Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №8

По дисциплине: «РПО»

Тема: “Простейшая игра”

Выполнил:

Студент 3 курса

Группы ПО-7

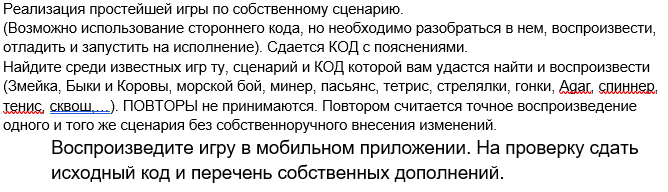
Комиссаров А.Е.

Проверил:

Козинский А.А.

2023

**Цель работы:** получить знания о создании игр на мобильных платформах на примере ОС Android.

** Ход работы.**

**Файл MainActivity.java**

**package com.example.lab10;**

**import android.os.Bundle;**

**import android.util.DisplayMetrics;**

**import android.view.Window;**

**import android.view.WindowManager;**

**import androidx.appcompat.app.AppCompatActivity;**

**public class MainActivity extends AppCompatActivity {**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN, WindowManager.LayoutParams.FLAG\_FULLSCREEN);**

**this.supportRequestWindowFeature(Window.FEATURE\_NO\_TITLE);**

**DisplayMetrics dm = new DisplayMetrics();**

**getWindowManager().getDefaultDisplay().getMetrics(dm);**

**Globals.SCREEN\_X = dm.widthPixels;**

**Globals.SCREEN\_Y = dm.heightPixels;**

**setContentView(new GamePanel(this));**

**}**

**}**

**Файл MainThread.java**

**package com.example.lab10;**

**import android.graphics.Canvas;**

**import android.view.SurfaceHolder;**

**public class MainThread extends Thread {**

**public static final int MAX\_FPS = 60;**

**private final SurfaceHolder surfaceHolder;**

**private GamePanel gamePanel;**

**private boolean running;**

**private static Canvas canvas;**

**private Object lock = new Object();**

**public MainThread(SurfaceHolder surfaceHolder, GamePanel gamePanel) {**

**super();**

**this.surfaceHolder = surfaceHolder;**

**this.gamePanel = gamePanel;**

**}**

**public void setRunning(boolean running) {**

**synchronized (lock) {**

**this.running = true;**

**}**

**}**

**public boolean isRunning() {**

**synchronized (lock) {**

**return this.running;**

**}**

**}**

**@Override**

**public void run() {**

**long targetTime = 1000/MAX\_FPS;**

**while(isRunning()) {**

**long startTime = System.nanoTime();**

**canvas = null;**

**try{**

**canvas = this.surfaceHolder.lockCanvas();**

**if (canvas != null) {**

**synchronized (surfaceHolder) {**

**this.gamePanel.update();**

**this.gamePanel.draw(canvas);**

**}**

**} else {**

**return;**

**}**

**} catch (Exception e) {**

**e.printStackTrace();**

**} finally {**

**if (canvas != null) {**

**try {**

**surfaceHolder.unlockCanvasAndPost(canvas);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**long timeMillis = (System.nanoTime() - startTime) / 1000000;**

**long waitTime = targetTime - timeMillis;**

**try {**

**if(waitTime > 0) {**

**sleep(waitTime);**

**}**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**}**

**Файл AboutActivity.java**

**package com.example.lab10;**

**import android.os.Bundle;**

**import android.view.MenuItem;**

**import android.view.animation.AnimationUtils;**

**import android.widget.Button;**

**import androidx.annotation.NonNull;**

**import androidx.appcompat.app.ActionBar;**

**import androidx.appcompat.app.AppCompatActivity;**

**public class AboutActivity extends AppCompatActivity {**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_about);**

