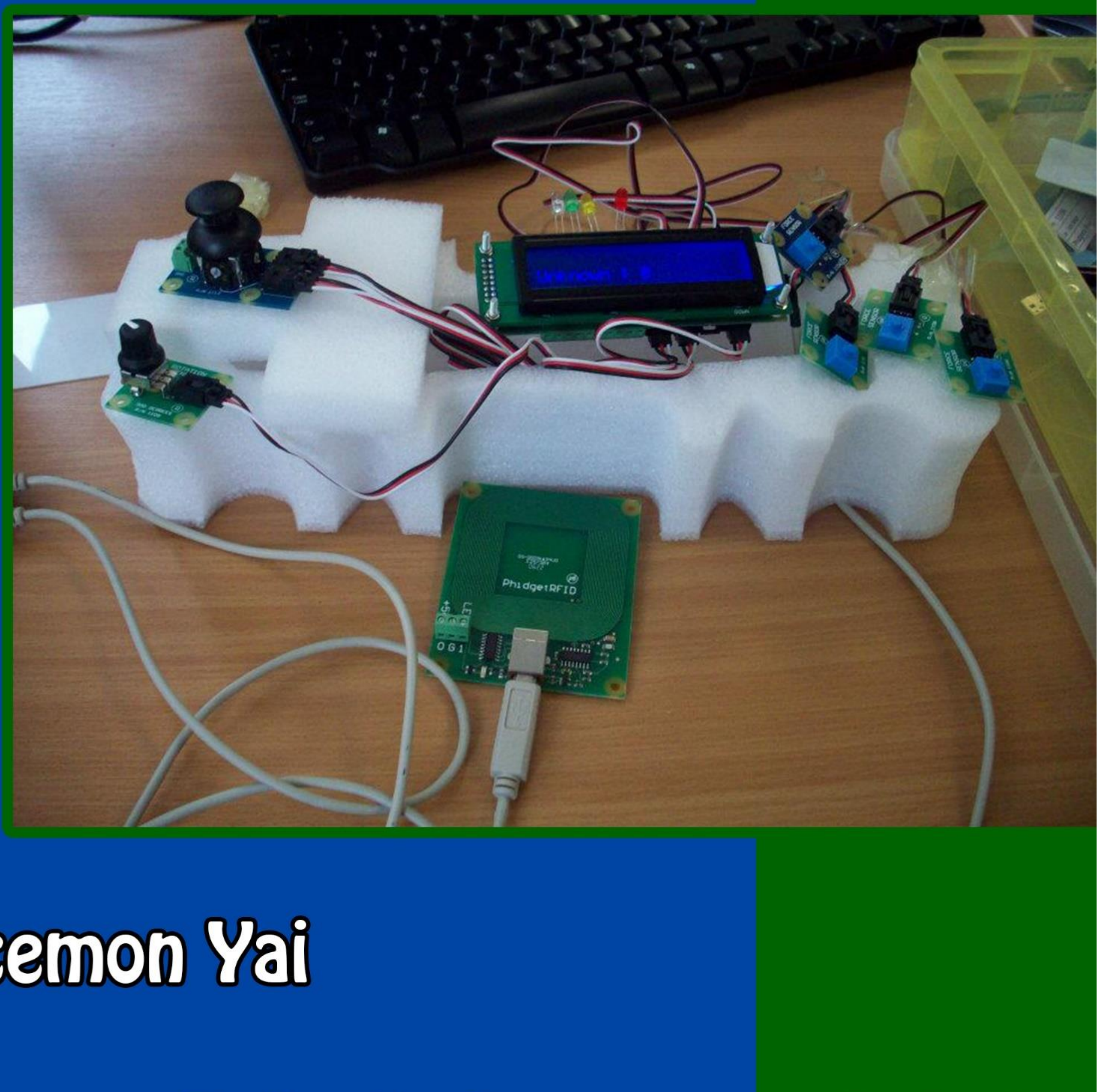
The NeceMoon Album

Technologies and Strategies to Keep Moving Forward

Chapter 9 Research and Case Studies



By **Necemon Yai**

First Edition 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

necemon@gmail.com

www.necemonyai.com

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 5

For All Practical Purposes 8

Chapter 9: Research and Case Studies 9

Phidgets: First Steps In Robotics? 10

Paper Prototyping 17

Dual Shock 3 - Part 1: History of game controllers 19

Dual Shock 3 - Part 2 : The amazing story of the Dual Shock 23

Dual Shock 3 - Part 3: Device Analysis and Specification 25

Business at the speed of game production: OpenGL or XNA? 28

GOOSE (GOOgle Supply search Engine) 29

Final Reminder 33

Conclusion 34

Share The Album with Your Mates 35

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 (or necemonyai.com/blog)
The NeceMoon™ Album can be downloaded in various file formats, in full from TheAlbum.NeceMoon.com (or 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 entreprenerd. 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
(or 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 Chapter 9 of The Album: “Research and Case Studies”. If you care, 8 other chapters are also available. 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
(or 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

