context sensitive help. This course starts with basic concepts like projects and solutions, shows you how to make Visual Studio look and work the way you want it to, and demonstrates how to use the most popular tool windows and dialogs. It goes further into tips and shortcuts that will save you time every day. Using Visual Studio is about more than writing code or reading code written by others. To be truly productive, you need to debug well and understand the designers that help you build your user interface. This course also shows you how to add helpful extensions that make Visual Studio even better. When you've completed it, you'll know how to use the tool itself and can focus on a specific language or framework as your next step.

### [Bootstrap 3](#)

Twitter's Bootstrap 3 can help you achieve a great looking and performing web site. Building great looking websites that work well with different sized devices can be a challenge. By utilizing Bootstrap 3 framework, you can meet that challenge head-on. Bootstrap 3 is a mobile-first responsive design framework for structuring your website's HTML. It includes a great grid system, responsive design, CSS typography and components to solve many of the most common design challenges that face web developers today.

### [Building Your First Xamarin.Android App from Start to Store](#)

[Xamarin is a cross-platform development tool](#). It solves dilemmas many developers face when developing cross-platform apps: separate coding languages and UI paradigms. With Xamarin, you can use C# for iOS, Android, and Universal Windows apps.

In this course, you'll learn how to build a complete, fully-working Xamarin.Android app using C#. Xamarin.Android is covered in a very practical, hands-on way that you can follow along.

### [More Effective LINQ](#)

Learn how to fully harness the power of LINQ by exploring best practices and avoiding common pitfalls by solving some fun and challenging problems. In this course, you will learn how to take full advantage of the power and capabilities of LINQ. You will see how the LINQ extension methods can be combined together to solve complex problems in a simple and succinct "pipeline." Throughout the course, you will learn to solve some "LINQ Challenges" and pick up lots of pro tips that will take your LINQ skills to the next level, including how to extend, debug, optimize, and test LINQ.

### [Async C#](#)

In this advanced series, Jon Skeet shows us the new Asynchronous goodness available in C# 5.0. Managing threads, awaiting asynchronous calls - these are all made simple with a few new keywords and types. Asynchronous coding in a static language is not exactly "simple," and rather than try to simplify complex topics, Jon have, instead, decided to go deep and see how things work at a deeper level. Again, this is an advanced production! If you are new to C# (or new to programming in general), you may want to become familiar with the basics of C# before you tackle this one.

### [TypeScript Fundamentals](#)

TypeScript is an open source language that provides support for building enterprise scale JavaScript applications. Although several patterns exist that can be used to structure JavaScript, TypeScript provides container functionality that object-oriented developers are familiar with, such as classes and modules. It also supports strongly-typed code to ensure inappropriate values aren't assigned to variables in an application.

In this course, John Papa ([link to his site](#)) will walk you through the key concepts and features that you need to know to get started with TypeScript, and use it to build enterprise scale JavaScript applications. You'll learn the role that TypeScript plays as well as key features that will help jump-start the learning process.

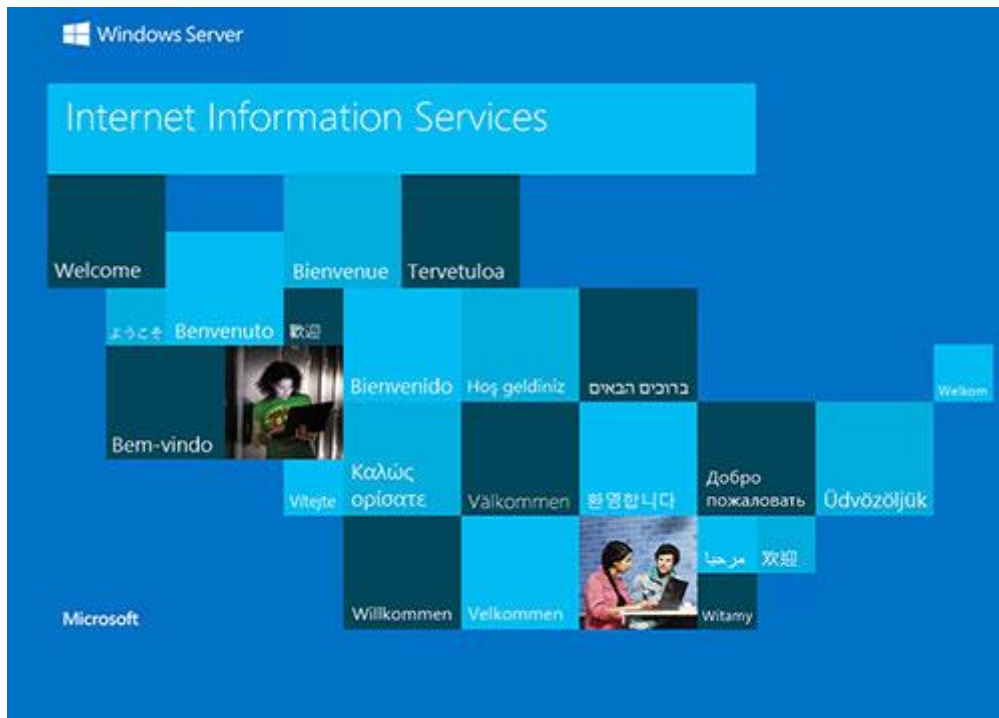
### [Understanding ASP.NET Core](#)

This course will help you quickly grasp how to get going with ASP.NET Core with a compact, practical presentation, covering all the best new features of ASP.NET since its release.

This latest release of .NET has a lot to be excited about, it not only supports development for a lightweight version of .NET called .NET Core, but for the first time you can also target non-Windows platforms. In this course, Understanding ASP.NET Core, you'll get to see what concepts are most important to get you quickly up to speed. You'll start with seeing what's changed within the project structure and mechanics in Visual Studio. Then, you'll get to see what's new in MVC for ASP.NET Core along with sharing code across existing .NET frameworks and newer .NET Core frameworks. Finally, you'll get to see it's versatility first-hand when an application is deployed to multiple operating systems. By the end of this course, you'll be ready to take advantage of the best new features of this latest release of .NET.

## Deploying a Web App to IIS / Windows Server In 7 Clever Steps

By [Necemon](#)



Here is a simple deployment routine, for a reliable and smooth release :-)

1. Create/configure [the website](#) and [its application pool](#)
2. Copy the files over : [setup an FTP\(S\) connection](#), as well as the client access ([FileZilla](#), [Visual Studio](#), etc.)
3. [Assign writing permissions](#) to the application identity on the relevant folders (for logging, files uploads, etc.)
4. Set time outs to appropriate values  
[Idle Timeout](#): you can change it from the default of 20 to however many minutes you want. You can also adjust the setting to 0 (zero) which effectively disables the timeout so that the application pool will never shut down due to being idle. This can be configured in the Advanced Settings of the application pool.  
[ConnectionTimeOut](#): specifies the time (in seconds) that IIS waits before it stops a connection that is considered inactive. This can be configured in of the Advanced Settings of the Administrative Tools (system.applicationHost/weblimits).

#### 5. Configure Auto-Start

A common problem is the need to perform initialization tasks and "warm up" tasks for a web application. Larger and more complex web applications may need to perform lengthy startup processing, prime in-memory caches, generate content, etc... prior to serving the first HTTP request. One way to fix this is to twist a couple of properties in the [application initialization module](#):

- Set the application pool StartMode property to AlwaysRunning
- Set PreloadEnabled to true, and specify the application pool and path

#### 6. Configure SQL Server / Set up automatic data Backups

Create a [maintenance backup plan with SQL Agent](#)

#### 7. HTTPS / SSL [Certificate Installation](#)

HTTPS provides [security, identity, SEO, access to HTML5 powerful features and more](#).

## Launching Your Application At Windows Startup Without Hacking The Registry

By [Holty Sow](#)

In this post, I will show you two methods to configure your .NET applications so that they would launch as soon as Windows starts. These two methods do not require any change to the registry, hence you don't need to worry about cleaning up that database if the user uninstalls your application.

### First method:

From the Windows Installer Deployment Project, follow these steps:

In the **File System** of the project, add a special folder called **User's Startup** Folder.

Then a shortcut of the project output is created in the Application Folder and renamed to give it a more meaningful name.

We cut (CTRL + X) the shortcut we just created and paste it (CTRL + V) in the User's Startup Folder.

### Second method:

The other solution is to trigger an automatic start from within the code of the application. To do this, I created an activation function, and another one for deactivation.

N.B.: For this code to work, you will need to add a reference to the **Windows Script Host Object Model** assembly in your project.

The activation function can be written as follow:

```
public void EnableApplicationStartup()
{
    string shortcutPath =
System.IO.Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Startup),
    "MyShortcut.lnk");
    if (System.IO.File.Exists(shortcutPath)) return;
    WshShell wshShell = new WshShellClass();
    IWshShortcut shortcut = (IWshShortcut)wshShell.CreateShortcut(shortcutPath);
    shortcut.TargetPath = Assembly.GetEntryAssembly().Location;
    shortcut.Description = "My first shortcut";
    shortcut.Save();
}
```

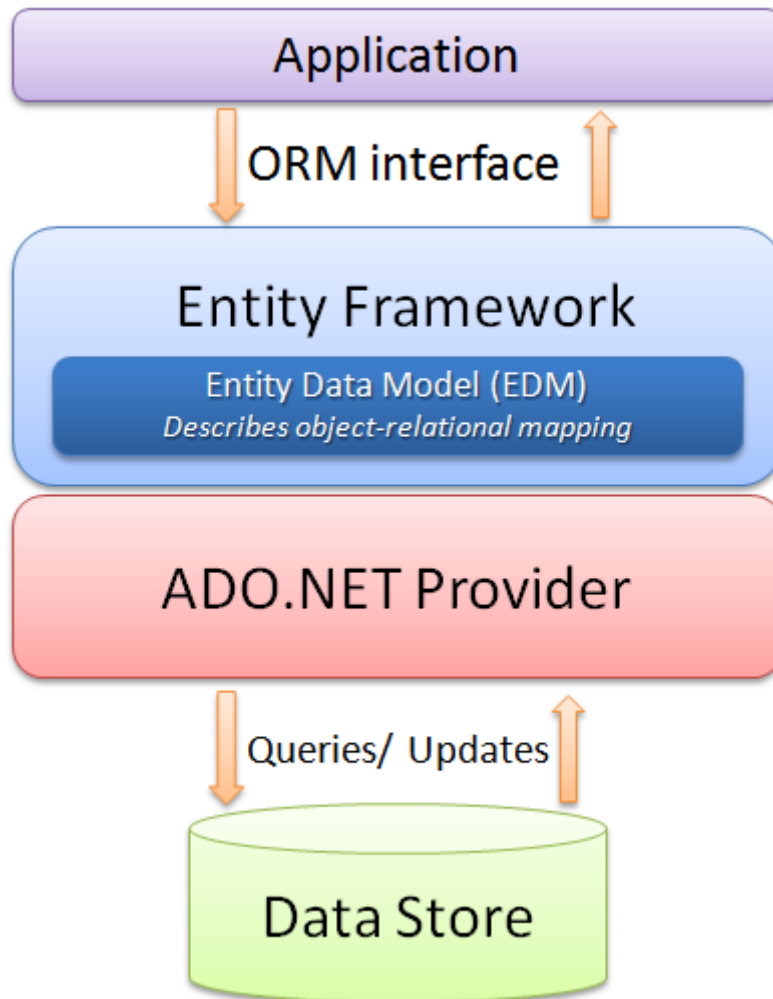
Here is the deactivation function:

```
public void DisableApplicationStartup()
{
    string shortcutPath =
System.IO.Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Startup),
    "MyShortcut.lnk");
    if (!System.IO.File.Exists(shortcutPath)) return;
    System.IO.File.Delete(shortcutPath);
}
```



## My 12 Favorite Entity Framework Tricks

By [Necemon](#)



The Entity Framework is an ORM system, a set of technologies that support the development of data-oriented software applications. It's part of the .NET Framework. I have been playing with it for a couple of years and here are my top tips :

### 1. Extending the Timeout value

When loading large amounts of data, the operations happen to fail every time they hit the default time out, so it may be a good idea to extend it to a few hours.

Doing it when initializing the Context object ensures that it's all set before any database call.

### 2. Second Level Caching

That's a caching of query results. The results of SQL commands are stored in the cache, so that the same SQL commands retrieve their data from the Cache rather than executing the query again against the underlying provider. This can have a performance boost for your application and results in less activity against your database. To enable this feature, you can [download and implement this open source project](#).

### 3. Dynamic Code First and database migration

EF allows you to program against a model without having to deal with a database directly. With the Code-First approach, you can focus on the domain design and start creating classes. Code-First APIs will create/update the database on the fly based on your entity classes and configuration.

By the way, while we are on the topic on configuration, you can do it all within your C# code. Here are a couple of handy properties:

`AutomaticMigrationEnabled` : Set it to true to enable the Code-First magic

`AutomaticMigrationDataLossAllowed` : a value indicating if data loss is acceptable during automatic migration. If set to false an exception will be thrown when data loss may occur as part of an automatic migration.

### 4. Previewing Code-First change details

There is an easy way to generate the SQL scripts that EF plans to execute, before/without actually committing those changes. It goes like this:

```
var configuration = new MigrationConfiguration();
var migrator = new DbMigrator(configuration);
var scriptor = new MigratorScriptingDecorator(migrator);
string script = scriptor.ScriptUpdate(null, null);
```

It could be useful if you want to make some changes to that script, or if you are debugging an issue or if you are just curious about what's going on :-)

### 5. MARS

Multiple Active Result Sets (MARS) is a feature that allows the execution of multiple batches on a single connection. To say it in a simple way, you can make a connection to server and then submit multiple requests to server at the same time, and then read results from these requests in whatever way you want. Just include "MultipleActiveResultSets=True" in your web.config file.

### 6. Setting all the properties constraints in a single place

For example, instead of adding an attribute to each string property, you could set a default string maximum length as follow:

```
protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    modelBuilder.Properties<String>().Configure(p => p.HasMaxLength(360));
}
```

## 7. Inheritance strategy

EF has many ways of dealing with inheritance

- [Table per hierarchy](#) : a single table with all based and derived class properties (A discriminator property being used to differentiate between them)

- [Table per type](#) : base properties on the base table, and for each derived class, derived properties on a separate table

- [Table per concrete type](#) : each derived class gets its own table with all (based or derived) properties.

Pretty extensive, the only case I am not sure about : is there a way to make EF not deal with inheritance at all ? I mean, assuming A derives from B, is there a way to make EF treats A and B as completely different classes and ignore the fact that that one inherits from the other ?

