HKUST Independent Project CSIT 6910A

Spring 2014

Report

Fast Paced Strategy Mobile Game

Submitted By,

Huang Jiefeng

Email: jiefeng218@gmail.com

MSc. Information Technology (2013), HKUST

Supervised by,

Dr. David Rossiter

Department of Computer Science and Engineering, HKUST

Content

1. Background....................3

1.1 Why Game?....................3

1.2 Why on Mobile Phone?....................4

1.3 Why Fast paced strategy?....................4

2. Introduction....................5

2.1 Brief description....................5

2.2 Demonstration....................5

3. Game Design....................11

3.1 Hardware & Software Employed....................11

3.2 Flow Chart of Game....................11

3.3 Main Class Diagram....................12

3.4 Implementation of Fight States....................12

3.5 Other Details....................15

4. Conclusion and Future work....................16

5. Appendix....................17

1. Background

As a game enthusiast, I want to develop an Online Fast Paced Strategy Mobile Game as my independent project.

1.1 Why Game?

"Gaming can make a better world" – Jane McGonigal, TED Talk[1].

Though using one game to change the world is idealistic, Gaming can actually affect player imperceptibly and make a better world.

How big is the gaming audience?

"No other sector has experienced the same explosive growth as the computer and video game industry." — Michael D. Gallagher, president and CEO, Entertainment Software Association.

Actually, the total number of people around the globe who play games is expected to surpass 1.2 billion by the end of 2013[2]. Furthermore, the global games market is currently worth $70.4 billion and is expected to grow at 6% a year[3].

In China, there are about 490 million game players till 2013. The total game market income has reached $13.7 billion in 2013, increased by 38%

[1] http://www.youtube.com/watch?v=dE1DuBesGYM

[2] Newzoo: Global Games Market Report

[3] 2013 State Of Online Gaming Report

than 2012.

The game industry has great potential of development.

1.2 Why on Mobile Phone?

Portability is the most important feature of mobile phone, which make it more suitable for some simple games than PC. These kind of games aim for players' time fragments. What's more, with development of mobile phone industry and increasing speed of mobile network (4G is available now in some country, and 3G is becoming popularization), better special effect and more social interaction on mobile game become possible. All of these enhance the game experience and attract more and more people playing mobile phone game.

In China, the number of mobile phone user is approaching 1.15 billion, and it keep increasing. The number of mobile gamers in China is expected to hit 192 million by the end of 2012, according to market researcher Niko Partners. This means that the number of mobile gamers now likely exceeds the number of PC players in China.

Mobile Game is now becoming the hottest one, and I think this situation will keep for more than 3 years, especially in China.

1.3 Why Fast paced strategy?

In my opinion, mobile game should aim for mobile phone users' time fragments. So it should be fast paced, suitable for several minutes, even seconds. That also means the game should be simple with easy operations and short story, unlike traditional RPG game.

Indeed, the most popular mobile game are chess game, card game and casual puzzle game, nowadays. My inspiration comes from these kinds of games, and mostly from Hearthstone, a fast-paced strategy card game from Blizzard Entertainment. But it is still a little complex and monotonous.

This project will creates a simple and funnier mobile game.

2. Introduction

2.1 Brief description

This game is a turn-based fighting Game. At the beginning, the player have only one sprite. And he can use it to go through different chapters to fight with different monsters. He can also use skill to capture the monsters to make up a bigger team (change the fighting sprites in the edit scene). After winning the fight, all the sprites that have attended the fight, will get some experience (sprites will increase their level after getting enough exp.) and goods.

2.2 Demonstration

- Main Scene: The first scene after start the game, player can choose different part of the game. After click the start button, there are several chapters.