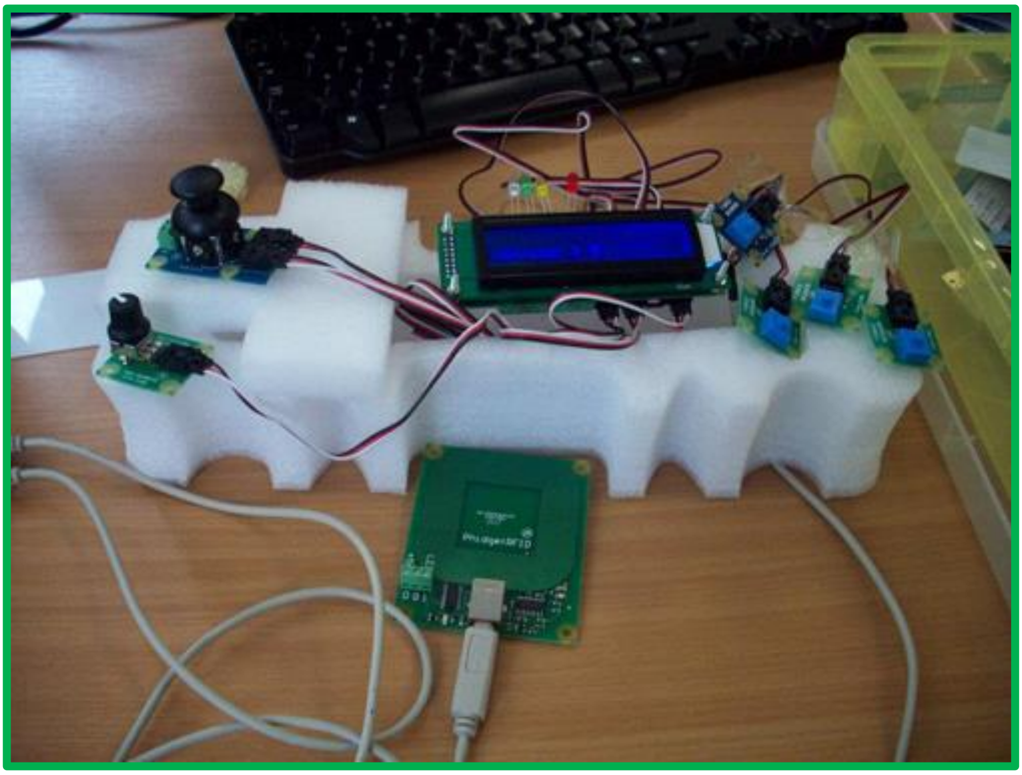
Chapter 9: Research and Case Studies

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

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)