Not a very common scenario, but just for the sake of avoiding duplication, I could want to use inheritance in the C# code so that I don't copy the properties twice in different classes but I wouldn't want to involve any reaction from EF. TPC comes close to that, but the classes still share the same primary key.

## 8. EntityFunctions Methods

When using LINQ to Entity Framework, your predicates inside the Where clause get translated to SQL, which doesn't have a translation for some complex constructions like `DateTime.AddDays()`. That's where [EntityFunctions methods come in the picture](#).

## 9. LINQKit

[LINQKit](#) is a free set of extensions [for LINQ to SQL and Entity Framework power users](#). I mostly use it to dynamically build predicates and insert expression variables in subqueries, but it also allow to :

- Plug expressions into EntitySets and EntityCollections
- Combine expressions (have one expression call another)
- Leverage AsExpandable to add your own extensions.

## 10. Lazy Loading

One of the most interesting Entity Framework features is the Lazy Load function, which is the process whereby an entity or collection of entities is automatically loaded from the database the first time that a property referring to the entity/entities is accessed.

To put it simply, lazy loading means delaying the loading of related data, until you specifically request for it. This could be enabled by setting the `Configuration.LazyLoadingEnabled` property to true.

## 11. AsNoTracking

Entity Framework exposes a number of performance tuning options to help you optimise the performance of your applications. One of these tuning options is [.AsNoTracking\(\)](#). This optimisation allows you to tell Entity Framework not to track the results of a query. This means that Entity Framework performs no additional processing or storage of the entities which are returned by the query.

There are significant performance gains to be had by using [AsNoTracking\(\)](#).

## 12. Staying away from performance traps and other EF gotchas

The object context manager can lead to situations where [EF behaves in surprising ways](#). It's good to [be aware of the patterns](#) that you can follow to [avoid these pitfalls](#).

## ASP.NET MVC: Avoid Polluting The Model Of The View With Error Messages

By [Holty Sow](#)

I recently answered a question that came up on the [StackOverflow website](#). Here was the situation.

Let's consider the following model:

```
public class ForgotPasswordMV
{
    [Display(Name = "Enter your email"), Required]
    public string Email { get; set; }
}
```

This model is used in a controller action as follows:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Search(ForgotPasswordMV viewModel)
{
    if(Temp.Check(viewModel.Email))
        return RedirectToAction("VerifyToken", new { query = viewModel.Email });
    else
    {
        ViewBag.ErrorMessage = "Email not found or matched";
        return View();
    }
}
```

The question was whether using the ViewBag dynamic property of the controller to expose the error message was a good practice, as the OP investigations suggested that it was necessary to expose properties from the model.

Obviously it is highly recommended NOT to use the ViewBag property since it does not provide any Strong Typing. If you want to communicate with the view you should always involve a typed model. The suggested solution is therefore legitimate in that respect, however, it is not a good practice when it comes to model error handling in ASP.NET MVC.

The solution for exposing error messages (as in the previous example, which does not use data annotation attributes) derives from the use of the AddModelError method, from the ModelStateDictionary class. We do not need to instantiate this class because a ModelState property containing an instance of this class already exists in the Controller.

Therefore, the right solution is the following:

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Search(ForgotPasswordMV viewModel)
{
    if
    {
        // ...
    }
    else
    {
        this.ModelState.AddModelError("Email", "Email not found or matched");
        return View(viewModel);
    }
}

```

It should be noted that this method receives as first parameter the name of the property of the model to which the error message is associated. So, in order to display this error message in the view near the Email field, you can simply add the following line next to it:

```
@Html.ValidationMessageFor(m => m.Email)
```

However it is possible to have a general error message for the whole model, that is, an error message which is not attached to any property of the model. For this, it would be necessary to use an empty String as first parameter:

```
ModelState.AddModelError(String.Empty, "Email not found or matched");
```

In the Razor View, use the following line:

```
@Html.ValidationSummary(true, "The following error has occurred:")
```

The first Boolean parameter indicates that you do not want to display error messages that are already attached to model properties.

In this post, we saw that we do not need to pollute our model, nor do we need to attach our error messages to specific properties. It's all about using ASP.NET MVC tools to make things easy.

I hope this post has been helpful to you.

## Limiting The Execution Of An Action To AJAX Requests Only

By [Holty Sow](#)

In ASP.NET MVC, we manipulate Views, among other things. Some of these Views represent full pages and some are just page parts. These parts or areas belonging to a view are called partial views, and they are also returned by controller actions. Since these partial views should only be used within a view, the ASP.NET MVC framework allows us to protect any call to these partial actions by decorating them with the **ChildActionOnly** attribute. This attribute makes sure that the action:

- cannot be used as an entire view and the application developers will always run it using the **HtmlHelper.Action** or **HtmlHelper.RenderAction** methods.
- has a URL that will not be accessible via the address bar, if a user somehow becomes aware of the existence of this URL.

However, as with any dynamic site, we will have AJAX requests that can also make requests for HTML content without having to load the page completely. This content also represents a part, and when we receive the response from the web server we have to embed this piece of HTML somewhere in the page. The AJAX request sent to the server will certainly invoke a controller action. This action, like those marked with the **ChildActionOnly** attribute, must have these constraints:

- should only go through AJAX requests.
- inaccessible via the browser address bar.

But the ASP.NET MVC framework does not offer any attributes that allow us to apply these restrictions to an action, but it gives us the tools to create them. For this, we need to code a filter that will be executed just before the execution of the action in question. If the incoming request complies with the requirements of a request made in AJAX then we let the action continue its way. In case the conditions are not met, we return a 404 page (as if the URL didn't exist).

The code of the new filter that I will call **AjaxOnlyAttribute** (derived from the **ActionFilterAttribute** class) is as follows:

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = false)]
public class AjaxOnlyAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        if (filterContext.HttpContext.Request.IsAjaxRequest())
        {
            base.OnActionExecuting(filterContext);
        }
        else
        {
            filterContext.HttpContext.Response.StatusCode = 404;
            filterContext.Result = new HttpNotFoundResult();
        }
    }
}
```

The new class only overrides the method that we are interested in, that is, the **OnActionExecuting** method that is called just before the start of the action invoked by the incoming request. The attribute can be set on the controller to handle the set of actions where an AJAX request is mandatory.

When getting new developers onboard, they might not immediately understand why those requests keep returning **404** messages. To avoid wasting time in recurrent explanations, I think it would be worth having the **DEBUG** mode enabled by default for developers. The code of our class would then look like this:



```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = false)]
public class AjaxOnlyAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        if (filterContext.HttpContext.Request.IsAjaxRequest())
        {
            base.OnActionExecuting(filterContext);
        }
        else
        {
            #if DEBUG
            filterContext.Result = new ViewResult { ViewName = "AjaxOnly" };
            #else
            filterContext.HttpContext.Response.StatusCode = 404;
            filterContext.Result = new HttpNotFoundResult();
            #endif
        }
    }
}
```

Since developers have to compile the application in **DEBUG** mode most of the time, that **AjaxOnly** View can provide them with some explanations and instructions. Note that this version of the class only works if an **AjaxOnly.cshtml** View has been added to the **Shared Views** directory of the application.

Alternatively, we do not have to return a partial view. We can simply replace the following code:

```
filterContext.Result = new ViewResult { ViewName = "AjaxOnly" };
```

With some simple text content:

