UML Model Design for Mobile Game on the Android OS

Zhu Quanyin, Jin Ying, Xu Chengjie, Geng Rui

1 Faculty of Computer Engineering, Huaiyin Institute of Technology, Huaian, 223003, China, hyitzqy@126.com
2,3,4 Faculty of Computer Engineering, Huaiyin Institute of Technology, Huaian, 223003, China, jinying@hyit.edu.cn, haxcj@126.com, 249576195@qq.com

Abstract

In order to promote the mobile games to keep up with the new technologies development, the mobile game of Gallant Fighter with Double Blade based on the Android OS designed in this paper applies various techniques, especially for the gravity sensing, such as object pool, multi-threaded, socket connection, maps and etc.. The game management, the service class, the sound manager class, the game view class, the pass tips window class, the rank window class, game activity class, skin class collision detection and bit map util class are designed by UML model respectively. Experiments demonstrated its performance and proved that this model is meaningful and useful to develop other online mobile games. The UML model supports the game development and provides happiness for players in the leisure time.

Keywords: UML Model, Mobile Game, Gravity Sensing, Android OS

1. Introduction

A mobile game is a video game played on a mobile phone, cell phone, PDA, handheld computer or portable media player. This does not include games played on handheld video game systems such as Nintendo DS or PlayStation Portable. Mobile games are played using the technologies present on the device itself. For networked games, there are various technologies in common use. Examples include text message (SMS), multimedia message (MMS) or GPS location identification. Young people are often comprised the majority of early adopters and most avid users of mobile gadgets and applications, especially mobile games. Mobile gaming is playing a more and more important role in the entertainment industry, partially due to the rapid development of mobile communication. Porting mobile web application engine to the Android platform can reduce the workload. Android is a widely anticipated open source operating system for mobile devices that provides a basic operating system, an application middleware layer, a Java software development kit (SDK), and a collection of system applications. Since the source code of Android was released to people, a large community of developers has organized around Android. Although many researchers interesting focus on the secure architecture, compensate modeling, educational modeling, mobile video games, belief game model, asymmetric information theory and so on. However, Android has a large community of developers writing applications ("apps") that extend the functionality of the devices even used to build mobile medical image viewer. Developers write primarily in a customized version of Java. There are currently more than 520,000 apps available for Android.

We focused on the case study of mobile games. Based on our relative researched works on the mobile games, the UML model for the mobile game of Gallant Fighter with Double Blade which is widely well come on the Interest is studied in this paper.

2. Game management

Game manager class is the game management class that contains all the pictures methods, such as the enemy's draw, boss, draw the effect of the explosion and the drawing of the machine etc. the painting thread, collision detection thread, collision detection methods of listening to events, such as a keyboard response of the monitor and touch screen phone surveillance are drawn. The game management UML shows as Fig. 1.
3. Gravity sensing

The Coordinate system of gravity sensing in the Android OS takes the upper-left of the mobile screen as the origin. The direct define is shown as Fig. 2.

Using the x, y and z values to get the trigonometric function value, and then the mobile phone moving state can be accurate measurement.

1) SensorManager sensorMgr=(SensorManager) getSystem Service (SENSOR_SERVICE) can get a hardware controller. For example, LocationManage can determine the location, AudioManager can use the audio player.

2) Sensor sensor = sensorMgr.getDefaultSensor (Sensor.TYPE_ALL) can get the x, y and z values of offset.

3) SensorEventListener lsn = new SensorEventListener() is used to get the value changing. When a SensorEvent has been declared to listen, the changing value can be gotten from TextView, and these values are float[] array. That is x, y and z value respectively.

4) Three parameters are listen, sensing device, and delicacy respectively. SENSOR_DELAY_FASTEST, SENSOR_DELAY_NORMAL and SENSOR_DELAY_UI represents the fast, normal and slowly respectively.

![Figure 1. The game management UML](image)

4. Collision detection

1) Rectangular and rectangles collisions. Rectangular and rectangles collision general rules of object collision all can handle rectangular, collision and realization principle is to test whether two rectangular overlap. The parameters of rectangular 1 hypothesis are as: the upper left corner of the coordinates (x_1, y_1), width and height is w_1, h_1 respectively; the parameters of rectangular 2 are as: the upper left corner of the coordinates is (x_2, y_2), width and height is w_2, h_2 respectively.

In the test, mathematics can be handled as a comparison of the coordinates of the center in the x and y directions distance and width of the relationship. That is two rectangular center in the direction of the
x the absolute value of the distance is less than or equal to the width and the rectangular one half of the distance, and y direction is less than or equal to the absolute value of the rectangular height and of the half.

The mathematical expression is as follows:

- x direction: \(|(x_{1} + \frac{w_{1}}{2}) - (x_{2} + \frac{w_{2}}{2})| < |(w_{1} + w_{2}) / 2|
- y direction: \(|(y_{1} + \frac{h_{1}}{2}) - (y_{2} + \frac{h_{2}}{2})| < |(h_{1} + h_{2}) / 2|

In the program, only required to convert the code above conditions can be met.