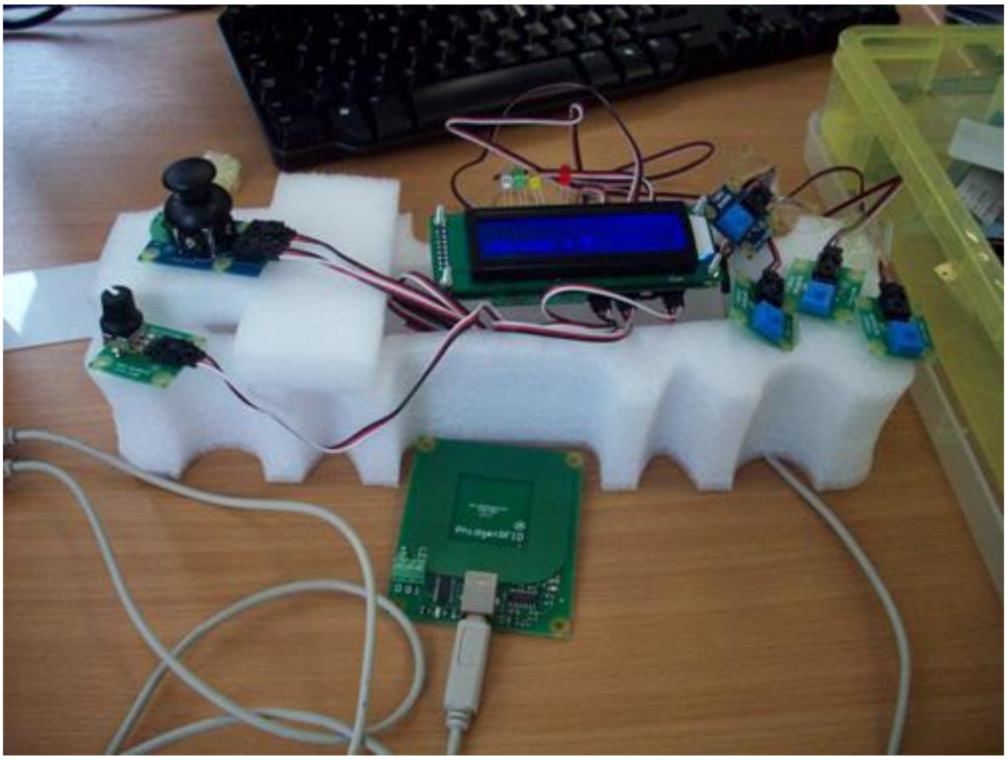
Where to find them ?

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

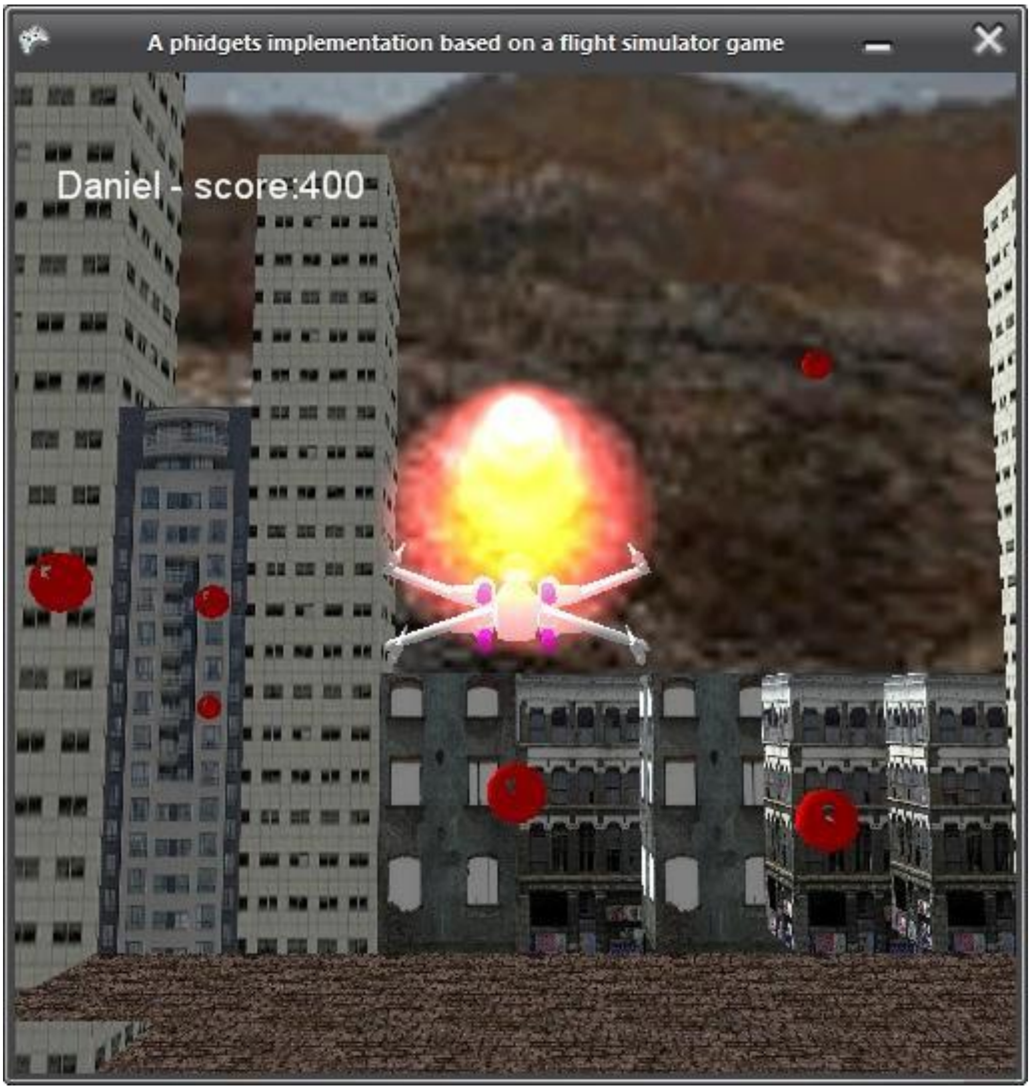
For example, as a C# developer, just download those and you are ready to go:
1. Getting Started Guide
2. Windows Installer (32-bit)
3. Code Samples