```
filterContext.Result = new ContentResult
{
    Content = $"This action '{filterContext.HttpContext.Request.RawUrl}' was designed for  
AJAX requests only"
};
```

However, the version that uses an actual View has an advantage, it's the ability to systematically render the relevant page "**\_Layout**", typically defined at the application level.

## [Interview] In The Bubble Of Holty Sow: "for more efficiency, I prefer to work in an agile team"



### **1. Quick (but effective) introduction of the personage and her achievements?**

My name is Holty Samba SOW. After having started my higher studies in Maths-Physics, then completed a professional degree in Computer Science in Senegal, I went on to further my studies in France and obtained a Master's degree in [Electronic Documents and Information Flow](#) in 2009. This allowed me to get into the professional world of computer programming. Currently I am a consultant and developer, mainly on [the .NET platform](#), at SoftFluent, which is a software publisher but also a service company.

### **2. What are your main goals in life?**

At the moment, my main goal is to keep my passion for computer programming for as long as possible by accumulating a maximum of experiences in the Web and Mobile world. I hope to enable my native country, Senegal, to benefit from this wealth of experience.

### **3. What tools and techniques do you use to accomplish things efficiently?**

My favorite IDEs: VS Code for anything related to CSS, JavaScript and TypeScript. For everything else, I use the Visual Studio IDE with the ReSharper extension for more efficiency.

My favorite technologies are generally anything that involves back-end technologies on the .NET platform, like ASP.NET MVC, ASP.NET Web API and recently ASP.NET Core.

The methodology I prefer to work in a team is Scrum/Agile.

#### 4. Any recommendations for your juniors?

A book: "C# in a Nutshell" to thoroughly understand the C# programming language.

Twitter and Blogs: following [influencers \(tech speakers, CTO, etc.\) via Twitter or through their blogs](#), can help in staying aware of some good stuff. [links]

Websites:

[docs.microsoft.com](https://docs.microsoft.com) (I think the new Microsoft documentation website is quite well developed);

[stackoverflow.com](https://stackoverflow.com) (I think that trying to answer questions from other developers can sometimes help in uncovering tips and tricks that we would not have thought of otherwise);

[pluralsight.com](https://pluralsight.com) (a lot of videos about programming, really interesting... The only catch : there's a charge for it);

[channel9.msdn.com](https://channel9.msdn.com) (same as Pluralsight, but it's free and the speakers are Microsoft employees).

#### 5. What is the best way to contact you?

Email: [holty.sow@gmail.com](mailto:holty.sow@gmail.com)

Blog : [hssow.wordpress.com](https://hssow.wordpress.com)

LinkedIn : [linkedin.com/in/holty-samba-sow-43042715](https://linkedin.com/in/holty-samba-sow-43042715)

Twitter: [twitter.com/CodeNotFound](https://twitter.com/CodeNotFound)

Facebook : [facebook.com/CodeNotFound](https://facebook.com/CodeNotFound)

## The Xamarin Revolution

By [Necemon](#)



What is Xamarin?

Xamarin is a free, cross-platform development tool. It solves dilemmas many developers face when developing cross-platform apps: separate coding languages and UI paradigms. With Xamarin, [you can use C# for iOS, Android, and Universal Windows apps](#). And with Xamarin Forms, interface design for all three platforms can be accomplished within its XAML-based framework.

The Xamarin platform consists of a number of elements that allow you to develop applications for iOS and Android:

C# language – Allows you to use a familiar syntax and sophisticated features like Generics, Linq and the Parallel Task Library.

Mono .NET framework – Provides a cross-platform implementation of the extensive features in Microsoft's .NET framework.

Compiler – Depending on the platform, produces a native app (e.g. iOS) or an integrated .NET application and runtime (e.g. Android). The compiler also performs many optimizations for mobile deployment such as linking away un-used code.

IDE tools – The Xamarin Studio IDE and the Xamarin in Visual Studio allow you to create, build and deploy Xamarin projects.

In addition, because the underlying language is C# with the .NET framework, projects can be structured to share code that can also be deployed to Windows Phone.

Here is [some documentation I would recommend](#) if you consider getting started with Xamarin :

[Developer Home Page](#)

[Cross Platform Guide](#)

[10 minutes introduction for Android](#)

[Android Guide](#)

[10 minutes introduction for iOS](#)

[iOS Guide](#)

[The Xamarin Show](#)

[Xamarin University](#)

[Pluralsight courses](#)

[Lynda Courses](#)

## Top 8 Tricks with Xamarin Programming

By [Necemon](#)



If you don't know what Xamarin is, may I suggest that you start by reading my [introductory notes](#) before proceeding with this?

For those in the now, let's get into it without further ado:

### [Xamarin Forms](#)

Xamarin.Forms is a cross-platform UI toolkit that allows developers to easily create native user interface layouts that can be shared across Android, iOS, and Windows Phone. You can build native UIs for iOS, Android and Windows from a single, shared C# codebase. This means that applications can share a large portion of their user interface code and still retain the native look and feel of the target platform. Xamarin.Forms allows for rapid prototyping of applications that can evolve over time to complex applications. Because Xamarin.Forms applications are native applications, they don't have the limitations of other toolkits such as browser sandboxing, limited APIs, or poor performance. Applications written using Xamarin.Forms are able to utilize any of the API's or features of the underlying platform, such as (but not limited to) CoreMotion, PassKit, and StoreKit on iOS; NFC and Google Play Services on Android; and Tiles on Windows. In addition, it's possible to create applications that will have parts of their user interface created with Xamarin.Forms while other parts are created using the native UI toolkit.

### [Custom Renderers](#)

Xamarin.Forms user interfaces are rendered using the native controls of the target platform, allowing Xamarin.Forms applications to retain the appropriate look and feel for each platform. Custom Renderers let developers override this process to customize the appearance and behavior of Xamarin.Forms controls on each platform.

Custom renderers provide a powerful approach for customizing the appearance and behavior of Xamarin.Forms controls. They can be used for small styling changes or sophisticated platform-specific layout and behavior customization. This article provides an introduction to custom renderers, and outlines the process for creating a custom renderer. Xamarin.Forms Pages, Layouts and Controls present a common API to describe cross-platform mobile user interfaces. Each page, layout, and control is rendered differently on each platform, using a Renderer class that in turn creates a native control (corresponding to the Xamarin.Forms representation), arranges it on the screen, and adds the behavior specified in the shared code.

### [SQLite](#)

Most applications have some requirement to save data on the device locally. Unless the amount of data is trivially small, this usually requires a database and a data layer in the application to manage database access.

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including [several high-profile projects](#).

iOS and Android both have the SQLite database engine "built in" and access to store and retrieve data is simplified by Xamarin's platform.

### [ACR User Dialogs](#)

A cross platform library that allows you to call for standard user dialogs from a shared/portable library : Actionsheets, alerts, confirmations, loading, login, progress, prompt, etc.

### [Xlabs](#)

Xamarin Forms Labs is a open source project that aims to provide a powerful and cross platform set of controls and helpers tailored to work with Xamarin Forms : AutoCompleteView, Calendar Control, ImageButton, HyperLinkLabel, etc.

### [UI Sleuth](#)

A Xamarin.Forms debugging tool and UI inspector. If you've ever made a web site, it's similar to Microsoft's F12 tools or Chrome Developer Tools. You can use it to efficiently track down layout issues, prototype a new design, and remotely control a device.

### [GenyMotion Android Emulator](#)

Fast and easy Android emulator. The most powerful Android emulator for app developers and testers. It's available for free or premium on Windows, Mac and Linux.

### [Remoted iOS Simulator](#)

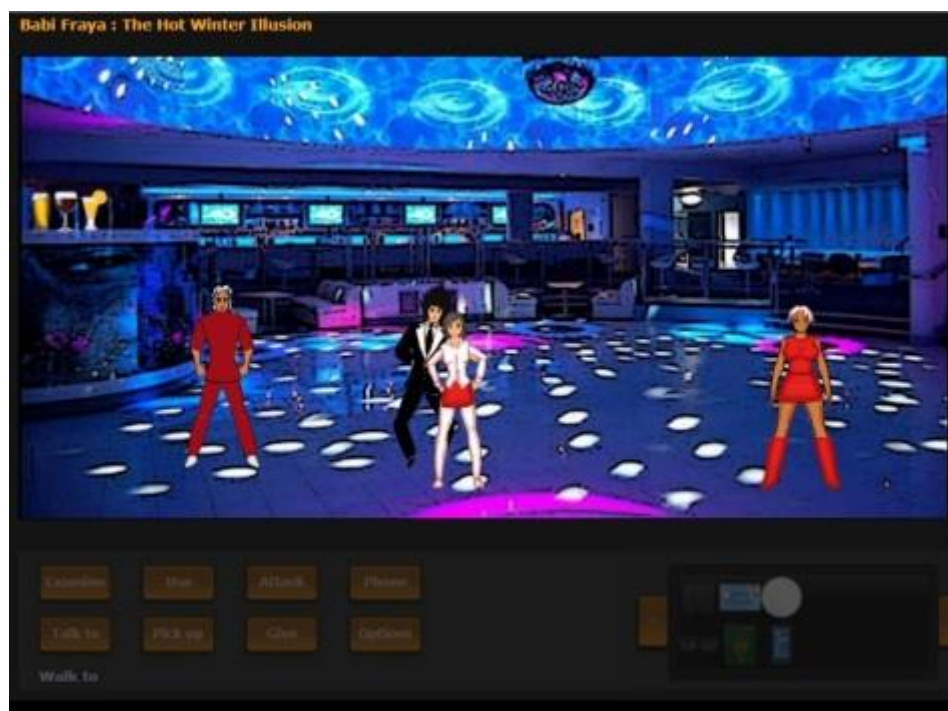
To test and debug iOS apps within Visual Studio on Windows. Most modern Windows computers have touch screens, and the remote iOS simulator lets you touch the simulator window to test user interactions in your iOS app. This includes pinching, swiping, and multiple-finger touch gestures - things that previously could only be easily tested on physical devices.



# Chapter 8

## Epic Prototypes, Classic Projects, Historic Genre

[necemon.com](http://necemon.com) | [babifraya.com](http://babifraya.com) | [evasium.com](http://evasium.com) | [kpakpatoya.com](http://kpakpatoya.com)



## Inside Bavardica - Part 1 : Discussion of the subject areas

By [Necemon](#), 2010

A modern instant messenger is composed of various features which include public conversations, private conversations, user picture and signature. When it's time to innovate and make an instant messenger that is so special that it is both interesting and standing apart from whatever program already available on the market, it is not too obvious to figure out what extra features could be attached. It feels like almost everything had been already thought of and implemented in some other application.

However, [Bavardica](#) is my attempt to make a communication platform like no other. Bavardica, derived from Bavardage, which actually means talking in French, has got mostly two assets over regular instant messengers. Firstly, there is the permanent presence of artificial beings that interact with other users. Secondly, the application relies on a two-dimensional space which represents a small world in which the users are inhabitants represented by animated characters. This project is a good chance to set the scene of what could become an innovative virtual world.



Many areas of web entertainments are involved in the Bavardica Project. In its initial form, Bavardica was basically a chat application, but as it is evolving, Bavardica also fits in the following fields.

## Artificial Intelligence

As taught to me by [Dr Phil Grant](#), the main objective of AI (Artificial Intelligence) is to try to make computers perform tasks that humans tend to be good at. The actual name “Artificial Intelligence” was coined by John McCarthy in the 60’s. There are a lot of topics which might nowadays be considered as part of the general subject of Artificial Intelligence. But for our current purpose, we shall only consider the themes of natural language understanding and representation of knowledge.

Now, what does it mean for a machine to be intelligent? Alan Turing thought about this problem in the 50’s and came up with the following test: A human would ask questions to both a computer and a human in some other rooms; if the interrogator is not able to figure out (from the answers) which is the machine then the computer is said to behave in an intelligent manner. A very early attempt to build a program which would pass the Turing test was the ELIZA program (by Joseph Weizenbaum). ELIZA was mimicking a conversation with a psychologist and was quite impressive at first glance. However, it was missing a common sense and an actual understanding of natural language (which are both pretty hard to acquire from a computer point of view). Nevertheless, many programs of this kind followed. They were based on tricks like:

- o The repetition of user’s statements verbatim (subject to pronominal adjustments)
- o Preceding the repeated statement with the introductory “Why do you need to tell me”
- o Asking “Why do you ask that”, which, in effect, change the topic or the level of the conversation.

