package com.hypermotion2d;

import android.content.Context;

import android.media.AudioManager;

import android.media.SoundPool;

import android.opengl.GLSurfaceView;

import javax.microedition.khronos.egl.EGLConfig;

import javax.microedition.khronos.opengles.GL10;

import com.roxiga.hypermotion2d.\*;

public class HyperMotion2D implements GLSurfaceView.Renderer {

private Context \_context;

public int \_width;

public int \_height;

public boolean \_touch;

private float \_xPos;

private float \_yPos;

private Sprite2D \_title = new Sprite2D();

private Sprite2D \_earth = new Sprite2D();

private static final int ENEMY\_NUM = 8;

public static final int BOMB\_NUM =8;

private Sprite2D[] \_enemy = new Sprite2D[ENEMY\_NUM];

private Sprite2D[] \_bomb = new Sprite2D[BOMB\_NUM];

private Vector2D[] \_enemyDelta = new Vector2D[ENEMY\_NUM];

public static final int GS\_TITLE=0;

public static final int GS\_MAIN=1;

public static final int GS\_GAMEOVER=2;

public int \_gameState = GS\_TITLE;

private float \_frequency;

private Sprite2D \_ship = new Sprite2D();

public Vector2D \_shipDelta = new Vector2D(0,0);

private static final int XY\_SIZE = 64;

private static SoundPool \_spExplode;

private int \_soundID;

public Sprite2D \_explode = new Sprite2D();

public Sprite2D \_explode2 = new Sprite2D();

//@override

public HyperMotion2D(Context context){

\_context = context;

\_spExplode = new SoundPool(1,AudioManager.STREAM\_MUSIC,0);

\_soundID = \_spExplode.load(context, R.raw.explode,1);

}

//@Override

public void onDrawFrame(GL10 gl){

gl.glClear(GL10.GL\_COLOR\_BUFFER\_BIT | GL10.GL\_DEPTH\_BUFFER\_BIT);

switch(\_gameState){

case GS\_TITLE:

\_title.draw(gl,getRatio());

break;

case GS\_MAIN:

\_earth.draw(gl,getRatio());

enemyMove();

bombMove();

shipMove();

enemyDraw(gl);

bombDraw(gl);

\_ship.draw(gl);

explode2Draw(gl);

break;

case GS\_GAMEOVER:

enemyDraw(gl);

bombDraw(gl);

\_ship.draw(gl);

explodeDraw(gl);

if(\_explode.\_width >350){

\_gameState = GS\_TITLE;

}

\_earth.draw(gl,getRatio());

break;

}

}

private float getRatio() {

return (float)\_width/600.0f;

}

private void enemyMove(){

int i;

for(i=0; i<\_enemy.length;i++){

\_enemy[i].\_pos.\_x += 8\*(Math.random()-Math.random())+\_enemyDelta[i].\_x;

\_enemy[i].\_pos.\_y += Math.random()-Math.random()\*2 + \_enemyDelta[i].\_y;

\_enemyDelta[i].\_x \*= 0.9;

\_enemyDelta[i].\_y \*= 0.9;

if(\_enemy[i].\_pos.\_x <0){

\_enemy[i].\_pos.\_x=0;

\_enemyDelta[i].\_x \*= -1;

}

if(\_enemy[i].\_pos.\_x > \_width-XY\_SIZE){

\_enemy[i].\_pos.\_x=\_width - XY\_SIZE;

\_enemyDelta[i].\_x \*= -1;

}

if(\_enemy[i].\_pos.\_y<0){

\_explode.\_pos.\_x=\_enemy[i].\_pos.\_x;

\_explode.\_pos.\_y=\_enemy[i].\_pos.\_y;

\_explode.\_visible =true;

\_gameState = GS\_GAMEOVER;

\_spExplode.play(\_soundID, 1.0F, 1.0F, 0, 0, 1.0F);

}

float x = \_ship.\_pos.\_x - \_enemy[i].\_pos.\_x;

float y = \_ship.\_pos.\_y - \_enemy[i].\_pos.\_y;

float x2 = \_shipDelta.\_x;

float y2 = \_shipDelta.\_y;

if(x\*x+y\*y<50\*50){

if(x2\*x2+y2\*y2>1\*1){

\_enemyDelta[i].\_x = \_shipDelta.\_x;

