

The Algorithm of Gravity Master Sensor Game Based on the Android Platform

Yang ahui

School of Computer Engineering, Jimei University,
Xiamen, China
yyyah@jmu.edu.cn

*Pu yunming

School of Computer Engineering, Jimei University,
Xiamen, China
yunmingpu@163.com

Abstract—The ball game is designed with 3D physical sensor technology based on the Android platform. There are three modules in the game, the presentation layer module, the background logic module and mobile phone posture calculation module, and the classes in the module are introduced, the algorithm of gravity master transducer are Analyzed, Sensor technology switch the game playing mode from the original mobile phone keypad or touch operation to a more liberal rejection, shake, turning and other space operations. As the experiments, the application of gravity master transducer ensures the game more attractive.

Keywords—component; Algorithm, Gravity Master Sensor, Android Platform, Sensor Computing model

I. INTRODUCTION

Android is a Google's mobile platform based on Linux kernel. The platform is a true mobile development platform which consists of operating system, middleware, user interface and application software. In 2007, Google officially announced the cooperation with other thirty three mobile phone manufacturers, hardware and software providers, mobile phone chip suppliers and mobile operators, which means the foundation of Open Handset Alliance. The alliance is going to set up a standardized and open software platform in the mobile industry [1]. In terms of software, Android, as a development platform for mobile devices, is composed of operating system, middleware and application. Android software can be divided into the following levels from top to bottom [2].

- (1)Application
- (2)Application Framework
- (3)A variety of Libraries and the Android operating environment (RunTime)
- (4)Layer of the operating system (OS)

In generally, the Android application consists of Activity, Broadcast Intent to Receiver, Service and Content Provider. But they are not all necessary in all programs, for example, a simple HelloWorld program is only related with Activity[3].

There's an XML description file in the Android project, named the AndroidManifest.xml. In this file, the above four categories of the component name should be mentioned, and

the description of necessary features and requirements for each component should also be involved [4].

II. THE REQUIREMENT OF THE BALL GAME

A. The Requirement of the game

This game is gravity sensitive. The ball is subject to gravity. The ball moves along with the movement of the mobile phone. During the course, the ball will encounter various obstacles, such as traps, inconsecutive road, and artillery, etc.

When the game starts, the welcome screen is firstly presented, which consists of three spherical buttons, they are start, on / off sound and quit.

The player should change the position of the mobile phone to control the movement of the ball. The background of the game interface is gradually moving stars and meteorites nebula. When players makes the ball reach the destination, the game will prompt "Great, click to the next level". Then the player will enter the next adventure. And the game will display "Congratulations on passing the test". Then the click will lead to the main menu. There are many obstacles in the game interface, when the small ball touched one of these obstacles, the life will decrease by one, and the small ball is to restart. When the life exhausted, the game will prompt 'Unfortunately, the ball disappeared into the vast universe ...', and the game is voer.

Within playing, click the 'menu' button on the phone keypad, the menu of the game will pop-up. The player can also choose to continue the game, on / off sound, help information, or return to the main menu.

B. The architecture of the game

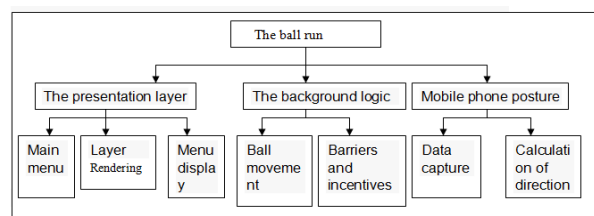


Fig.1 The Architecture of the game

The overall structure of the game is the structure of the various functional modules of the game, including the presentation layer module, the background logic module, mobile phone posture calculation module, the functional structure of each module [5], as shown in Figure 1.

The presentation module is responsible for rendering the game UI, including drawing the main menu, the drawing of each layer in the interface of the game and the game menu drawing.