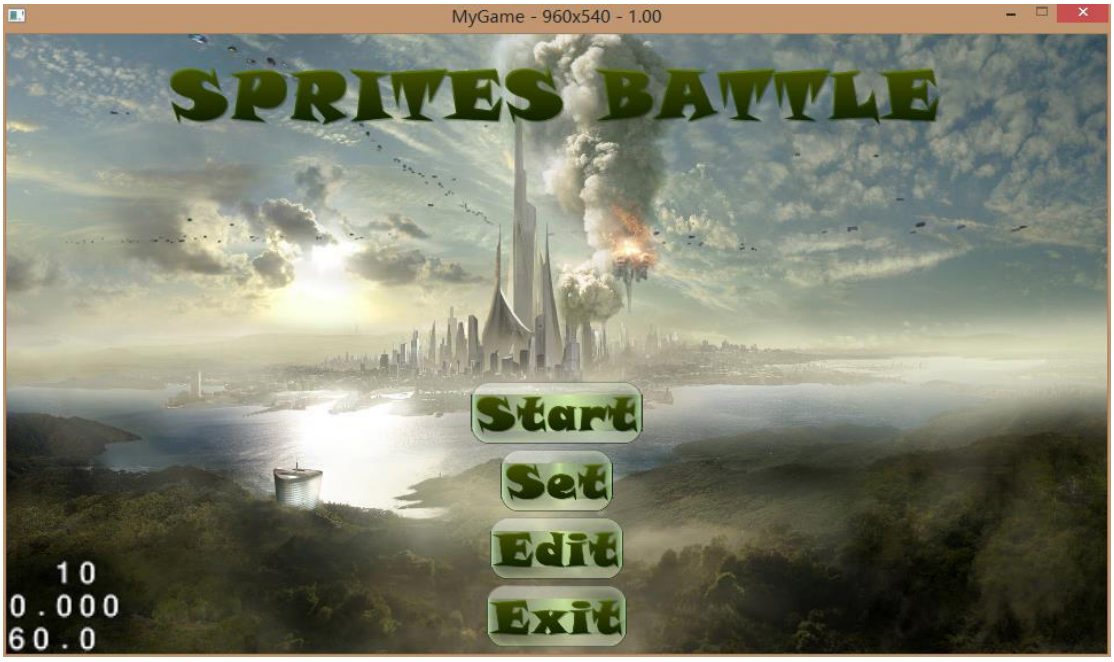

Fig.1 Main Scene

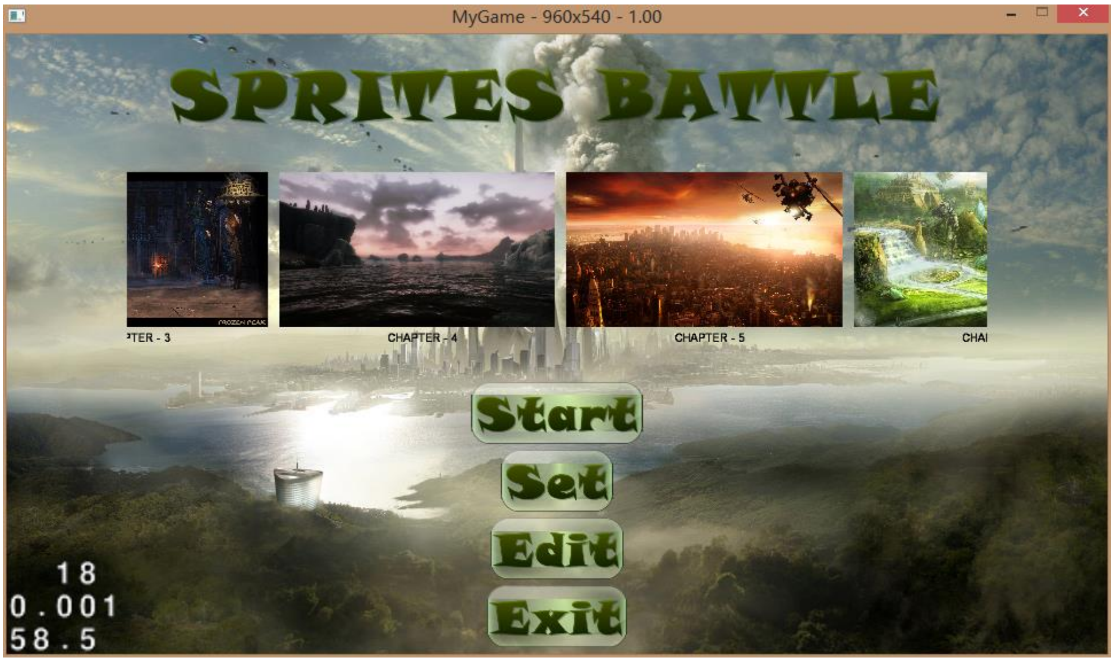

Fig.2 Choose Chapter

- Set Scene: Player can enable or disable the background music and sound effect, or change their volume.

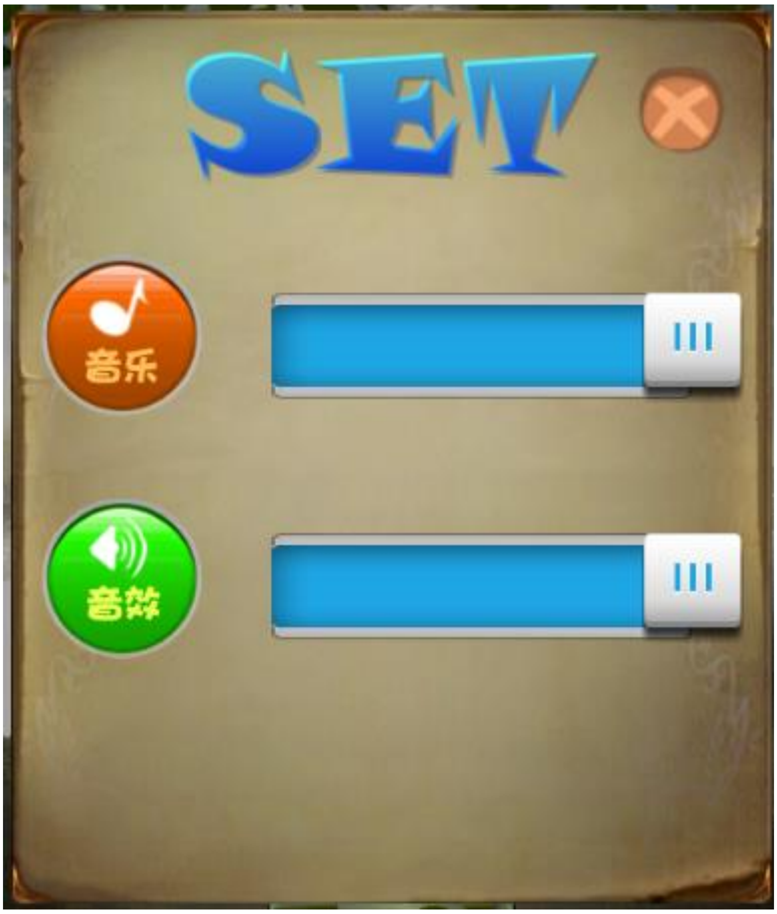

Fig.3 Set Scene

- Fight Scene: The most important part of the game.