Later on, Colby and AI wrote a program called PARRY based on ideas in ELIZA which had more complex properties like elements of emotion and more knowledge of grammar. It was intended to mimic a paranoid patient who believed he was being persecuted. In a real experiment, 33 psychiatrists were asked to rate the degree of paranoia of patients who were communicating through typed transcripts. Three of them were from human patients and two generated by PARRY. None stated that they thought there was anything unusual. Later the psychiatrists were told that some of the transcripts were machine generated, but they were, on the whole, unable to recognise which ones were from PARRY. Of course this whole situation is a bit odd anyway, as human paranoids behave in strange ways. The point here is that it may be difficult to differentiate a human talk from a machine talk (Phil Grant).

Moreover, throughout the time, AI systems are becoming more and more advanced, in fact, more intelligent. Currently, one of the strongest programs of this type is ALICE (Artificial Linguistic Internet Computer Entity) by Richard Wallace. According to Wikipedia, that’s a program inspired from ELIZA that engages in a conversation with a human by applying some heuristical pattern matching rules to the human’s input. The program uses an XML schema called AIML (Artificial Intelligence Markup Language). I use a similar model for implementing the AI feature in this project.

**Virtual worlds:**

A virtual world can be defined as a synthetic reality, an environment that is completely generated by a computer system and that is completely transportive in that the participant controls an avatar immersed in an artificial world, according to the book *Networking and Online Games* (Armitage, Claypool and Branch, 2006). It says that virtual worlds are intended for his users to inhabit and interact. Sometimes, the user can also manipulate elements in the said world and therefore experience a kind of telepresence. As the rate of people using virtual worlds is increasing by 15% every month and as there are nearly 600 million people worldwide registered in virtual world today (according to K Zero, a virtual word consultancy service), I felt that it would be a good plan to learn about this trend and give it a try. In its current form, Bavardica is a virtual world in the sense that each bavard (avatar) is aware of the other bavards and of the scene boundary.

**Multimedia:**

In a web browser, operations like timing, streaming, playing music forth and back, playing a list of music files etc. are not at all trivial using basic html and javascript. One big advantage of a browser plugin like Silverlight is that it gives much more ease about operations on media files. The book 'Accelerated Silverlight 3' had been really valuable by teaching me how the `System.Windows.Controls.MediaElement` control provides media playback capability. That control can handle both audio and video in a variety of formats like mp3, wmv, asx, etc. There is even a support for third-party codecs and for new media formats such as H.264, Advanced Audio Coding and mp4. Furthermore, the media element in Silverlight has a lot of properties, methods and events that allows a lot of control over the media. Some of the properties are `AutoPlay`, `CurrentState`, `CanSeek`, `IsMuted`, `Position`, `NaturalDuration`, `NaturalVideoHeight` and `Volume`. The methods of the `MediaElement` control include `Pause`, `Play`, `SetSource` and `Stop`. Some of the events are `CurrentStateChanged`, `MediaEnded`, `MediaOpened`, `MediaFailed`, the later allowing some more error handling in case the media cannot be found or in the case the format is incorrect.

This control allows us to bring a symphonic soul to this project. So another aspect of the Bavardica project is the audio streaming, that is now in its most basic form with an mp3 music file playing in the background during the conversation. There are also sound effects that serve as alerts informing that a new user just logged in or that someone just left a new message.

We will now proceed and [describe how the application has been build.](#)

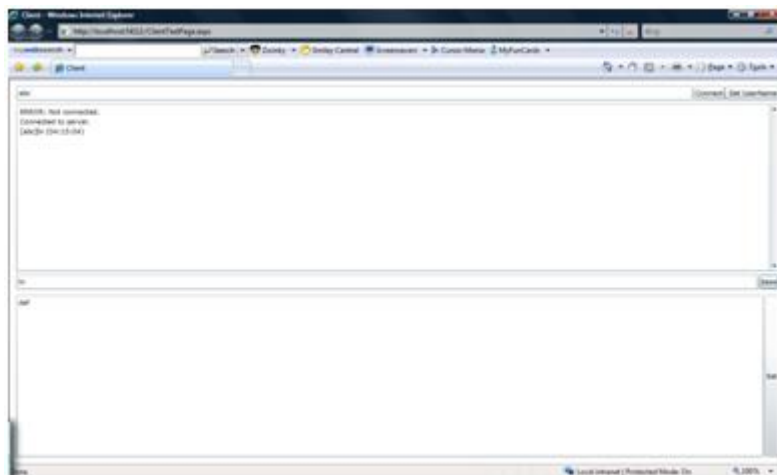
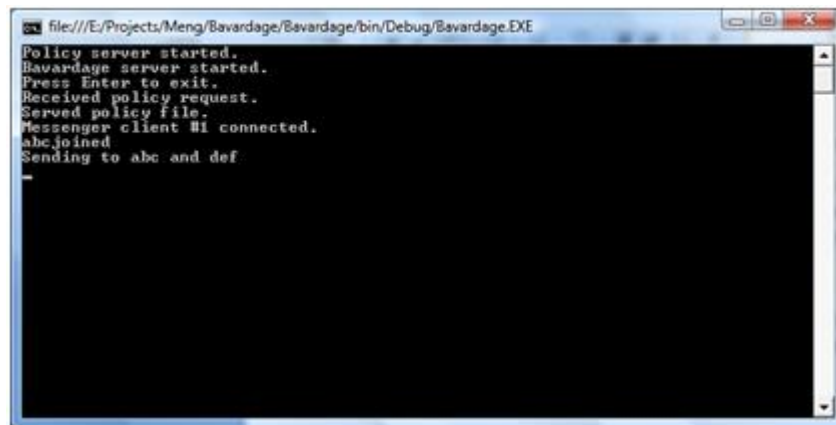
## Inside Bavardica - Part 2 : Progress and Achievement of the Project

By [Necemon](#), 2010

As we discussed the [subjects areas](#) related to [Bavardica](#), we can now go through the steps in the actual achievement of the project.

### 1. Sending a public message over the TCP network from a Silverlight client

The model selected for this project is the prototyping model. Starting first with a throwaway prototype, I could first send simple TCP packets using Silverlight and a console application as a server. It was a very basic chat that let a user, say A send a message to a user B so that no one else will receive the message apart from A and B. The server is a .Net console application and the client is a Silverlight application hosted in an ASP.NET page.



### 2. First contacts with Silverlight animation

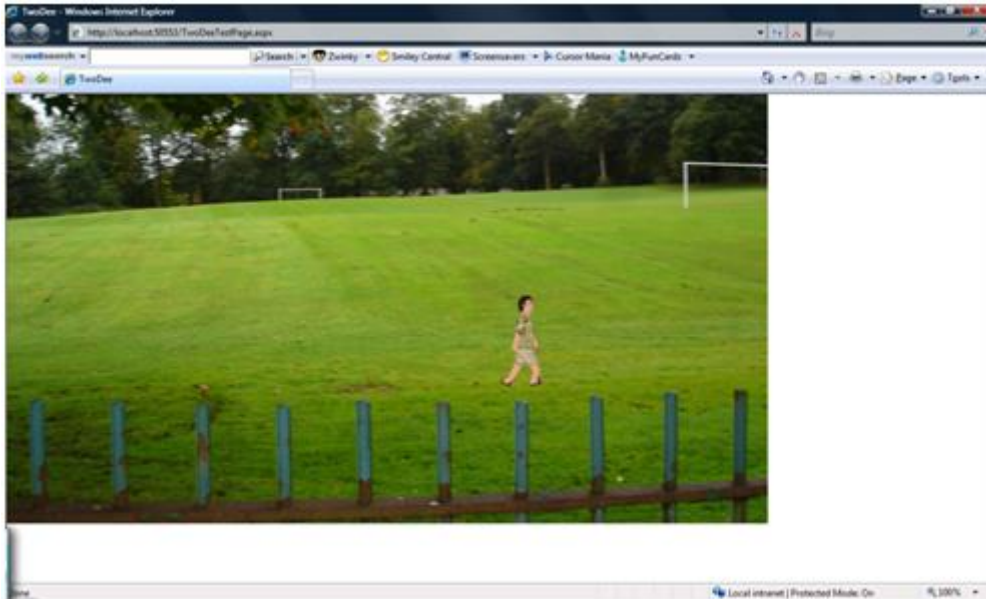
The second program I wrote was a simple Silverlight application with an animated 2d character walking around a park that was actually the background of the web page. This experiment had for sole objective the study of the motion of a character over a 2D scene.



The first key point of this prototype is the way the avatar is animated when a direction key is pressed:

The second key point is that the model used a sprite sheet or frame by frame animation. It consists of sliding horizontally a sequence of adjacent frames with a transparent background, one frame at a time when a directional key is hit.

This approach used `KeyAnimationUsingKeyFrames` but will be dropped later in the project as it involved no interpolation and that it would have required a lot of time drawing each character in every single position.



### 3. Making the server lies silently in the background (As a web service)

It was planned to integrate the two prototypes so that users could both chat and move around the 2D scene. It was also planned that the server application lies silently in a web service so that a console application need not to be kept open throughout the operation. Both goals have been reached in the next prototype.

The fact is that I thought about building a more robust server application based on which I would build an evolutionary prototype that will gradually grow with the project. That server had two main requirements:

- It had to get rid of the console application. It should just be a silent process in the background. For this purpose, I just used web services.
- The server had to push data to the clients. That is sending data even when they are not requesting. In the classic HTTP model, first the client makes a request to the server and then only the server can respond. A solution to this issue was to use the WCF (Windows Communication Foundation) Duplex web services; and that's what I did. That's an asynchronous two ways communication model.

It's worth observing how the [WCF Duplex web services work](#)

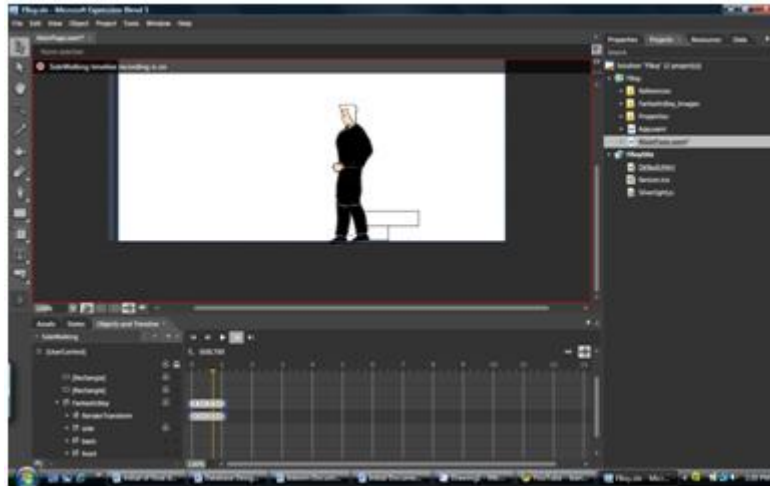




So even if this is mainly a programming project, and not a graphic project, it's worth mentioning that quite a lot of time and effort goes in drawing and animating original characters. Customization is also an important feature of a virtual world. Users should be able to choose how they look and how they are dressed. For this project, I chose to draw four basic characters (2 males and 2 females) and at least two costumes and hair styles for each character. Due to the time constraint again, I could not draw and animate more than that. The operation involves fully drawing and painting the bodies then drawing clothing over. Three basic skin colours are available and the users can change the colours of their clothes and hair. Here is a preview of one of the character:



The characters are basically animated to walk, to sit and to stand. The problem with animations is that it had to be done every time from the three angles (front, back, side). There is also an animation to make them look alive even when they are standing idle. Those models used the `DoubleAnimationUsingKeyFrames` model by opposition to the previous one who used `KeyAnimationUsingKeyFrames`. Microsoft Expression Blend had been used to process the animations:



There are two main advantages to the animation system that had been used:

- Silverlight's interpolation makes things silky smooth
- The characters are customisable. They can just shift boots or hair style (or anything else) as the users want.

Unfortunately, handling characters in a virtual world is not only about design. It involves a lot of coding. Let's start with the creation of a character.

A few array lists are maintained, offering a variety of outfits to the user, as well as shapes and genders

According to the choice of the user, the relevant parts are exposed and the parts that are not relevant are hidden. However, the changes in colour follow another logic.

What happens here is that the source file for each part of the body is fetched from a different directory that changed according to the choice of the user. Later in the application, when the user can change colours about all the parts of their outfits, the same logic is applied.

This was about the character creation. Once the character is created, details about its outlook are stored in the database as expressed in the database design. From there, every time the user comes online, the server can send all its details to all the other clients. As soon as a client is aware of another user online, he has to represent the relevant character in his scene according to the data that had been submitted:

Setting the other characters details work as in the character creation. It's mainly about hiding and showing images.

Now that we have seen how the code involved in characters appearance works, we will now attack the code involved in animation and motion. It starts with a key pressed by a user that is handled by Silverlight. In our example, we will consider the right key:



## 6. The Cybavard

After the character design and animation, an emphasis has been given to the Cybavard, the artificial intelligence part of the project. The logic behind the Cybavard is inspired from the A.L.I.C.E bot introduced earlier in this presentation. What is interesting about A.L.I.C.E is that it was based on an XML dialect called AIML (Artificial Intelligence Markup Language). Due to the universal nature of XML, it can be used with a wide variety of language. The point is that, as this is a .Net project, AIML could also fit to the Cybavards. It was all about finding a C# AIML interpreter. For this project, I used a .net library file, AIMLBot.dll (by Nicholas Toll), released under the GNU General Public License. That program offers a lot of advantages:

Very small size: about 52 k

Very fast: can process 30 000 categories in under one second

Means of saving the bot's brain as a binary file

A simple and logical API

The inner working of this program is also fairly simple. It's based on pattern matching: for each sentence entered, it checks in its knowledge base which stimulus response would fit best:

So what I am doing is adding a reference to the binary file from my web service, I also upload the knowledge base on the server. The knowledge based is mainly a set of AIML files that I can edit as per my requirements. I could also add my own AIML files. From the cybavard class, what I had to do was creating a bot object and loading the AIML files.

In the following part, we will go through [the design of the characters](#).

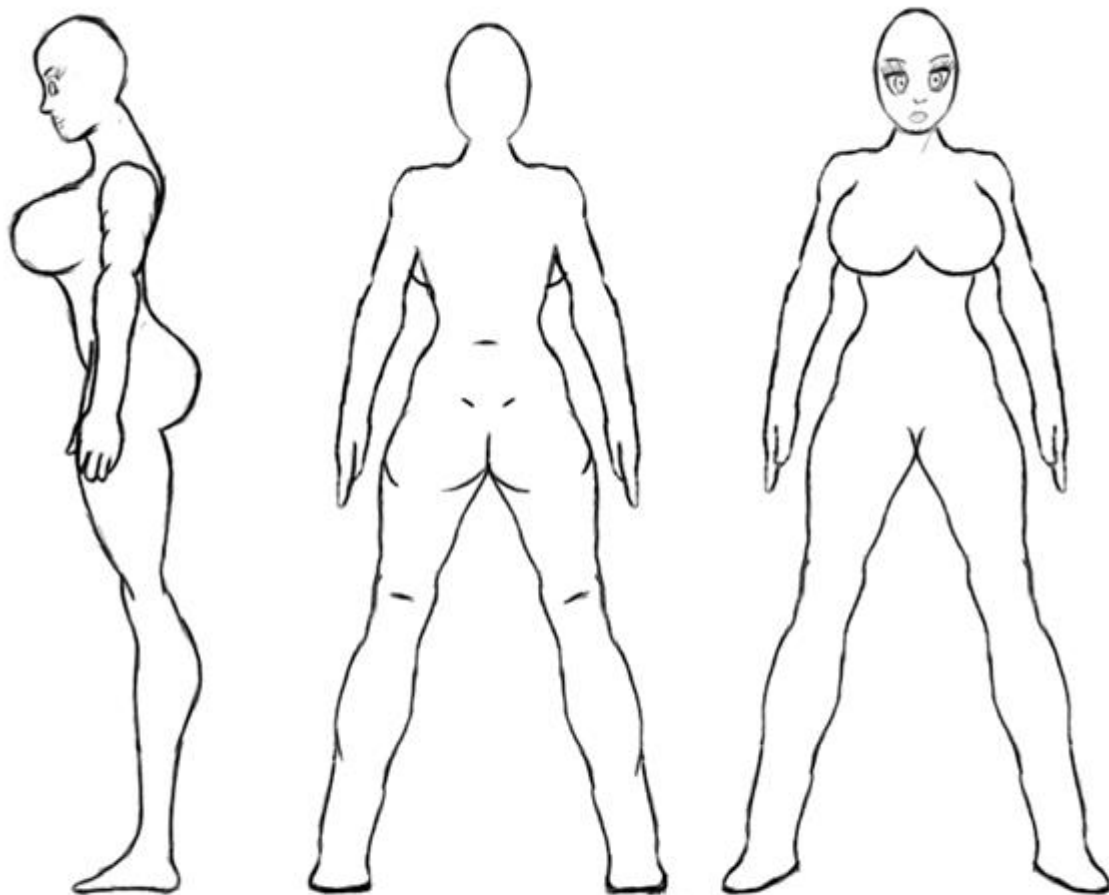
## Inside Bavardica - Part 3 : Design of the 2D characters

By [Necemon](#), 2010

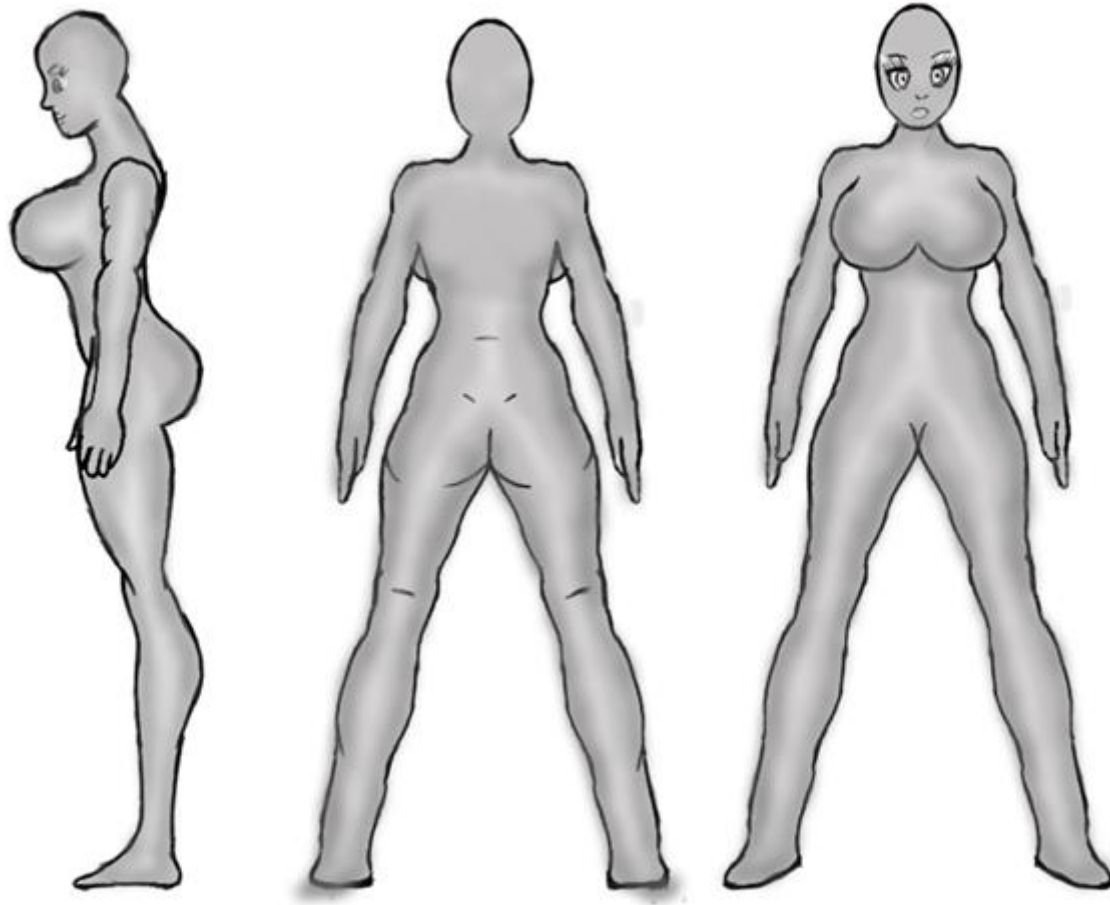
### How I draw the 2D characters

Due to the nature of the drawing art, which is not really computer science related, I thought it was best to describe this process in another part right after [the progress description](#). This work had been a full part of the project and even though it does not directly concern programming (before we get into animation), it is also a computer work involving a pen tablet and some image processing software (Adobe Photoshop). The following screenshots describe the steps in the drawing of one character (out of four).

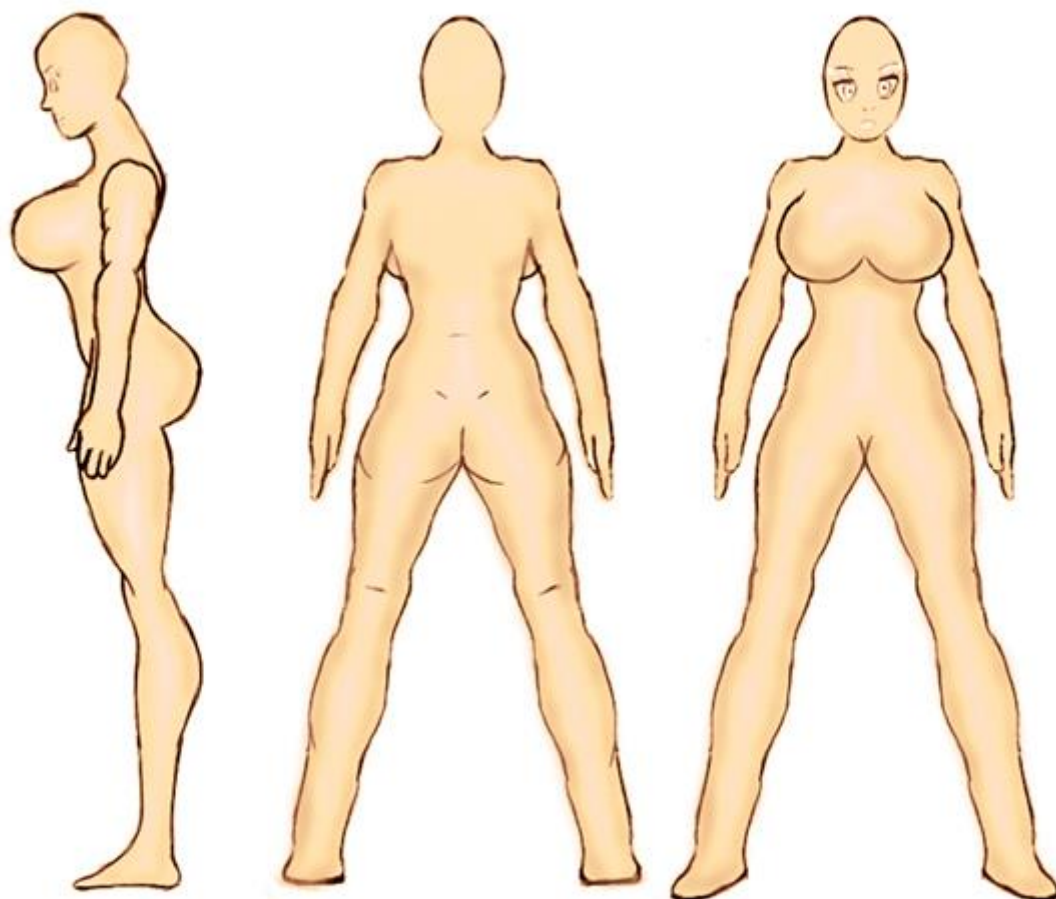
#### A. Sketching



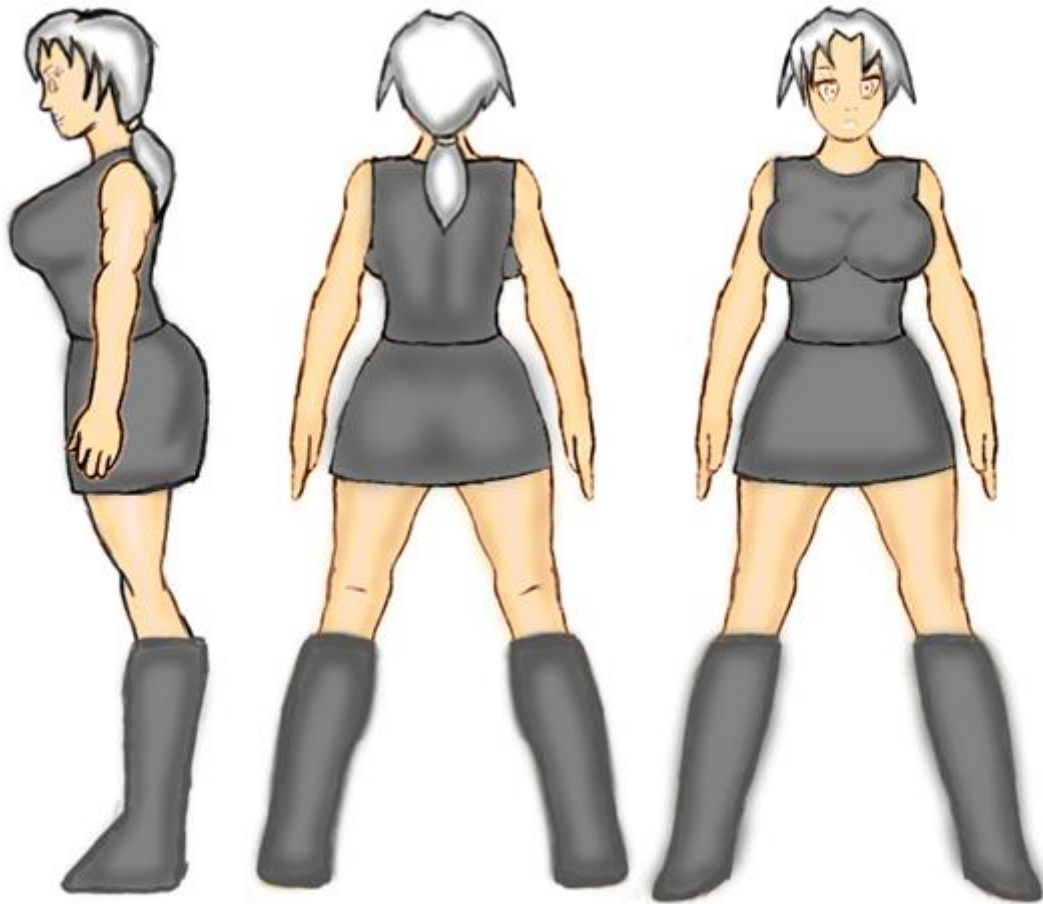
**B. Filling**



### C. Coloring



#### D. Clothing



For details about animation techniques, you can check [those tutorials that Darren Mart wrote](#). To check out what's coming next with Bavardica, [move on to the next part](#).



## Inside Bavardica - Part 4 : Scope for future improvements

By [Necemon](#), 2010