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 where 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;
```

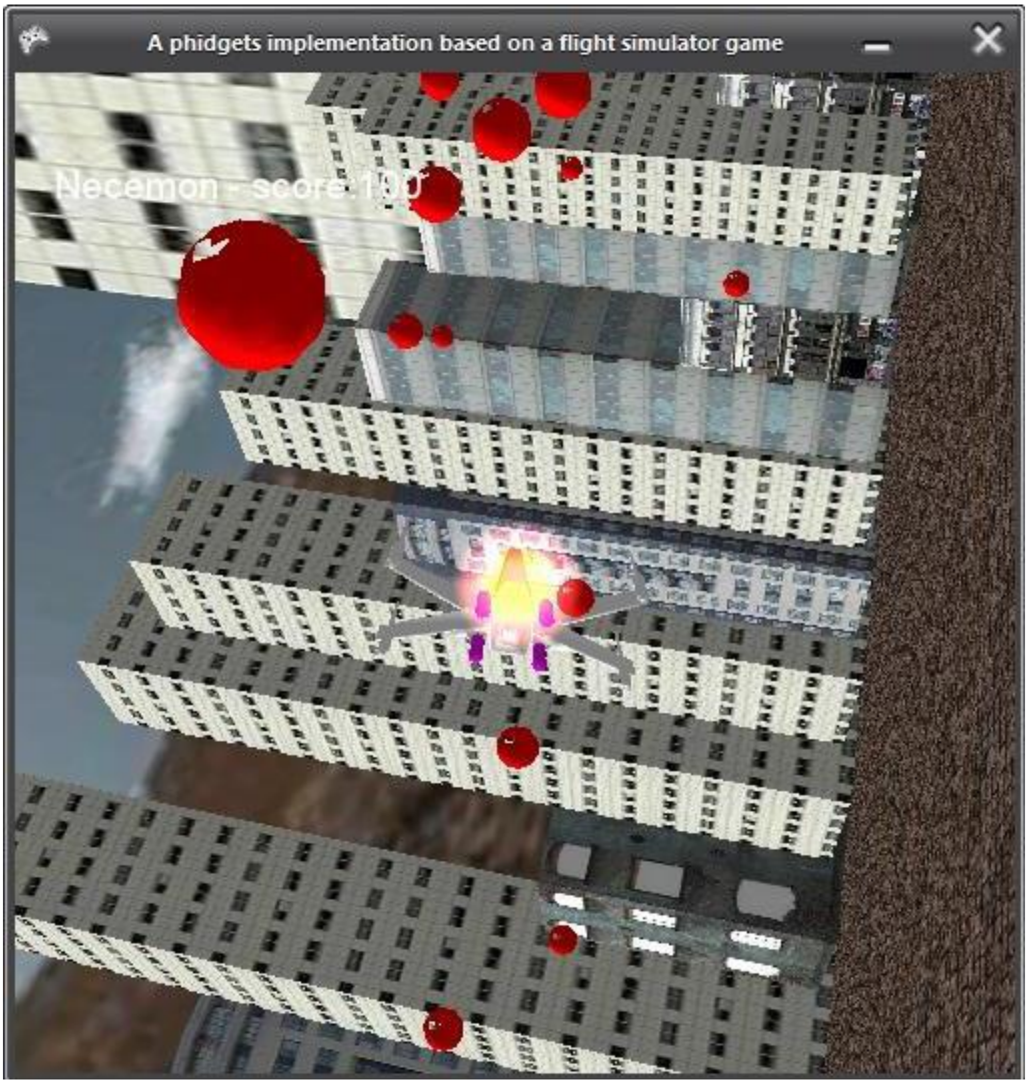
The following code creates a new bullet every time the space bar is pressed provided the previous bullet was fired more than 100 milliseconds ago