  The monsters are on the left hand side, while the player’s sprites are on the right. There are 4 different commands: “attack”, “skill”, “goods” and “escape”.

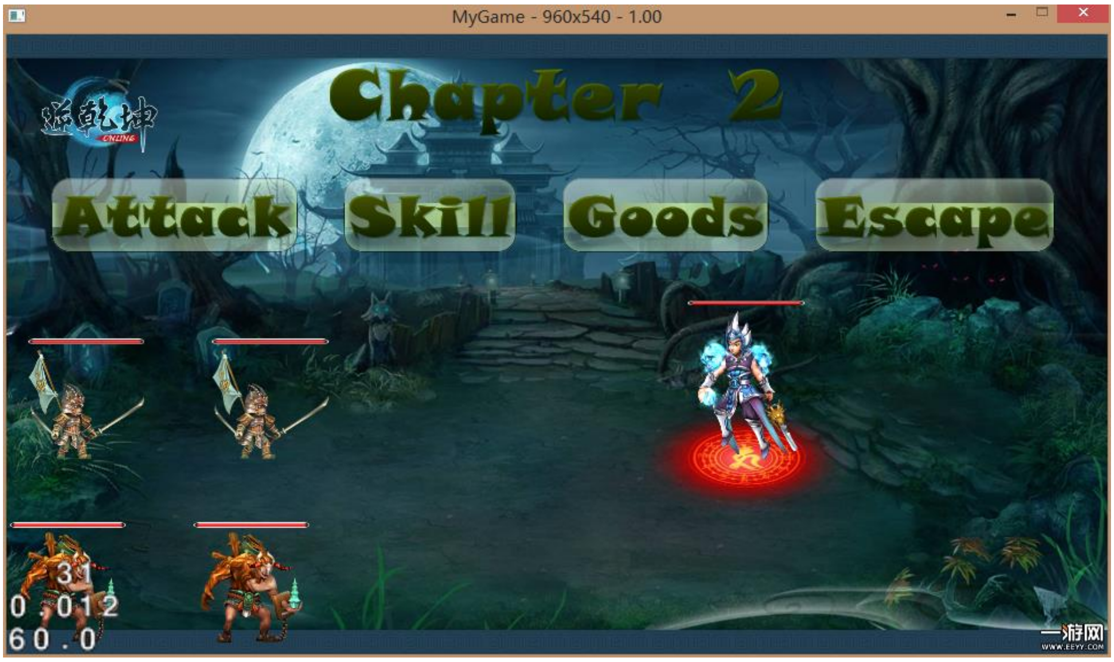

Fig.4 Fight Scene

Attack is the basic command, click on it and next click on one of the monsters, the sprite will attack it.

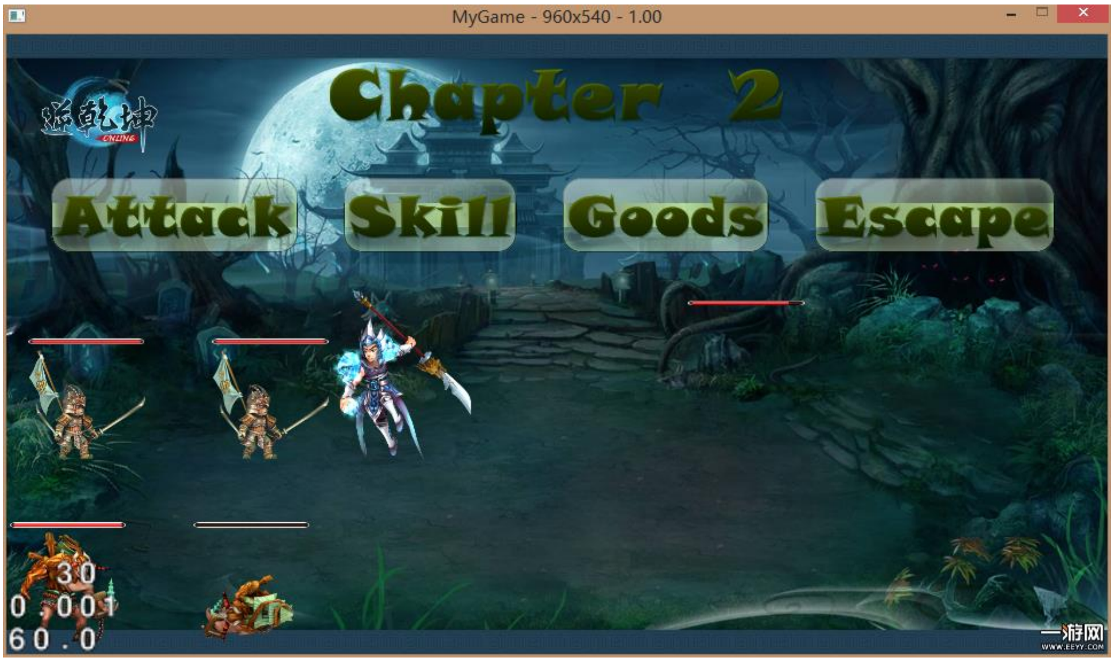

Fig.5 Sprite attack

After clicking on the “skill” and “goods” button, the skills or goods will appear. **(You can catch new sprites)**