Background logic module is mainly used to control the movement of the ball. It is also responsible for the detection of ball collision with the obstacles and incentives, and further action for these movements.

Phone posture calculation module includes a data capture module and a calculation module. The data capture module presents the change of the posture, the calculation module will present the direction of the ball by calculating the data captured.

C. The class structure of the ball game

The class structure in the presentation layer is shown in Figure 2. The presentation layer provides data model for the UI rendering, to modify the data model, or to draw the data model.

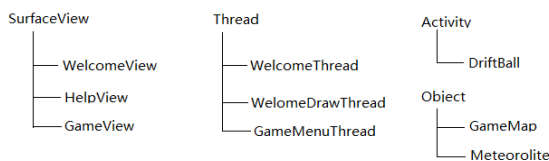


Fig.2 The structure of the presentation module

The WelcomeDrawThread and WelcomeThread are a subsidiary of the Thread class of WelcomeView. WelcomeDrawThread serves to redraw WelcomeView at a regular time, while WelcomeThread serves to modify the contents.

HelpView class is relatively simple, which is responsible for displaying the help information.

GameView class is inherited from SurfaceView, serves to render of the main game screen, while DrawThread serves to redraw GameView at a regular time.

GameMenuThread is responsible for the slide / slide-out effect in the game, which means that every time you click the menu button, the GameMenuThread is set up.

GameMap class is responsible for providing the map in the game. It stores all the information of checkpoints, and externally provide methods used to obtain the map information of specified checkpoints.

Meteorolite class inherits from Object. Each of Meteorolite object represents a meteorite. A number of meteorites will be displayed in the interface of the game.

Meteorolite class encapsulates the meteorite location and the animation frame index information.

DriftBall class inherits from Activity, mainly responsible for the switch between the various Views, and for handling the user clicks on the screen event.

The class structure of the background logic module is shown in Figure 3. It consists of two thread classes which are inherited from two Object subclasses which are developed independently.

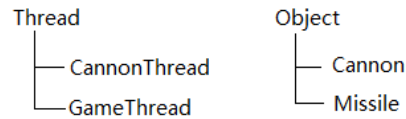


Fig. 3 the structure of background logic module

Cannon class inherits from Object, which encapsulates the artillery position, range and other information, as well as the methods used to launch shells. CannonThread is inherited from the Thread class, whose main function is to monitor the position of the ball. When the ball falls into its range, it would be fired .

Missile class belongs to shells class. Its object is created by the Cannon object in CannonThread. Each of the Missile object encapsulates its direction, coordinates, as well as the method, which is responsible for moving the shells and detect the collision.

The GameThread is in charge of two things, to modify GameView in meteorites and traps animation frame index, and to move the ball according to the direction, and the corresponding collision detection and processing.

The class structure of the cell phone posture calculation module is shown in Figure 4. The module only contains the BallListener class which inherited from SensorListener and the self-developed RotateUtil class.

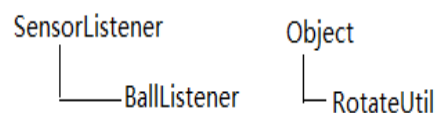


Fig. 4 The structure of the posture calculation module

(1) BallListener is inherited from SensorListener, which is responsible for the monitoring changes parameter of the posture, and transmitting the changes to RotateUtil for further analysis and calculation.

(2) RotateUtil is responsible for receiving the phone posture data from BallListener, and calculating the direction of the movement of the ball.

III. THE DESIGN OF PRESENTATION LAYER AND BACKGROUND LOGIC

In main menu, after click the start button, the Handler message was sent to DriftBall to create class Gameview,

then switch to Gameview screen. There are many classes in the game UI, they are GameView, GameThread, and GameMenu. The Gameview is the core class of UI, which is responsible for the screen display and game control.

