A UML Model for Mobile Game on the Android OS

Zhu Quan-yin, Jin Yin, Xu Chengjie, Gen Rui

Faculty of Computer Engineering, Huaiyin Institute of Technology, Huaiian, 223003, China

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, and the rank window 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.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of ICAE2011.

Keywords: UML model; mobile game; Gallant Fighter with Double Blade; 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. [1]

Young people are often comprised the majority of early adopters and most avid users of mobile gadgets and applications, especially mobile games. [2] Mobile gaming is playing a more and more important role in the entertainment industry, partially due to the rapid development of mobile communication. [3] Porting mobile web application engine to the Android platform can reduce the workload. [4] 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. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. 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. 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.

![Game Management UML Diagram](image)

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) getSystemService(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.
4. The main game UML models

4.1. Service UML model

Udpthread 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.3.

![Fig. 3. The game service UML model](image)

4.2. 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.
4.3. 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.

Fig. 5. The music play UML model

Fig. 6. The game view UML model
4.4. 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 pasttipswindow class is the passed tips class. It used to define the opening, closing, or the initial rest the game respectively. See the Fig.7.

5. 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. 8 (a), (b), (c) respectively.

Fig. 7 (a) rank window UML model; (b) pass tips window UML model

Fig. 8 (a) The First fighting window; (b) the Boss window; (c) the Second fighting widow
6. 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.

References