This is the last part of the [series](#) on [Bavardica](#). Apart from a user friendly and colourful user interface, the other features include the edition of the [character outlook](#) in real time (while on the scene). Here again, hair style, skin, clothes and shoes can be changed. Additionally, colours of all those garments can also be changed. The way it works is pretty similar to the way a character is created.

In the meanwhile, some other functions had been added:

**Collision detection:** This is to make sure that bavards don't get out of the scene boundary. Also, they don't walk inside each other when move vertically. This feature is based on simple "if conditions" that check whether the move doesn't not lead outside the scene or at a position occupied by another character.

**Bubbling:** That function is used to process text entered in bubble (graphic mode). Every time a user sends a message, it appears on the scene as a flying bubble that gradually disappears. An animation had been designed for this purpose. So both motion and opacity animations are applied to the bubble object. Each key pressed is processed individually as the bubble does not rely on a textbox. The main issue in this task was that the positions of special characters change according to the keyboard type. Finally the bubble handles pretty smoothly conversations from American and british keyboards.



More rooms

The current application contains a single scene where all the public conversations happen. A good extension could be to have many scenes and a map to navigate among the scene.

### More music

The music background is always the same mp3 track playing all over again. It is intended in the next version to have a playlist of tracks as music background. Further, since there will be many rooms, each room will have a different theme and a different kind of music.

### More cybavards

Due to the large amount of time and data required to simulate the talk of a single cybavard, it has not been possible to provide more than one of those cyber bavards. The process in their creation is pretty simple. It's mainly a matter of time. For instance, the cybavard in this demo has a knowledge base of nearly 4 megabytes (of text).

### More animations for a better interaction

The only animations available at the moment are walking in all 4 directions and breathing while the character is idle. It is planned to add more animations like dancing, running, jumping. There will be some moves to express the character's mood. At a later stage, there could be animations that involve more than one character, like hug or hand shake. It is also planned to have more animations for characters to express their moods.

A working version is available at <http://bavardica.com>

## 10 things I learned from building Bavardica

By [Necemon](#), 2010

This is just a summary of the most important lessons I learned from building and publishing Bavardica. I sincerely hope you get some useful information (or reminder if you already knew all of this) from that. ( PS: a working version of bavardica is available at <http://bavardica.com> )

**1. Nobody will use an application unless it provides some added value**, something interesting they don't already find in what they normally use. This is so true, even for free applications. They might try it but they would give it up quite soon if there is no actual incentive.

**2. The very fact that they are asked to register will discourage many people from trying a web application.** The most obvious causes might be that they don't want to waste time doing it, or they don't want to give away their details for privacy purpose. So at least a trial version should be offered to visitors without registration so that they can see what it's all about before they make any decision.

**3. The various commands and features are (only) obvious to the developer who built the application.** However, they may appear as strange and complicated to use for some users. The application should therefore be as simple as possible and there should be some hints for every command (right inside the application, not on an external file).

**4. Google is your friend** (well, I guess search engines in general). I spent a lot of time reading about Silverlight at the beginning of the project but whenever I had a doubt, I relied more on web searches than on books. I could find most of the answers I was looking for in technical forums. I could also find related work from senior developers (mostly [Anoop Madhusudanan](#), [Tomasz Janczuk](#) and [Darren Mart](#) in this particular occasion).

**5. Client-server architecture and silent web services.** This one is purely technical. I learned how to make an application lies silently in the background as a web service or a windows service; And I learned how to push data to the client over HTTP using a WCF Duplex Polling service.

**6. The more choice, the merrier.** Users enjoy having a lot of content to choose from. They like to express their personality and. to customize their character. Art and media requirement consume much more time than actual coding, but isn't it worth it? In the coming versions I would definitely be adding more rooms, more clothes and more animations. But...

7. **Not all at once.** It's important to move one step at the time not to get overwhelmed. It's all about priority management.

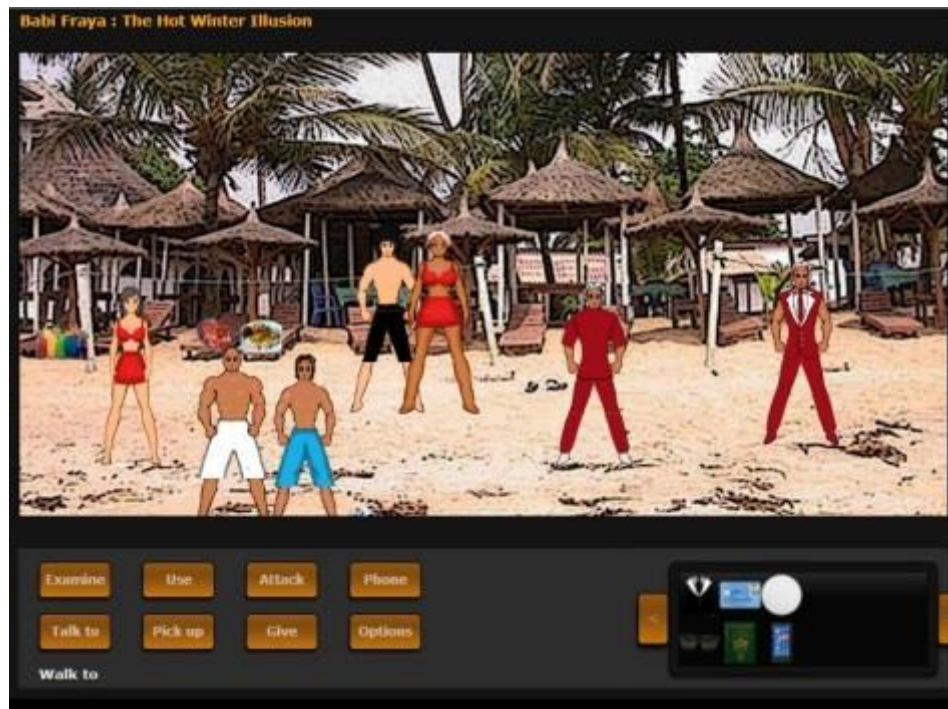
8. **Work results can be rewarding.** I am mostly talking here about the satisfaction and sense of accomplishment one's gets from work. It feels good to accomplish something, even if it's just a small thing for now. It motivates me to keep going.

9. **Feedback is priceless.** I could not learn all those lessons I am describing in these posts if I didn't have some guys and girls to show me my mistakes, to tell me what they didn't like about my work and how I could improve it. I would like to take this chance to thanks all those who tried Bavardica. Thank you for your feedback.

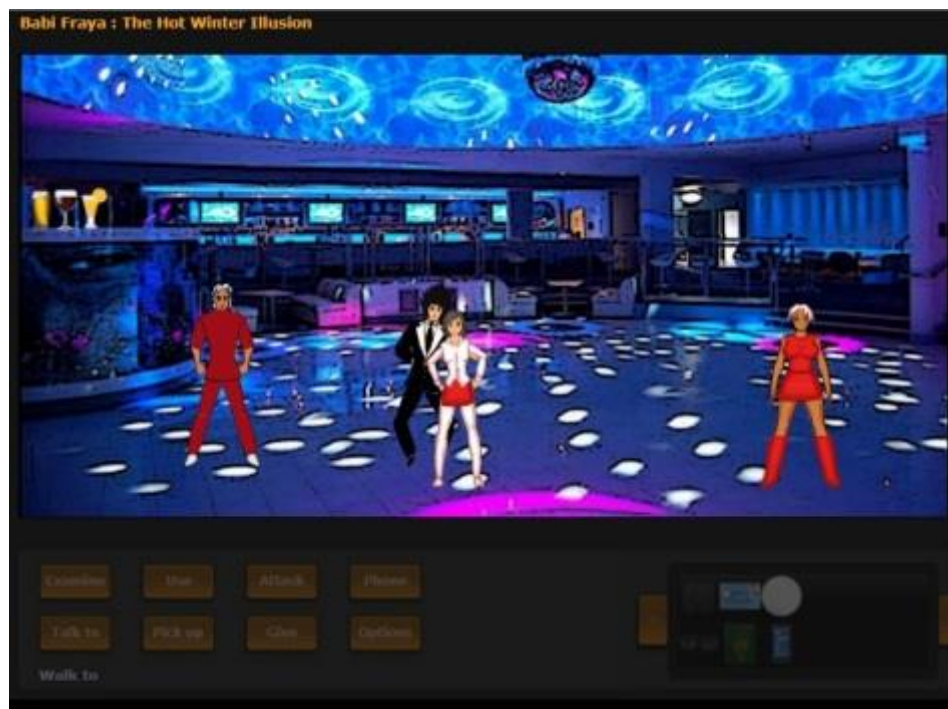
10. **There is a long way ahead** (to build a decent virtual world).

## **Babi Fraya: The Hot Winter Illusion**

By [Necemon](#), 2013



I just published a beta version of my [first complete game](#). If you have a moment to try it, I hope you will enjoy it : <http://babifraya.com/>



**Babi Fraya™** is a ["Point & Click" adventure game](#), therefore a game in which the player assumes the role of protagonist in an interactive story driven by exploration, dialogs and puzzle-solving. It's also a fiction which tells a story, even though the player can affect the story, as there are different screenplay branches.



## Scenario

Winter 2010. Just another random New York student heading back to Babi where he is originally from. Great time ahead for Lewis ! Partying all night in Rue Princesse (Princess Street), enjoying the garba on the sunny beaches, hanging out with old buddies and new encounters... Until a sudden change of atmosphere. What could have possibly gone wrong? Nobody knows. It all went really fast... Within a few minutes everything went from harmony to drama: screaming, bombing, looting, violent clashes, food shortage, closure of borders. When Babi Joy becomes Babi Frayeur, what was meant to be a dream holiday increasing looks like a commando mission. What to do now? Fight or Fraya? Fight to Fraya? [You choose](#).





In short, I just have 3 things to say about it:

- This is a [point-and-click adventure game](#), built on [Microsoft Silverlight](#).
- The story offers an off-the-wall perspective of the [#CIV2010 Babi Battle](#), turning dream holidays into a commando mission.
- The game is quite easy and a few parts could potentially be improved but it works fairly well...

[Babifraya.com](http://Babifraya.com). Any feedback is welcome,

N.



## Relaunch : Kpakpatoya 2.0 as an Interactive Platform

By [Necemon](#)

# #Kpakpatoya

### 1. Introduction

[Kpakpatoya™](#) is a repository for what's new, popular and relevant on the Internet with a focus on Africa. Users like you can submit content and decide, through voting, what's good (kiffs) and what's junk (zaps). Links that receive community approval raise towards the top, so the front page is constantly in motion and (hopefully) filled with fresh, interesting links. Kpakpatoya™ is part of the [Evasium Network](#).

"Kpakpatoya" is a term from the Ivorian slang which indicates the act of spreading rumors, passing on supposed stories relating to surrounding things, events and people. In our context, it would therefore be appropriate to understand "kpakpatoya" with regard to scoops, gossips and even compelling news.



## 2. Historical Context

April 2011. The Ivory Coast was then emerging from a major crisis, a crisis in which information and communication technologies have played an important role in providing practical assistance to related casualties. Beyond simple commenting, hashtags such as #CIV2010 and #CIVSocial were used to spread critical news and provide concrete solutions to these victims. Through donations, call centers and notably Twitter, which then comes up as the ideal tool for broadcasting urgent information in real time.

The Ivorian community on Twitter was growing and booming, with more and more registrations, tweets and contributions. But that community was still relatively small in terms of numbers. For such a small group on Twitter, how to get together, and exchange around common themes? Through the #Hashtags again, of course... On online social networks, the Hashtag is a word or set of words preceded by the symbol '#'. The hashtag is used to centralize messages around a specific term. It acts as a keyword so that users can follow, comment on a conversation or get together around a topic, for example the measures of refueling in times of crisis or even medical emergencies.

But in this post-war period, it was time to switch to less sinister tags. [#Kpakpatoya](#) was quite timely for this small community seeking to gather in an atmosphere that could be described as "cool". It was the tag that these Ivorian "influencers" and "pioneers" on Twitter used to do without local news, inside jokes, viral content, or other kinds of interesting information, including sports events, (technology) conferences, or day-to-day life.

Detecting a chance to improve some socio-technological conditions, I launched the [kpakpatoya.com](#) website in that same period. At the beginning, it was just [a flow of tweets around the #Kpakpatoya feed + a few extra features](#), including a summary of local and international news published in real time. Then I tried with varying degrees of success to offer some applications such as some discussion forums, the blog and the ability to tweet directly from the web site. I had also opened the Twitter account [@kpakpatoya](#) and arranged the mobile application implementation, which still present the news, as well as the Facebook page and the website.

Through this form of ecosystem, and a combination of technical and social skills, the hashtag #Kpakpatoya grew in popularity. By following the thread #kpakpatoya, it was easy to find CIV culture sympathizers, their news, as well as the most popular local themes, even for a newcomer.



3 years later, the Ivorian Web scene turned into a completely different form...

On the one hand, this virtual community has expanded a lot, which is most welcomed, since there is a lot more content, web projects, web entrepreneurs, bloggers, users, etc. Except that a simple hashtag is not enough anymore for an effective exploration of the content. It would not be practical to communicate around a single hashtag. Of course, there is always the option to "break down", to use the alternative hashtags that have emerged in the meantime (although #Kpakpatoya remains one of the most popular tags). That's a good solution, provided one can constantly be aware of new users and tags to follow, so as not to get lost in all this volume of content, and not to miss the most relevant and recent information..

On the other hand, there has been very little evolution when it comes to the #Kpakpatoya project, whether it be in terms of technology, computer graphics, communication, infrastructure, etc. The Facebook page has less than 2000 fans, the Twitter page has around 8000 followers, the mobile application is obsolete, and until recently, the components of the website that had a lot of success in the first few months have been almost neglected (I will spare you the numbers related to the decrease in web traffic). There are almost no contributors to the progress of the project, there are just the users. This could have been a decent community project, but instead it was reduced to a latent state.

