

XI`AN TECHNOLOGICAL UNIVERSITY

实验报告

实验课程名称 ANDROID应用程序开发

专 业： 物联网工程

班 级： 16060616

姓 名： 田宇龙

学 号： 16060616107

实验学时： 4

指导教师： 周江卫

成 绩：

2018 年 11 月 23 日

西安工业大学实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 专业 | 物联网工程 | 班级 | 16060616 | 姓名 | 田宇龙 | 学号 | 16060616107 |
| 实验课程 | Android应用程序开发 | 指导教师 | 周江卫 | 实验日期 | 2018.11.23 | 同实验者 |  |
| 实验项目 |  | | | | | | |
| 实验设备及器材 | Pc机 | | | | | | |

**一：实验目的**

贪食蛇的设计与实现

**二：实验内容**

布局文件

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="MainActivity">

<Button

android:id="@+id/buttonRight"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_alignParentRight="true"

android:text="→" />

<Button

android:id="@+id/buttonDown"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/buttonRight"

android:layout\_toLeftOf="@+id/buttonRight"

android:text="↓" />

<Button

android:id="@+id/buttonLeft"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/buttonDown"

android:layout\_toLeftOf="@+id/buttonDown"

android:text="←" />

<Button

android:id="@+id/buttonUp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/buttonDown"

android:layout\_toRightOf="@+id/buttonLeft"

android:text="↑" />

<Button

android:id="@+id/buttonPause"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/buttonLeft"

android:layout\_alignParentLeft="true"

android:layout\_toLeftOf="@+id/buttonLeft"

android:text="暂停" />

<Button

android:id="@+id/buttonStart"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/buttonUp"

android:layout\_alignLeft="@+id/buttonPause"

android:layout\_alignRight="@+id/buttonPause"

android:text="开始" />

<TextView

android:id="@+id/textView\_Score"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/buttonStart"

android:layout\_alignParentLeft="true"

android:text="分数：0 最高分数：0"

android:textAppearance="?android:attr/textAppearanceLarge" />

<com.example.myapplication.SnakeView

android:id="@+id/snakeView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_above="@+id/textView\_Score"

/>

</RelativeLayout>

Mainactivity.java:

package com.example.myapplication;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

Button button\_start;

Button button\_pause;

Button button\_up;

Button button\_down;

Button button\_left;

Button button\_right;

TextView textview\_score;

SnakeView snakeview;

int score = 0;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

button\_start =(Button) findViewById(R.id.buttonStart);

button\_pause =(Button) findViewById(R.id.buttonPause);

button\_up =(Button) findViewById(R.id.buttonUp);

button\_down =(Button) findViewById(R.id.buttonDown);

button\_left =(Button) findViewById(R.id.buttonLeft);

button\_right =(Button) findViewById(R.id.buttonRight);

textview\_score =(TextView)findViewById(R.id.textView\_Score);

snakeview = (SnakeView)findViewById(R.id.snakeView);

button\_start.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

snakeview.StartGame();

}

});

button\_pause.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

snakeview.PauseGame();

}

});

button\_down.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

snakeview.ControlGame(SnakeView.DIR\_DOWN);

}

});

button\_up.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

snakeview.ControlGame(SnakeView.DIR\_UP);

}

});

button\_right.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

snakeview.ControlGame(SnakeView.DIR\_RIGHT);

}

});

button\_left.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

snakeview.ControlGame(SnakeView.DIR\_LEFT);

}

});

snakeview.setOnSnakeDeadListener(new SnakeView.OnSnakeDeadListener() {

@Override

public void OnSnakeDead(int foodcnt) {

Toast.makeText(MainActivity.this,"游戏结束，您的本次得分为："+Integer.toString(score),Toast.LENGTH\_LONG).show();

}

});

snakeview.setOnSnakeEatFoodListener(new SnakeView.OnSnakeEatFoodListener() {

@Override

public void OnSnakeEatFood(int foodcnt) {

score++;

textview\_score.setText("分数："+Integer.toString(score)+" 最高分："+Integer.toString(score));

}

});

}

}

SnakeView.java:

package com.example.myapplication;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.Point;

import android.os.Handler;

import android.os.Message;

import android.util.AttributeSet;

import android.util.Log;

import android.view.View;

import java.util.Random;

import java.util.Timer;

import java.util.TimerTask;

/\*\*

\* Created by Lenovo on 2016/9/22.