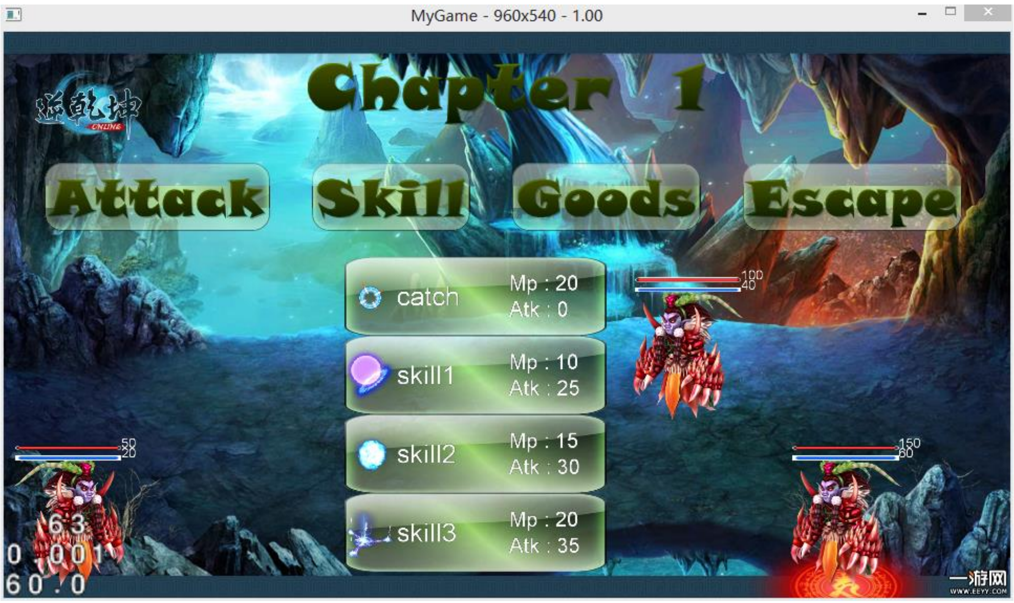

Fig.6 Skills

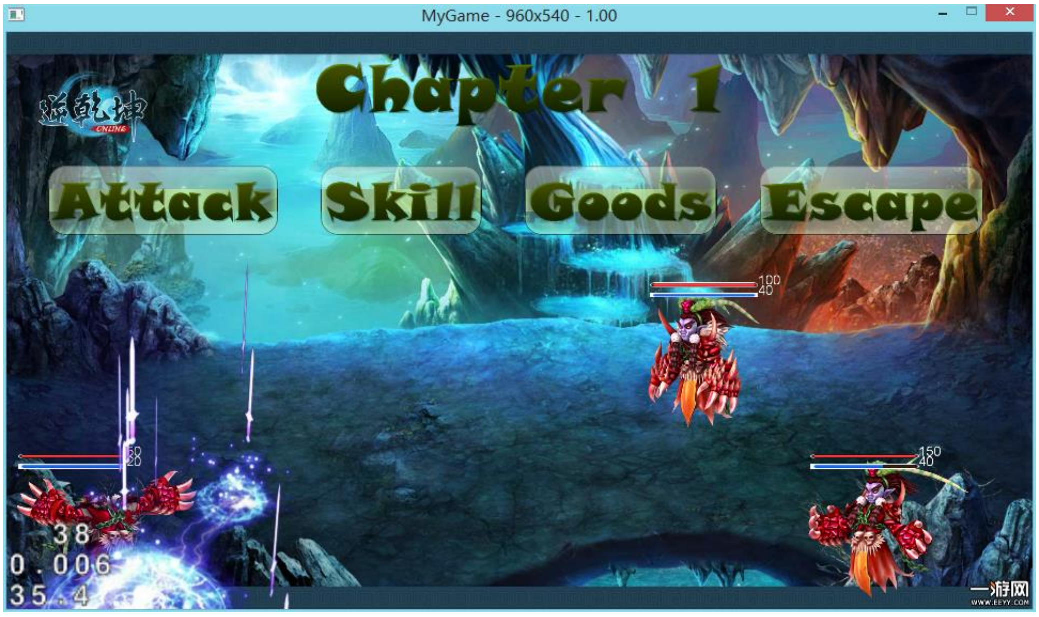

Fig.7 Skill effect

After victory, player can get some items, like HP (Item that can increase sprite’s healthy point) and MP (Item that can increase sprite’s magic point). Also his sprites, which participated in the battle, will get some EXP (Experience, which can be used to increase sprite’s level).