\_enemyDelta[i].\_y = -\_shipDelta.\_y;

\_shipDelta.\_x \*= -1;

\_shipDelta.\_y \*= -1;

\_spExplode.play(\_soundID, 1.0F, 1.0F, 0, 0, 1.0F);

}

else{

\_enemy[i].\_pos.\_x += x/2;

\_enemy[i].\_pos.\_y -= y/2;

}

}

}

}

private void enemyDraw(GL10 gl){

for(int i=0; i< \_enemy.length;i++){

\_enemy[i].draw(gl);

}

}

private void bombMove(){

int i;

for(i=0;i<\_bomb.length;i++){

\_bomb[i].\_pos.\_x += 10\*(Math.random()-Math.random());

\_bomb[i].\_pos.\_y += Math.random()-Math.random()\*1.8f;

if(\_bomb[i].\_pos.\_x<0){

\_bomb[i].\_pos.\_x=0;

}

if(\_bomb[i].\_pos.\_x > \_width-XY\_SIZE){

\_bomb[i].\_pos.\_x = \_width - XY\_SIZE;

}

if(\_bomb[i].\_pos.\_y<0){

\_explode.\_pos.\_x=\_bomb[i].\_pos.\_x;

\_explode.\_pos.\_y=\_bomb[i].\_pos.\_y;

\_explode.\_visible = true;

\_gameState = GS\_GAMEOVER;

\_spExplode.play(\_soundID, 1.0F, 1.0F, 0, 0, 1.0F);

}

float x = \_ship.\_pos.\_x - \_bomb[i].\_pos.\_x;

float y = \_ship.\_pos.\_y - \_bomb[i].\_pos.\_y;

if(x\*x + y\*y < 50\*50){

\_explode.\_pos.\_x = (\_ship.\_pos.\_x + \_bomb[i].\_pos.\_x)/2;

\_explode.\_pos.\_x = (\_ship.\_pos.\_y + \_bomb[i].\_pos.\_y)/2;

\_explode.\_visible = true;

\_spExplode.play(\_soundID, 1.0F, 1.0F, 0, 0, 1.0F);

\_gameState = GS\_GAMEOVER;

}

for(int j=0;j<\_enemy.length;j++){

float x2 = \_bomb[i].\_pos.\_x - \_enemy[j].\_pos.\_x;

float y2 = \_bomb[i].\_pos.\_y - \_enemy[j].\_pos.\_y;

float x3 = \_enemyDelta[j].\_x;

float y3 = \_enemyDelta[j].\_y;

if(\_enemy[j].\_pos.\_y<\_height && \_bomb[i].\_pos.\_y <\_height && x2\*x2+y2\*y2<50\*50 && x3\*x3+y3\*y3<5\*5){

\_explode2.\_pos.\_x = (\_enemy[j].\_pos.\_x + \_bomb[i].\_pos.\_x)/2;

\_explode2.\_pos.\_y = (\_enemy[j].\_pos.\_y + \_bomb[i].\_pos.\_y)/2;

\_explode2.\_width = 32;

\_explode2.\_height = 32;

\_explode2.\_visible = true;

\_enemy[j].\_pos.\_x = (float)Math.random()\*(\_width - XY\_SIZE);

\_enemy[j].\_pos.\_y = \_height+100+(float)Math.random()\*\_frequency;

\_bomb[j].\_pos.\_x = (float)Math.random()\*(\_width - XY\_SIZE);

\_bomb[j].\_pos.\_y = \_enemy[j].\_pos.\_y+(float)Math.random()\*200;

if(\_frequency>100){

\_frequency-=10;

}

\_spExplode.play(\_soundID, 1.0F, 1.0F, 0, 0, 1.0F);

}

}

}

}

private void bombDraw(GL10 gl){

for(int i=0;i<\_bomb.length;i++){

\_bomb[i].draw(gl);

}

}

private void shipMove(){

\_ship.\_pos.\_x += \_shipDelta.\_x;

\_ship.\_pos.\_y -= \_shipDelta.\_y;

if(\_ship.\_pos.\_x<0){

\_ship.\_pos.\_x =0;

\_shipDelta.\_x \*=-1;

}

else if(\_ship.\_pos.\_x > \_width-XY\_SIZE){

\_ship.\_pos.\_x = \_width-XY\_SIZE;

\_shipDelta.\_x \*=-1;

