Yuan Yifan

ACCELERATOR BASED GAME PROGRAMMING ON ANDROID MOBILE PHONE

Bachelor´s Thesis
Information Technology

May 2013

MIKKELIN AMMATTIKORKEAKOULU
Mikkeli University of Applied Sciences

DESCRIPTION

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Date of the bachelor's thesis** May 13th 2013
Author(s) Yuan Yifan	**Degree programme and option** Information Technology

Name of the bachelor's thesis

Accelerator Based Mobile Phone Game Programming on Android

Abstract

Today mobile phones play a more and more important role in our daily life. Not only because of the function of communication but because of the new features of the smart phones today. People use smart phones because smart phones are able to help people do a lot of things. With the development of the hardware of mobile phones the quality and popularity of mobile phone games have increased. Android platform takes the biggest part of the smart phone market. The aim of my thesis is to get familiar with Android smart phone and implement an application that can run on an actual Android smart phone.

To achieve the aim, I plan to develop a mobile phone game on Android platform.The idea of the game is that there is a drunk chicken going back to his home. But actually this is a endless running game, so the chicken will never really arrive his home. I make the game using accelerometer to keep the balance of the chicken. And I will make it not so easy to control the balance by adding a lateral acceleration.

Uniquely, I decide to use a new software to develop this game. Normally the Android SDK is the common way to program a Android game. However, in this project I use Adobe Flash to achieve the same goal and compare the advantages and drawbacks between Flash and Android SDK.

Subject headings, (keywords)

Android, Flash, ActionScript, Accelerator, Smart Phone

Pages	**Language**	**URN**
48	English	NBN:fi:amk-201305219955

Remarks, notes on appendices

Tutor	**Employer of the bachelor's thesis**
Matti Koivisto	Mikkeli University of Applied Sciences

CONTENTS

1. INTRODUCTION

Today the smart mobile phones are no longer simply a tool for communication. With various applications, mobile phones have numerous functions. Sending E-mail, surfing on the internet, chatting, taking photos, recording videos, translating text and so on are some typical application areas. However, mobile games nowadays are playing one of the most important role in all of those functions. We can not deny that these applications can brighten up our boring days and become one of your relaxation's activities that you can easily participate with your finger tips.

The trends of mobile phones are changing and evolving all the time, so do the games. No one could have imaged that we can play a 3D game on mobile phone with a touch screen while they were playing single games like "Snake" with an old monochrome Nokia phone. The categories of available games naturally depends on the type of the mobile phone. The hardware of our mobile phone is more and more powerful today so that the games have also become more and more attractive and realistic.

However, the graphics are not the only important issue for a game. Games are for playing, so the most important point of a game is how to control the game. We used the keyboard to play the "Snake". Nowadays, we mainly play with touching the screen because only few of the latest mobile phones still contain keyboards. The problem is that touch-screen is not good at controlling an action game because people easily touch the wrong place on the screen. Therefore, the popularity of accelerometer based games is obviously increasing. An accelerometer is a device that can measures proper acceleration which has multiple applications in industry and science. Accelerometer is now becoming an essential component on a mobile phone.

The aim of my project is to get familiar with the process of accelerometer based mobile phone gaming. That is the reason why I decided to make a game application in this project. This game is called "Pixel Drunk Chicken", the idea of the game is that Pixel Drunk Chicken is going back to his home. He can not keep the balance while walking. You have to help this chicken walk to his home safely. My intention is to make the game using an accelerometer, so the players can keep the balance of the chicken by moving mobile devices in their hands.

The structure of the thesis is following:

In Chapter 2, I introduce the history of mobile gaming. It contains the most popular categories of mobile games in each period and shows the advantages and drawbacks of them.

In Chapter 3, I introduce the Adobe Flash platform, which I used to develop my game. This chapter include the history of Adobe Flash, the main features and the new functions of the latest version and how those function helped me to develop my application.

In Chapter 4, I demonstrate the logic relations between each part of the game. Then I introduce the content of the game which includes the special modes, items, roles, and buildings.

In Chapter 5, The effects of the main classes and values are mentioned. Also, I explain the principles of the main functions in this game.

In Chapter 6, there are the results of the tests of this game. Testing includes both simulator test and practical tests with a real device.

In Chapter 7, I demonstrate somethings that have yet to perfect in the game and some further contents that I want to implement in this game.

2. THE DEVELOPMENT OF MOBILE PHONE GAMES

2.1 Overview

With the development of communication and electronic technology in recent years, mobile phone has become a necessary device in our daily life. The things we need to take every day when we go out changed from “keys and wallet” to “keys, wallet and mobile phone”. With the more powerful hardware and 3G even 4G network connections, we can almost do everything with our smart phone. However, it seems like that one function is getting more and more interest, which is the games.

A pie chart in Figure 2.1 shows the percentage of time spent per application category among Android and iOS smart phone owners.

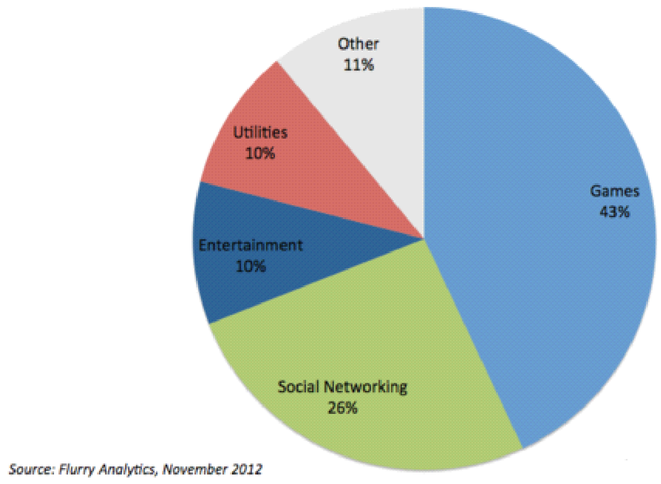


Figure 2.1 Global iOS &Android Time Spent Per App Category [1]

As we can see from Figure 2.1, games take up 43% percent of people’s smart phone time. It is the biggest part among all categories of applications. And the situation is that not only with the smart phones. I believe that all kinds of mobile phones have game function, even the most basic phone. Therefore, everyone is able to play a game with their mobile phone.

The main reason which increases the popularity of games is that people really have time and need some form of recreation. For instance in Beijing, after a whole day work, most of the citizens have to go back home by public transportations. So what can people do while they are waiting for a bus or taking a metro? The answer is that most of them play games even though some people prefer to read an ebook or surf on the internet. On the other hand, because of the low capacity of network in crowded buses and subways, offline games have further strengthened the top position of games.

Just like the computer games grow together with computers, mobile phone games also develop with mobile phones. Mobile gaming has more users and is growing more rapidly than any other gaming platform in the US, according to new data from eMarketer [2]. All in all, the market of mobile phone games is a very considerable market. It is worth to spent time in this field.

2.2 Embedded Mobile Phone Games

Because of the limitations of the mobile phone systems and hardware in late 90s, people were not able to create and install an application in a mobile device by themselves. So, embedded games were the only type of game which could run on the old original mobile phone.

Embedded games are installed in the microchips of the mobile phones. Users are not able to modify, add or delete embedded games. Mostly, this kind of games appeared on old or basic mobile phones.

One game that I have to notice is Snake, see Figure 2.2. No one can deny that Snake is one of the most important member in the mobile phone games family. Appeared in 1997, the ancestor of all mobile phone games, Snake was the first mobile phone game in the world.

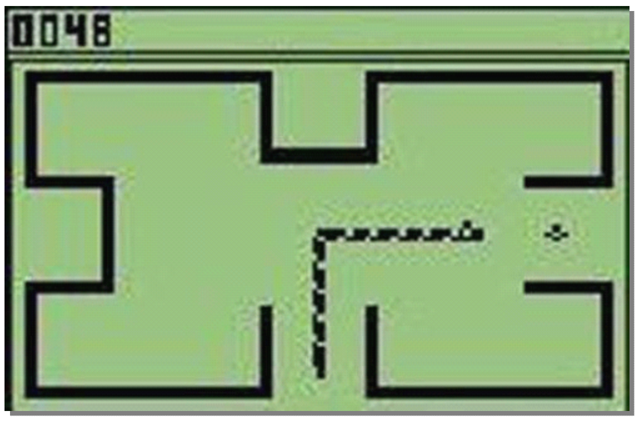


Figure 2.2 Snake, 1997 [3]

Before Snake, no one could image that we can play games with a basic mobile phone like Nokia 6110. Undisputedly, Snake is an embedded mobile phone game.

2.3 Java Based Mobile Phone games

After the period of monochrome phones, mobile phones with color displays entered the markets. Embedded games were no longer the main actor in mobile phone games. because mobile phones began to have flash memory and faster processors. So we were able to install some custom applications and games even though we still did not have a real operating system in mobile phone.

Before smart phones became popular. Java based mobile phone games were the dominant player in mobile phone game world. A great number of mobile gaming companies appeared in this period while some of the computer game companies started to join the mobile gaming market. Gameloft, EA and CAPCOM are good examples of them.