There are four levels in the ball game, including Map, Star space, Meteorolite, Nebula. In the map coverage, the GameMap is the important class, it declares a static 3-dimension byte array, which has five 2-dimension arrays, used for store map data. The following is declaring of GameMap class:

```
public class GameMap{

    static byte [][] map = { //map, 0is null, 1 is road,2 is
    home,3 is life added, 4 is ball burning,5is trap,6 is gun

    { //first layer
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        .....//some map data omitted

        {0,0,0,6,1,1,1,1,1,1,1,1,1,1,1,1,0,0},
        {0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0}

    },
```

In this game, the background logic module is served for the movement of the ball and the collision detection module. It also includes a control module on the process of the game.

The movement of the ball is realized by the checkAndMoveBall of GameThread. Its principle is to detect the movement direction of the ball in the x and y directions. After each move on the x or y direction, it will detect if the collision happens, if yes, the movement is revoked. The movement direction of the ball is shown in Figure 5. The four directions, up, down, left and right, are represented by 0,2,4,6 respectively, while the rest by 1,3,5,7.

Each collision detection in checkAndMoveBall, the member method of GameThread named checkCollision will be called. A Boolean value will be returned, if the value "true", the successful collision.

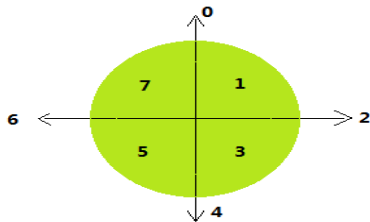


Fig. 5 the directions of ball game

In the collision detection, the corresponding column and row in the map matrix is calculated which is used to identify whether the collision will happen or not, and the primitive type is 0, If yes, the value of "true" is returned.

IV. THE ALGORITHM OF SENSOR COMPUTING MODULE

A. the structure of the module

The function of the sensor module is to capture the cell phone attitude, and according to the received data to calculate the direction of the ball, but more complex to implement. The module is related to BallListener and RotateUtil. The BallListener class inherits from SensorListener, and its main function is to monitor the phone's attitude changes. The onSensorChanged class fixed time intervals to obtain the phone gesture, and call analysis data method is calculated after the end of each calculation, the current system time recording to startTime members of the variable re-timing. Values array method will analysis and modify data, getDirectionCase calls the RotateUtil method to calculate the direction of the ball.

RotateUtil class captured phone posture data calculated from the direction of movement of the ball, these calculations involve some knowledge of mathematics and physics, but also more difficult to understand part of this game, here introduce RotateUtil calculation principles

B. The algorithm of Gravity Master Sensorr

Sensors capture the data of the mobile phone along the Yaw axis, Pitch-axis Roll-axis turning angle. The ball is affected by gravity and exercise, regardless of the phone how to flip the ball out along the component of the acceleration due to gravity on the phone screen. So just ask the projection of the acceleration of gravity vector on the phone screen, we can obtain the direction of movement of the ball. In phone gesture system, the direction of gravity acceleration vector is always contrary to the Yaw axis. Phone screen rotation in any plane, calculate the acceleration of gravity vector on the phone screen projection will be more difficult. So a reversal method to seek the vector projection is the key, the steps are as follows.

(1) Initial case, the phone has been carried out in any flip, the acceleration of gravity vector showed a certain angle with the cell phone screen plane.

(2) Maintain constant acceleration of gravity vector with the phone screen plane relative, which is equivalent to the acceleration of gravity vector welded to the mobile phone screen. Take the phone to turn back along the trajectory just turned, for example, when the phone was originally turned 30 ° around the Yaw axis counterclockwise, now along the Yaw axis clockwise 30 °.

(3) When the phone along the three directions of rotation are restored, it will be in the initial state, for example, Yaw, Pitch and Roll axis corresponds to the space coordinates z, x and y axis.

(4) At this point, the acceleration of gravity vector will be a vector intersects on the phone screens, the vector projected onto the x-y plane can be derived, according to the direction of the projection vector direction of movement of the ball.

The above algorithm is not a constant acceleration of gravity vector projection to the all of the phone screen.