But the rectangular collision is only a coarse detection collision method, because a lot of the actual object might not be a rectangle of rules.

2) Round collision

Round with round collision should be one of the simplest collisions, in mathematics, for two rounds if an overlap, it only should to calculate the distance between the two circles center. That is calculate the distance between the two circles is less than the sum of two round radius.

Assume 1 circle round the coordinates \((x_{1}, y_{1})\) with round \(r_{1}\), and the another circle round the coordinates \((x_{2}, y_{2})\) with round \(r_{2}\). Because of MIDP1.0 can not calculate the floating-point data, on the other hand, the floating-point operation is slower, so the calculation can be changed as follows.

Mathematical expression: \((x_{1} - x_{2})^{2} + (y_{1} - y_{2})^{2} < (r_{1} + r_{2})^{2}\).

5. The main game UML models

5.1. Menu activity UML model

The game menu includes games begin, cancelled by game, help, exit and game setting button. When the start or the exit functions are called, then activity's finish from jump and follows method. ControlActivity control page is the jump, this page realized by switch button corresponding. RadioGroup correspond the voice of components of the overall control. With the sound the single example to the entire object model only form an in overall control. Help pages and the list is a PopupWindow components. PopupWindow as a customer reminder and its spending than to small is from follows the activity exist a small component. The Menuactivity UML shows as Fig. 3.

![Figure 3. The MenuActivity UML model](image)

Events surveillance, events such as the click corresponding events that has two kinds of monitoring method, one is a click on the corresponding events on-click events surveillance, the other is a on-touch event monitoring. The latter can realize to the press mobile and raise three movements of the control, but not the former fast speed of the corresponding. The game only realized simple button response, and need in response of the three movement of the control of the background of the conversion. All options are a touch by listening in.SetListeners.

Custom listening to events, because of the system can transfer an event value in the value which has in all three movements in the press movement at event.getAction() == MotionEvent.ACTION_DOWN. It will convert the backgroundrankButton.set and BackgroundDrawable(buttonDown). When event.getAction() == MotionEvent.ACTION_UP, the background will be change, so that they can get click on an action get different background events.

```java
beginButton.setOnTouchListener(new OnTouchListener() {
    public boolean onTouch(View v, MotionEvent event) {
        // Handle touch events
        return true;
    }
});
```
public boolean onTouch(View v, MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        beginButton.setBackgroundDrawable(buttonDown);
    } else if (event.getAction() == MotionEvent.ACTION_UP) {
        beginButton.setBackgroundDrawable(buttonUp);
        Intent intent = new Intent(GameConfig.mActivity,
                                   ProgressActivity.class);
        GameConfig.mActivity.startActivity(intent);
    }
    return false;
}

5.2. Service UML model

Thread inherits the thread class and realization the method of run. Use while cycle control can keep on to test whether the client in connecting to the server. Thunderserver class connect to the client depend on the roles of provided IP and according to TCP, UDP interface. Running in the thread when the client connection, and put it in a container. The arraylist <client> class inherits from the arraylist class. It can accommodate connection of the client. For the client class according to the server’s IP and TCP, UDP interface to connect the server. See the Fig. 4.

![Figure 4. The game service UML model](image)

5.3. Sound management UML model

SoundManager class is a small clever class, it can be accurately controlled. However, the class has loaded files can’t be more than one million seconds, because when the explosion of the time we need an exact file loading in the plane of the explosion was so with this method of loading voice.

Single example mode to achieve their instantiation own static methods, if the object does not exist very empty then instantiation yourself, if there are calls for static object. The voice of the pool with system defined, the control method of sound belly is full, stop playing method for the stop. The voice getStreamMaxVolume methods obtain the voice of the system.

The MusicPlayer class is the public class of MediaPlayer awakened operation. This kind of doesn’t limit the size of the file in the continuous play but there will be a delay in the background music. See the Fig. 5.

5.4. The game view UML model

With the SurfaceHolder.Callback function calls back the canvas, only when the canvas blockade to lock on the top operations such as update and painting. The canvas is submitted to unlock the process...
of canvas. See the Fig. 6.

5.5. The rank and pass tips window UML model

Window bag is used in on the game popwindow collective management. The Rankwindow is a high marks popwindow and used to list achievement high grades of the display.

LayoutInflater mLayoutInflater = (LayoutInflater) GameConfig.

mActivity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

The passTipsWindow class is the passed tips class. It used to define the opening, closing, or the initial rest the game respectively. See the Fig. 7.