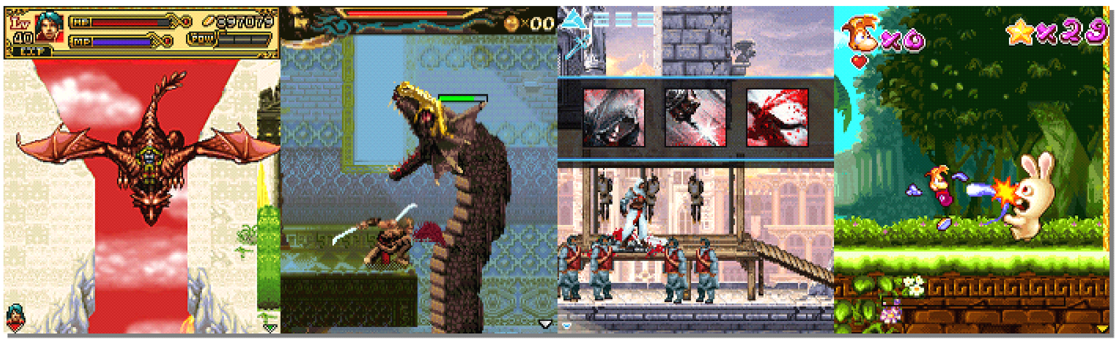


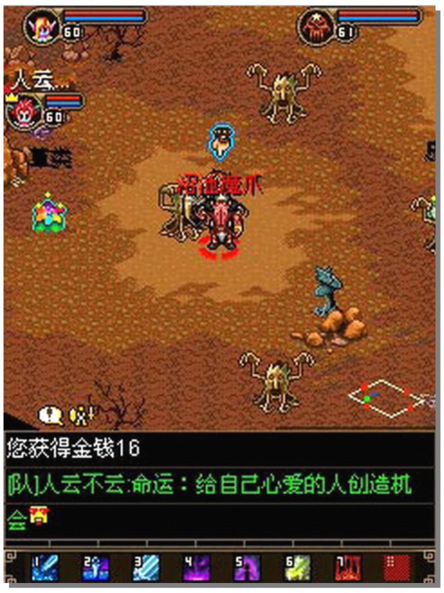
Figure 2.3 Some Java based games from Gameloft [4]

As we can see from Figure 2.3, mobile games were not color blind any more. The graphics and game content were improved a lot. Even today, Java games still take up a small part of the market but it is not the most popular mode one anymore. Over a time users' expectations continue to grow. As the cost of wireless data communication reduced, a new market was founded by mobile game companies - online games.

2.4 Wireless Application Protocol

Wireless Application Protocol is a technical standard for accessing information over a mobile wireless network. Before the introduction of WAP, mobile service providers had limited opportunities to offer interactive data services [5].

With WAP, J2ME and BREW gaming got another evolution. Even some instant messaging applications and browsers came out with WAP support. For mobile gaming, the biggest change was that users were able to play online games with a WAP supported phone.



您获得金钱16
[队]人云不云:命运：给自己心爱的人创造机
会

Figure 2.4 China online mobile phone game - Tian Jie

Shown in Figure 2.4, Tian Jie is a excellent example to show how successful the first generation mobile phone online game was. It had more than 7,000,000 registered users and tens of thousands of players online at the same time.

However, because of the low data rate of 2G network, online games did not dominant the mobile gaming market at that time.

2.5 Symbian OS and Nokia N-Gage

Not the first but the first popular smart phone operating system was Symbian. People started to know Symbian OS from NOKIA 7650 (shown in Figure 2.5) in 2002.

Figure 2.5 NOKIA 7650

At that time, JAVA based applications were still controlling the market. So maybe this is the reason why, as a smart phone system which has its own application format, Symbian also supported JAVA applications. But beyond people's imagination, the popularity of Symbian increased sharply and became the hero of the market in 2004. Mobile gaming companies began to make specific games for Symbian OS. And then Nokia N-Gage showed at the right time.

N-Gage (Figure 2.6) was a mobile phone and handheld game system by Nokia, based on the Nokia Series 60 platform [6].

Figure 2.6 N-Gage QD

Although N-Gage was not a very successful product but the key is that it made people feel like "Yes, we can play games very well with Symbian system! ". However, a mobile phone is always a mobile phone, not a game center. So after N-Gage QD, Nokia gave this serial up. It did not means Nokia give games up. On the contrary, Nokia found the balance point of the mobile games and mobile phones, N81 (Figure 2.7) was the best example of this.

Figure 2.7 NOKIA N81

Not like the N-Gage, the appearance of N81 is more like a mobile phone instead of a game handheld. However, the amazing thing was it still has the function of a game handheld. People were still able to play the specific games for N-Gage. Based on those advantages, N81 became the most popular phone in young people group.

From this period, JAVA based games started to decay. More and more companies began to focus on Symbian. Then plenty of excellent games appeared on Symbian. Figure 2.8 shows a 3D game - Spiderman on Symbian.

Figure 2.8 Spiderman 3D [7]

No one could foresee that Symbian suddenly fell down from the top when Google Android drifted into view. And soon, Apple also enrolled to this competition with its IOS.

2.6 The latest mobile phone games

People already tasted the sweetness of smart phones from Symbian. Therefore, when Android appeared, compared with the inflexible user interface of Symbian, Android's attractive user interface was so likable for the smart phone users. On the other hand, this open source system attracted a lot of companies, including HTC, Samsung, Sony and some other small companies who were not able to produce smart phones. With the support both from users and companies, Android fastly replaced the top spot of Symbian.

With Android, mobile gaming entered an unprecedented era of great development. Using the improved hardware of the new mobile phones, game companies were allowed to be more open to creative ideas on their games. And because of the Android software development kit is accessible for everyone. So not only those professional mobile gaming companies but also some small groups even individual person were able to make an Android application.

Figure 2.9 shows the Android game Asphalt 7 from Gameloft. As we can see, with the duel-core even quad-core processor and individual graphic processing unit, mobile phones now are able to run a huge size of 3D game.

Figure 2.9 Asphalt 7 [8]

Although, Android is very popular is does not dominate all the market. A powerful competitor of Android is Apple IOS.

On the contrary, IOS is a completely closed operating system. So Apple bravely opened the first commercial application store - App store. Which mean the gaming companies were not the only who can earn money from the games they made. However, it seems like this is a win-win situation. As a closed OS, IOS does not allows users to download any applications without App store. So not like Android, a big part of the applications in App store are not free. Companies can earn more money from that. This is the reason why Android has the largest number of users but most of the games are released on IOS first. For example the Angry birds in Figure 2.10. Angry birds now supports more than 10 platforms but it first showed in Apple App store.

Figure 2.10 Angry birds

Now, the thing every game company should to do is to make a game for multiple platforms. Android, IOS and even Windows phone is coming. The companies are in a kind of dilemma that if they make a game support multiple platform, the cost increase or they lose the users who are using the certain platform.

This has lead companies to find out platform independent solutions to implement mobile games. One interesting solution is Adobe Flash which is covered in more details in next chapter.

3. ADOBE FLASH AND ACTIONSCRIPT

3.1 The History of Adobe Flash

Adobe is one of the most successful computer software companies in the world. The company has historically focused upon the creation of multimedia and creativity software products, with a more-recent foray towards rich Internet application software development [9].

The software I am going to introduce here is Adobe Flash. Adobe Flash is a multimedia and software platform used for authoring of vector graphics, animation, games and Rich Internet Applications (RIAs) which can be viewed, played and executed in Adobe Flash Player [10].

Flash was originally created by Future Wave, and at that time its name was Future Splash. Macromedia purchased Future Wave in 1996 and renamed it to Flash. Then Adobe purchased Macromedia in 2005. At first, the purpose of Flash was to easily make 2D animation and play the animations on the website. This was a very creative and amazing thing around 1996 while most of the websites did not even use JPG images. Flash successfully attracted the eyes of Microsoft and Disney. That was one of the main reason why Flash grows rapidly. However, Flash was still defined as a simple victor graphic software at that time.

There was a big change from Flash 2.0 after Macromedia purchased it when it perfected with the first scripting language of Flash, ActionScript 1.0. With ActionScript 1.0, Flash was able to realize human-machine interaction. In 2000, ActionScript 2.0 was shown with Flash 5.0. Since then Flash was no longer only a victor graphic software. With ActionScript 2.0 and ability to play video and music, Flash has become to a real multimedia platform.

After that, Flash became more and more popular. Because of the simplicity of using Flash, a great number of people started to focus on making Flash animation and games.

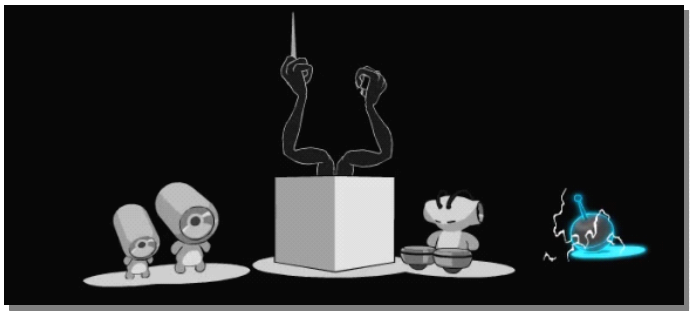


Figure 3.1 Tokyoplastic 2003

The Tokyoplastic in Figure 3.1 was one of the best Flash animations in 2003. Not only Flash animation, but also a lot of Flash games appeared. Increasing number of creative Flash games were made by using ActionScript 2.0 even though ActionScript 2.0 was still not a real programming language.

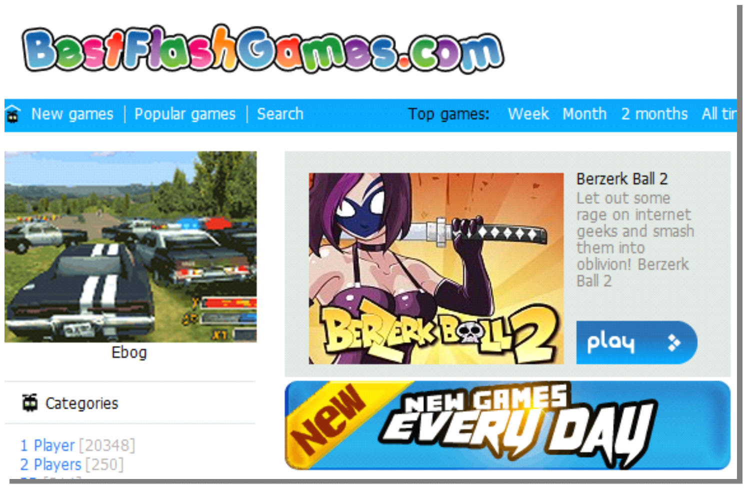

BestFlashGames.com
New games | Popular games | Search
Top games: Week Month 2 months All ti
Ebog
Berzerk Ball 2
Let out some rage on internet geeks and smash them into oblivion! Berzerk Ball 2
play
Categories
1 Player [20348]
2 Players [250]
New
NEW GAMES EVERY DAY

Figure 3.2 BestFlashGames.com [11]

Figure 3.2 shows a famous Flash game website - BestFlashGames.com. The appearing of this kind of website proofs that there are really a large amount of Flash games in the world. Because typically Flash games are played on the website, it is not possible to make large games with it. This seems like a drawback of Flash platform. However, this is also the reason why there are a lot of small fun games which make people addicted to them. From this point of view, Flash games are similar with mobile phone games.