```
      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

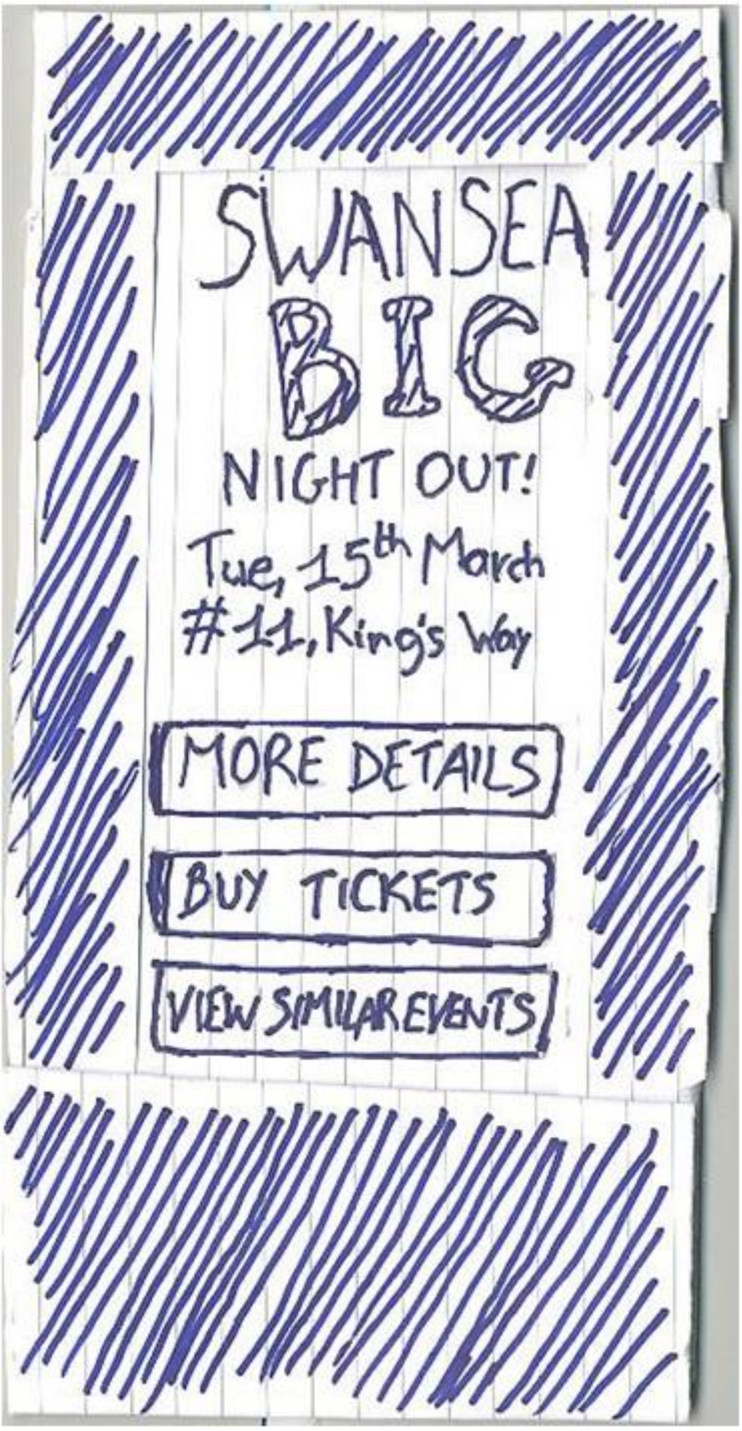
```
if (keys.IsKeyDown(Keys.C) || iw.Force3 > 25)
            {
                //make the lights blink
                iw.ifKit.outputs[0] = (!iw.ifKit.outputs[0]);
                iw.ifKit.outputs[1] = (!iw.ifKit.outputs[1]);
                iw.ifKit.outputs[2] = (!iw.ifKit.outputs[2]);
                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;

                }

            }
```