**Button star = findViewById(R.id.button\_star);**

**star.setOnClickListener(v -> v.startAnimation(AnimationUtils.loadAnimation(getApplicationContext(), R.anim.spin)));**

**ActionBar actionBar = getSupportActionBar();**

**if (actionBar != null) {**

**actionBar.setDisplayHomeAsUpEnabled(true);**

**}**

**}**

**@Override**

**public boolean onOptionsItemSelected(@NonNull MenuItem item) {**

**if (item.getItemId() == android.R.id.home) {**

**onBackPressed();**

**return true;**

**}**

**return super.onOptionsItemSelected(item);**

**}**

**}**

**Файл GameObject.java**

**package com.example.lab10;**

**import android.graphics.Canvas;**

**import android.graphics.Paint;**

**abstract class GameObject {**

**abstract void draw(Canvas canvas, Paint paint);**

**}**

**Файл GamePanel.java**

**package com.example.lab10;**

**import android.annotation.SuppressLint;**

**import android.content.Context;**

**import android.content.Intent;**

**import android.graphics.Canvas;**

**import android.graphics.Color;**

**import android.graphics.Paint;**

**import android.graphics.Point;**

**import android.graphics.Rect;**

**import android.view.MotionEvent;**

**import android.view.SurfaceHolder;**

**import android.view.SurfaceView;**

**public class GamePanel extends SurfaceView implements SurfaceHolder.Callback{**

**private MainThread thread;**

**private Player player;**

**private Point playerPoint;**

**private PipeManager pipeManager;**

**private Rect rectangle = new Rect();**

**private boolean movingPlayer = false;**

**private boolean playerStart = false;**

**private boolean gameOver = false;**

**private long gameOverTime;**

**Paint paint = new Paint();**

**public GamePanel(Context context) {**

**super(context);**

**getHolder().addCallback(this);**

**thread = new MainThread(getHolder(), this);**

**player = new Player(new Rect(100,100,200,200), Color.rgb(0, 210, 0));**

**playerPoint = new Point(Globals.SCREEN\_X /2 , 3\* Globals.SCREEN\_Y /4);**

**player.update(playerPoint);**

**pipeManager = new PipeManager(400, 500);**

**setFocusable(true);**

**}**

**public void reset() {**

**playerPoint = new Point(Globals.SCREEN\_X /2 , 3\* Globals.SCREEN\_Y /4);**

**player.update(playerPoint);**

**pipeManager = new PipeManager(400, 500);**

**movingPlayer = false;**

**playerStart = false;**

**}**

**@Override**

**public void surfaceCreated(SurfaceHolder surfaceHolder) {**

**thread = new MainThread(getHolder(), this);**

**thread.setRunning(true);**

**thread.start();**

**}**

**@Override**

**public void surfaceChanged(SurfaceHolder surfaceHolder, int format, int width, int height) {**

**}**

**@Override**

**public void surfaceDestroyed(SurfaceHolder surfaceHolder) {**

**boolean retry = true;**

**while (retry) {**

**try {**

**thread.setRunning(false);**

**thread.join();**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**retry = false;**

**}**

**}**

**@SuppressLint("ClickableViewAccessibility")**

**@Override**

**public boolean onTouchEvent(MotionEvent event) {**

**switch (event.getAction()) {**

**case MotionEvent.ACTION\_DOWN:**

**if(!gameOver) {**

**playerStart = true;**

**movingPlayer = true;**

**}**

**if(gameOver && System.currentTimeMillis() - gameOverTime > 200) {**

**showScoreboard();**

**reset();**

**gameOver = false;**

**}**

**break;**

**case MotionEvent.ACTION\_MOVE:**

**if (!gameOver && movingPlayer) {**

**playerPoint.set((int)event.getX(), (int)event.getY());**

**}**

**break;**

**case MotionEvent.ACTION\_UP:**

**movingPlayer = false;**

**break;**

**}**

**return true;**

**}**

**public void update() {**

**if(!gameOver && playerStart) {**

**player.update(playerPoint);**

**pipeManager.update();**

**if(pipeManager.playerTouchedPipe(player)) {**

**playerStart = false;**

**gameOver = true;**

**gameOverTime = System.currentTimeMillis();**

**}**

**}**

**}**

**@Override**

**public void draw(Canvas canvas) {**

**super.draw(canvas);**

**pipeManager.draw(canvas, paint);**

**player.draw(canvas, paint);**

**if(!playerStart && !gameOver) {**

**paint.setTextSize(100);**

**paint.setFakeBoldText(true);**

**paint.setColor(Color.GREEN);**

**drawCenterText(canvas, paint, "Move the square");**

**}**

**if(gameOver) {**

**paint.setTextSize(100);**

**paint.setFakeBoldText(true);**

**paint.setColor(Color.RED);**

**drawCenterText(canvas, paint, "Game Over");**

**}**

**}**

**private void drawCenterText(Canvas canvas, Paint paint, String text) {**

**paint.setTextAlign(Paint.Align.LEFT);**

**canvas.getClipBounds(rectangle);**

**int cHeight = rectangle.height();**

**int cWidth = rectangle.width();**

**paint.getTextBounds(text, 0, text.length(), rectangle);**

**float x = cWidth / 2f - rectangle.width() / 2f - rectangle.left;**

**float y = cHeight / 2f + rectangle.height() / 2f - rectangle.bottom;**

**canvas.drawText(text, x, y, paint);**

**}**

**private void showScoreboard() {**