As said earlier in 2006, Macromedia was purchased by Adobe and Flash became a part of Adobe's Creative Suite. The next year, ActionScript 3.0 came out with Flash 9.0. That was the biggest step of Flash ever. Unlike its ancestors, ActionScript 3.0 is a complete object-oriented programming language. I will cover in ActionScript 3.0 later in more details.

In 2010 with the introduction of Adobe Flash CS5, Adobe Flash was a very professional animation, graphic and game multimedia platform. At this point, the common thing of all the versions of Flash is that Flash can only run in a desktop operating system. However, as I mentioned in the previous chapter, smart phones already had became popular at that time. Especially Android and iOS mobile phones. Adobe sensitively realized that they need to pay more attentions on mobile platforms.

So in the next two versions of Adobe Flash - Adobe Flash CS5.5 and CS6. We found that we can make iOS and Android applications by Adobe Flash. Before this, each of the mobile platform has its own programming tool, for example the Android SDK for Android devices but now, Adobe Flash was the first third-party mobile application development platform.

3.2 Adobe Integrated Runtime

Adobe AIR is a cross-platform runtime system. The Adobe AIR runtime enables developers to package the same code into native apps for iPhone, iPad, Kindle Fire, Nook Tablet, and other Android devices [12]. The Adobe AIR is one of the key technologies to make Flash accessible to mobile platforms.

I believe Adobe Flash with Adobe AIR could be a great and efficient tool for mobile application development. There are some reasons. First of all, Adobe Flash CS6 has specific and powerful functions for mobile application development. The second and more important is that the application companies have no need to pay more research and development funding for different platforms. Let us think about the following situation.

Figure 3.3 FINAL FANTASY DIMENSIONS for iOS

The game shown in Figure 3.3 is named Final Fantasy Dimensions. It is a very excellent RPG mobile game. However, it is a iOS-only game. It means all the users who are using the other platforms are not able to play this game. And the reason why the game company does not develop the game in the other platforms maybe is just because they do not want to pay more development money.

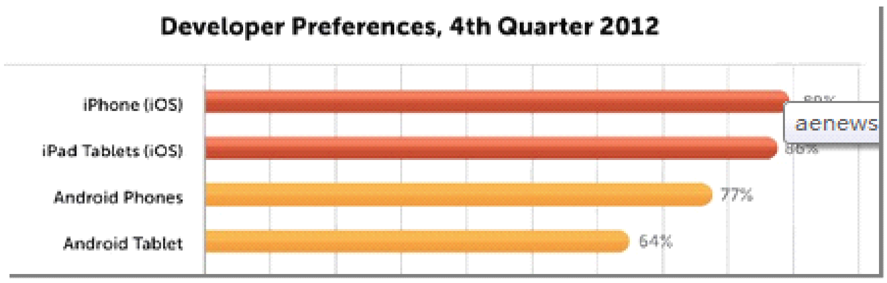


Figure 3.4 Developer Preferences [13]

And then let us look at Figure 3.4, we can see that developers prefer iOS over Android. The main reason is that they can get more benefits from iOS than from Android. Therefore, there are some of the applications like the game in Figure 3.3. The company only release their applications on iOS platform because they think they can not get enough benefit from the other platforms.

However, this problem may be solved by Adobe Flash and Adobe AIR. Because with Adobe AIR, the code of the applications for different platforms is the same. Companies have no need to intentionally make different version for different platforms. I think this is the best improvement Adobe was made.

3.3 Adobe Flash Professional CS6

Released in 2012, Adobe Flash Professional CS6 is the latest version of Adobe Flash. It is also the software which I use to implement this project.

3.3.1 Typical Functions of Adobe Flash Professional

At first I am going to introduce some of the basic functions of Adobe Flash Professional.

The most important roles of Flash animation are frames and components (shown in Figure 3.5). All the Flash animations consist of fames and components.

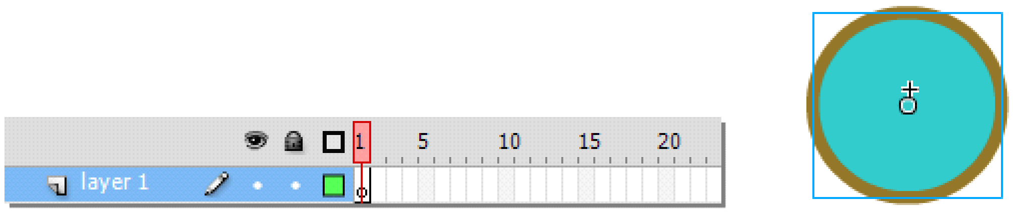


Figure 3.5 Frames and components

There is a common idea of game development. To make a game with graphics, there are always two essential parts to be considered. They are the coding part and the graphic part. The graphic part directly shows the interface and the content of the game to people through the screen. While the coding part consists of a lot of commands which the CPU can read and execute to control the graphics and utilize the specific component of the devices.

An immense advantage of Flash is that we can easily design and make the graphic part without coding anything. As a powerful animation software, Adobe Flash professional has multiple tools to draw or import and edit images as components. And the graphic part are comprised of components and images. However if we use some of the typically programming languages like C++ to implement the graphic part we have to code some commands to order the device to draw every frames. Comparing these two ways, obviously, using Flash is more intuitive for the application designer.

The second but also a vital feature of Flash is the MovieClip. A MovieClip can be considered as integrated animation. We can even to import another Flash animation as a MovieClip in our own project. We can put any MovieClip on any stage as a component in the project.

MovieClip makes it much easier to produce the whole animation or game. For example, if we want to make a man walking on the street it will be too complicated and messy if we put all the components (include the legs, arms and shoes) on the stage. However, we can make a MovieClip of a walking man, which means now the walking man is an integrated component. Then we can put the whole walking man component on the stage.

Actually, a MovieClip is a subclass of the stage which contain the MovieClip. For an example situation: The stage contains MovieClipA. MovieClipA contains MovieClipB. And MovieClipB has a value V. We can use ActionScript command MovieClipA.MovieClipB.V to get the value of V.

3.3.2 New Features of Adobe Flash Professional

Compared to the earlier versions of Adobe Flash Professional, the version of CS6 has some very useful new features especially for mobile phone programming. Firstly, also the most vital function that I can use for mobile game development is the Adobe AIR mobile simulator.

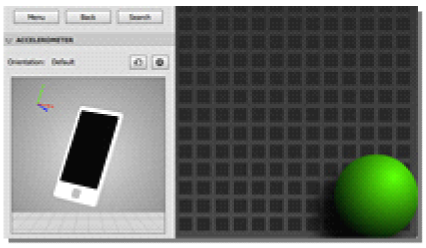


Figure 3.6 Adobe AIR mobile simulator

Shown in Figure 3.6, Adobe AIR mobile simulator can simulate common mobile application interactions like screen orientation, touch gestures, and accelerometer to help speed up testing [14].

With Adobe AIR mobile simulator, we have no need to download the special simulator of the platform you want to use. Also, we have no need to export our application every time when we want to test it in the simulator. Just press CTRL + Enter to start the simulator. It saves substantial amount of my time.

The next is code snippets. Code snippets is a extremely useful new function in Adobe Flash Professional CS6. It contains ActionScript 3 snippets that are commonly used in projects such as banner advertisements and games.

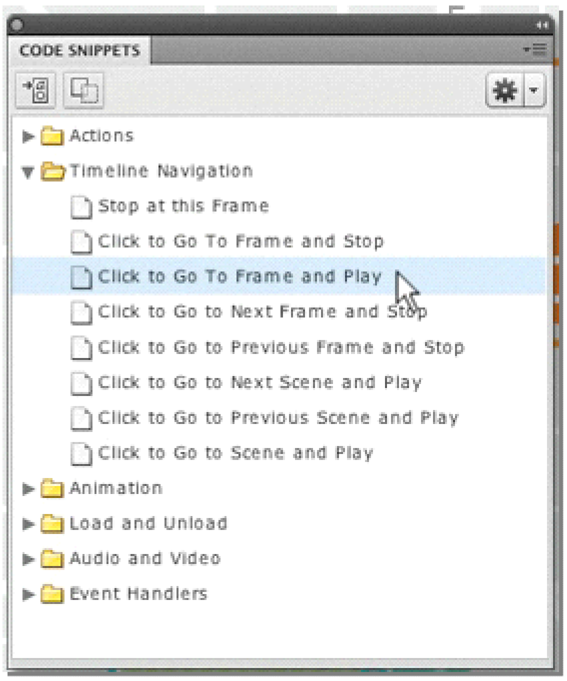

CODE SNIPPETS
Actions
Timeline Navigation
Stop at this Frame
Click to Go To Frame and Stop
Click to Go To Frame and Play
Click to Go to Next Frame and Stop
Click to Go to Previous Frame and Stop
Click to Go to Next Scene and Play
Click to Go to Previous Scene and Play
Click to Go to Scene and Play
Animation
Load and Unload
Audio and Video
Event Handlers

Figure 3.7 Code snippets

Figure 3.7 shows the code snippets window in Adobe Flash. As we can see, it is very simple to use the available code snippets. Just click the component you want to control and choose the snippets. This is a very efficient way to connect the graphic part and coding part.