It is true that the community was formed in a rather organic way, there was no concrete strategy, everything happened gradually and naturally, which perhaps explains the cool-down after the buzz effect had passed, each party being concerned with their own projects. I must admit that the applications I wrote as part of this project were essentially in maintenance mode and that I had moved on to other things myself. Anyway, no one stepped up.

A wise man once said: If there is anything potentially interesting and you do not understand why nobody does it, it's because you haven't done it either. That's DIY: Do It Yourself.

I therefore take over the lead on this project to try and produce something useful. In recent weeks, my team and myself have been doing some renovations, trying to produce a more interactive platform (with some added value, if at all possible) in order to facilitate meetings, exchanges and content management within the community.



### 3. Description

Kpakpatoya™ is powered by [Evasium®](#), a small team of technology enthusiasts. We basically make Internet software. We focus on building edutainment-oriented web/client applications.

On Kpakpatoya™, you can discuss pretty much anything, but the most welcomed topics are : Buzz, Clash, Economics & Politics, Culture, Insolite, Local News, Africa, Sport, Diaspora.

Beyond the option to read news, it is now a matter of adding a touch of interactivity, where everyone can share links, discuss, comment and vote on the existing notes. A Kiff is a positive vote (like a "like", a "+1") for a note. A Zap is the opposite, it is a negative vote for a note. Any Kpakpatoya user may kiff or zap a note, and since the score of the note depends on the difference between his kiffs and his zaps, each vote helps to establish the position of the note in the Kpakpatoya ranking. The notes could be listed in order of popularity, to allow the next visitors to spot the best content.

Nothing too complex, it's just a sharing system as found everywhere on the Web. For example, tech entrepreneurs have [Hacker News](#), Americans came up with [Reddit](#), French cybernauts have [tapemoui.com](#). The concept has existed for a while, it's just that there was no Ivorian/African version.

Kpakpatoya™ is available in 2 languages.

In English:

<https://kpakpatoya.com/?lang=en>  
<https://www.facebook.com/KpakpatoyaNews>  
<https://twitter.com/kpakpatoyanews>

In French:

<https://kpakpatoya.com/?lang=fr>  
<https://www.facebook.com/Kpakpatoya>  
<https://twitter.com/kpakpatoya>

Kpakpatoya™ may (or may not) become a reference point in terms of news and "affaires" for Ivorian web users. But that's not the main goal here. The current goal is to provide a useful tool for expanding the sharing options for local content, and to implement the long overdue Kpakpatoya renovation.

On a more personal note, I think we should rather see this project as an experience that could potentially help to optimize the visibility of online African content.

For having managed the website, the Facebook account and the Twitter account of Kpakpatoya™ over the last 3 years, it seems to me that many followers expect a better interaction system and they strive for a better visibility for their content, namely their projects, links published, or simple opinions and comments. Therefore, it is possible that this version of the site meets their expectations to some extent.

There are still many things we could add or improve, we can do it as we go along, while taking users feedback into consideration, but the prototype of this new version is relatively stable, and what is needed now is trying it. That is to say, create your account, test the different forms of interaction, provide your feedback and share the link with your contacts so that they can do the same.





## Flux Project : Social News Web Applications

By [Necemon](#)



This is just a quick update on my [ongoing side projects](#).

I have been working on a series of [social news web applications](#) where users can submit content and decide, through commenting and voting, what's good (kiffs) and what's junk (zaps). Links that receive community approval raise towards the top, so the first pages are constantly in motion and (hopefully) filled with fresh, interesting links.

Nothing impressive, but it gives me a good basis for making a robust web platform, as in, the common features:

- Security (cryptography, membership, login, registration, password reminder, etc.)
- Database layer / caching (or how to manage an ever growing volume of "big" data)
- Back end services (email senders, pings, http requests, logging)
- A stable [ASP.NET MVC](#) application
- User actions/management : voting, commenting, publishing, moderation
- Assets management (images, javascript, css, minification/bundling)
- Bonus: I am learning a lot when it comes to deployment (IIS settings, virtual directories, ports, application pools, etc.)

In short, the kind of things that could be useful in a modern [web application](#). Actually, this is my most advanced web application so far.

On [Extreme Programming](#) mode

I am still interested in building [virtual worlds](#) and [online games](#), but I thought I would release something (anything) at regular intervals. That way, I can improve as I go, learning/shipping at every step, instead of working on one thing for many years and come up with something that's not really practical/useful. So, that's the idea : getting stuff out there early, as soon as I've got something that covers the basics, and then build it up. If my experience with web projects has taught me anything, it's that people who use apps will suggest or ask for things that the developers wouldn't have thought of doing for themselves, and some of them turn out to be really good ideas even if we don't always quite see the potential at the time. Also it helps getting a better idea of priorities.

Anyway, that basic version might already be quite useful in some contexts, as news portal, one of the flavours being [kpakpatoya.com](#). you can think of it as a small bilingual Reddit for African news. It's starting to have a decent audience (with [8000+ followers on Twitter](#) and [2000+ fans on Facebook](#)) and some of the followers may welcome an opportunity to browse more content and to interact around it (share, comment, vote and add their own content).

So that's what I have been up to recently, each site having its own topics and they are all available online (in English and French) :

[flux.evasium.com](#) (Education, Science and Virtual Worlds)

[notiflux.com](#) (an exploration of online content on Personal Development and Winning at life)



[kpakpatoya.com](http://kpakpatoya.com) (a repository for what's new, popular and relevant on the Internet with a focus on Africa)

[degamage.com](http://degamage.com) (essentially geeky entertainment)

[appuyage.com](http://appuyage.com) (an exploration of psychology, sexology and interpersonal relationships at the emotional level)

So [I did a game](#) the year before, shipping these web portals this year and the coming months sounds promising in terms of technical improvements and new features. Let's see.

## Stay up to date with your favorite topic on social platforms

By [Necemon](#)



Having written [Notiflux™](#) and [AUDE](#) (Automatic Updates Distribution Engine) some time last year, I thought I would play around with the [Facebook](#) and [Twitter](#) APIs and keep posting links to some fresh news and articles in real time.

So depending on the kind of things you like, you might want to join or follow some of those pages:

[Degamage™ UK](#) : PC & Web gaming news.

->[Facebook](#) ->[Twitter](#)

[Evasium Park](#) : A [2D Virtual World](#). We are currently working on the version 0.2. In the meanwhile, here are regular news about virtual worlds.

->[Facebook](#) ->[Twitter](#)

[Notiflux™](#) : Your flux of notes and notifications. All about winning : strategy, tactics, philosophy, marketing, business, etc.

->[Facebook](#) ->[Twitter](#)

[Evasium®](#) : Posting news about fun learning, education through technology.

->[Facebook](#) ->[Twitter](#)

[Evasium Developers](#) : The nerdy one. Latest and best technology articles about programming in general and Microsoft C# .NET in particular.

->[Facebook](#) ->[Twitter](#)

Take your pick!

N.

## Furtivue (Or How To Send Furtive Messages Like A Ninja)

By [Necemon](#), 2012



Furtivue : a furtive view over your messages. ([www.furtivue.com](http://www.furtivue.com))

Furtivue is a web service that allows you to send temporary (self-destructing) emails, as in, you can control how long the message is going to stay with the recipient.

When they open your Furtivue message, it appears for a specified number of seconds, then self-destructs. The duration will depends upon the length of the message (Or you can choose how long you want the message to be displayed).

### **How it works:**

User A writes a message to user B, a link is sent to B. When B clicks on the link, the link is de-activated (therefore can't be used thereafter), the message is shown from a Silverlight canvas and disappear after the predefined time is over.

### **Purpose:**

This could be used for sending confidential/secret information (such as passwords), or any other data you don't want the receiver to keep permanently in their e-mail box : for some reason, you may want to express yourself without leaving a permanent record.

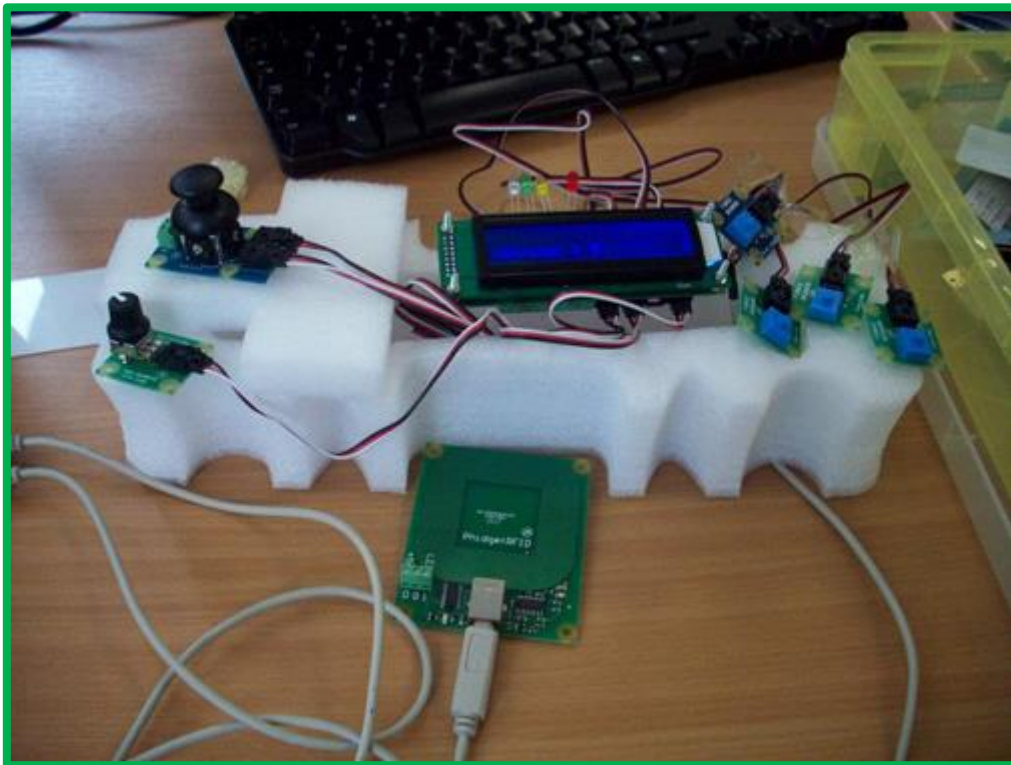
### **Now what?**

You can [try it here](#) or [download/explore the source code from here](#).

# Chapter 9

## Research and Case Studies

*L'informatica è un'esperienza che ti arricchisce molto  
Almeno stando a quello che dice il mio estratto conto !*



## Phidgets: First Steps In Robotics?

By [Necemon](#)



Phidgets are a set of "plug and play" building blocks, some low-cost electronic components that you can control from your personal computer via USB. They provide you with some [extra input/output methods beyond the classic](#) mouse + keyboard + screen.

As [Harold Thimbleby](#) mentioned in his book [Press On](#), phidgets are a very nice way to get into hardware programming, as you may want to build real systems, not on screen or web browser simulations : phidgets are so-called because they are the physical equivalent of on-screen widgets (Windows Gadgets).

Phidgets = Physical + Widgets  
(Widgets = Windows + Gadgets)

### Where to find them ?

there are various official phidgets dealers [spread across the world](#). For example, if you live in the UK, you can check out [robotshop.com/uk](#).

### **Getting started**

All the USB complexity is managed behind the Phidget API (Application Programming Interface). Applications can be developed quickly by programmers using their favorite language: C/C++, C#, Cocoa, Delphi, Flash AS3, Flex AS3, Java, LabVIEW, MATLAB, Max/MSP, MRS, Python, REALBasic, Visual Basic.NET, Visual Basic 6.0, Visual Basic for Applications, Visual Basic Script, Visual C/C++/Borland.NET, etc. (If you can't code you can also use software such as Microsoft Robotics Studio or even Microsoft Excel).

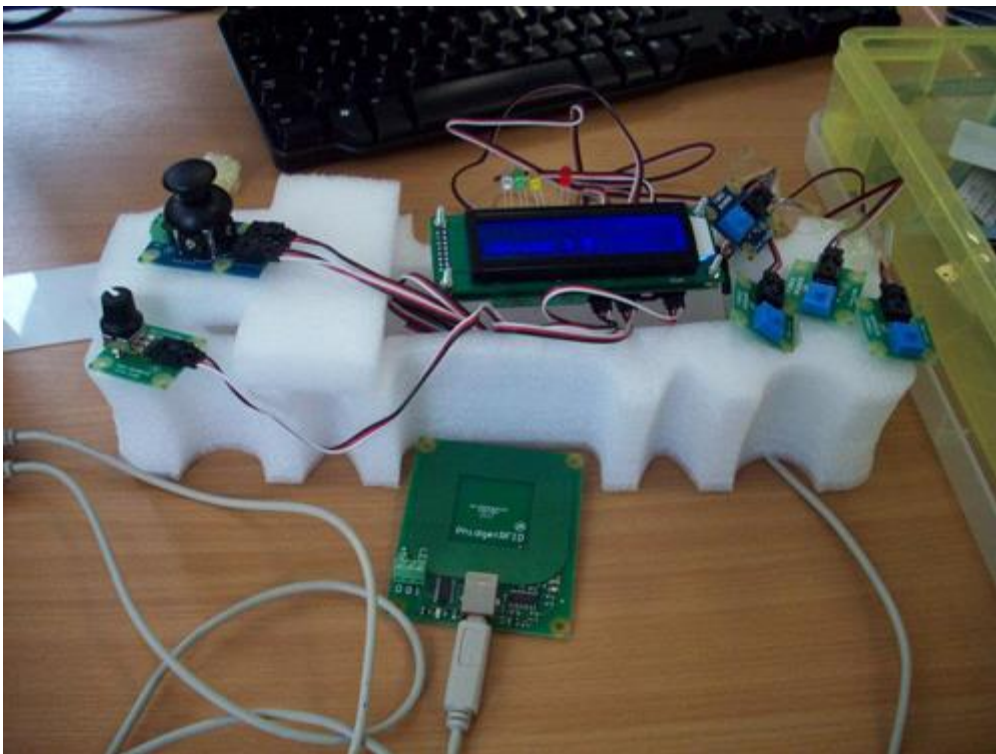
All you need is the drivers that you can get by downloading [the relevant installer](#). Similarly, you can [access the various manuals, the samples and the API sorted by language from here](#)

For example, as a C# developer, just download those and you are ready to go:

1. [Getting Started Guide](#)
2. [Windows Installer](#) (32-bit)
3. [Code Samples](#)

### **A practical example**

Allow me to introduce the Quick-&-Dirty-Pad (while I think of a better name), a "game controller" I built with my partner [Daniel Williams](#), while we were back at [University](#) during the [Interaction Technologies](#) classes led by [Dr Parisa Eslambolchilar](#).



Yes I know, it's not that great, but hey, everyone needs to start somewhere...

We used it primarily to go along with a flight simulation game we programmed, we used force sensors for shooting, a joystick for direction, RFID tags for user authentication and a rotation sensor to regulate the game speed. The game events were reflected through a LCD screen and a few LED indicators.





(Thanks for reading so far. The rest is for the nerds only. This is the part where we briefly discuss the key technical details.)

The game itself was written in C#. It was inspired from a [Riemers XNA tutorial](#). The key parts of the code include the navigation system:

```
if (keys.IsKeyDown(Keys.Right))
    leftRightRot += turningSpeed;

if (keys.IsKeyDown(Keys.Left))
    leftRightRot -= turningSpeed;

float upDownRot = 0;

if (keys.IsKeyDown(Keys.Down))
    upDownRot += turningSpeed;

if (keys.IsKeyDown(Keys.Up))
```

```
upDownRot -= turningSpeed;
```

The aircraft keeps moving forward automatically and the above code is used to control the rotational movements. One of our challenges was to incorporate the phidget code for the joystick into this part of the code such that the joystick behaves as arrow keys.

First of all we created a new input wrapper which is the class that contains the sensor change handler to detect any changes in the movement of the joystick. The input wrapper class implements the interface kit object which detects any input from any phidget

```
InputWrapper iw = new InputWrapper();

ifKit.SensorChange += new SensorChangeEventHandler(ifKit_SensorChange);
```

The first section of the next code controls the x axis of the joystick. The output values of the joystick are between 0 and 999, so for any value greater than 500 the plane would rotate to the right and any value less than 500 the plane would rotate to the left. The second part of the code is used to control the Y axis of the joystick. This time any value greater than 500 would rotate the plane so it's facing upwards while any value less than 500 will rotate the plane so it's facing downwards.

```
if(iw.XJoystick>550)

    leftRightRot = (iw.XJoystick-500)/10000;

if (iw.XJoystick < 450)

    leftRightRot = (iw.XJoystick-500)/ 10000;

if (iw.YJoystick > 550)

    upDownRot = (iw.YJoystick - 500)/10000;

if (iw.YJoystick < 450)

    upDownRot = (iw.YJoystick - 500) / 10000;
```

The LCD screen was used to display the current player and their score. The user must scan their id to start the game or the game wouldn't start. The RFID phidget was used to detect the identity of the current playing user. First of all we created a text LCD wrapper and an RFID wrapper which contain the various event handlers. The RFID detects if a tag is present; if not a message appears on the LCD screen asking the user for their tag

```
TextLCDWrapper tlw = new TextLCDWrapper();

RFIDWrapper rw = new RFIDWrapper();

while (rw.tag == null || rw.tag.Length < 2)

{

    tlw.JustTyped("Tag, Please!");

}
```

When the user scans their tag the RFID will detect if the user is registered to play the game or not and display the relevant message on the LCD screen

```
private void DrawText()
{
    if(rw.tag != null && rw.tag.EndsWith("cb"))
    {
        tlw.JustTyped("Necemon : " + score);

        spriteBatch.DrawString(font, "Necemon - score:" +
score, newVector2(20, 45), Color.White);
    }

    else if (rw.tag != null && rw.tag.EndsWith("8a"))
    {
        tlw.JustTyped("Daniel : " + score);

        spriteBatch.DrawString(font, "Daniel - score:" +
score, newVector2(20, 45), Color.White);
    }

    else if (rw.tag != null)
    {
        spriteBatch.DrawString(font, rw.tag + " - score:" +
score, newVector2(20, 45), Color.White);

        tlw.JustTyped("Unknown : " + score);
    }
}
```

The rotation sensor will be used to control the speed of the plane. The values of the rotation sensor are obtained through the input wrapper class, previously mentioned when describing the joystick. The more you turn the rotation phidget the more the speed increases

```
gameSpeed = iw.RotationValue/100;
```

The following code creates a new bullet every time the space bar is pressed provided the previous bullet was fired more than 100 milliseconds ago

```
if (keys.IsKeyDown(Keys.Space))
{
```

```

double currentTime = gameTime.TotalGameTime.TotalMilliseconds;

if (currentTime - lastBulletTime > 100)
{
    Bullet newBullet = new Bullet();

    newBullet.position = xwingPosition;

    newBullet.rotation = xwingRotation;

    bulletList.Add(newBullet);

    lastBulletTime = currentTime;
}
}

```

Our task was to implement four sensors to control the bullets being fired from the plane. The values of the force sensor are also obtained through the input wrapper class. The following code now creates a new bullet of type 3 every time the force sensors are pressed

```

if (keys.IsKeyDown(Keys.C) || iw.Force3 > 25)
{
    //make the lights blink

    iw.ifKit.outputs[0] = (!iw.ifKit.outputs[0]);

    iw.ifKit.outputs[1] = (!iw.ifKit.outputs[1]);

    iw.ifKit.outputs[2] = (!iw.ifKit.outputs[2]);

    iw.ifKit.outputs[3] = (!iw.ifKit.outputs[3]);

    double currentTime = gameTime.TotalGameTime.TotalMilliseconds;

    //within a certain frequency

    if (currentTime - lastBulletTime > 300)
    {
        //Create the bullet and add it to the bulletlist

        Bullet newBullet = new Bullet();

        newBullet.position = xwingPosition;

        newBullet.rotation = xwingRotation;
    }
}

```

```
bulletList3.Add(newBullet);  
  
//play a sound  
soundEffect1.Play();  
  
lastBulletTime = currentTime;  
  