\*/

public class SnakeView extends View {

private int blocksize = 40;

private int halgblocksize = 20;

private int width,height;

private int offsetX,offsetY;

private int snakeLen;

Point[] snakePoints = new Point[100];

private int snakeDir;

private Point food = new Point();

Paint ptBackground = new Paint();

Paint ptHead = new Paint();

Paint ptBody = new Paint();

Paint ptFood = new Paint();

Paint ptBorder = new Paint();

static public final int DIR\_UP = 0;

static public final int DIR\_RIGHT = 1;

static public final int DIR\_DOWN = 2;

static public final int DIR\_LEFT = 3;

private Timer mTimer = null;

private TimerTask mTimerTask = null;

private Handler mHandler = null;

private final int SNAKE\_MOVE = 1;

private int mFoodCnt;

private int mGameStatus;

private final int STATUS\_RUN = 1;

private final int STATUS\_DEAD = 2;

private final int STATUS\_PAUSE= 3;

public SnakeView(Context context, AttributeSet attrs, int defStyle) {

super(context, attrs, defStyle);

// TODO Auto-generated constructor stub

InitGame();

}

public SnakeView(Context context, AttributeSet attrs) {

super(context, attrs);

// TODO Auto-generated constructor stub

InitGame();

}

public SnakeView(Context context) {

super(context);

// TODO Auto-generated constructor stub

InitGame();

}

public void InitSnake(){

snakeLen = 4;

for(int i=0;i<100;i++)snakePoints[i] = new Point();

snakePoints[0].x= 3;

snakePoints[0].y = 0;

snakePoints[1].x = 2;

snakePoints[1].y = 0;

snakePoints[2].x = 1;

snakePoints[2].y = 0;

snakePoints[3].x = 0;

snakePoints[3].y = 0;

food.x = 4;

food.y = 4;

snakeDir = DIR\_RIGHT;

mFoodCnt = 0;

snakeDir = DIR\_RIGHT;

mGameStatus = STATUS\_PAUSE;

}

public void InitGame()

{

ptBackground.setColor(Color.argb(255, 0, 0, 0));

ptHead.setColor(Color.argb(255, 255, 0, 0));

ptBody.setColor(Color.argb(255, 255, 211, 55));

ptBorder.setColor(Color.argb(255, 218, 178, 115));

ptFood.setColor(Color.argb(255, 0, 11, 255));

InitSnake();

mHandler = new Handler()

{

@Override

public void handleMessage(Message msg)

{

switch (msg.what)

{

case SNAKE\_MOVE:

SnakeMove();

break;

default:

break;

}

}

};

if (mTimer == null)

{

mTimer = new Timer();

}

if (mTimerTask == null)

{

mTimerTask = new TimerTask()

{

@Override

public void run()

{

Message message = new Message();

message.what = SNAKE\_MOVE;

mHandler.sendMessage(message);

}

};

}

if(mTimer != null && mTimerTask != null )

mTimer.schedule(mTimerTask, 300, 300);

}

@Override

protected void onDraw(Canvas canvas) {

// TODO Auto-generated method stub

super.onDraw(canvas);

canvas.drawRect(offsetX, offsetY,

(width)\*blocksize+offsetX, (height)\*blocksize+offsetY, ptBackground);

canvas.drawRect(offsetX-halgblocksize, offsetY-halgblocksize,

width\*blocksize+offsetX+halgblocksize, offsetY, ptBorder);

canvas.drawRect(offsetX-halgblocksize, offsetY-halgblocksize,

offsetX, height\*blocksize+offsetY+halgblocksize, ptBorder);

canvas.drawRect(width\*blocksize+offsetX, offsetY-halgblocksize,

width\*blocksize+offsetX+halgblocksize, height\*blocksize+offsetY+halgblocksize, ptBorder);

canvas.drawRect(offsetX-halgblocksize, height\*blocksize+offsetY,

width\*blocksize+offsetX+halgblocksize, height\*blocksize+offsetY+halgblocksize, ptBorder);

canvas.drawRect(food.x\*blocksize+offsetX, food.y\*blocksize+offsetY,

(food.x+1)\*blocksize+offsetX, (food.y+1)\*blocksize+offsetY, ptFood);

for(int i = 0; i< snakeLen; i++)