Code snippets are useful not only individually, but they can be mixed and matched to rapidly create complex interactive projects. Advanced programmers can use this panel to create and store custom snippets for later use [15]. This is the first time for me to program with ActionScript 3.0. It is generally known that the principles of different programming language are similar but the grammars are different. As a programmer, I know the principles but I am not familiar with the grammars of ActionScript 3.0. Therefore, I like the code snippets because they can help me to learn the grammars efficiently.

3.4 ActionScript 3.0

ActionScript is the programming language for the Adobe Flash Player and Adobe AIR run-time environments. It enables interactivity, data handling, and much more in Flash, Flex, and AIR content and applications. ActionScript 3.0 offers a robust programming model that is familiar to developers with a basic knowledge of object-oriented programming [16].

ActionScript 3.0 executes in the ActionScript Virtual Machine. The Adobe AIR which was mentioned in the previous chapter contains the ActionScript Virtual Machine. Now we have the new version of ActionScript Virtual Machine, AVM2. ActionScript 3.0 has much better performance than version 2.0 while running in the AVM 2. ActionScript 3.0 code can execute up to ten times faster than legacy ActionScript code.

In addition, the scripting capacities of ActionScript 3.0 are superior to the previous versions. As I mentioned before, ActionScript 2.0 is not considered as a real object-oriented programming language. But ActionScript 3.0 is. It is designed to facilitate the creation of highly complex applications with large data sets and object-oriented, reusable code bases.

Some of the contents of ActionScript 3.0 are similar to the previous versions such as the basic classes. On the opposite, as a new version, there are also many differences from the previous version because of the new kernel of ActionScript 3.0.

It is well known that errors are the most common things that we meet while we were testing our program. No one can develop a integrated program without any errors not even for the best programmer in the world. Therefore, a better description of the errors can assist the programmer to fix the program efficiently. ActionScript 3.0 reports more error conditions than previous version. It improves the experience of debugging and enabling our application. We can also use the stack tracer to easily find where the error exactly happened.

When developing an application, to reduce the using of CPU is a difficult but important task. Choosing the right type of value is one of the ways. For example, we should not use a double-precision number if we can use a boolean value. ActionScript 3.0 offers three kind of numbers: Number, int and unit. Number means a double-precision number. The unit type is a nice choice for loop counters. Int type is a integer which can make ActionScript code take advantage of the fast integer math capabilities of the CPU. We can use the most suitable type of number while we are programming.

Class is a useful feature in most of the programming languages, including ActionScript. However, too many and complicated classes make programmer confused. ActionScript 3.0 have simplified this problem.

There are still so many useful classes for API in ActionScript 3.0 that permit you to control objects at a low level. But the structure of language in this ActionScript is easier to write and understand than the previous one. It just deleted some of the useless classes in the old versions.

ActionScript 3.0 offers API for accessing the display list. It is a tree that has visual elements in the application. The classes in this function will work with visual primitives.

The Spite class is a lightweight building which is a base class for visual elements like a user interface component. The Shape class represents raw vector shapes. These classes are an examples of the new operator for this function which also can be reparented whenever you want.

Depth management is automatic. It provides methods for specifying and managing the stacking order od objects.

ActionScript 3.0 includes the function of loading and handling assets and data in the application. These methods are easy and consistent to use in API. The loader class has a single mechanism for loading SWF files and image assets and also give a way to reach the information about loaded content. The URLLoaer class has a separate mechanism to load text and binary data in data-driven applications. The Socket class can read and write binary data to sever socket in any format.

A flash.text with all text-related APIs package is included in Actionscript 3.0. Start with the TextLineMetrics class which provides the details about metrics, it can be used for a line of text within a text field. This class has been replaced the TextFormat.getTextExtent() method in the previous version of ActionScript. Next is the TextField class for low-level methods it contains information about a line of text or single character in a text field. For example, the getCharBoundaries() method returns a representing the bounding box of a character. The getCharIndexAtPoint() method returns the index of the character at a specified point. The getFirstCharInParagraph() method returns the index of the first character in a paragraph. Line-level methods include getLineLength(), which returns the number of characters in a specified line of text, and getLineText(), which returns the text of the specified line. Lastly, the Font class is for managing embedded fonts in SWF files.

The classes in the flash.text.engine package will make up the Flash text Engine. This set of class is mainly for a lower-level control over text and creating text frameworks and component.

4. WORKING PRINCIPLE OF THE APPLICATION

4.1 Description

The game is named Pixel Drunk Chicken because the hero of the game is a chicken. I want to imitate the old 8 bit game so I used a special pixel style for the graphic part of this game. We can see the idea of the style from Figure 4.1.

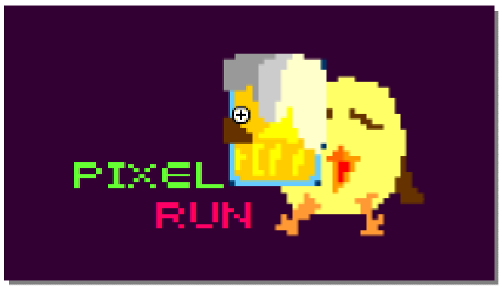


Figure 4.1 Pixel Style

The story of the game is that a fat chicken is drunk in the night, and he wants to go back to his home. However, he can not keep his balance well. The player need to help him to keep his balance while he is walking back by using the accelerometer. There are also some items on the way home for collecting. They may help the chicken but they also might make it more difficult. This is a infinity running game, so the chicken will never arrive his home.

This is a game on Android platform and the development platform is Adobe Flash Professional CS6. The components of mobile phone the game uses are a touch screen and accelerator.

4.2 The Chicken

The first thing I will introduce is the main role that we need to control in the game, the drunk chicken. Figure 4.2 shows the component of the chicken role. I used much smaller pixel for drawing the chicken than any other elements in this game in order to make it cuter and fatter.

Figure 4.2 The Chicken

The chicken is the default character we need to control in the game. I calculated a math function to imitate the real speed-acceleration of a object and used the function expression on the chicken to define the way he move. The exact math function will be explained in the next chapter. And I also make a value of the difficulty. The value increases with the running time of the game. As a result, the chicken will be more and more difficult to control. The game is over whenever the chicken touches the buildings on the street sides.

4.3 Buildings

The buildings on the street sides are in a individual MovieClip. There are some code in the MovieClip to make the buildings and items show randomly. All the building elements are shown in Figure 4.3. Figure 4.4 shows the buildings in the running game.

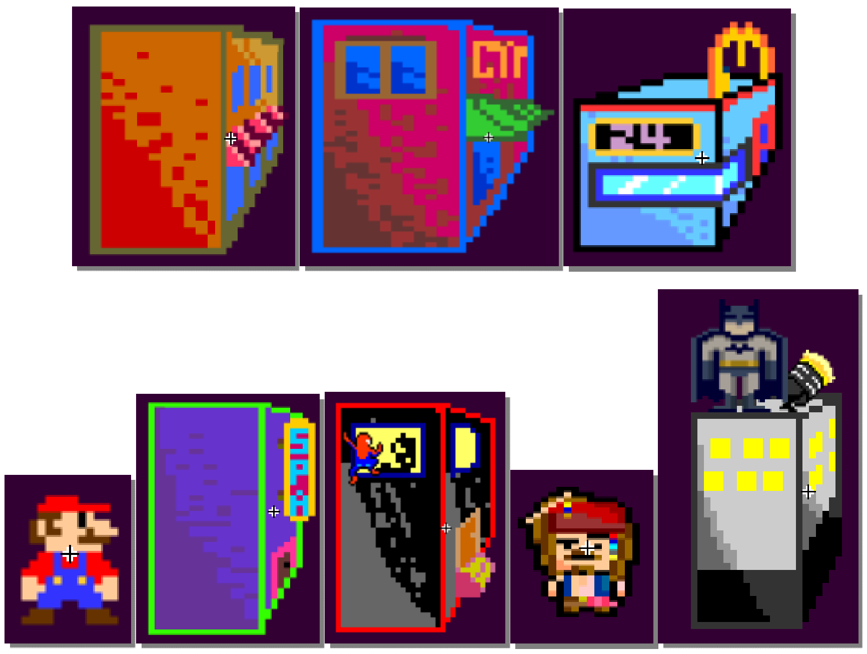


Figure 4.3 All the buildings

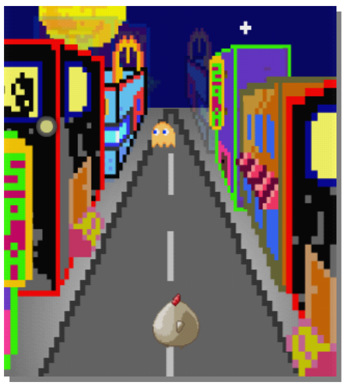


Figure 4.4 The building in the running game

4.4 Items and modes

There are four kinds of items and three different modes in this game.

4.4.1 Superman and Superman mode

The superman is a icon that we can collect in the game. Whenever you get the superman icon, the chicken will change into superman mode for a limited period and add 500 extra points. In superman mode, the role will not be able to move. However the role will be invincible. Figure 4.5 shows the superman icon and the superman role.

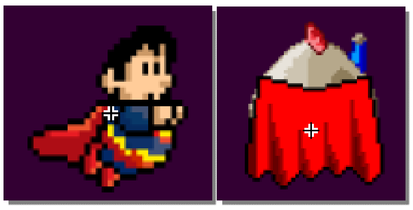


Figure 4.5 Superman

4.4.2 Ghost and Ghost mode

Not like the superman, the ghost mode is more like a debuff. To transform into the ghost mode, you need to get a ghost icon on the street. Also, similar to the supermen mode, the ghost mode remains for a short period of them. In the duration, the effect is inversing the moving direction. See Figure 4.6.

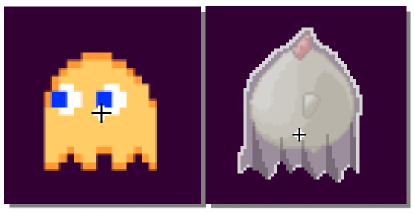


Figure 4.6 Ghost

4.4.2 Bread and sober mode

The accelerated moving is the key point of this game. It is also the feature to make the game difficult. Eating bread is the only way to change the pattern of movement. After you eat the bread, you will enroll into the sober mode for a while. In sober mode, the acceleration will be disabled. So that you can easily control the role for a short time. There is no special mode role for sober mode. The Bread is shown in Figure 4.7.