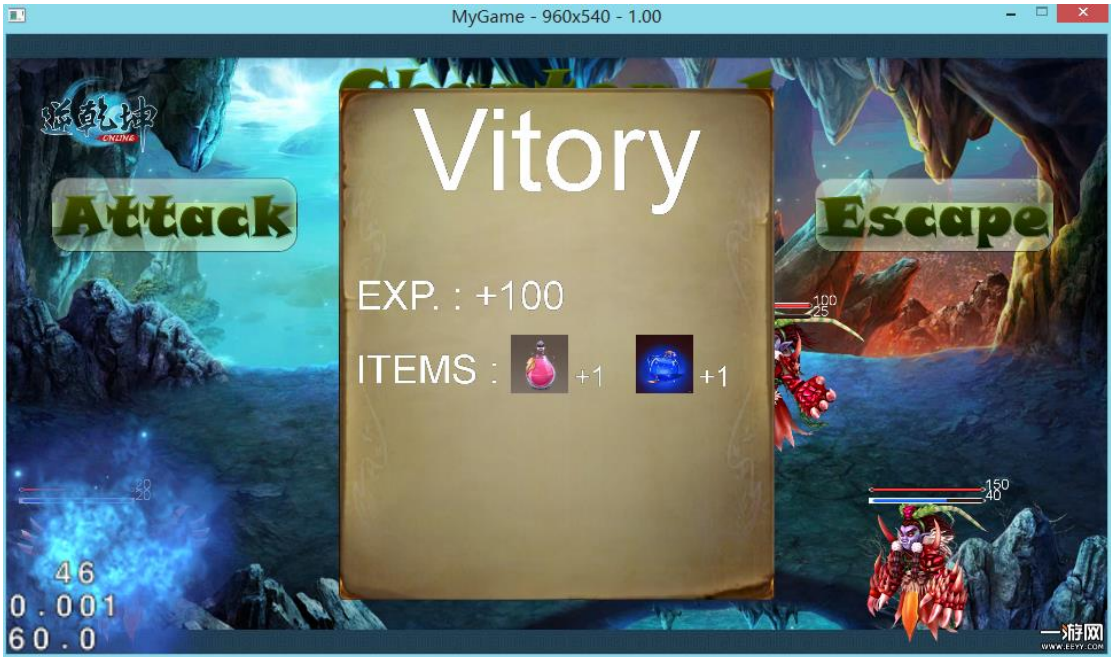

Fig. 8 Victory

- Edit Scene: In this scene, player can check his own sprites and make different formation. The maximum number of sprites that can attend in battle is 4.

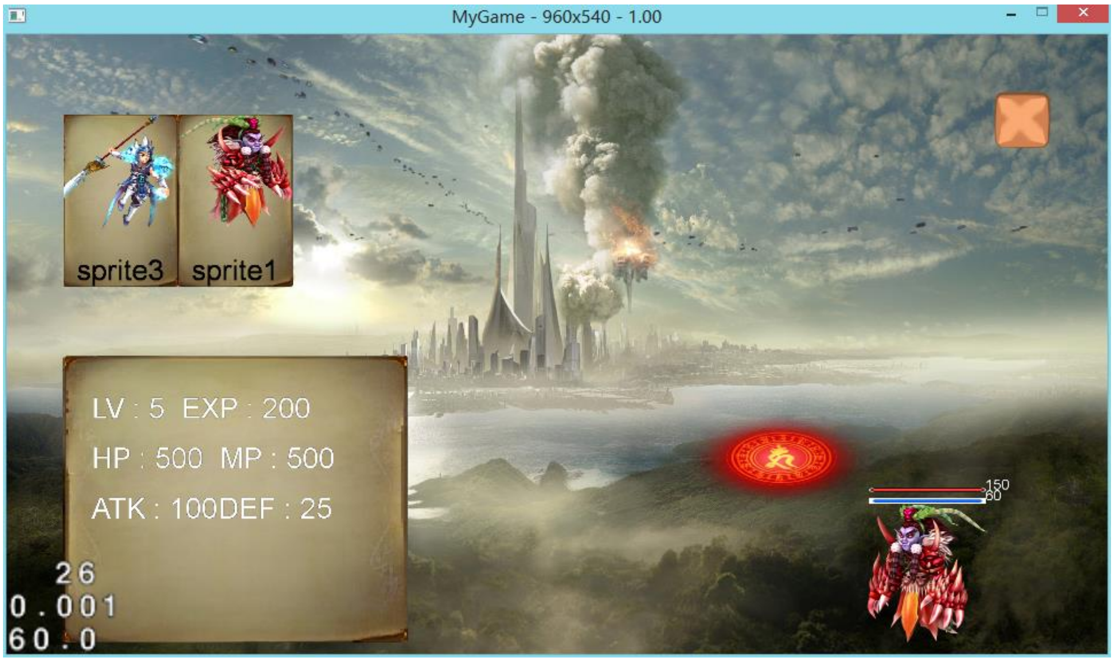

Fig. 9 Edit Scene

3. Game Design

3.1 Hardware & Software Employed

- Platform: Android
- Game Engine: Cocos2dx (version2.2.1)
- Development environment: VisualStudio2012, CocoStudio (version 1.1.0.0), JDK&JRE, Eclipse with ADT, NDK
- Develop language: C++