}  
  
}
```





## Paper Prototyping

By [Necemon](#)



Last week, I had to make a [paper prototype](#) for a mobile application I was working on. I ended up building a quite nice thing (picture above) but at the beginning I was not sure about how to go about it. So I just thought I would write about it to share how I proceeded.

The first thing was to do a little bit of research, just to see how people normally do it. The website [paperprototyping.com](http://paperprototyping.com) had been quite inspiring for that matter (as well as [that video](#)).

I then headed to the office stationery to grab the necessary stuff. For the whole process I just used a pencil, a pen, a gluestick, some record cards and scissors. That's it.

The next steps were to cut some of the cards to the scale of a (big) mobile phone, wrap one of them around the others and make a hole on each side so that "screens" could go through:



All this may sound kind of obvious but I guess that's the article I wish I could have read when I was starting.

Thanks for reading.



## Dual Shock 3 - Part 1: History of game controllers

By [Necemon](#), 2011



This study is about a device which is probably the best joypad of all times: the Dual Shock. It has many variants and we can expect it to keep evolving in the coming years, however, we will focus here on the current version, the Dual Shock 3.


In this article, we will go through the history of the game controller in general and then focus on the history of the Dual Shock itself, so that we understand how it was built. In a next parts, we will carry on with [the context in which the device was built](#) and then [analyse the controller by studying its specifications](#).

### **Basics**




There are many categories of game controllers, among which we will retain the most popular one, the gamepad, as that's exactly what the Dual Shock is. Gamepads, also known as joypads, can have many action buttons combined with one or more omnidirectional control sticks or buttons. They are held using both hands with fingers (typically thumbs) used to provide input; as mentioned on Wikipedia, most modern game controllers are a variation of a standard gamepad. Common additions include shoulder buttons placed along the edges of the pad, joysticks, centrally placed buttons and internal motors that provide haptic feedback (as in, they typically vibrate).

## Some history

In his article intituled "History of the game controller", Catalin Ivan stated "Ever since the very beginning of video gaming, the controller has been the best (and usually the only) way of man-machine interaction." That's actually an interesting fact, and it would be good to follow the evolution from that "very beginning" to see how we reached the current gamepad technologies. There had actually been many, many gamepads released throughout the history, we are not going to study all of them, we are just going to review those who appear like important milestones that we observed from a study by John Honnibal and that we are summing up in the following table:

Controller Name	Description	Picture
The TV Tennis Game Paddle	<ul style="list-style-type: none"> <li>- An analogue control based on a monostable</li> <li>- Game-start buttons were located on the console, not on the paddles</li> <li>- One variant was the use a slider control</li> <li>- Some TV games didn't even put the paddle on wires but place them on the front of the game itself</li> <li>- Generally, with the early models, there were no buttons on the pads</li> </ul>	

<p>The Atari 2600 Joystick</p>	<ul style="list-style-type: none"> <li>- A digital four-way joystick with a single fire button</li> <li>- Again, game-start button on the console itself</li> </ul>	
<p>The Atari 7800 Joystick</p>	<ul style="list-style-type: none"> <li>- A digital four-way joystick with two fire buttons</li> <li>- Simply add an extra-button to the 2600 Joystick</li> </ul>	
<p>The 8-bit Home Computer Joystick</p>	<ul style="list-style-type: none"> <li>- An analogue joystick with a single fire button</li> <li>- Here, the game could be started using the keyboard</li> </ul>	
<p>The Vectrex Joystick</p>	<ul style="list-style-type: none"> <li>- An analogue joystick with FOUR buttons</li> <li>- Games were started from the controller using the normal fire buttons</li> <li>- The fire buttons are labelled '1' to '4'</li> </ul>	
<p>The Nintendo Entertainment System Joypad</p>	<ul style="list-style-type: none"> <li>- A digital joypad with two game-start buttons and two fire buttons</li> <li>- introduced the idea of placing active components and circuitry inside the joypad.</li> <li>- The shape of the game pad, however, is a very basic rectangular box.</li> </ul>	

	<ul style="list-style-type: none"> <li>- The two fire buttons are both red in colour and are labelled 'A' and 'B'.</li> </ul>	
The Super Nintendo Entertainment System Joypad	<ul style="list-style-type: none"> <li>- A digital joypad with two game-start buttons and six fire buttons.</li> <li>- introduced the idea of shoulder buttons</li> <li>- shaped much more naturally to fit the player's hands.</li> </ul>	
The Sony PlayStation I Joypad	<ul style="list-style-type: none"> <li>- A digital joypad with two game-start buttons and eight fire buttons.</li> <li>- adds more shoulder buttons to the SNES pad</li> <li>- an even more ergonomic shape</li> </ul>	
The Sony PlayStation II Joypad	<ul style="list-style-type: none"> <li>- An update to the PlayStation I controller that adds analog controls and a pair of vibration motors.</li> <li>- the analog-to-digital converter (ADC) is in the controller and the interface is digital.</li> <li>- The two motors are fitted with eccentric weights of different sizes, and can be independently controlled by the game.</li> </ul>	

From this brief summary, we can see the different steps in the history of the game controller before the Dual Shock took over. A few deductions we can make are firstly that we slowly went from controllers with zero buttons to controllers with one, two buttons and so on. Those buttons became more and more meaningful with colours and labels. We also observe that the commands slowly moved from the game console to the controller. Further, we note that controllers got more and more ergonomic with time.

Now, let's revisit [the amazing story of the DualShock](#).

## Dual Shock 3 - Part 2 : The amazing story of the Dual Shock

By [Necemon](#), 2011



We just reviewed the [evolution of the game controller](#) in general. Now let's dip in the very interesting story of the creation the PlayStation Controller. The facts reported here mainly come from [Kevin Gifford who wrote the "PlayStation 1 Design" article](#) and from [Reiji Asakura who wrote the book "Revolutionaries at Sony – The Making of the Sony PlayStation and the Visionaries Who Conquered the World of Video Games."](#)

To get into the right context, allow me to introduce a few Japanese public figures who are keys in this story. First, Ken Kutaragi, who was chairman of Sony Computer Entertainment. Then we also have the late Norio Ohga (1930-2011) who was president of the Sony Corporation by the time the story happened. And there was, Teiyu Gotoh, the main designer of the PlayStation Project, which is the hero of this story. It's also worth mentioning that the Sony PlayStation was a huge success as it was released. It was actually the largest selling single model the electronic equipment market had ever seen (with 40 million units worldwide without a model change within three years). "One of the main factors that led to the success of Playstation", said Ohga, was the design of the main unit and controller. That's what Gotoh confirmed by stating "I'm convinced that the design contributed to the runaway success of the PlayStation, We had to start the Playstation business with no track record. A major issue was whether software companies would publish titles for it. I thought the first hurdle was to ensure that game creators would like the hardware design. After all, game creators are the machine's first users. We've got it right if they think they want a platform that looks like ours."

For Gotoh, design was the very expression of content. He was aiming for a simple yet original and trendless design. Gotoh was originally a designer of television sets. He was well known for his sophisticated design style and excellent performance on projects like the Sony Profile PRO and the Sony miracle, the VIAO PC. However, the PlayStation project was a completely different matter, as Gotoh himself said: "TVs have a fixed shape and use the same devices. Within those constraints, I struggled to find ways to create innovative designs that are different from other products, how to emphasize the difference. But the game machine was completely different from conventional products, because the PlayStation is a one-and-only product. Although it competes against other platforms in the game-machine market, there are no other PlayStations. Other Sony products categories have their own history and fixed design process, but there were no examples to follow with the PlayStation, only a concept of doing something different from other products in the same genre."

In our study, we are more concerned with the PlayStation controller than the console itself. But the interesting part is that Gotoh designed not only the main parts of the PlayStation but also its peripherals such as the memory card and eventually the controller. However, at the beginning, the controller design didn't know a unanimous approval at Sony. While Ohga approved it, Kutaragi, the technologist, immediately rejected it: "What's this? The shape is original, but it doesn't look very easy to use." It was indeed very different from the conventional controllers which were flat, planar while Gotoh's controller was three-dimensional. He recalls "I have worked at Sony for twenty-one years now, but there has never before been such a difficult product. The controller was considerably harder to design than the console." It took him more than a year to design it. He proceeded by carving a lump of acrylic foam and gripping it repeatedly to see how it felt. The shape of the controller was refined through countless repetitions of this process. A few issues with the flat controllers were that they had to be held tightly during use as the palm was not in contact with the device. This tended to be uncomfortable and stressful. Further, those controllers could not properly fit different types and sizes of hands. That's how Gotoh came with some grips so that the controller could be held as naturally as possible.

At some point, Gotoh conducted an experiment, by gathering some children and make them test his controller. They found it a bit weird at the beginning as they were used to the flat controllers. However, once they learned how to use it, they seemed to really enjoy it. This was encouraging because children are honest in their feedback. If they like something, or if they don't they just say how they actually feel about it. Further, there was another very interesting point in that experiment. [As Harold Thimbleby said](#), children are better testers for devices than adults, as they tend to behave like trolls. They keep trying new things without fearing breaking the device and the fact that they are good at getting rid of any technology preconceptions made them avoid any design assumption. This made them the perfect testers for a revolutionary device like the PlayStation controller, spearheading a new generation of game controllers.

## Dual Shock 3 - Part 3: Device Analysis and Specification

By [Necemon](#), 2011

Now that we understood [the context and the motivations](#) that led to the creation of the PlayStation Dual Shock from the first part, we will get to know more about the device itself.



The dimensions of the controller are roughly 6 x 3.5 x 2 inches.

The Dual Shock has on the left side, the same directional pad found on most controllers. However, on the right, the action buttons are labelled with original symbol; this is also an innovation by Gotoh while conventional controllers normally used numbers and letters to label the buttons, as he states himself: "Other game companies at the time assigned alphabet letters or colors to the buttons. We wanted something simple to remember, which is why we went with icons or symbols, and I came up with the triangle-circle-X-square combination immediately afterward." In an interview reported by Games Radar, Gotoh explained how it came up with the symbols "I gave each symbol a meaning and a color. The triangle refers to viewpoint; I had it represent one's head or direction and made it green. Square refers to a piece of paper; I had it represent menus or documents and made it pink. The circle and X represent 'yes' or 'no' decision-making and I made them red and blue respectively."



Further there are also four shoulder buttons on the top, for the middle fingers and two joysticks that improve the experience in some game categories like racing or flight simulation.



The vibration is an interesting effect as the device rumbles when some dramatic events happen during the game. That provides a great game experience.

However, the joysticks and the haptic effects didn't appear in the original version of the PlayStation controller. As a clarification, it's worth mentioning that the Dual Shock is actually an advanced version of that original version.

Now coming to the latest version, which is the one we are mostly concerned with here, the Dual Shock 3, which itself is an advanced version of the Sixaxis, the original game controller for the Sony PlayStation 3. Both Sixaxis and Dual Shock 3 looks the same but they are different in that the Dual Shock 3 is heavier, more opaque and it rumbles.

Even though the Dual Shock 3 can be plugged to the console with a removable USB cable, it's also a wireless device which can work over a Bluetooth connection. An autonomy of 30 hours is ensured by an internal battery. The wireless ranged is believed to top out over 20 meters. (PS3 Fanboys 365)

Ergonomically, all the Dual Shock are pretty much the same. And honestly, they are the most comfortable game controllers I personally had the chance to play with (I had been trying a wide range of controllers over the years). However, one downside of the Dual Shock 3 could be that it is quite lighter than its predecessors. But here again, the player get used to it after some time.

### Putting the DualShock 3 in perspective with competitors



Technically, the Xbox 360 provides an interesting option when it comes to batteries. They use AA batteries which can therefore be easily changed. On the opposite, the Dual Shock internal battery can't be changed (Not by the gamer, at least). But when it comes to cables, the Dual Shock 3 uses standard USB cables while the 360 controller requires a specific cable that is unique to that controller. All those factors affect the overall cost of the device from the gamer point of view. However, the Dual Shock 3 is more expensive on the market than the 360 controller (£39 against £29 by the time I purchased mine).

Now when it comes to playability, each controller has its own strengths and weaknesses. Mike Ferro once compared the playability of controllers based on the fighting, racing and shooting categories. If we consider racing for example, The key difference here is in the shoulder buttons that the racers normally use to decelerate. They are much more comfortable on the 360 controller as they show less resistance.

For shooting games, the critical controls are the joysticks. The Dual Shock 3 analogue stick has a slightly smaller dead zone, which offers a greater precision and a better response time to the player. Coming to fighting games, the use of the directional pad is critical to make the character execute precise moves. The Dual Shock 3 is clearly superior in that matter because of the way the touches are partitioned. On the contrary, moves are hard to execute on the 360 controller because the user literally feels they are trying to move a single piece of plastic, as opposed to independent up, down, left and right angles.



## Business at the speed of game production: OpenGL or XNA?

By [Necemon](#), 2012



A common start ups question in the gaming industry is: Which graphical environment should we use? As time is money, a crucial factor in a choice is time: not only the time it takes to build complete games with each API but also a fluid timing during the gaming experience to make it enjoyable. In this analysis, we compare the OpenGL industry standard to the Microsoft XNA game framework on the speed factor. The study includes learning curve, programming, modeling and rendering speeds on both sides.

[Download the report here](#)

## GOOSE (GOOgle Supply search Engine)

By [Necemon](#)



Google GOOSE is a product I came up with during a marketing exercise. We were asked to think of a new product from a well-established brand that could fulfil customers satisfaction. Please note that the GOOSE product is purely fictional. I don't work at Google and the following product description is not endorsed by Google or any of its employees.

### **The problem**

Shopping is a time consuming process; the rhythm of our lives, jobs and other activities goes so fast that we hardly find time to shop properly. The current technologies provide us solutions to speed up the process by shopping online. But even online, shopping remains a task that involves some effort, thinking and time for every purchase. Sometimes you just don't feel like doing it; or you got a lot of other things to do. Further, we are sometimes so busy that we may just forget to do it.

Another thing is that it may be such a pain. Sure, it's fun to shop for new clothes and shoes. But it's not fun to go get vegetables and meat every week.

### **The solution**

What if you could just decide once what should be there in your fridge at all times and not worry about it again? You could just have what you want at all time without having to regularly spend the effort, time and thinking that food shopping require. And that's what the new Google Supply Search Engine (GOOSE) is all about.

The GOOSE (GOOgle Supply search Engine) is a software platform for a new technology of home fridges which systematically:

- knows what food items you need
- knows when it's missing from your fridge

- check the best offers for those items on the internet
- order the missing food online
- trigger the delivery of those items back to your fridge (actually to your doorstep)

### **How it works**

- On the server side:

The GOOSE consists of a web service where each customer can register and provides a list of the items he needs at all times. The customer can also set special items that will be only delivered under specific circumstances (for example, a birthday cake to be delivered only once at a particular date). All that information is stored in the GOOSE database.

The customer can also set on which frequency he wants his fridge to be filled up, say every Friday afternoon. So, when the time comes, the GOOSE web service orders the missing items that had been communicated from the shop the customer indicates. But an interesting thing is that the customer can give the freedom to the service to choose from which shop to buy. It obviously can't be too far from the customer home but the key is that the GOOSE service can compare the offers from different shops and choose the best offer. The goods are then delivered to the customer's house.

- On the client side:

The GOOSE software works on a new model of home fridge system that contains some special components:

- A touch screen device for the user to set up his options. The screen also let the user see and choose the promotional offers that Google sends him, (by the way, this brings additional revenues since advertisers can pay to show some ads to the users).
- Additionally, all the user inputs can also be done from a regular computer system which is connected to the GooSE account of the said customer. All the changes made will be updated to the GooSE database and communicated back to the fridge and vice-versa.
- The fridge is equipped with some devices that track the fridge content and the checks in and checks out of any food items. Those devices are mainly camera, infrared devices and sensors. Bar codes could also be used to identify complex items.

### More details

The software would have to interact with the internet for online purchasing and content recognition to some extent. This means the fridge should have access to a Wi-Fi connection. The insides of the fridge would have to be made in a specific manner for content recognition. For example the egg tray, the compartments for bread, vegetables and fruits need to fit a small sensor to recognize if the item is present or not. For example, a beam of light could be cut when a particular product is present uninterrupted when the product is absent. However this wouldn't completely solve the problem as the software would have to recognize what kind of item is present. For example if we take the case of fruits and vegetables, the software would have to know which fruit or vegetable it is, this could be done by having a camera in your fridge which is also capable of counting the number of fruits or vegetables. A barcode reader could be used for those items that come with a bar code. Barcode readers and cameras can be integrated in their usage. Once the camera recognizes the barcode it is then sent online and checked with online databases. The same technology applies for the barcode reader application on Smartphone's.

Now that we have come to an understanding on how content is recognized, what do we do with this information? Based on this information a shopping list is generated and stored temporarily until next purchase is made. The shopping list can also be categorized based on various sections such as:

- o Most wanted items such as soft drinks or any item the customer would be keen on having at all times.
- o Least important items which are items that are customer or not very particular they should have at all times.
- o Short lived items such as milk and other such products that have a short life span. The software would display the date on which the last purchase was made and would give you the option to add the number of days for which that product would expire.

The software would also be capable of reminding you prior to the day of expiry that the item needs to be consumed.

When the software comes to use for the first time a general shopping list is created regardless of categories. Google having kept every bit of information of past purchases then gradually learns what type of customer you are and offers a shopping list based on auto categorization which you either accept or deny.

### Potential customer base:

Goose is a software product that searches and orders food items, based on the purchase history of the items which were previously stored in the fridge. It automatically searches and orders the product on a scheduled date, so people who are working could use it at their convenience. Further GOOSE mainly targets

- The young people, who are the most familiar with the internet technology and online shopping. It would perfectly fit students who move away from their parents and are not used to food shopping.
- The parents, who are responsible for their whole family food supply and sometimes have a hard time dealing with it (especially the single parents).
- The developed countries: places with high internet penetration and widely spread home delivery.
- Additionally, GOOSE could equally assist both genders: men are typically the most uncomfortable with food shopping so it could be a great help to them. Conversely, women who often have all the burden of this task would be relieved in a high extent.

Our product Goose doesn't order the product in the single store, it compares the product in the many stores available in the market and automatically order the cheapest or best product. However, some stores can be promoted by paying some advertisement fee. That would be the second form of profit for GOOSE after the fridge software licensing fees.

## Challenges

The first challenge met in the implementation would be for Google to find a partner that would manufactures the fridges that would support the GOOSE System, as Google is not in the hardware industry. It should be fairly feasible to collaborate with a well establish fridge manufacturer (like LG, Samsung, etc.) thanks to the notoriety of Google as a brand.

Another challenge would be the market penetration. A lot of market research need to be done to know how to build a satisfying product and once the product is ready, it will need to be promoted. But promotion is not really a problem, Google is the king of online advertisement after all.



## Final Reminder

Before we conclude, I just thought I would remind you that this document contains the complete Album and that you may download, (re-)read or share any of the various chapters and volumes independently/separately, depending on your interests. The PDF, EPUB, MOBI/AZW3/KF8 (Amazon Kindle) and MP3 formats are available.

To get them, just click on any of the links you like below or go to [TheAlbum.NeceMoon.com](http://TheAlbum.NeceMoon.com) (or [necemonyai.com/Blog/page/The-Album.aspx](http://necemonyai.com/Blog/page/The-Album.aspx))

## The NeceMoon Album (complete)

### Volume 1: Moon Light (softcore)

[Chapter 1: Strategy and Tactics](#)

[Chapter 2: Digital Marketing and Web Visualisation](#)

[Chapter 3: Corporate Worlds and Emerging Markets](#)

[Chapter 4: Quick Wins, Tricks and Tips](#)

[Chapter 5: Transition - Extra Thoughts and Sharp Fantasy](#)

### Volume 2: Full Moon (hardcore)

[Chapter 6: Software Development and Engineering](#)

[Chapter 7: C# .NET Programming](#)

[Chapter 8: Epic Prototypes, Classic Projects, Historic Genre](#)

[Chapter 9: Research and Case Studies](#)

The Album is available in French as well at [Album.NeceMoon.com](http://Album.NeceMoon.com) (or [necemonyai.com/Blog/page/L-Album.aspx](http://necemonyai.com/Blog/page/L-Album.aspx))

## Conclusion

This is Good Bye. However, we can stay in touch. Feel very welcome to add me on [LinkedIn](#), [Twitter](#) or [Facebook](#). If you also have any tools and tactics that help you achieve things efficiently, I would like you to tell me about that. My email address is [necemon@gmail.com](mailto:necemon@gmail.com). You are more than welcome to write to me and report any possible mistake in this document, or to suggest any improvement, or to tell me about your favourite parts. However, if you don't like The Album, don't bother writing to me.

Anyways, I wish you all the best for your ongoing and next projects. Thank you for reading. And many thanks to [Wikipedia](#), [MSDN](#), [IconFinder](#) and [FreeDigitalPhotos](#) for the clarifications and the graphic resources. Special thanks to the Evasium Team. Thanks to all those who contributed, thank you Ahou The African Chick, Antoine Mian, Cyriac Gbogou, Darren Mart, Edith Brou, Holty Sow, Israel Yoroba, Jean Luc Houedanou, Jean-Patrick Ehouman, Karen Kakou, Nanda Seye, Nnenna Nwakanma, Olivier Madiba, Vanessa Lecosson and Yehni Djidji.

Thank you Monty Oum, rest in peace.

## [Share The Album with Your Mates](#)

Just click on any relevant icon below. It only takes a couple of seconds and it's free for everyone.

Alternatively, you can share this link on any social media website or email it to your contacts that would benefit from reading it: <http://TheAlbum.NeceMoon.com>