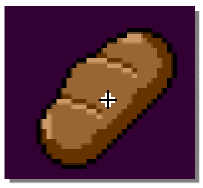


Figure 4.7 Bread

4.4.3 Milk

You will not get into any special mode after you drink milk in the game. The effect of milk is to reset the difficulty value. It seems like not as useful as the previous items however it might be the most important thing in the later game to make you survive. The icon is in Figure 4.8.

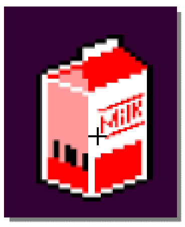


Figure 4.8 Milk

4.5 Game Operation Principle

The game is an ActionScript Flash game. So unlike the normal Flash animations that have a great amount of frames there are only four main frames in this game. They are the start frame, setting frame, playing frame and game over frame. Figure 4.9 shows the logical relations between those frames.

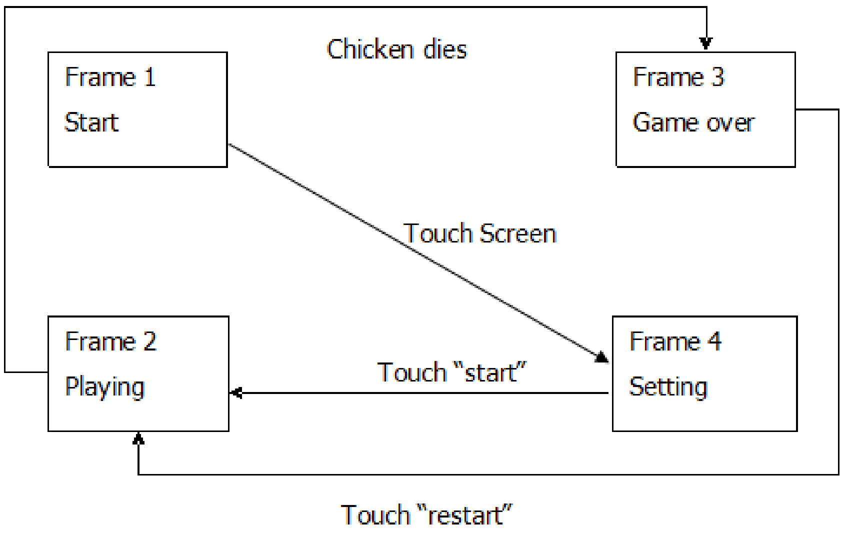


Figure 4.9 logical relations

4.5.1 Start frame

The frame 1 is used as a start frame. It is not a very important frame for this game. However, it contains the name of the game. In addition, the start frame also gives people the first impressions (See Figure 4.10). I tried several kinds of the first frames and finally, I decided to use a lot random squares with different size, angle rotation and alpha. The purpose of using the squares is echoing the theme of the pixel style. The only function of the frame is that when you touch anywhere in the screen, the game will go further to the setting frame.

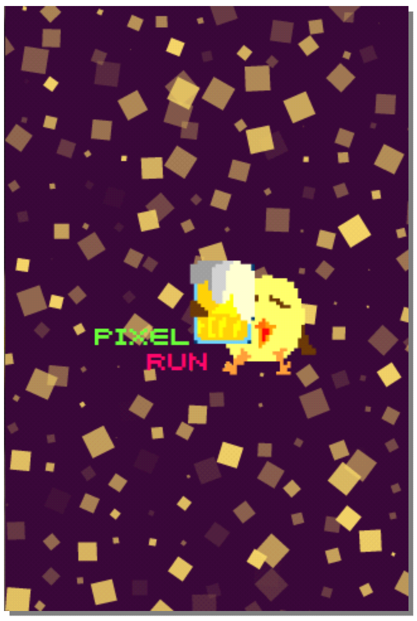


Figure 4.10 The start frame

4.5.2 Playing frame

The second frame is for the playing frame. This is the main frame of the game. It has the biggest number of layers in order to display all the elements which include the roles, items, buildings, road, score, timeline and the sky. Figure 4.12 shows the layers of this frame.

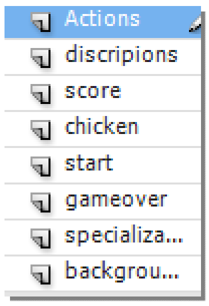


Figure 4.12 The layers

Also, the playing frame contains the main ActionScript code. The code mainly include the function of showing the buildings, showing the items, judgement of getting item and judgement of game over. Also all the elements are in the pixel style. I will cover in more details of the code in next chapter. The screen of playing frame is shown in Figure 4.13.

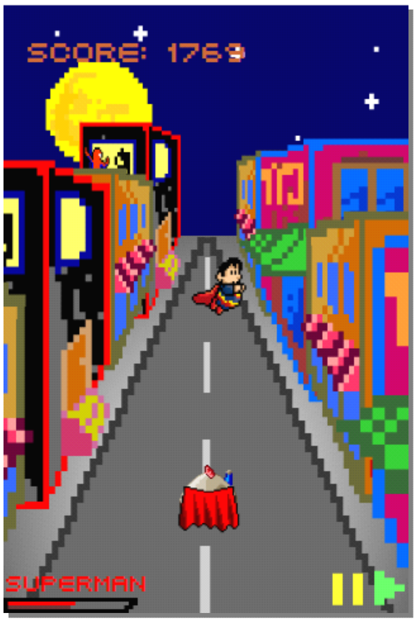


Figure 4.13 The playing frame

4.5.3 Game Over frame

The Game over frame takes the frame 3,it appears when the chicken touches the buildings at either the left or right side. It means this time of game is over. The game over frame contain the current score when the chicken die and the content of "GAME OVER" in a big text field. Also, of course we have the most useful buttons in this frame which are replay and quit. When the player touches the replay button, the game will go back to the frame 2 - playing frame. See Figure 4.14.

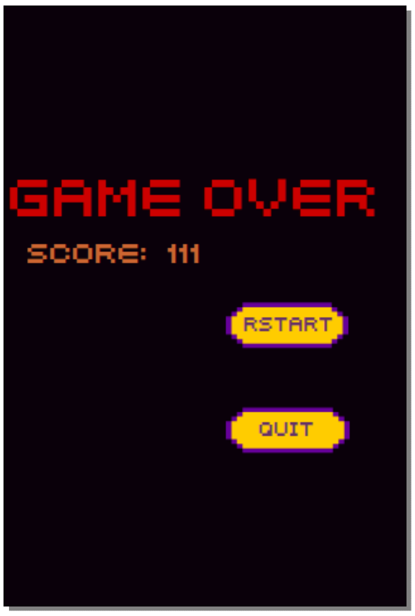


Figure 4.13 The game over fraem

4.5.4 Setting frame

I took the frame 4 as the setting frame. To implement showing this frame, I wrote the script in the start frame that when touch the screen, go to frame 4. A setting frame is a common and necessary part of a game. Typically, it contains for example buttons for starting, stage choosing, help, music switch, about game and so on. For my application, the setting frame has the start game button, music switch, and about me. And for those buttons, I also used the pixel button frame and pixel font to unified the style of the whole game (See Figure 4.11).

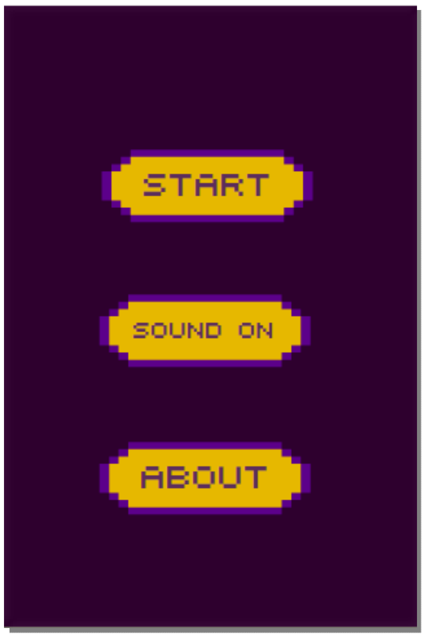


Figure 4.11 setting frame

5. IMPLEMENTATIONS

5.1 Main Components and MovieClips

As I introduced in previous chapter. A Flash application consists of components and MovieClips. So in this part, I will introduce the main components and MovieClips and then explain what are their duties and how they work.

5.1.1 Buffs

Buffs is a MovieClip which contain all the items' icons. Let us see Figure 5.1 first.

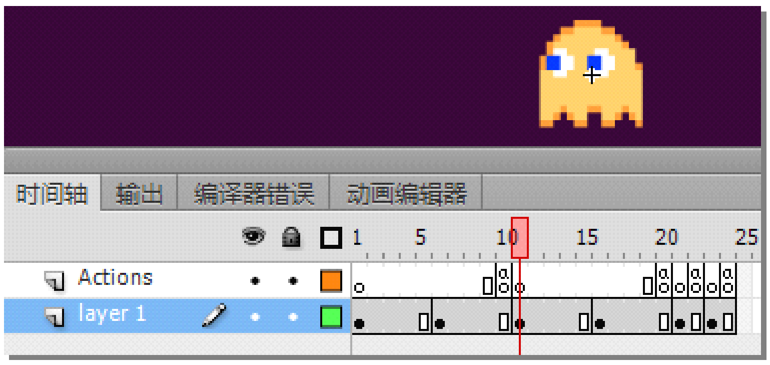


Figure 5.1 Buffs

As we can see from Figure 5.1. There are 24 frames in this MovieClip. Frames 1-10 are Superman, frames 11-20 are Ghost, 21 and 22 are Bread then the last two for Milk. The reason why I draw them in one MovieClip is that in this way, I have no need to put four individual components in the stage. So this method saves the random access memory.