3.2 Flow Chart of Game

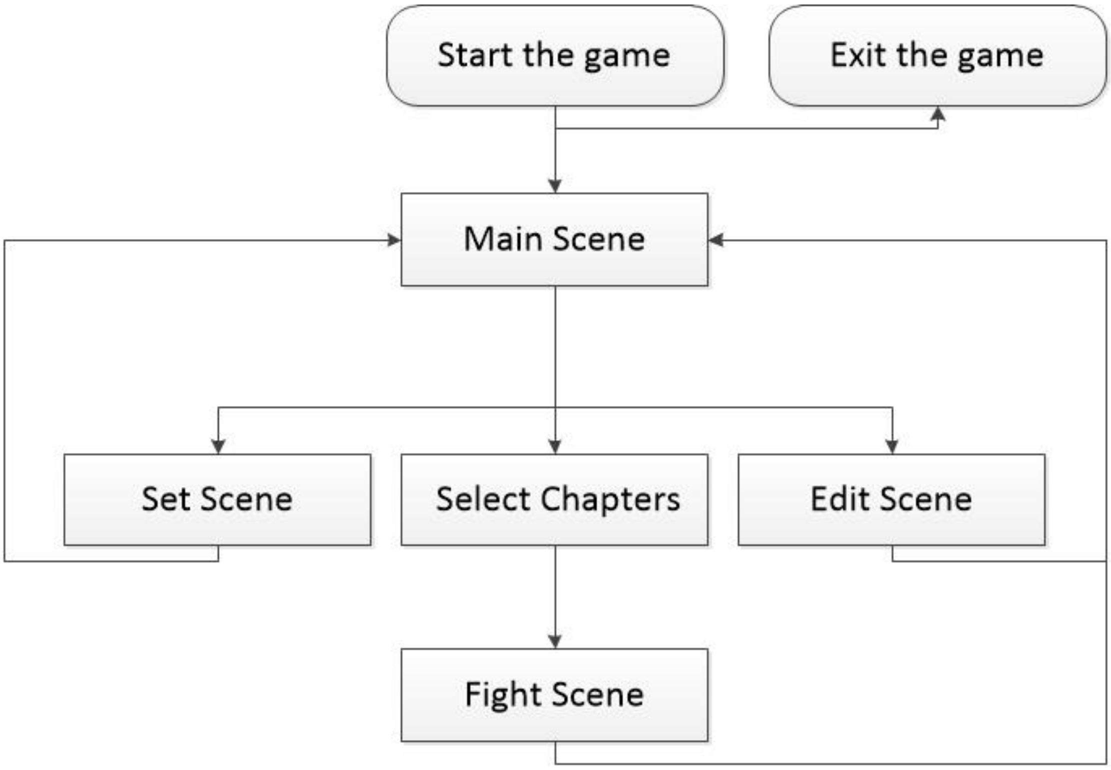

Fig.10 Flow Chart

3.3 Main Class Diagram

Fig.11 shows the dependency of the main classes. (The simple version, taking Fighting Scene as example)

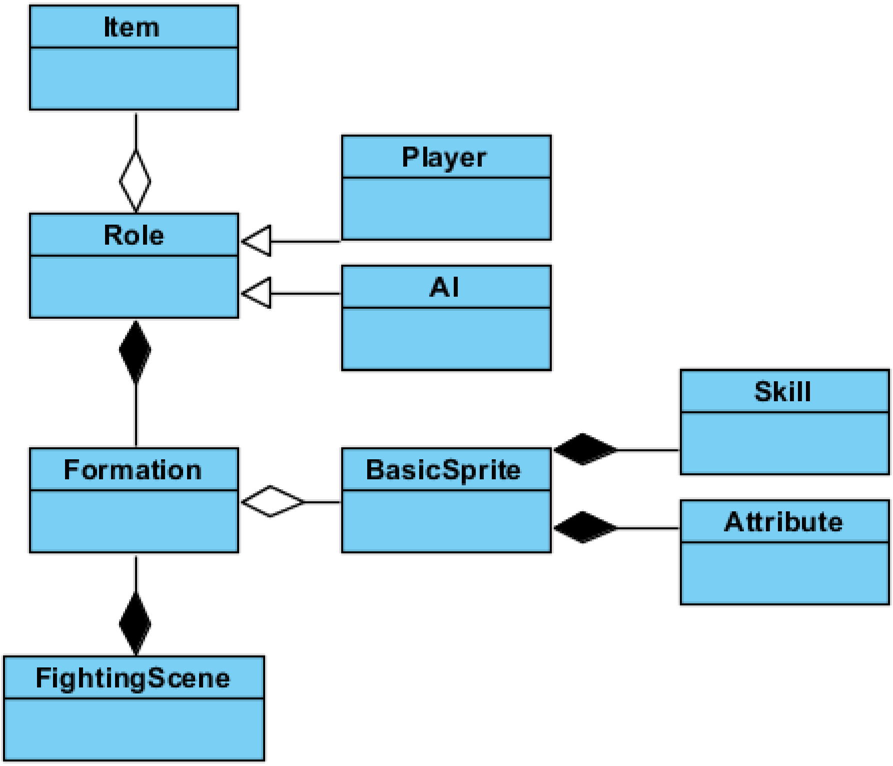

Fig.11 Main class diagram

3.4 Implementation of Fight States

The most important part in this game are the logic states of sprites. There are many states of each sprite, and the states of them are almost the same. Considering these characteristics, they can be implement as finite-state machine. The state diagram of sprite is like this:

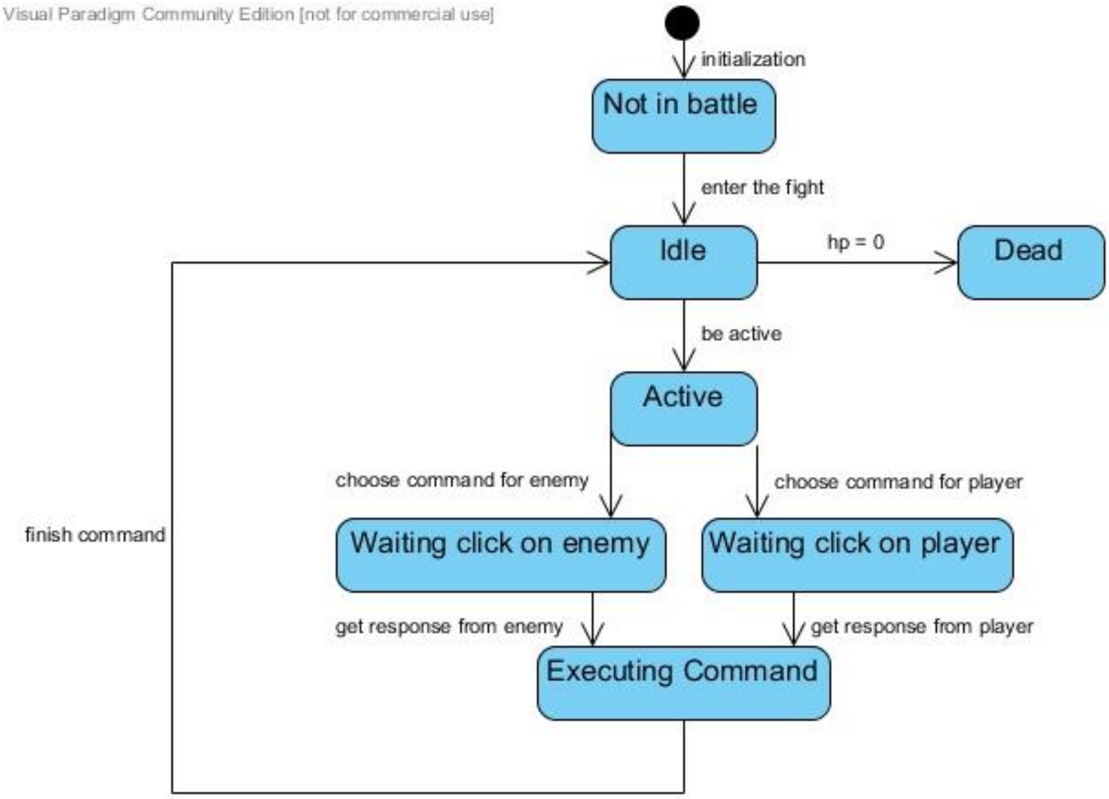

Fig.12 state diagram

This is implemented by using **State Pattern** and **Observer Pattern.** Fig. 13 show the class diagram of **State Pattern.**

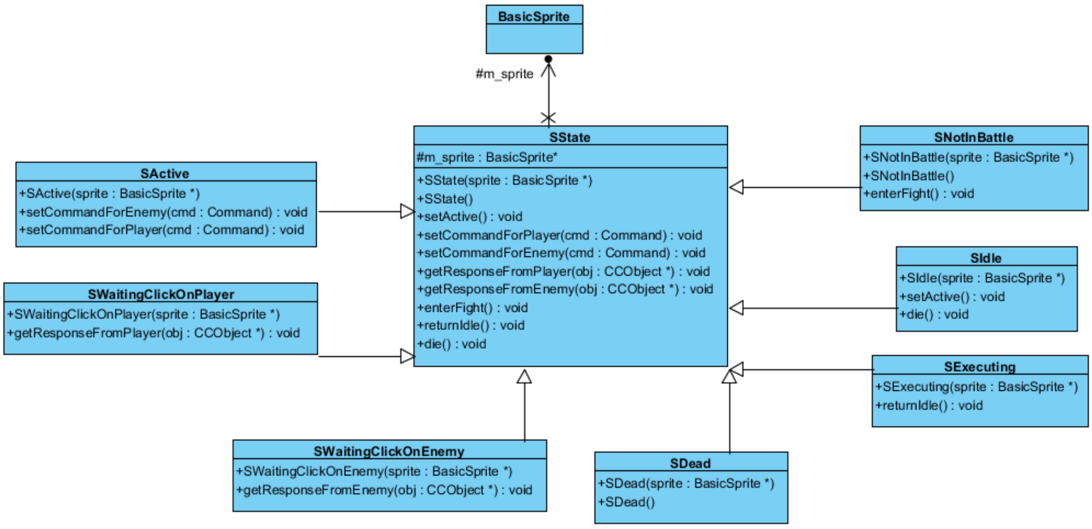
BasicSprite
#m_sprite
SState
#m_sprite : BasicSprite*
+SState(sprite : BasicSprite *)
+SState()
+setActive() : void
+setCommandForPlayer(cmd : Command) : void
+setCommandForEnemy(cmd : Command) : void
+getResponseFromPlayer(obj : CCObject *) : void
+getResponseFromEnemy(obj : CCObject *) : void
+enterFight() : void
+returnIdle() : void
+die() : void
SActive
+SActive(sprite : BasicSprite *)
+setCommandForEnemy(cmd : Command) : void
+setCommandForPlayer(cmd : Command) : void
SWaitingClickOnPlayer
+SWaitingClickOnPlayer(sprite : BasicSprite *)
+getResponseFromPlayer(obj : CCObject *) : void
SNotInBattle
+SNotInBattle(sprite : BasicSprite *)
+SNotInBattle()
+enterFight() : void
SIdle
+SIdle(sprite : BasicSprite *)
+setActive() : void
+die() : void
SExecuting
+SExecuting(sprite : BasicSprite *)
+returnIdle() : void
SWaitingClickOnEnemy
+SWaitingClickOnEnemy(sprite : BasicSprite *)
+getResponseFromEnemy(obj : CCObject *) : void
SDead
+SDead(sprite : BasicSprite *)
+SDead()

