A BnB Mobile Game Online Based on J2ME and J2EE

Quanyin Zhu, Lijuan Zhao  
Faculty of Computer Engineering  
Huaiyin Institute of Technology  
Huai’an, China  
E-mail: hyitzqy@gmail.com

Suqun Cao  
Faculty of Mechanical Engineering  
Huaiyin Institute of Technology  
Huai’ai, China  
E-mail: caosuqun@126.com

Jianchun Shen, Shuai Zhang  
Faculty of Computer Engineering  
Huaiyin Institute of Technology  
Huai’ai, China  
E-mail: shenjianchun@gmail.com

Abstract—Through the example of Bubble and Bubble (BnB) mobile game, the whole development process of mobile game based on J2ME and J2EE is showed. First, Game API in the Mobile Information Device Profile (MIDP) 2.0 is used to build the game engine. Through the optimization and compatibility design of process, middleware framework is selected. Second, the game classes and components design is introduced in detail. Third, the server performance for detonation on the client, the communications time between the clients with the server and the procedure performance of the server have been tested. Various techniques, such as object pool, multi-threaded, socket connection, sprite, Maps etc., are applied in BnB’s development. Experiment demonstrates its performance and proves this case is meaningful and useful for other online mobile game development. Some propositions for further research is also suggested.

1. INTRODUCTION

The software product of mobile games is developed very quickly recently. Many kinds of methods have been researching by many researchers which are depending on their interesting in this domain. The case study is main approach which are reported on some papers [1,2]. Several research activities have been done describing frameworks or architectures [2-5] and different methods haven been introduced [6-8]. Based on the multiplayer online games (MOG), a WiMAX link model, which use the orthogonal frequency division multiple access (OFDMA) is detailed in the reference [9], and the impact of system parameters are simulated in the reference [6]. Domestic mobile online games started in 2004, which has got such great development that on the whole the gap between the international mobile online games is not obvious. According to the latest report, it shows that until 2011 the scale of China’s mobile games market will reach 3.7 billion while compound annual growth rate (CAGR) of the mobile gaming market will be 48%. By the first half of 2008, the number of Chinese mobile netizen has reached more than 50 million, up by 2.6 times as much as that over the same period of last year. With the 3G networks up and put into use, this figure will be further swollen. The development of the mobile phone networks will retake the road of PC network, and the mobile online game will step onto a new level and become a rising industry.

The industry professionals have analyzed that today’s online game is the tomorrow of mobile online games. At present, the market of domestic mobile phone games only takes a small portion of the games industry, but it shows great potential growth. Major mobile communications operators, mobile manufacturers, network operators and game developers take mobile games as next growth point to increase revenue. Huge group of mobile phone users also determine that the mobile games market is of great potentials [10]. However, many applications based on 3G are so limited and developing slowly in China. So some mobile games have been transformed only on the Internet games and some mobile games have been developed on the simulation platforms [11-15], and much more of them are accomplished on Role-Playing –Game (RGP) such as Java Me MIDP platform. Because of the RAM is limited, and the implement faculty and transacting capability of CPU are very lower, so, the mobile games are mostly simply but should be interested enough, such as Dig gem, Climb, Hunt, Feeding and Kongfu [8], some of them can play both the mobile and the PC [3]. How to transfer the wildly played in internet games to fitting mobile is very interesting and would be satisfied the player playing it in everywhere.

In this paper, we focused on a very interested game which has been played widely on Internet that is Bubble and Bubble (BnB). The mobile online game client architecture which is depended on the jQuery framework and the server architectures which are depended on the Struts and Hibernate framework are introduced. Between the client and the server communications are used the middleware platform which are developed on the Socket API. The service performance has been measured and the optimizations are discussed in last section.

II. GAME CLASS AND FRAMEWORKS

A. Class Design

The process of the game development is based on the idea of object-oriented, so that the function of each module should be as independently as possible. There are
a total of 16 classes designed for the game depend on the Java game class:
1) BombGame.Start
   Start Class inherits from javax.microedition.midlet.MIDlet, procedure starts when the constructor Start() is called and ends when its destroyApp(boolean arg0) method called. Start is the core part of the entire game, and all the visual objects must call its setCurrent(Displayable nextDisplayable) method to make themselves appear on the screen. Start Class is the manager of the whole procedure.

In Start Class a lot of constants are also defined, which are declared to be public static final, let any class in the game obtain access to the value of these constants by the way like that int tick=Start.MILLIS_PER_TICK (obtain the screen refresh time). In this way the configuration of the game becomes very flexible, and when it is time to compile the program, these constants will be inline by dotfuscator, and the efficiency will not be affected.