Instead, the virtual projection of the acceleration due to gravity through the reversal of cell phone gesture to in the phone screen space coordinate system x-y plane. So that the main calculation is how the phone along the direction turned the angle of switching back. Here, you need to pay special attention to restore order along the three spatial directions, Yaw, Pitch and Roll axis of the rotation angle. This algorithm, the order must be for Yaw, Pitch and Roll. Through the icon below, these three axes of the angle of recovery instructions. Initial state as shown in Figure 6, the phone has three axes of rotation.

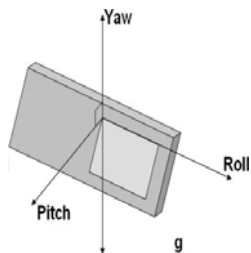


Fig. 6 Rotation initial State

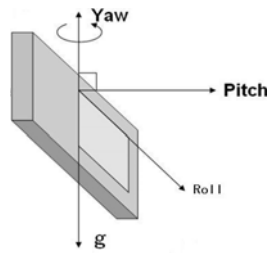


Fig. 7 Yaw Axis Recovery

Of course, Yaw axis always points to the vertical upward direction, Z-axis coordinate system and the standard space, it can be a negative directly to the rotation angle of the angle recovery, shown in Figure 7.

Along with the Yaw axis turned the angle, Pitch axis at this time becomes space coordinates in the x-axis shown in Figure 7.

Along with the Pitch axis (it has become a space coordinate system x-axis in this case) turned the angle of recovery, the Roll axis on the amendments to the y-axis in order to space coordinates, shown in Figure 8. Finally, according to the y-axis will be turned to the angle of recovery, then the phone screen where the plane becomes the space coordinates in the x-y plane, attached to the acceleration of gravity in the plane of cell phone is a phone plane intersecting vectors which shown in Figure 9.

Later, projecting the vector onto the phone plane, and calculating the ball rolling direction through the projection point. If you do not change the angle in accordance with the above order recovery, the calculation of the spatial coordinates of conversion would be very complicated. Each step of the angle of recovery is based on the standard spatial coordinate system axis.

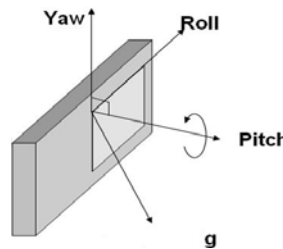


Fig. 8 Pitch Axis Recovery

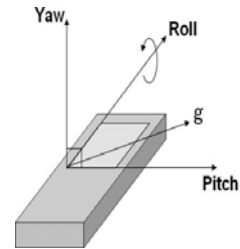


Fig. 9 Roll Axis Recovery

V. THE RESULTS OF THE EXPERIMENTS

Run the game, the Welcome screen showed in Figure 10. Click the Start button, and enter the loading interface.



Fig.10 The Running of the game

ACKNOWLEDGMENT (HEADING 5)

The author would like to thank for the funding and support from the science project of Fujian province, China , Xiamen science bureau support.

REFERENCES

- [1] Yafeng Wu, Yaguang Shu. 2010, Android2.0 game developers combat Collection. People's Posts and Telecommunications Press.
- [2] Zhi-Long Yu, Yu-Hsun Chen, Mingjie Zheng, and Xiaofeng Chen. 2009, the complement of the development of the Android SDK example. People's Posts and Telecommunications Press, July 2009.
- [3] Yan Jin, Shanglong Yao. 2009, Google Android SDK Getting Started with the actual combat. People's Posts and Telecommunications Press,
- [4] Fengsheng Yang. 2010, Uncover the secrets of the Android application development, Mechanical Industry Press,
- [5] Super Korea, Quan Liang. 2010, the principles and development points of the Android system. Press of Electronics Industry,
- [6] Ai Shier (U.S.), 2007, Thinking in java. The fourth Edition. Mechanical Industry