Fig.13 class diagram of State Pattern implementation

Observer Pattern is used when sprites are waiting response from others. For example, when attack command is chose, the active sprite will waiting for click on enemy. The following code is for observing and posting:

//add an observer

```
CCNotificationCenter* t_pNotiCenter = CCNotificationCenter::sharedNotificationCenter();
SEL_CallFuncO t_oCallFuncO = callfuncO_selector(BasicSprite::getResponseFromEnemy);
t_pNotiCenter->addObserver(m_sprite, t_oCallFuncO, "enemy", NULL);
m_sprite->setState(m_sprite->waitingClickOnEnemy);
```

//post a message

```
void Formation::ccTouchEnded(CCTouch *pTouch, CCEvent *pEvent) {
    BasicSprite* bs = NULL;
    if (containsCCTouchPoint(pTouch, bs) && isAlive(bs->getTag())) {
        char* name = NULL;
        if (side == PLAYER_SIDE)
            name = "player";
        else
            name = "enemy";
        CCNotificationCenter::sharedNotificationCenter()->postNotification(name, bs);
    }
}
```

3.5 Other Details

Constans.h --- including all the file path

AudioManager --- music related interface package

```
/* preload */

static void preLoadBgMusic(const char* music);

static void preLoadEffect(const Sound& sound);

/*    called at AppDelegate    */

static void init();

/* play */

static void playMusic(const char* music,bool loop = true);

static void playEffect(Sound& sound, bool loop = false);

static void playEffect(const char* path,bool loop = false);

/* pause */

static void pauseMusic();

static void pauseEffect(const Sound& sound);

/* stop */

static void stopMusic(bool release = false);

static void stopEffect(const Sound& sound);

static void stopAllEffect();

/* resume */

static void resumeMusic();

static void resumeEffect(const Sound& sound);

/* end */

static void end();
```

AnimationUtil --- animation related interface package

```
//add a sprite action frame

static void createSpriteAnimWithFrameName(const char* spriteName, const char*
actionName, CCAnimation* & animation, float delay = 0.1f, unsigned int iLoops = 1);

//add all the action frame of a sprite

static void addSpriteAnimationsWithName(const char* spriteName);

//add skill effect

static void addSkillAnimationsWithName(const char* skillName, float delay = 0.1f, unsigned
int iLoops = 1);

//get sprite animation

static CCAnimation* getSpriteAnimationWithName(const char* spriteName, int action);

//get skill animation

static CCAnimation* getSkillAnimationWithName(const char* skillName);
```

4. Conclusion and Future work

In this project, I have finished the single-player part of a turn-based fight game, the whole fighting scene and edit scene. It can easily adding more sprites or items by import more pictures or frame animations resource into the project, the code will automatically read these resources, as mention before.

But there are a lot of elements that can be added. For example, the sprite add attributes like Critical Strike and Dodge Rating, which can increase the diversity of the game.

And still, this game is a single-player game, it can be change to be a multi-player online game. That will make it funnier and more attractive.

5. Appendix

● Minutes of the 1st Project Meeting

Date: Friday, 21 March 2014

Time: 1:10 PM **Place:** Room 3512

Attending: Prof. Rossiter, HUANG Jiefeng

Absent: None

Recorder: HUANG Jiefeng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

HUANG Jiefeng finished the project demo, including the main scene, set scene and part of the fight scene.

Discussion Items and Things to Do

- Change the game to single-player one
- Finish the fight scene

Meeting adjournment

The meeting was adjourned at 2:00 PM.

● Minutes of the 2nd Project Meeting

Date: Thursday, 24 April 2014

Time: 1:30 PM **Place:** Room 3512

Attending: Prof. Rossiter, HUANG Jiefeng

Absent: None

Recorder: HUANG Jiefeng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

HUANG Jiefeng finished the main part of the fight scene, including adding skills, goods and sprites' attributes.

Discussion Items and Things to Do

- Finish the edit scene
- Write the first version of report

Meeting adjournment

The meeting was adjourned at 2:30 PM.

Minutes of the 3rd Project Meeting

Date: Wednesday, 14 May 2014

Time: 1:00 PM **Place:** Room 3512

Attending: Prof. Rossiter, HUANG Jiefeng

Absent: None

Recorder: HUANG Jiefeng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

HUANG Jiefeng finished and showed the whole project, including the code, report and video.

Discussion Items and Things to Do

- Modify and submit the project

Meeting adjournment

The meeting was adjourned at 1:30 PM.

Minutes of the 4th Project Meeting

Date: Friday, 16 May 2014

Time: 2:00 PM **Place:** Room 3512

Attending: Prof. Rossiter, HUANG Jiefeng

Absent: None

Recorder: HUANG Jiefeng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

HUANG Jiefeng showed the final video and report.

Discussion Items and Things to Do

- None

Meeting adjournment

The meeting was adjourned at 2:30 PM.