Figure 5. The music play UML model

Figure 6. The game view UML model
5.6. Game activity UML model

Game activity is the front page of the main game page of control. The game loading this follows the custom visualization game view. The generation of game page inherited surfaceview. All that can be operated in Android in surfaceview components are drawing components of the realization of the photographic class, such as video camera, video class, etc. Surfaceview inheritance to realize three abstract methods: surfaceCreated method, when the method of generating calls; SurfaceChanged (when the picture changes call the method); and SurfaceDestroyed (when the picture destroyed call the abstract methods) respectively. See the Fig. 8.

Scores storage, when surfaceview destroyed scores of stored in processing RankInfoStore invokes the current user name, if (GameResult.score more then current score), The current scores and scores of the existing to more if the existing high test scores, then call RankInfoStore. This store tools storage methods for game data storage when the game just generating the canvas to restart the need of thread, or when canvas method of destruction at the end of the game painting thread destroyed.

5.7. Skin UML model

1) Skin is the father of defaultskin class, the provisions of the static attribute by its subclasses to achieve. See the Fig. 9 (a).

2) defaultskin this kind of role is to use bitmap for instantiation of the plane, with set and the get method for calls inside of the acquired bitmap.

3) preparedskin is defaultskin classes, the function is according to the epitome of instantiation to have a picture the search for her. See the Fig. 9 (b).

5.7. UML model of collision detection and bit map util

The toolkit is used for collision diction and bit map util on the games UML built, such as the game result processing tools, boundary processing tools, image adjusted the tools.
1) GameResult. This kind of game result processing tools: definition win or lose, the value apart into integer array method scores.

2) CollisionDetection. The tools of boundary treatment: inside this judge the elements is out, all the collision detection methods and tools in collision in effect, such as life minus one, the change of the score of the sounds, or produce.

3) BitmapUtil. The tools of image processing has designated areas graphics, from a larger version of thumbnails, according to intercept screen density enlarge images processing, a larger version to return to cut figure elements array and screen density enlarge images processing tools method.

Make the interception, regions in the image of the method is mainly used in the bitmap creatbitmap several parameters is the first pictures of the original resources to intercept. The second is the beginning of the screenshots of the coordinates of the pictures and the picture in width and height.

According to the screen density enlarge images processing, in order to make graphics in the LCD, but have not distortion can full screen we put them in a density to amplify or narrow. Tools class CollisionDetection. If the test fly out of boundary key code:

```java
public static boolean checkEdge(Component c) {
    int x = c.getXPos();
    int y = c.getYPos();
    int width = c.getBitmap().getWidth();
    int height = c.getBitmap().getHeight();
    return (x>0) && (x<(GameConfig.screenWidth-width))&& (y>0) &&(y<(GameConfig.screenHeight-height));
}
```

Figure 9. (a) The game Skin UML model (b) The game PreparedSkin UML model

Inspection a component collided with each other. In the game colliding time introduced the first
game coordinate system, if the model is the rectangular so vertex coordinates is upper left point coordinates, or if it is a round circle of words coordinates. When a hit from the upper left of the collision and plus it’s tall to be hitting, that is \((x_1 + \text{width1} > x_2) && (y_1 + \text{height1} > y_2) && (x_2 + \text{width2} > x_1) && (y_2 + \text{height2} > y_1)\).

Deal with bullets and plane collision, when judge after the collision, the bullet set to death, death will encapsulate the bullet to a container of concentrated to deal with. The collision diction and bit map util UML are shown as Fig. 10 (a), (b) respectively.

![Figure 10. (a) The game CollisionDetection UML (b) The game BitmapUtil UML](image)

6. The game implement

Five static constants (Sign in, Start, Menu, Help, and Sign out) are defined in the MainMenu class which is used to identify the selected state. Six sub-static constants (Sky, Message, Music, Tool, Flight and Cloud) are defined in the ConstantUtil class which is used to identify the selected sub-state. When the players enter the game from the start window of the main interface, the initial state is Fighting window. The First fighting window, the Boss window, and the Second fighting widow are shown as Fig. 11 (a), (b), (c) respectively.

![Figure 11. (a) The First fighting window; (b) the Boss window; (c) the Second fighting widow](image)

7. Conclusions

There are many technologies involved in the development of mobile games, such as game state machine, object pool, multi-threading, wizard, maps and so on. Through program optimization and design of compatibility, it develops the game engine and simple server procedure which are suitable for games that have single-screen maps. And it has a good reference for those same kinds of games. As a language, Java is easy to learn and master, and it follows a strong, secure, portable, and scalable platform. All these elements make Java a perfect development tool in the field of small device.

3G will accelerate the integration between mobile networks and traditional Internet. Meanwhile, network integration will enable the existing version of Internet games to have the edition of mobile terminals. It can be foreseen that the entire game industry will eventually realize the integration of PC and mobile phones.
8. References


[17] Quanyin Zhu, Hong Zhou, Yunyang Yan, and Chuanchun Yu, “Happy Farm an Online Game for Mobile Phone”, Communications in Computer and Information Science, Vol.234, no.4, pp.120-128, 2011.