Normally, when you put a MovieClip on the stage, it will play in a continuous loop. That is not what I want. Because the purpose is that I want it only to play the part of the Superman when I want it show the animation of the Superman. And that is why I used the Actions layer.

This is a common way in several MovieClips in my application. We can see from the Figure 5.1, there are ActionScript commands at 10, 20, 22 and 24 frames. I used gotoAndPlay(); command in those frame. The result is that when the MovieClip plays to those frames, it will follow the command gotoAndPlay();. For example at frame 10, I wrote gotoAndPlay(1);. So that it will return to frame 1 when it at frame 10. Also the same idea in the other parts.

So the idea is that I use the gotoAndPlay(); command to control which icon I want to show in the stage.

5.1.2 Buildings

Building MovieClip has the same idea as Buffs. It contains all kinds of building that will be showed in the game. The building icons are showed in Figure 4.3.

5.1.3 The components of roles

There are three different role components in this game. They are chicken, ghost and superman. Each references one of the special modes.

These three role are all always in the stage. However, I used the name.visible = true; and name.visible = false; to control which one is shown on the screen.

I made a MovieClip type value for the roles by using var op:MovieClip; command. Then I can use op = role's name; command to easily change the object to control.

5.1.4 Movingbg

Movingbg is the MovieClip of showing the buildings. I wrote the ActionScript code at the first frame to make a random number. Then use switch(the random number) to achieve randomly showing the buildings. In addition, I defined two value temp1 and temp2 to store the last two numbers in order to prevent that the current building is to be the same as either of the previous two buildings.

5.1.5 Movingbuffs

This MovieClip has all the rights on showing the items on the street. It randomly choose left or right and what item will be showed in the game. Just like in the movingbg, I also used random number and switch to implement this function.

5.1.6 Timeblood

Timeblood actually is used for the timeline of each special mode. See Figure 5.2.

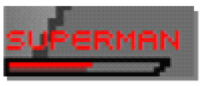


Figure 5.2 Timeline

5.2 Main Values

5.2.1 Timer class

The timer class is for measuring the time. A value of this class should be defined in the following format:
Var value:Timer = new Timer (unit time);
Since we defined the value of timer, we can add a listener of the timer:
value.addEventListener(TimerEvent.TIMER,function name);
The listener will be triggered once per unit time. As a result, the function will be executed in each unit time. We can add the code that we want in the function. However, the timer may not be exactly accurate. It depends on the performance of the CPU and the complexity of the code which have to be executed in the function.

The following two value are the value I used in this game. I will explain what are their duties.

time:
The unit time of this value is 100 ms. This time is for the functions which do not need to be executed very frequently. The functions bellow to this timer include the score counting, showing items, modifying difficulty value, game over judgement and getting items judgement.

time2:
This timer has a small unit time as 20 ms. The only responsibility is to refresh the position of the current role. The 20 ms unit time makes the roles moving smoothly. It equivalentis equal to 50 frames per second. It's well known that the standard of smooth for animation and game is 24 frames per second. However practically in a game, the frame rates higher than 40 frames per second can guarantee a smooth operations.

5.2.2 Numbers and Ints

Number and int are the most common types of value that we use in ActionScript language. In order to reduce the use of CPU and RAM, normally we should firstly consider int type. If an integer can not content the requirement of a value, then we secondly consider type of Number.

The first I will introduce the int values.

score: for storing the current score of this game. In each 100 ms, this value is refreshed and showed on the screen by using txt.text = "SCORE: " + score.toString(); command.

Stime: to store the duration time of each special mode. This number is just for counting, so integer is enough.

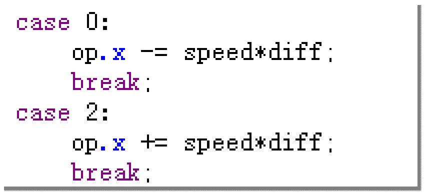
inmode: This value is for the game to identify the special mode of the role. Because different modes have different ways to control, I use switch (inmode) to choose which control we use for the current specific mode.

rd, temp1, temp2, lr, yyf: All these values have the similar function. So explain them together. They all use for temporarily store the random values to show the buildings and items randomly. Temp1 and temp2, as I said before, they are used to record the last two building number. So that I can make the building shows not repeatedly.

And secondly, the Number values.

acctemp: acctemp is for recording the current x direction parameter. The parameter references the x direction acceleration. The value is between 0-1 therefore I have to use Number type to store this value.

diff: diff is the value of the difficulty.



```
case 0:
    op.x -= speed*diff;
    break;
case 2:
    op.x += speed*diff;
    break;
```

Figure 5.3 The code of movement

Figure 5.3 shows a part of code of the game. This duty of these commands is to control the role. The either od these two command will be executed once per unit time. Therefore, when the diff increase, the distance that the role move will also increase. The initial of diff is 0.25. It increases by 0.1 per 150 frames.

speed: speed references the current speed of the role. In physics, we use the following formula to calculate the speed. The value v,a and t represent velocity, acceleration and time.

$$v = at$$

However, I have tried this formula in my game but it was not working correctly. The reason is this formula is for the situation that the acceleration is static. On the opposite, the acceleration in this game3 is changing all the time while the player is moving the mobile phone. Then I chose the next expression to fix this problem.

$$v = \sum_{i=0}^{i\to\infty} a_i$$

"a" in this formula references the acceleration in each unit time. The unit time of my game here is 20 ms. So then I can use the command "speed += acctemp;" to calculate the accurate speed.

5.2.3 Accelerator Class

I have only one accelerator type value in this game - accl. Obviously, this value is for recording the current accelerator parameters. The class of accelerator contains three parameters. They are x, y and z. These parameters change when the position of a mobile phone changes. However, we only use the parameter x in this game because the chicken can only move left or right. In addition, just like the timer, the value of the accelerator also needs a listener to execute the commands.

I use accl.addEventListener(AccelerometerEvent.UPDATE, updateHandler); to add a listener. Since we have the listener, we can add the commands in the function "updateHandler". Each time the accelerator parameters change, the updateHandler will be executed.

5.2.4 MovieClip Class

MovieClip Class is a specific class in Flash. And it is different from the other type of values. A member of MovieClip in Adobe Flash CS6 has no need to be defined before we use it. More accurately, the Flash software will automatically define it when we put a MovieClip on the stage and give it a instance name. For example in Figure 5.3.

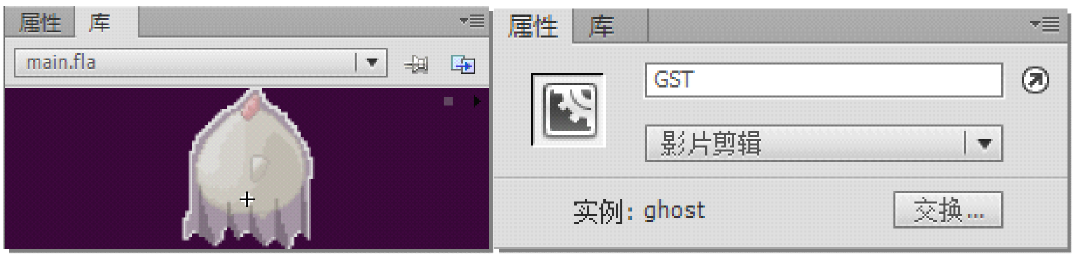
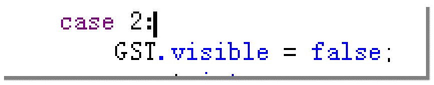


Figure 5.3 Instance name

Figure 5.3 shows the instance name of the ghost MovieClip. The instance name as we can see is GST. So then let us see the code.



```
case 2:
    GST.visible = false;
```

Figure 5.4 instance name

See in Figure 5.4. We can use GST. Sth to visit the value of GST movie clip. It means the GST MovieClip has already be defined.

The idea of all MovieClips are similar. So I will not talk about the other MovieClip members here. However, we should know that all Flash games consists of Movieclips and components.

5.2.5 Music Class

We can add music class member to play the music which in the same folder of the game. And when we export the program as an app file, we can include the music files. In this way we are able to guarantee that we the game can still find the music after we installed it in a real mobile phone.

When initializing the music value we have to give the path of the location of the music file.

For our mobile phone game the path should be only the name of music file because the music files are in the same location as the program. Then we can use commands such as "play" to control the music.

5.3 Main Functions

The functions also are called subroutines. It means a sequence of program instructions that perform a specific task, packaged as a unit. This unit can then be used in programs wherever that particular task should be performed. Subprograms may be defined within programs, or separately in libraries that can be used by multiple programs [17].

In this part, I will explain the principles and responsibilities of the main functions in this game. The functions in ActionScript 3.0 have no need to be declared in the beginning of the code page.

function superman():void

This function will be executed when the player collect the superman buff. The mode will be changed into superman mode. Concretely, the function will do the following things:

● Set the duration time and reset the counter.

● Set the "inmode" value as 1 to tell the program which mode the role is.

● Show the superman role and make the original chicken role invisible. The superman role will be showed in the middle place of the screen.

● Change the current object to the superman role.

● Show a text description on the left bottom corner of the screen.

● Set the boolean value "positive" to false. Which means the game will not able to control this role and in addition, it will also skip the judgement of game over in the duration of superman mode. As s result, in the superman mode, we can not move or die.

● Add 500 score point.

function ghost():void

This function will be executed when the player collect the ghost buff. It can change the mode into ghost mode. The concrete effect of this function are:

● Set the duration time and reset the counter.

● Set the "inmode" value as 2 to tell the program which mode the role is. Then the showtime2 function will change the way to control the current object. As I introduced before, the moving direction will be inversed.

● Reset the current speed. I need to explain a little for the reason why I write this command here. There is a situation that before the player get the ghost buff, he already has a very high moving speed. So if I do not reset the speed, the role will suddenly change the moving direction and move with the same high speed. It will be too difficult to control.

