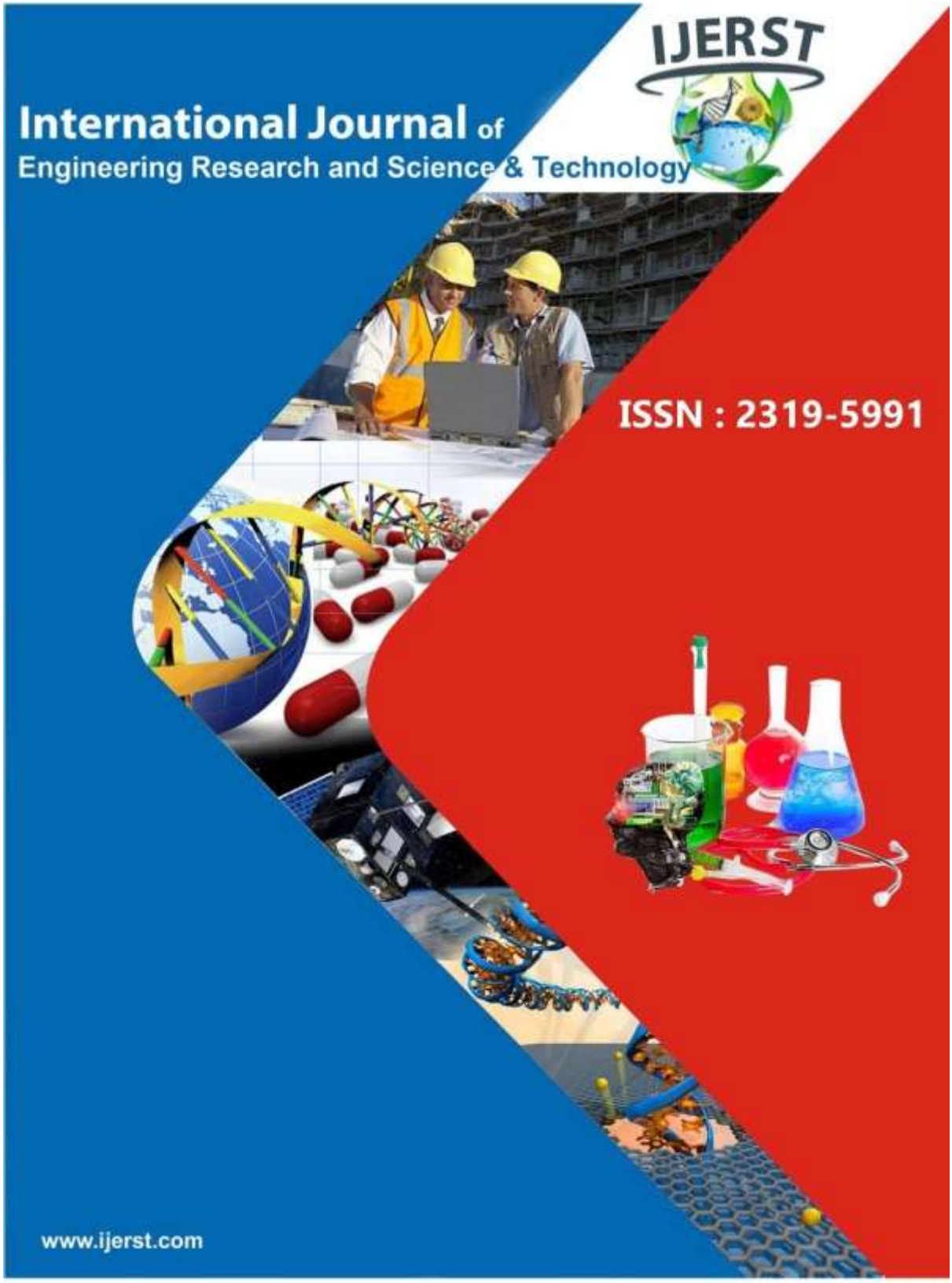

Email: editor@ijerst.com or editor.ijerst@gmail.com

This article can be downloaded from http://www.ijerst.com/currentissue.php

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 14, Issuse.2,April 2021

An Android-Based Mobile Game's UML Model

M VENKATA RATHNAM [1], G VENGALARAO [2]