2) BombGame.SplashCanvas
   SplashCanvas Class inherits from javax.microedition.lcdui.Canvas, and meanwhile implements the java.lang.Runnable interface. It is the Splash Screen of the game and also the first visual object. In the beginning of the implementation of procedures, SplashCanvas starts a new thread to show the relevant information of the graduation project, while allowing the procedure to initialize other programs resource in the background (the specific realization is detailed in section III).

3) BombGame.Menu
   Menu Class inherits from javax.microedition.lcdui.game.GameCanvas, which is the main menu of the game. When the Menu is being constructed, many state constants are defined to mark the current states so as to enter the different functional interfaces.

4) BombGame.Option
   Optionb Class realizes the setting method in the game, in which draw(Graphics g) method is used to draw the setting menu, optionSelect(int lastKeyPressed) is used to accept user’s input, and deal with the game logic processing in accordance with the user’s input.

5) BombGame.GameInfo
   GameInfo Class is used to draw the help and about interface, which defines the drawHelp(Graphics g, int index) and drawAbout(Graphics g) method.

6) BombGame.MainCanvas
   MainCanvas Class inherits from javax.microedition.lcdui.game.GameCanvas and meanwhile implements java.lang.Runnable interface. It is the "arena" of the game that to deal with most work of the game, including the detection of user input, processing the game logic and drawing the game screen.

7) BombGame.GameMaps
   GameMaps Class defines a number of layers, the methods of obtaining map information and each layer of cells filled with the corresponding elements.

8) BombGame.MapsInfo
   MapsInfo Class is used to describe the special data structure of the game map, such as the map’s layout, information about where the players were born and so on.

9) BombGame.MyRole
   MyRole Class inherits from javax.microedition.lcdui.game.Sprite, used to deal with the movement and the state changes of the role, initialize information of the role and the corresponding information of the Bomb, as well as the methods for the role to obtain the Bomb.

10) BombGame.ExplosionSprite
    ExplosionSprite Class inherits from BombGame.PropSprite, defines the process of role extinction on the condition that surrounded.

11) BombGame.Bomb
    Bomb Class inherits from javax.microedition.lcdui.game.Sprite, and defines the methods of the showing and hiding Bomb Objects.

12) BombGame.Explosion
    Explosion Class is an abstract of the bubble explosion, which is not an agent, because it is difficult to use agent to achieve the effect of bubble explosion, and bomb explosion is achieved by the modified picture in the Map.

13) BombGame.PropSprite
    PropSprite Class inherits from javax.microedition.lcdui.game.Sprite categories, defines the methods of showing and hiding objects of prop.

14) BombGame.AudioPlayer
    AudioPlayer Class is used to play sound for the game, and implements java.lang.Runnable interface. After running a program, an independent thread starts, to play background music for the game. Meanwhile, it defines the method of changing the volume and mute setting.

15) BombGame.ClientConnect
    ClientConnect Class inherits from java.lang.Thread, which is used to send and receive network news in the game. In ClientConnect, send(Msg msg) and parse(InputStream is) are the two methods to achieve sending and receiving information in the game.

16) BombGame.Msg
    Msg Class is the communication interface defined in engineering, which is implemented unanimously by other messages class, facilitates the network to send and receive data. The relationship between the whole projects is shown in Figure 1.

B. Game Frameworks
The framework for the entire game is divided into three kinds of states: waiting state, running state (which is divided into multiple sub-states), and the over state, all of which throughout the game. Framework of the entire game is based on the operation of state machine, when the game runs, a variety of different forms are differentiated as a state separately, and any time there...
only a state executed. As a result of taking it into account that nested state in games, all the status is distinguished between their respective. During the main thread of the game, they are constantly judged in order to achieve the conversion between the states. Game structure is shown in Figure 2.

![Game Structure](image)

### III. GAME DESIGNS

#### A. Splash Screen

The game produces a Splash Screen interface (SplashCanvas), used to show the game version, copyright, authors information and so on. SplashCanvas Class inherits from Canvas, implements Runnable interface, and set to full-screen mode, when the procedure starts, it starts a new thread for the Splash Screen rendering.

The Fade(int [] raw, int alphaValue, int maskColor, int notMaskColor) calls the chroma value of Alpha image to achieve the fade effect. In the draw(int [] rawInt, int alpha, int value, Image img, int sleep ) method, it defines the content for rendering the Alpha image. In order to achieve adaptability to resolution, first to use the methods of getHeight() and getWidth() to obtain the height and width of the screen, and then, with which to calculate the coordinates of the starting point that makes the contents show at the position of vertical center, so that the splash screen images can be center shown. Splash screen displays and the menu shown in Figure 3.