A phidgets implementation based on a flight simulator game

Paper Prototyping

By Necemon

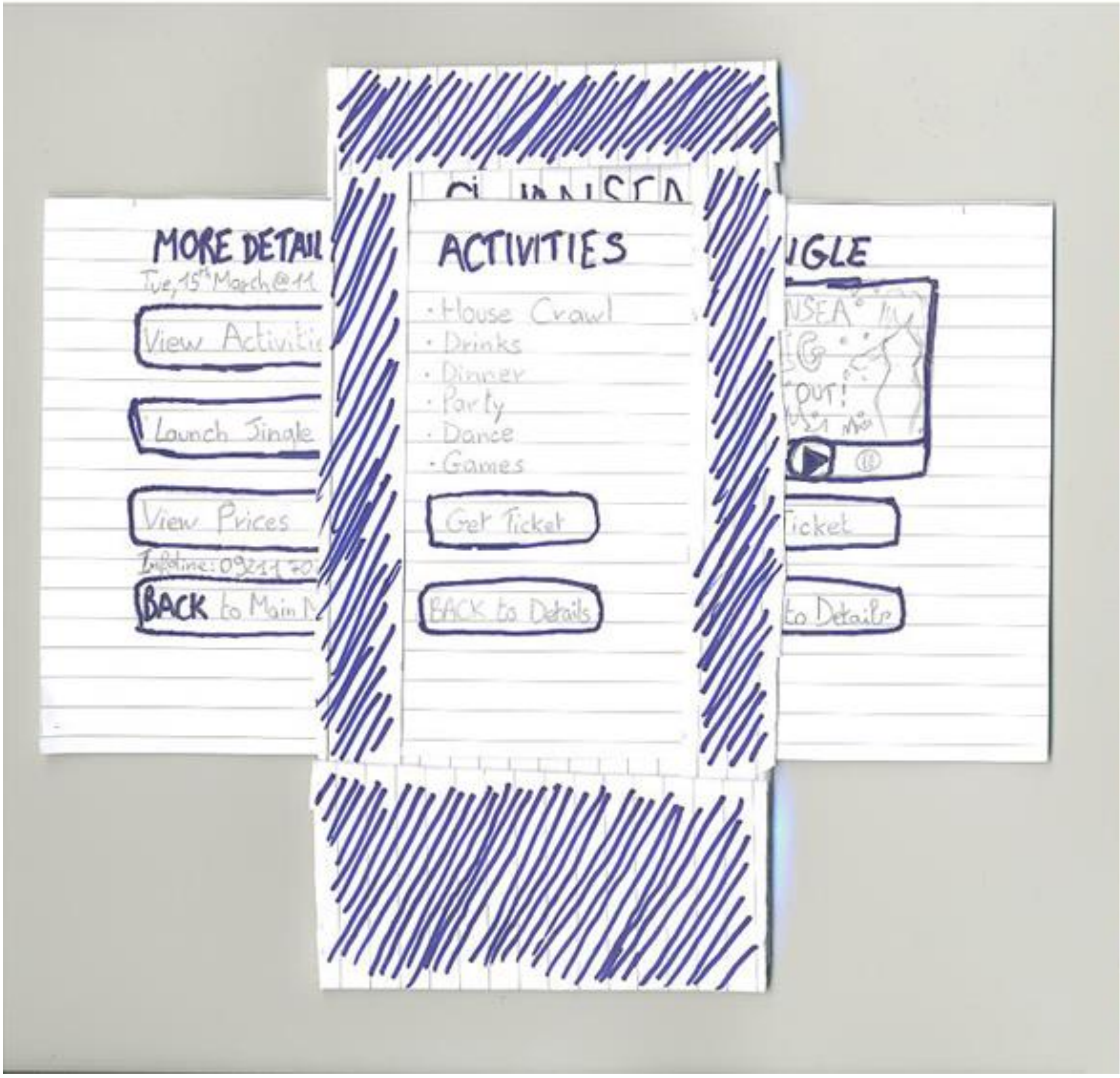


Last week, I had to make a paper prototype for a mobile application I was working on. I ended up building a quite nice thing (picture above) but at the beginning I was not sure about how to go about it. So I just thought I would write about it to share how I proceeded.

The first thing was to do a little bit of research, just to see how people normally do it. The website paperprototyping.com 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.