Abstract

The Android OS-based mobile game of Gallant Fighter with Double Blade designed in this paper makes use of a number of cutting-edge techniques, including object pool, multi-threaded, socket connection, maps, and etc., to encourage mobile games to keep up with the progress of technology. The UML model was used to create the game's administration, service class, sound manager class, game view class, pass hints window class, and rank window class. The results of the experiments showed that the approach worked well and could be used to create new online mobile games. The UML model aids in game design and makes gamers' free time more enjoyable.

Keywords: ***Android OS; UML model; gallant fighter double blade mobile game; gravity sensing***

Introduction

Mobile games are video games that may be played on mobile devices such as mobile phones, PDAs, handheld computers, and portable media players. Not included are portable consoles like the Nintendo DS or the PlayStation Portable, on which video games may be played. Mobile games make use of the hardware and software already installed on the user's device. There are several different technologies now in use for online gaming. Sending a text message, sending a multimedia message, or identifying your precise position using GPS are all good examples. When it comes to mobile technology, particularly mobile games, it's frequently the young that are the first to embrace and the most dedicated users. The explosive growth of mobile connectivity has helped propel mobile gaming to the forefront of the entertainment market. Bringing the power of the mobile web application engine to the Android platform helps lighten the burden, as we see in [3]. [4] Android is a highly anticipated open-source mobile operating system that offers a complete OS, a layer for implementing apps, a Java SDK, and a suite of system programs. A sizable community of programmers has formed around Android ever since its source code became publicly available.[5] Many programmers have contributed to the Android platform by creating programs (or "apps") that enhance the capabilities of Android smartphones. Programming is done mostly in a modified form of Java.[6] There are already around 520,000 Android applications available. [7][8] . The mobile gaming industry was our case study of choice. This study examines the Unified Modelling Language (UML) model for the popular mobile game Gallant Fighter with Double Blade. the game's administration is handled by a class called "game manager," which is responsible for everything from drawing enemies and bosses to creating explosion effects and sketching machines. Collision detection threads and techniques for listening to events, such a monitor's keyboard response or a touch-screen phone's listening capabilities, are sketched. In Fig. 1, we see the UML for game administration.

Assistant professor1,2
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
P.B.R.VISVODAYA INSTITUTE OF TECHNOLOGY & SCIENCE
S.P.S.R NELLORE DIST, A.P , INDIA , KAVALI-524201