#### B. Game Map

The map is the core part of the game, besides basic background, it is also related to the collision scene with the wizard, the interaction with the shock wave, and the production of props. GameMaps Class defines the needed layers of the map, as well as the method to fill layer element. The diagram of GameMaps Class is shown in Figure 4.

![GameMaps Class](image)

1) The Basic Structure

In the game the building would disappear after destruction and expose to the surface, so there should be at least two picture layers—the surface layer and the building layer. The surface layer displays the ground surface, such as grassplot, road (shown in Figure 5). The building layer is a generalized concept, which shows the buildings above surface, such as boxes, trees and so on. The surface layer is barrier free, whose effect is just a map, however, the building layer have their own attributes, such as buildings, trees, which can not be passed through and indestructible, while the boxes and stones cannot be passed through but can be destroyed, so during the game design, three layers are used to compose the game map, that is, the surface layer (groundTL), the building layer (barrierTL, Figure 6) and the box layer (boxTL, Figure 2) After all the layers overlap, they are to form a complete map of the scene.
The game scene is made up of a \( N \times N \) grid, and it is easy to think of using an array to show the scenes, with each element of an array mapping a cell. The following array is the Data Model of Figure 8.

```java
mission[0] = new byte[]{
    25,28,28, 1, 1,18,18,18,18,18, 1, 1,28,28,25,
    25, 1, 1, 1, 1, 1,18, 1, 1, 1,18, 1, 1, 1, 1,18,
    28, 1, 1, 1, 1, 1, 1,18, 1, 1, 1, 1,18, 1, 1,28,
    1,1,16, 3, 3, 3, 3, 3, 3, 3, 3, 3,16, 1, 1,
    27,27, 1, 1,16, 1, 1, 1,14, 1, 1, 1,16, 1, 1,27,27,
    1, 1, 1, 1, 1, 1,26, 1, 1, 1,26, 1, 1,24, 1, 1,
    24, 1, 1, 1, 1,26, 1, 1, 1,26, 1, 1,24, 1, 1,
    24, 1, 1, 1, 1, 1,26, 1, 1, 1,12, 1, 1, 1, 1,26,
    27,27, 1, 1, 1, 1, 1, 1,16, 3, 3, 3, 3, 3, 3, 3, 3,16, 1,
    25, 1, 1, 1, 1,14, 1, 1, 1,16, 1, 1,12, 1, 1,
    25, 1, 1, 1, 1,18, 1, 1, 1,18, 1, 1, 1, 1,18,
    28,28,28, 1, 1,18,18,18,18,18, 1, 1,28,28,28,25,
};
```

2) Mapping

The map design of a considerable part of mobile games is similar to brick structure mapping of the array. In order to simplify the process of its drawing, Sun released TiledLayer Class in MIDP2.0. A TiledLayer object contains all the image resources of blocks, as well as a map matrix. All images of blocks can be stored in a big PNG picture, by specifying the width and height of each tile, TiledLayer can automatically split the image resources of these bricks. To change the map cell, it only need to use TiledLayer.SetCell(int col, int row, int tileIndex) method to design the cell image index, and it is merely LayerManager work to handle the mapping.

```java
public void drawMaps(){//Mapping to the buffer zone
    for(int i = 0;i < total;i++) {
        int mapValue = map[i];
        int column = i%columns;// To obtain the current rows or columns
        int row = (i - column) / columns;
        if(mapValue == 2)
            groundTL.setCell(column, row,2);// To fill the surface of the road
        else if(mapValue == 3)
            groundTL.setCell(column, row,3);
        else
            groundTL.setCell(column, row,1); // To fill the surface of the grass plot
        if(mapValue > 20)
            boxTL.setCell(column, row,(map[i] - 20));// To fill the building layer
        else if(map[i] > 10)
            barrierTL.setCell(column, row,(map[i] - 10));
        }
    }
}
```

For this game, TiledLayer has not only solved the problem of mapping, through the flexible design, but also achieved animation effect of bricks breaking up and bubble explosion.

C. Game Scene

In this game, the scenes call MainCanvas achieves the function of thread, and is also the game screen, in charge of checking user’s input, realizing the game logic, and drawing the updated user interface.

