Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра ПУРИС

Лабораторная работа №9

по дисциплине «Программирование мобильных устройств»

Студент группы 0ВТб-1 Н.Д. Малышев

Преподаватель В.А. Тихомиров

2023

1. **Разработка приложения**

По заданию необходимо cделать так чтобы сверху, из случайных координат сыпятся прямоугольники случайных размеров.

В активности MyGdxGame (листинг 1.1) написан код для того, чтобы переключить экран на пользовтельскую процедуру.

Далее устанавливается пользовательский вид lab9 (листинг 1.2).

Листинг 1.1 – MyGdxGame

package com.lab9;  
  
import com.badlogic.gdx.Game;  
  
public class MyGdxGame extends Game {  
  
 @Override  
 public void create() {  
 //Переключение экрана на пользовательскую процедуру  
 setScreen(new lab9());  
 }  
}

Листинг 1.2 – lab9

package com.lab9;  
import com.badlogic.gdx.Gdx;  
import com.badlogic.gdx.Screen;  
import com.badlogic.gdx.graphics.GL20;  
import com.badlogic.gdx.graphics.OrthographicCamera;  
import com.badlogic.gdx.graphics.Texture;  
import com.badlogic.gdx.graphics.g2d.SpriteBatch;  
import com.badlogic.gdx.graphics.g2d.TextureRegion;  
import com.badlogic.gdx.math.Vector2;  
import com.badlogic.gdx.physics.box2d.Body;  
import com.badlogic.gdx.physics.box2d.BodyDef;  
import com.badlogic.gdx.physics.box2d.Box2DDebugRenderer;  
import com.badlogic.gdx.physics.box2d.ChainShape;  
import com.badlogic.gdx.physics.box2d.FixtureDef;  
import com.badlogic.gdx.physics.box2d.PolygonShape;  
import com.badlogic.gdx.physics.box2d.World;  
import com.badlogic.gdx.utils.Array;  
import java.util.ArrayList;  
public class lab9 implements Screen {  
 private World world;//переменная для управления миром  
 private Box2DDebugRenderer rend;//отладочный отрисовщик тел Мира  
 private OrthographicCamera camera;//видеокамера  
 private ArrayList<Body> rect = new ArrayList<>();//тело прямоугольника  
 Body w;  
 private int Nblock;  
 private float Nf;  
 SpriteBatch batch;  
 Texture recTexture;  
 Texture wallTexture;  
 TextureRegion region;  
 float width;  
 float height;  
 Vector2 pozB;  
 float alfa;  
  
 //Процедура создания тела прямоугольника  
 private Body createRect(){  
  
 BodyDef bDef= new BodyDef();//Структура геометрических свойств тела  
 bDef.type= BodyDef.BodyType.DynamicBody;//задать телу тип динамического тела  
 bDef.position.set((int)(Math.random()\*15f+1f), 14);//задать позицию тела в Мире – в метрах X и Y  
 Body rect =world.createBody(bDef);//создание тела в Мире  
 width = (float)(Math.random() + 0.2f) \* 2;  
 height = (float)(Math.random() + 0.3f) \* 2;  
  
 PolygonShape shape = new PolygonShape();//Создать эскиз контура тела в виде прямоугольника  
 shape.setAsBox(width,height);  
  
 rect.setUserData(new Vector2(width, height)); // сохранить размеры в пользовательских данных тела  
  
 //Структура физических свойств тела  
 FixtureDef fDef=new FixtureDef();  
 fDef.shape=shape;//вид контура тела  
 fDef.density=2; //плотность тела г/см3  
 fDef.restitution=0.7f;//упругость  
 fDef.friction=0.1f; //коэф-т трения  
 rect.createFixture(fDef);//свойства за телом  
  
 return rect;  
 }  
 //создания внешних стен  
 private void createWall() {  
 BodyDef bDef= new BodyDef();  
 bDef.type= BodyDef.BodyType.StaticBody;  
 bDef.position.set(0,0);  
  
 w = world.createBody(bDef);  
 ChainShape shape = new ChainShape();  
 shape.createChain(new Vector2[]{new Vector2(1,1), new Vector2(19,1)});  
  
 FixtureDef fDef=new FixtureDef();  
 fDef.shape=shape;  
 fDef.friction=0.1f;  
 w.createFixture(fDef);  
 }  
 @Override  
 public void show() {  
 world = new World(new Vector2(0,-10), true);//Создание нового мира – задан вектор гравитации в Мире  
 camera = new OrthographicCamera(20,15);//Создать камеру с охватом холста 20х15 метров  
 camera.position.set(new Vector2(10,7.5f),0);//Позиционировать камету по центру холста  
 camera.update();//Обновление состояния камеры  
 //rend = new Box2DDebugRenderer();//Создать отладочный отрисовщик  
 batch = new SpriteBatch();  
 recTexture = new Texture("badlogic.jpg");  
 wallTexture = new Texture("red.png");  
  
 createWall();//Создание внешних стен  
 }  
 @Override  
 public void render(float delta) {  
 Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);//Очистка экрана  
 batch.setProjectionMatrix(camera.combined);  
 //rend.render(world, camera.combined);  
 pozB=w.getPosition();  
 alfa=w.getAngle();  
 world.step(1 / 60f, 4, 4);  
 //Создание тел прямоугольников  
 if (Nblock < 1000) {  
 Nf += delta;  
 if (Nf > 0.9f) {  
 Body rectangle = createRect();  
 rect.add(rectangle);  
 Nblock += 1;  
 Nf = 0;  
 }  
 }  
 batch.begin();  
 //Отрисовка текстур прямоугольников  
 for (Body rectangle : rect) {  
 region = new TextureRegion(recTexture, 0, 0, recTexture.getWidth(), recTexture.getHeight());  
 Vector2 userDataVector = (Vector2) rectangle.getUserData();  
 // Получить значения ширины и высоты из Vector2 объекта  
 width = userDataVector.x;  
 height = userDataVector.y;  
 //batch.draw(region, rectangle.x, rectangle.y, rectangle.width, rectangle.height);  
 batch.draw(region, rectangle.getPosition().x - width, rectangle.getPosition().y - height,  
 width, height,  
 width\*2, height\*2,  
 1, 1,  
 rectangle.getAngle()\*180/3.14f);  
 }  
 //Отрисовка текстуры пола  
 region = new TextureRegion(wallTexture, 0, 0, wallTexture.getWidth(), wallTexture.getHeight());  
 batch.draw(region, pozB.x+1, pozB.y+0.7f, 1, 1,  
 18,0.3f, 1, 1, alfa\*180/3.14f);  
 batch.end();  
 }  
 @Override  
 public void resize(int width, int height) {}  
 @Override  
 public void pause() {}  
 @Override  
 public void resume() {}  
 @Override  
 public void hide() {}  
 @Override  
 public void dispose() {  
 //Удаление всех тел Мира  
 Array<Body> bodies = new Array<Body>();  
 world.getBodies(bodies);  
 for(int i = 0; i < bodies.size; i++) world.destroyBody(bodies.get(i));  
 rend.dispose();  
 world.dispose();  
 batch.dispose();  
 recTexture.dispose();  
 wallTexture.dispose();  
 }  
}

Работа приложения представлена на рисунке 1.1.



Рисунок 1.1 – Пример работы приложения

# Список использованных источников

1. РД ФГБОУ ВО «КнАГТУ» 013-2016. Текстовые студенческие работы. Правила оформления. – Введ. 2016-03-10. – Комсомольск-на-Амуре: ФГБОУ ВО «КнАГТУ», 2016. – 55 с.