This article can be downloaded from http://www.ijerst.com/currentissue.php

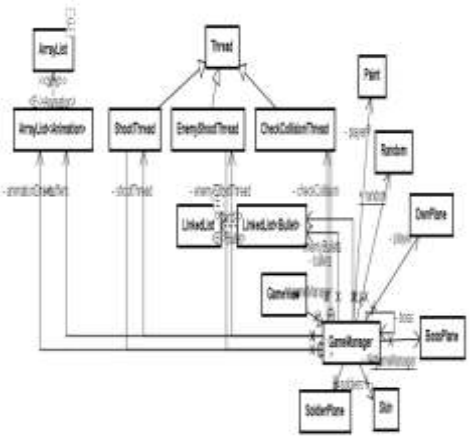

Fig. 1. The game management UML

Gravity sensing

The upper-left corner of the smartphone screen serves as the origin for the Android OS's gravity sensing coordinate system. Figure 2 depicts the straightforward definition. The trigonometric function value may be calculated from the x, y, and z coordinates, allowing for precise measurements of the mobile phone's motion. 1) A hardware controller may be obtained by using sensor=(Sensor Manager) get System Service (SENSOR_SERVICE). A location may be determined by Location Manage, and an audio player can be controlled by Audio Manager. The x, y, and z offset values may be obtained from the sensor by using the formula: sensor = sensorMgr.getDefaultSensor (Sensor. TYPE_ALL). To observe a shift in value, we may use SensorEventListener lsn = new SensorEventListener(). TextView provides access to the dynamic value when a SensorEvent has been defined to listen; the values are stored in a float[] array. That's the x, y, and z values, individually. 4) The listen, sensing device, and delicate parameters. The fastest, normal, and slowest delays are indicated by SENSOR_DELAY_ FASTEST, SENSOR_DELAY_NORMAL, and SENSOR_DELAY_UI, respectively

.

Fig. 2 The Coordinate system

UML models for the meat of the game

Modeling Services using UML

Upthread is a realization of the thread class and the run method. whether you want to check whether the client is successfully communicating with the server, use the while cycle control's keep-on option. The IP addresses and port numbers given by the client determine how the Thunderserver class implements the TCP/UDP interface. Thread-based processing and containerization upon client

connection. There is a lineage from the array class to the arraylist class. The client's connection may be made. Depending on the kind of client and the server's IP address, TCP or UDP may be used to establish a connection. As seen in Fig.3.

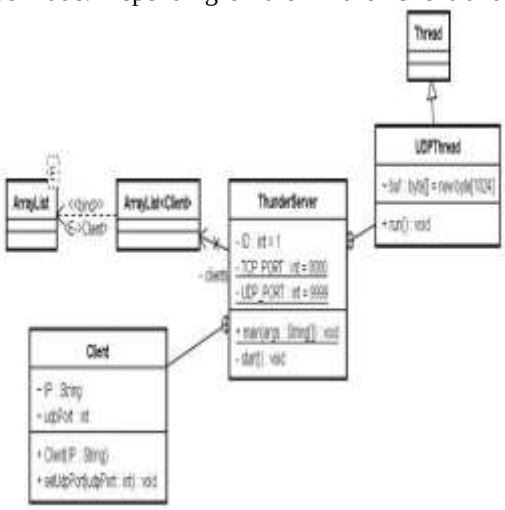

Fig. 3. The game service UML model

The UML Model of Sound Management

he SoundManager class is a compact intelligent entity that can be precisely managed. However, the maximum time a class may spend loading files is one million seconds. This is because, with this way of loading speech, we require a very precise file loading in the plane of the explosion. If the object does not exist or is extremely empty, and there are requests for a static object, the single-instance mode may be used to accomplish this goal via the use of static methods. The system's voice has been specified, and its methods of controlling sound volume and turning off the music when the stomach is full have both been implemented. The system's voice may be retrieved using the getStreamMaxVolume functions. MediaPlayer's public class after awakening is the MusicPlayer class. This kind of thing doesn't restrict the file size during continuous play, but the background music will be delayed. Refer to Fig.3

This is a UML model of the rank and pass tips window.

The cooperative game popwindow makes use of the window bag mechanic. When a user receives a high score, their accomplishments may be shown in a popup window called the Rankwindow. Inflater Layout mLayoutInflater = InflaterLayoutThe mActivity in GameConfig. The given tips class is the passtipswindow class, which is obtained by usingGetSystemService(Context.LAYOUT_INFLATER_SERVICE). It formerly marked the beginning, end, or first break in play. Look at Fig.7.

The gaming piece

The Main Menu class defines five static variables that are used to determine the current state: Sign in, Start, Menu, Help, and Sign out. The ConstantUtil class defines six sub-static constants (Sky, Message, Music, Tool, Flight, and Cloud) that are utilized to determine the current sub-state. When players launch the game from the main menu, they are immediately sent to the Fighting window. The First Fighting Window (a), the Boss Window (b), and the Second Fighting Window (c) are shown in Fig. 8.