MainCanvas class is built as an open stage at initial time, when informed of start() it begin to arrange the scene layout. In the process the roles of the game (the wizard), the background (map), the props and the bomb are added in turn. Once the thread starts, the state machine and the keyboard input will update the states of objects together, and finally draw all the states by the method of draw().

The main interface of the game is shown in Figure 9.

In MainCanvas class, besides run() method, the definition of the logic(), drawTime(), CheckCollision() methods are also included. The UML diagram of MainCanvas is shown in Figure 10.

D. Game Props

In the game development, we have designed PropSprite Class to create objects of the props, which inherited from Sprite, and the props have two attributes, the visibility and location. In PropSprite, it defines a variety of methods for the prop object to display or hide itself. The effect of the props is shown in Figure 11, and resources of the game props are shown in Figure 12.

E. Bombs (Bubbles)

This game program designed a Bomb Class as an abstract of bomb, which inherited from PropSprite Class and is the subclass of props Class. Bomb has some unique attributes such as visibility, position, power, delay, the player to lay it and state, having their own explosion, detonating other bombs, clearing effect of the explosion and so on.
1) The creation and recycle of bomb

Bomb is created by the role’s release, and disappears with the completion of explosion. The maximum number of bombs in the screen depends on the number of the roles and their capacity of shells carrying. In the screen there may not have a bomb, but also each cell may have a bomb, so the game should generate the dynamic bombs. Java is object-oriented programming language, and to create and release the object will occupy a large amount of resources, whereas in Java in many cases it can not be achieved if no objects used. However, in the J2ME environment, memory is seriously limited, so that the garbage collection can’t be implemented excessively. Once the garbage collection implemented, the program will stagnate, and so many objects generated will lead to memory swell greatly. Once the memory exhausted, the program will collapse. Therefore, the procedure should make full use of the distributed things beforehand, rather than distribution in using. In the program, the technology of the object-pool is used, which the objects that needed to generate dynamically are generated stately at first, and then be recycled at the right time, which is an effective solution to the performance loss brought by the creation and release objects.

2) Explosion, detonation and removal effect of explosion

The process of bomb explosions is divided into the following three steps:

a) draw the effect of explosion
b) damage the props within the scope of explosion
c) detonate the bombs within the scope of explosion.

In the procedure, it is difficult to realize the effect of explosion by wizard, so here the realization of the explosion effect is achieved by modifying the cells of explosion layer (blastTL). Therefore, Explosion Class is defined separately in the procedure for the realization of explosion, detonation, and the removal effect of explosion, in which a clock variable is defined to record the lasting time of explosion, explode() method to realize the effect of explosion, and the clearExplode() to remove the explosion effect. The effect of explosion, detonation and the removal effect of explosion are shown in Figure 13.

(a) Explosion  (b) Detonation  (c) Removal

Figure 13. Effect of Bomb

Explosion.explode() method is first to change the effect explosion of explosion center.

blastTL.setCell(col, row, 1);

The next is to the update the shock wave from the four directions, and in each direction, cycle is used to traverse all the cells within the scope of explosion. When the cells are passable, draw the shock wave, while the cells are impassable, the Map object is informed to destroy them and interrupt the cycle.

Map.destroy(int row, int col) method is used to destroy buildings within the designated cells, besides the destruction of the block buildings, the shock wave of explosion will detonate other bombs within the scope, which will be handed over to Explosion.detonate(int row, int col) to deal with.

F. Games Role

The states of the game’s roles themselves are rather complicated, and almost are interactive with all objects in game scenes, which is the core of the game. The roles have four directions which include right direction, left direction, forward and backward, and each direction have three moving-states controlled by the direction keys of...
mobile phone. The elements resource of the protagonist is shown in Figure 14.

![Figure 14. The Elements s Resource of the Protagonist](image)

Form the Table 1 to Table 3, we can find the procedure performance of the server is so good, but the Bubble place and detonation times is longer, it need to progress in the future work. Anotherfuture work is to progress the performance of the server.

IV. EXPERIMENT RESULT AND FUTURE WORK

In order to prove our work is meaningful and useful for other online mobile game development. We compared the BnB game and have test the performance for detonion on the client and the communications time between the clients with the server, the results show as in Table 1 and Table 2, and the Table 3 shows the procedure performance of the server.

![Figure 15. The States Conversion of the Roles](image)

**REFERENCES**


[9] Yang Yu, Zhu Li, Larry Shi, Ethan Yi-Chiun Chen, and Hua Xu, MyRole is a larger class in the procedure, in which many variables and methods are defined to describe the states and attributes of the roles. The UML diagram of MyRole is shown in Figure 15.