}

else if(\_ship.\_pos.\_y < 0){

\_ship.\_pos.\_y = 0;

\_shipDelta.\_y \*= -1;

}

if(\_ship.\_pos.\_x > \_height-XY\_SIZE){

\_ship.\_pos.\_y = \_height-XY\_SIZE;

\_shipDelta.\_y \*=-1;

}

else{

\_shipDelta.\_x \*= 0.9f;

\_shipDelta.\_y \*= 0.9f;

}

}

private void explodeDraw(GL10 gl){

if(\_explode.\_visible){

\_explode.\_pos.\_x--;

\_explode.\_pos.\_y--;

\_explode.\_width+=2;

\_explode.\_height+=2;

}

\_explode.draw(gl);

}

private void explode2Draw(GL10 gl){

if(\_explode2.\_visible){

\_explode2.\_pos.\_x--;

\_explode2.\_pos.\_y--;

\_explode2.\_width+=2;

\_explode2.\_height+=2;

if(\_explode2.\_width>200){

\_explode2.\_visible = false;

}

\_explode2.draw(gl);

}

}

//@Override

public void onSurfaceCreated(GL10 gl, EGLConfig config) {

int i;

gl.glClearColor(0.6f,0.58f, 1.0f, 1.0f);

gl.glDisable(GL10.GL\_DITHER);

gl.glEnable(GL10.GL\_DEPTH\_TEST);

gl.glEnable(GL10.GL\_TEXTURE\_2D);

gl.glEnable(GL10.GL\_ALPHA\_TEST);

gl.glEnable(GL10.GL\_BLEND);

gl.glBlendFunc(GL10.GL\_SRC\_ALPHA, GL10.GL\_ONE\_MINUS\_SRC\_ALPHA);

\_title.setTexture(gl,\_context.getResources(),R.drawable.title);

\_title.\_texWidth = 600;

\_title.\_width = 600;

\_earth.setTexture(gl,\_context.getResources(),R.drawable.earth);

\_earth.\_texWidth = 600;

\_earth.\_width = 600;

\_ship.setTexture(gl, \_context.getResources(), R.drawable.ship);

for(i=0;i<\_enemy.length;i++){

\_enemy[i] = new Sprite2D();

\_enemy[i].setTexture(gl,\_context.getResources(), R.drawable.enemy);

}

for(i=0;i<\_bomb.length;i++){

\_bomb[i] = new Sprite2D();

\_bomb[i].setTexture(gl, \_context.getResources(), R.drawable.bomb);

}

\_explode.setTexture(gl,\_context.getResources(),R.drawable.explode);

\_explode2.setTexture(gl,\_context.getResources(),R.drawable.explode);

}

public void init(){

int i;

for(i=0;i<\_enemy.length;i++){

\_enemy[i].\_pos.\_x = (float)Math.random()\*(\_width-XY\_SIZE);

\_enemy[i].\_pos.\_y = 500+i\*300+(float)Math.random()\*100;

\_enemyDelta[i] = new Vector2D(0,0);

}

for(i=0;i<\_bomb.length;i++){

\_bomb[i].\_pos.\_x = (float)Math.random()\*(\_width-XY\_SIZE);

\_bomb[i].\_pos.\_y = \_enemy[i].\_pos.\_y+(float)Math.random()\*100;

}

\_explode.\_width = 32;

\_explode.\_height = 32;

\_explode.\_visible = false;

\_explode2.\_visible = false;

\_shipDelta = new Vector2D(0,0);

\_ship.\_pos.\_x=(\_width-XY\_SIZE)/2;

\_ship.\_pos.\_y=0;

\_frequency = 2000;

}

public void actionDown(float x, float y){

\_xPos=x;

\_yPos=y;

}

public void actionMove(float x,float y){

}

public void actionUp(float x,float y){

switch(\_gameState){

case GS\_TITLE:

init();

\_gameState = GS\_MAIN;

break;

case GS\_MAIN:

\_shipDelta.\_x = (x-\_xPos)/8;

\_shipDelta.\_y = (y-\_yPos)/8;

break;

case GS\_GAMEOVER:

if(\_explode.\_width >200){

\_gameState = GS\_TITLE;

}

break;

}

}

//@Override

public void onSurfaceChanged(GL10 gl, int width, int height) {

// TODO Auto-generated method stub

}

}