● Record the current positon and change the current object to the ghost role. Not like the superman mode, the ghost role will not be shown in the middle of the screen but in the current position of the the chicken role. So we need to record the original current position and then use this position on the ghost role.

● Disable the current role and show the ghost role.

● Show a text description on the left bottom corner of the screen.

function bread():void

This function will change the role into conscious mode after eat a bread in the game. The conscious mode only changes the way to control the role. So we do not have to change the current object to any specific role.

● If the current role is not the normally chicken, the bread will be canceled. In another word, the players can not eat a bread when they are in the special modes.

● Reset the counter and set the duration time.

● Set the “inmode” value to 3. It means the game will use the specific way to control the role. This is also the key of the conscious mode. In this way, the acceleration will be disabled while the role are moving. It will be much easier to control in this conscious mode.

● Add 200 scores.

function milk():void

The only function of milk is reset the diff value. It will reduce the speed of movement.

function quitmode():void

Every mode has a duration time. When time is out, the chicken will move from the special modes back to chicken mode. The quitmode function is made for this. There is a “switch (inmode)” statement in this function to identify which mode the role will quit from. The common idea is to hide the specific role and show up the original chicken role. Then erase the description of those modes.

function showtime2(evt:TimerEvent):void
showtime2 is the function which executed by the listener of time2, a member of timer class. It will be executed every 20 ms. Because the high refresh rate, this function consumes a lot of CPU resources. Therefore, we should write as little codes as we can in this function. So the only duty I gave to this function is to control the movement of the roles. There will be only two statements executed each time.

function showtime(evt:TimerEvent):void
showtime is the most important function in this game. It belongs to the listener of "time" value. In the earlier version of the game, showtime and showtime 2 were not two individual functions. Also there were only one listener. However, with the game getting bigger and adding more and more functions I found out that the performance of the game suddenly reduced. After I searched the method that how to improve the performance I found that the problem is in the showtime function.

In order to make the game smooth, I set 20ms as the unit time of the listener. However, when the program got more complicated. There were too many statements need to be executed in each 20 ms. So I made the second listener. This showtime contains the functions that have no need to be executed very frequently. The unit time of this listener is 100 ms.

In the following part, I will introduce every function individually.

The first function is to display the building continuously. It seems like the buildings are shown one by one in the game while we are playing it. However the principle is not like this. Actually, each three continuous building are from the same MovieClip. We have two similar building MovieClips in this game. To explain the reason why we implement two of them, we need to see the Figure 5.5.

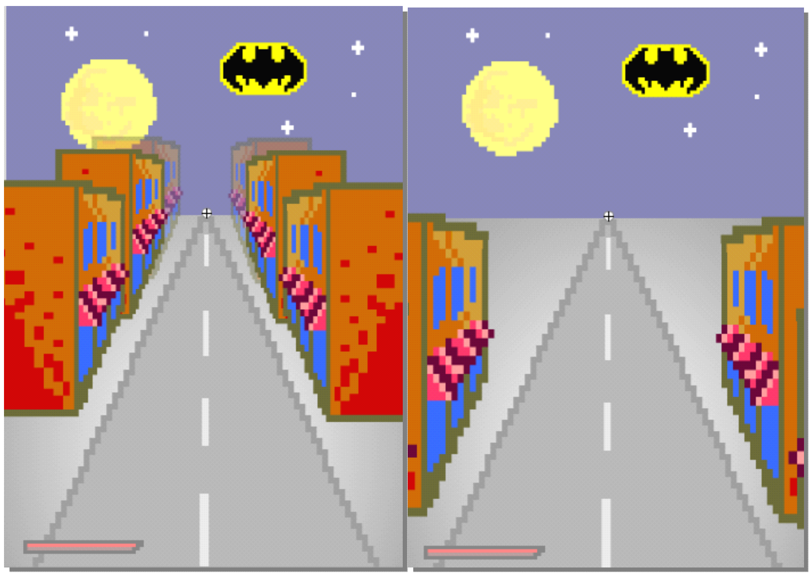


Figure 5.5 The movingbg MovieClip

The two screenshots are token from movingbg MovieClip (because I played this MovieClip without start the whole program so the buildings are not shown randomly). As we can see, each MovieClip contains only three buildings. In the first frame the buildings are at the far end of the street and in the last frame the third building has been passed and is no longer visible. It means the building will not be displayed continuously with only one MovieClip. To solve this problem, I used two overlapping movingbg MovieClip on the stage. Therefor we can create an illusion where the street never ends but new buildings keep appearing all the time.

The principle to implement this function is that we play the next MovieClip when the current frame of the previous one is 150. And then switch in the next round.

The same idea also have used in the movingbuffs MovieClip. This is for displaying the items in the game. This is the second function of showtime.

The third function is to judge the situation of game over. The rule is that the position of current role has to be between 20 and 300 (the total of horizontal pixels of this game is 320). This judgement will be executed every 100 ms while the boolean value “positive” is true. So for example in superman mode, the “positive” is false. Then the role can skip this judgement in the duration time to achieve the undead ability of superman mode. If the judgement statement of game over is true, then stop everything in this frame and turn to the frame three -

game over frame.

The last function of showtime is the judgement of getting items. The principle of this function is to judge whether the role is in the specific range of horizontal positions while the frame of the movingbuffs is between 146 to 161. Let us see Figure 5.6.

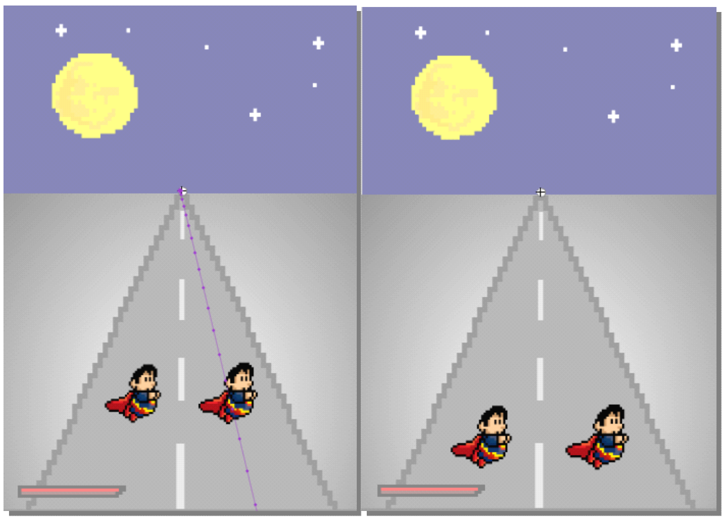


Figure 5.6 Frame 146 and 161 in movingbuffs MovieClip

If the judgement returns true, then the program will use L1.yyf (L1 is the instance name of movingbuffs and yyf is the value to store the random number of the buffs) to get the value of yyf. Once it gets the value of yyf, the program knows which item has displayed and which mode the role should change to.

function updateHandler(evt:AccelerometerEvent):void

This function belongs to the listener of accelerator. The effect of this function is to rotate the role. The role will be rotated up to 45° by turning the mobile phone. Figure 5.7 shows the practical effect.

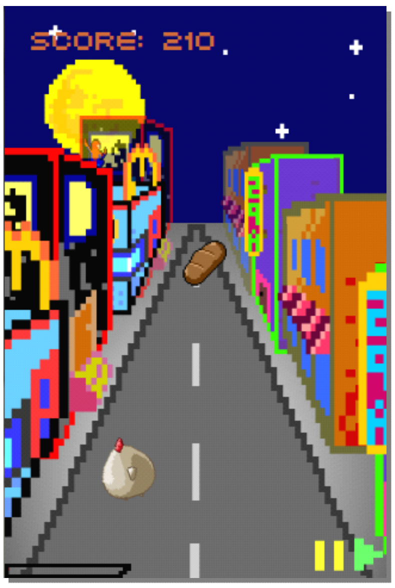


Figure 5.7 Rotation

In addition to these main functions there are two simple functions which are:

function fl_GenerateRandomint(limit:int):int

To make a random number.

function fl_FadeSymbolIn(event:Event)

To display the random squares in the start frame.

6. PROGRAM TESTS

After implementing the program was time to test it. I tested the program in the two environments. First using the simulator and second the real phone.

6.1 Simulator Test

The testing environment I used for simulator test was the same one that I used for development. The description of used hardware is shown in Table 1.

Table 1:

Hardware	Laptop: Alienware m11x R3 CPU: Intel i7 2617 Memory: 8GB Video card: NVIDIA GT 540m
Software	Adobe Flash CS 6 Adobe AIR 3.2

Figure 6.1 shows the interface of the simulator test. For the right part, there is a individual window for simulating the screen of mobile phones. And the left part is the controller. You can either drag the mobile phone in the screen or input values directly to control the accelerator. You can also simulate tapping by clicking the “TOUCH AND GESTURE” button.