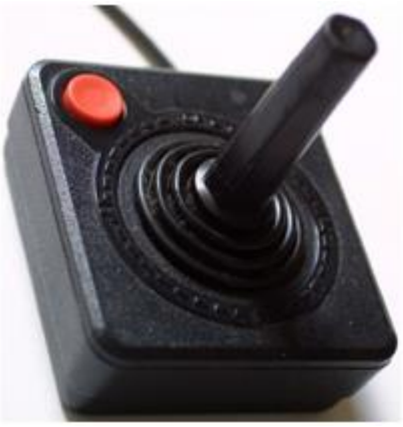
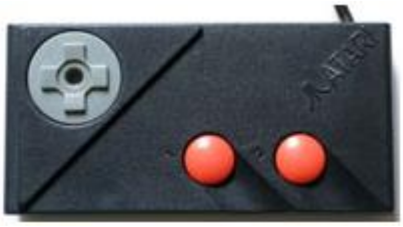
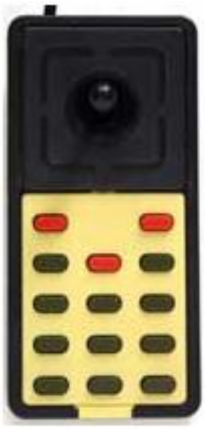
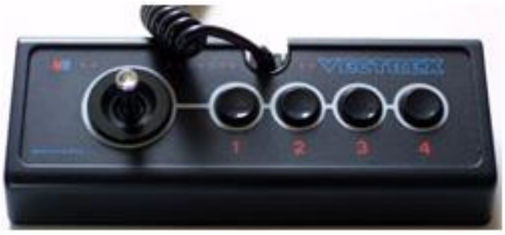
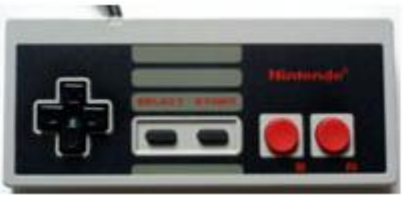
Basics

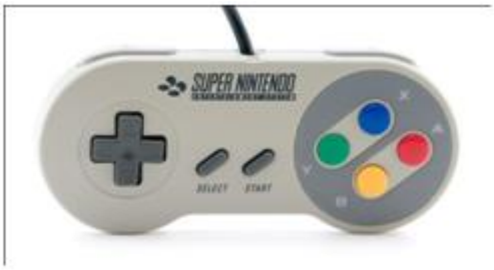
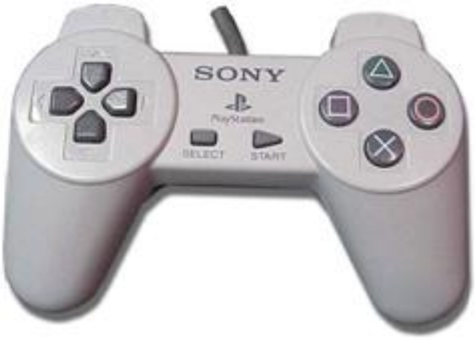
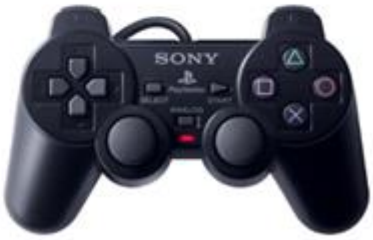
There are many categories of game controllers, among which we will retain the most popular one, the gamepad, as that's exactly what the Dual Shock is. Gamepads, also known as joypads, can have many action buttons combined with one or more omnidirectional control sticks or buttons. They are held using 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	- An analogue control based on a monostable - Game-start buttons were located on the console, not on the paddles - One variant was the use a slider control - Some TV games didn't even put the paddle on wires but place them on the front of the game itself - Generally, with the early models, there were no buttons on the pads	

The Atari 2600 Joystick	- A digital four-way joystick with a single fire button - Again, game-start button on the console itself	
The Atari 7800 Joystick	- A digital four-way joystick with two fire buttons - Simply add an extra-button to the 2600 Joystick	
The 8-bit Home Computer Joystick	- An analogue joystick with a single fire button - Here, the game could be started using the keyboard	
The Vectrex Joystick	- An analogue joystick with FOUR buttons - Games were started from the controller using the normal fire buttons - The fire buttons are labelled '1' to '4'	
The Nintendo Entertainment System Joypad	- A digital joypad with two game-start buttons and two fire buttons - introduced the idea of placing active components and circuitry inside the joypad. - The shape of the game pad, however, is a very basic rectangular box.	