.

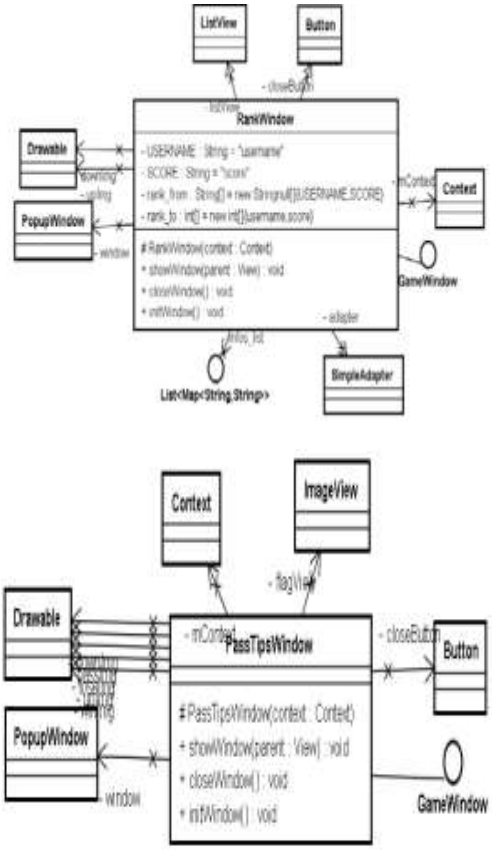

Fig. 7 (a) rank window UML model; (b) pass tips window UML model

Fig. 8 (a) The First fighting window;

(b) the Boss window;

(c) the Second fighting widow

Conclusions

Technologies like game state machines, object pools, multi-threading, wizards, maps, and so on are used in the creation of mobile games. It creates a game engine and straightforward server process well suited to games with single-screen maps by optimizing the code and designing for compatibility. In addition, it provides a useful resource for games of the similar genre. Java has a solid, secure, portable, and scalable platform and is simple to learn and master. For these reasons, Java is an ideal platform for creating new handheld gadgets. The adoption of 3G will hasten the convergence of mobile and fixed-line networks. Mobile terminal editions of established Internet games will be possible via network integration. It's predictable that the whole gaming business will come to terms with PC and mobile phone integration at some point.

References

[1] ZHANG Xin-meng JIANG Sheng-yi JIANG Ling-min. Design and Implementation of mobile network games Based on j2me/socket. Microcomputer Information.2010.Vol.26(12-3),p.236-238

[2] Stan Kurkovsky.Can Mobile Game Development Foster Student Interest in Computer Science?.Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's. London, UK. 2009,p.92-95

[3] Weishan Zhang, Dong Han, Kunz, T. Hansen, K.M. Mobile Game Development: Object-Orientation or Not. 31st Annual International Computer Software and Applications Conference. Beijing, China. 2007 ,p.601 – 608

[4] Yonghong Wu, Jianchao Luo, Lei Luo. Porting mobile web application engine to the Android platform.. 2010 IEEE 10th International Conference on Computer and Information Technology (CIT). Bradford, United Kingdom 2010,p.2157-2161

[5] Maoqiang Song, Wenkuo Xiong, Xiangling Fu. Research on Architecture of Multimedia and Its Design Based on Android. 2010 International Conference on Internet Technology and Applications. Wuhan, China. 2010, p.1 – 4.

[6]Shankland, Stephen (12 November 2007). Google's Android parts ways with Java industry group. CNET News. http://www.news.com/8301-13580_3-9815495-39.html.

[7] Barra, Hugo (10 May 2011). Android: momentum, mobile and more at Google I/O. The Official Google Blog. http://googleblog.blogspot.com/2011/05/android-momentum-mobile-and-more-at.html. Retrieved 10 May 2011.

[8] Android Stats. 25 Sep 2011. http://www.androlib.com/appstats.aspx. Retrieved 25 Sep 2011.