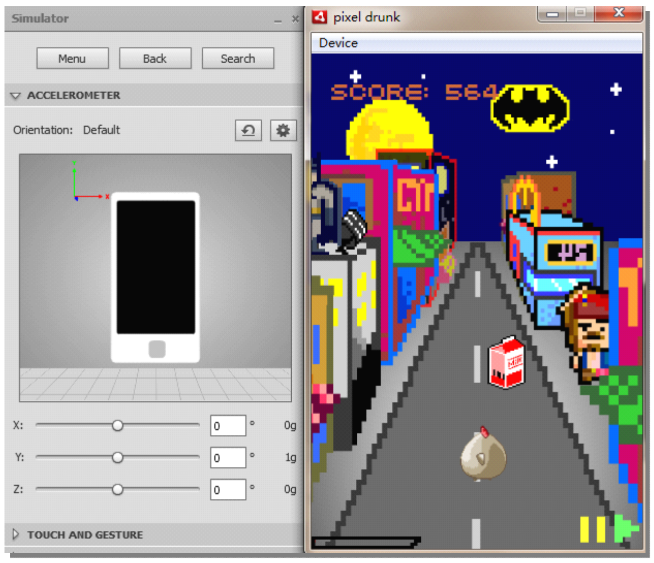


Figure 6.1 The interface of simulator

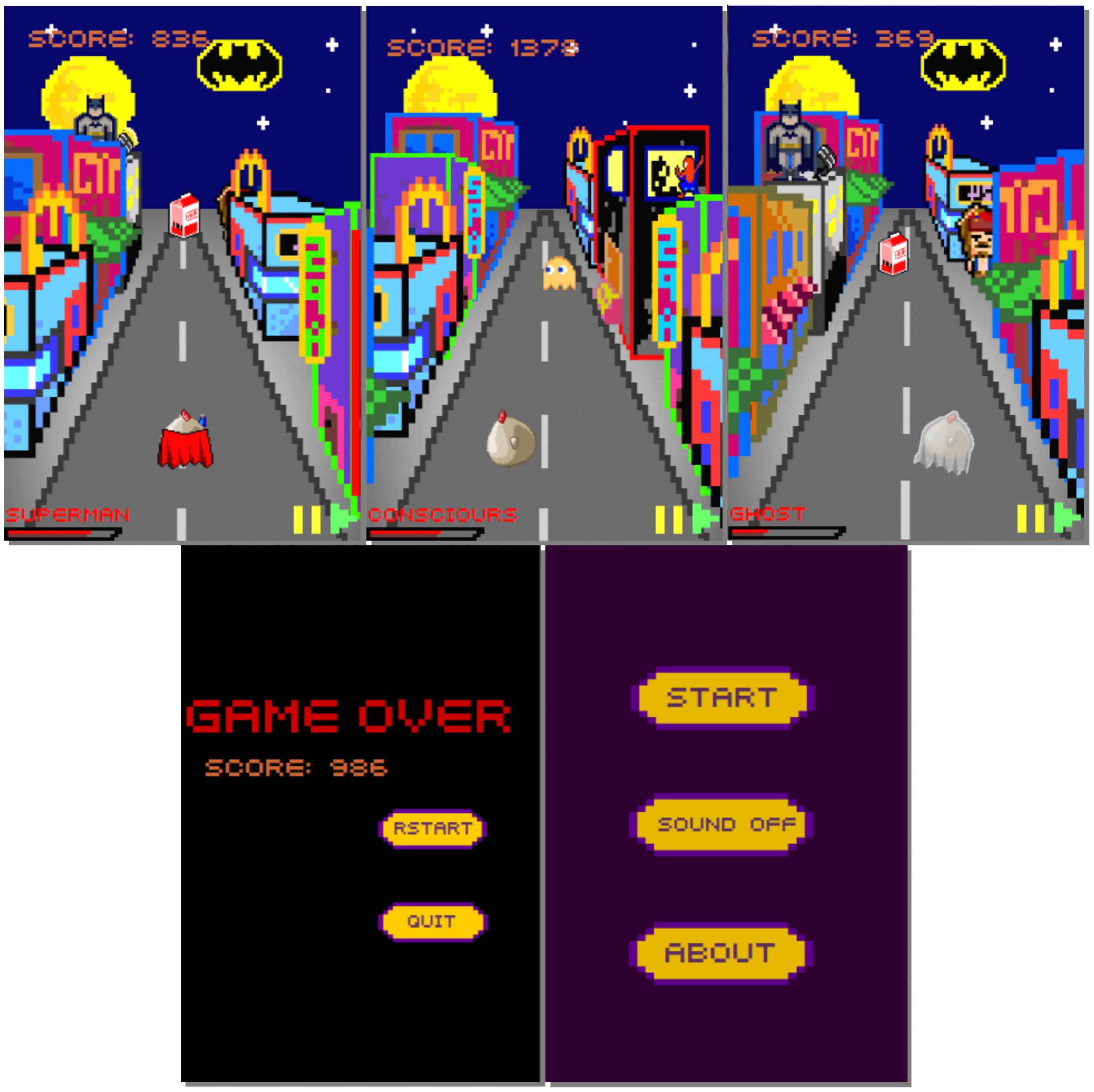


Figure 6.2 Results of the test

I tested the following functions:

●Enter the superman mode.

●Enter the ghost mode.

●Enter the conscious mode.

●The effect of milk

●Sound switch.

●The judgement of game over.

●Pause.

●Restart.

●Quit game.

As we can see from Figure 6.2. We can enter the special modes successfully as well as quit those modes. The other functions can not be shown in screenshots. However all the functions were achieved in the game.The game runs properly, all the functions are implemented. The simulator test is successful.

6.2 Practical Mobile Phone Test

The aim of my project was to make a application which can run on a real mobile phone. Therefor the practical mobile phone test is necessary. The information of the testing device is shown in Table 2.

Table 2:

Hardware	Mobile phone: Galaxy S plus CPU: MSM8225T (1.4GHz) Memory: 512 MB
Platform	Android 2.3

I tested the same functions as in simulator test. There are two functions that I need to pay more attentions on mobile phone. they are music switch and quit game. Because the path of assistant files in mobile phone are not the same as in PC, so I should try if the music file is valid. And for the function of quit game, I need to check whether the game still runs in the background after I quit it. Figure 6.2 shows the game while it is running on the real mobile phone.

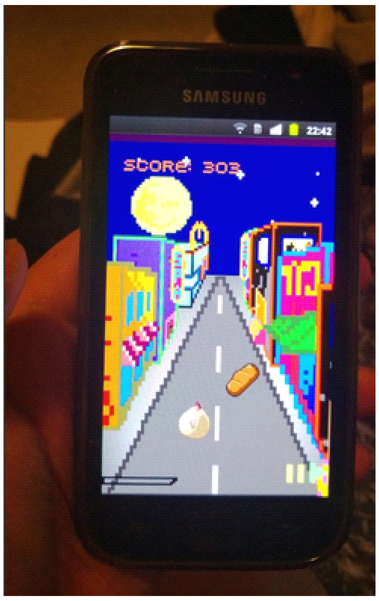
I 

Figure 6.2 The game running on a real mobile phone

The result is that all the functions were working correctly. No problem on playing music and the game does not run in the background after I quit it. In addition, the game displayed smoothly on the testing device even though the performance of mobile phone is not as good as the performance of computer.

7. CONCLUSIONS

The aim of my final thesis is to develop an application for Android platform and test it on an actual smart phone. As a result, I developed the game which is called Pixel Drunk Chicken for Android platform. And the result of the test proves that this game can successfully run on an actual smart phone. Therefor, the aim is achieved.

Before I choose this topic, I knew neither about Android application programming nor ActionScript language. So that I took very long time to practice and learn how to develop an Android application with ActionScript. Because Flash is a new way to develop Android applications so that there were not a lot of available documents and informations that I can utilize. There were a lot of errors and mistakes that I had to test and correct them by myself. However, after I overcame and finished this game, I really learned a lot of knowledge.

According to the limitation of the time. I am not able to implement more various of functions in my game. Even though this game has met the requirements of the project and it looks not bad there are still somethings that are not perfect in this game.

The first is about optimization. I have improve the performance of the game by reducing the use of CPU. However, I did not try to reduce the use of memory. A good program should use as less as memory it can even though the result shows that the memory of the mobile phone is enough to this game.

The second problem is resource reclamation. This problem happens because of a defect of ActionScript. There is no efficient way to completely delete a MovieClip and reclaim the memory they have used. So the result is that the game will use more memory when the player replay the game a lot of times. Because each time the game starts, the elements in this game ask for memory but never return the memory back.

The last is about the content of the game. I want to make more interesting mode and some special items in this game in the future.

[1] Flurry Analytics - Time Spent Per Application Category [Referred 13/2/2013]
<URL:http://news.xinhuanet.com/info/2013-01/30/c_132137699.htm >

[2] The real time report. [Referred 14/2/2013]
<URL:http://therealtimereport.com/2012/06/12/mobile-gaming-to-rise-26-social-gaming-up-10-in-2012/ >

[3] Wikipedia - Snake. [Referred 14/2/2013]
<URL:http://en.wikipedia.org/wiki/Snake_(video_game) >

[4] Techcn - The history of mobile gaming [Referred 14/2/2013]
<URL: http://www.techcn.com.cn/index.php?doc-view-133500.html >

[5] Wikipedia - WAP [Referred 15/2/2013]
<URL:http://en.wikipedia.org/wiki/Wireless_Application_Protocol >

[6] Wikipedia - N-Gage [Referred 26/3/2013]
<URL:http://en.wikipedia.org/wiki/N-Gage_(device) >

[7] Symbian^3 game -Spiderman [Referred 26/3/2013]
<URL: http://www.sjvip.com/symbian/game/47035.html >

[8] Asphalt 7 [Referred 26/3/2013]
<URL http://androidxgame.blogspot.fi/2012/08/asphalt-7heat-apk-data-qvga-hvga-wvga.html >

[9] Adobe [Referred 26/4/2013]
<URL:https://en.wikipedia.org/wiki/Adobe_Systems >

[10] Adobe Flash [Referred 26/4/2013]
<URL:https://en.wikipedia.org/wiki/Adobe_flash#History >

[11] BestFlashGams [Referred 26/4/2013]
<URL:http://www.bestflashgames.com/ >

[12] Adobe AIR [Referred 26/4/2013]
<URL:http://www.adobe.com/products/air.html?promoid=DINNY >

[13] Developer preferences [Referred 26/4/2013]
<URL:http://www.artengineer.cn/bbs/forum.php?mod=viewthread&tid=32175 >

[14] Adobe AIR mobile simulator [Referred 27/4/2013]
<URL:http://www.adobe.com/products/flash/features.html >

[15] Code snippets [Referred 27/4/2013]
<URL:http://www.adobe.com/devnet/flash/articles/code_snippets_panel.html >

[16] ActionScript [Referred 27/4/2013]
<URL:http://help.adobe.com/en_US/as3/learn/WSf00ab63af761f170-43fa6dce12937d272e9-8000.html>

[17] Subroutine [Referred 4/5/2013]
<URL:http://en.wikipedia.org/wiki/Subroutine >