	- The two fire buttons are both red in colour and are labelled ‘A’ and ‘B’.	
The Super Nintendo Entertainment System Joypad	- A digital joypad with two game-start buttons and six fire buttons. - introduced the idea of shoulder buttons - shaped much more naturally to fit the player's hands.	
The Sony PlayStation I Joypad	- A digital joypad with two game-start buttons and eight fire buttons. - adds more shoulder buttons to the SNES pad - an even more ergonomic shape	
The Sony PlayStation II Joypad	- An update to the PlayStation I controller that adds analog controls and a pair of vibration motors. - the analog-to-digital converter (ADC) is in the controller and the interface is digital. - The two motors are fitted with eccentric weights of different sizes, and can be independently controlled by the game.	

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 Entertaiment. 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."

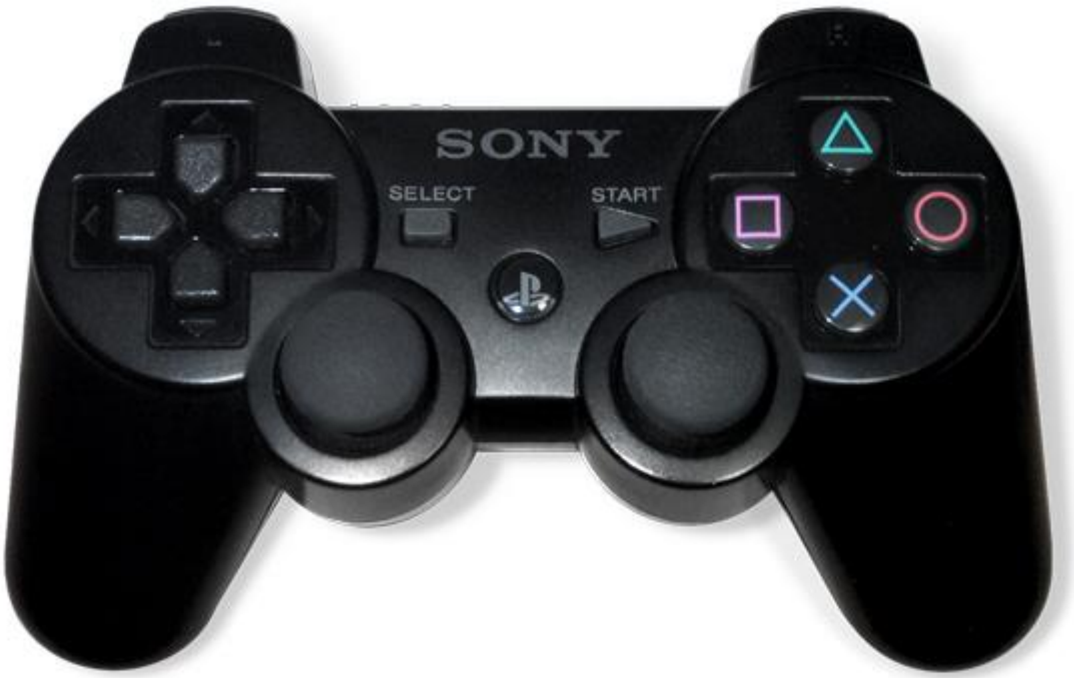
In our study, we are more concerned with the PlayStation controller than the console itself. But the interesting part is that Gotoh designed not only the main parts of the PlayStation but also its peripherals such as the memory card and eventually the controller. However, at the beginning, the controller 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 that 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 as 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)

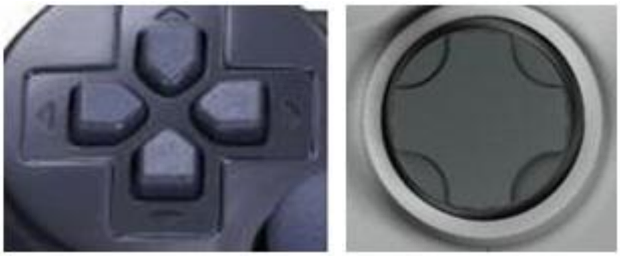
Ergonomically, all the Dual Shock are pretty much the same. And honestly, they are the most comfortable game controllers I personally had the chance to play with (I had been trying a wide range of controllers over the years). However, one downside of the Dual Shock 3 could be that it is quite lighter 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 matter 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 Chapter 9 of the Album, and that 8 other chapters are also available. 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
(or 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
(or 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. 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