{

if(i ==0)

{

canvas.drawRect(snakePoints[i].x\*blocksize+offsetX,snakePoints[i].y\*blocksize+offsetY,

(snakePoints[i].x+1)\*blocksize+offsetX, (snakePoints[i].y+1)\*blocksize+offsetY, ptHead);

}

else

{

canvas.drawRect(snakePoints[i].x\*blocksize+offsetX, snakePoints[i].y\*blocksize+offsetY,

(snakePoints[i].x+1)\*blocksize+offsetX, (snakePoints[i].y+1)\*blocksize+offsetY, ptBody);

}

}

}

@Override

protected void onSizeChanged(int w, int h, int oldw, int oldh) {

// TODO Auto-generated method stub

super.onSizeChanged(w, h, oldw, oldh);

width = w / blocksize - 1;

height = h / blocksize - 1;

offsetX = (w - width\*blocksize) / 2;

offsetY = (h - height\*blocksize) / 2;

}

public interface OnSnakeEatFoodListener {

void OnSnakeEatFood(int foodcnt);

}

public interface OnSnakeDeadListener {

void OnSnakeDead(int foodcnt);

}

private OnSnakeEatFoodListener mOnSnakeEatListener;

private OnSnakeDeadListener mOnSnakeDeadListener;

public void setOnSnakeEatFoodListener(OnSnakeEatFoodListener listener)

{

this.mOnSnakeEatListener=listener;

}

public void setOnSnakeDeadListener(OnSnakeDeadListener listener)

{

this.mOnSnakeDeadListener=listener;

}

public void StartGame()

{

switch(mGameStatus)

{

case STATUS\_DEAD:

InitSnake();

mGameStatus = STATUS\_RUN;

if(mOnSnakeEatListener != null)

mOnSnakeEatListener.OnSnakeEatFood(mFoodCnt);

break;

case STATUS\_PAUSE:

mGameStatus = STATUS\_RUN;

break;

default:

break;

}

}

public void PauseGame()

{

if(mGameStatus == STATUS\_RUN)

{

mGameStatus = STATUS\_PAUSE;

}

}

public void ControlGame(int dir)

{

if(mGameStatus != STATUS\_RUN)

return;

switch(dir)

{

case DIR\_UP:

case DIR\_RIGHT:

case DIR\_DOWN:

case DIR\_LEFT:

snakeDir = dir;

break;

default:

break;

}

}

public void SnakeMove()

{

if(mGameStatus != STATUS\_RUN)

return;

Point point=new Point();

switch(snakeDir) {

case 0:

point.x = snakePoints[0].x;

point.y = snakePoints[0].y - 1;

break;

case 1:

point.x = snakePoints[0].x + 1;

point.y = snakePoints[0].y;

break;

case 2:

point.x = snakePoints[0].x;

point.y = snakePoints[0].y + 1;

break;

case 3:

point.x = snakePoints[0].x - 1;

point.y = snakePoints[0].y;

break;

}

if(point.x <0 ||point.x >= width ||

point.y <0 ||point.y >= height)

{

//Game Over

mGameStatus = STATUS\_DEAD;

if(mOnSnakeDeadListener != null)

mOnSnakeDeadListener.OnSnakeDead(snakeLen);

return;

}

if(point.x == food.x && point.y == food.y)

{

//Eat mFood

Random random = new Random();

food.x = random.nextInt(width-1);

food.y = random.nextInt(height-1);

snakeLen++;

mFoodCnt++;

if(mOnSnakeEatListener != null)

mOnSnakeEatListener.OnSnakeEatFood(mFoodCnt);

}

for(int i= snakeLen-1; i>0; i--)

{

snakePoints[i].x = snakePoints[i-1].x;

snakePoints[i].y = snakePoints[i-1].y;

}

snakePoints[0].x = point.x;

snakePoints[0].y = point.y;

invalidate();

}

}

**三：实验运行效果：**