**Intent intent = new Intent(this.getContext(), PlayerNameActivity.class);**

**intent.putExtra(ScoreboardActivity.EXTRA\_SCORE, pipeManager.getScore());**

**this.getContext().startActivity(intent);**

**}**

**}**

**Файл Globals.java**

**package com.example.lab10;**

**public class Globals {**

**public static int SCREEN\_X;**

**public static int SCREEN\_Y;**

**public static int PIPE\_X = 120;**

**}**

**Файл Pipe.java**

**package com.example.lab10;**

**import android.graphics.Canvas;**

**import android.graphics.Paint;**

**import android.graphics.Rect;**

**public class Pipe extends GameObject {**

**private Rect pipeRect;**

**private Rect pipeRect2;**

**private int color;**

**public Pipe(int startX, int startY, int playerGap, int color) {**

**this.color = color;**

**this.pipeRect = new Rect(startX, startY + playerGap, Globals.PIPE\_X + startX, Globals.SCREEN\_Y);**

**this.pipeRect2 = new Rect(startX, 0, Globals.PIPE\_X + startX, startY);**

**}**

**@Override**

**public void draw(Canvas canvas, Paint paint) {**

**paint.setColor(color);**

**canvas.drawRect(pipeRect, paint);**

**canvas.drawRect(pipeRect2, paint);**

**}**

**public void decrementX(Float x) {**

**pipeRect.left -= x;**

**pipeRect.right -= x;**

**pipeRect2.left -= x;**

**pipeRect2.right -= x;**

**}**

**public boolean playerCollide(Player player) {**

**return Rect.intersects(pipeRect, player.getRectangle()) || Rect.intersects(pipeRect2, player.getRectangle());**

**}**

**public Rect getRectangle() {**

**return pipeRect;**

**}**

**public void setColor(int color) {**

**this.color = color;**

**}**

**}**

**Файл PipeManager.java**

**package com.example.lab10;**

**import android.graphics.Canvas;**

**import android.graphics.Color;**

**import android.graphics.Paint;**

**import java.util.ArrayList;**

**public class PipeManager {**

**private ArrayList<Pipe> pipes;**

**private int playerGap;**

**private int pipeGap;**

**private long startTime;**

**private long initTime;**

**private int score=0;**

**float speedMultiplier;**

**private int pipeColor = Color.BLACK;**

**private int backGroundColor = Color.WHITE;**

**public PipeManager(int playerGap, int pipeGap) {**

**this.playerGap = playerGap;**

**this.pipeGap = pipeGap;**

**startTime = System.currentTimeMillis();**

**initTime = System.currentTimeMillis();**

**pipes = new ArrayList<>();**

**populatePipes();**

**}**

**public void draw(Canvas canvas, Paint paint) {**

**canvas.drawColor(backGroundColor);**

**for (Pipe pipe: pipes) {**

**pipe.draw(canvas, paint);**

**}**

**paint.setColor(Color.BLUE);**

**paint.setTextSize(50);**

**canvas.drawText("Score: " + score, 60, 60 + paint.descent() - paint.ascent() , paint);**

**}**

**public void populatePipes() {**

**int pipeCursorX = 5 \* Globals.SCREEN\_X / 4;**

**for (int i = 0; i < 5; i++){**

**int yStart = (int) (Math.random() \* (Globals.SCREEN\_Y - playerGap));**

**pipes.add(new Pipe(pipeCursorX, yStart, playerGap, pipeColor));**

**pipeCursorX += pipeGap + Globals.PIPE\_X;**

**i++;**

**}**

**}**

**public void update() {**

**boolean newPipeWasIntroduced = false;**

**startTime = System.currentTimeMillis();**

**float speed = Globals.SCREEN\_X / 260.f;**

**speedMultiplier = (float) (Math.sqrt(1 + (startTime - initTime) / (10000.f)));**

**speed \*= speedMultiplier;**

**for (Pipe pipe : pipes) {**

**pipe.decrementX(speed);**

**}**

**if (pipes.get(0).getRectangle().right <= 0) {**

**int yStart = (int) (Math.random() \* (Globals.SCREEN\_Y - playerGap));**

**pipes.add(new Pipe(pipes.get(pipes.size() - 1).getRectangle().right + pipeGap, yStart, playerGap, pipeColor));**

**pipes.remove(0);**

**score++;**

**newPipeWasIntroduced = true;**

**}**

**if(score != 0 && score %5 == 0 && newPipeWasIntroduced) {**

**int red= (int) (Math.random() \* 255);**

**int green = (int) (Math.random() \* 255);**

**int blue = (int) (Math.random() \* 255);**

**pipeColor = Color.rgb(red, green, blue);**

**for (Pipe pipe: pipes ) {**

**pipe.setColor(pipeColor);**

**}**

**backGroundColor = Color.rgb(255 - red, 255 - green, 255 - blue);**

**}**

**}**

**public boolean playerTouchedPipe(Player player) {**

**for(Pipe pipe : pipes) {**

**if(pipe.playerCollide(player)) {**

**return true;**

**}**

**}**

**return false;**

**}**

**public int getScore() {**

**return score;**

**}**

**}**

**Файл Player.java**

**package com.example.lab10;**

**import android.graphics.Canvas;**

**import android.graphics.Paint;**

**import android.graphics.Point;**

**import android.graphics.Rect;**

**public class Player extends GameObject {**

**private Rect rectangle;**

**private int color;**

**public Player(Rect rectangle, int color) {**

**this.rectangle = rectangle;**

**this.color = color;**

**}**

**public void update(Point point) {**

**rectangle.set(point.x - rectangle.width()/2, point.y - rectangle.height() / 2, point.x + rectangle.width()/2, point.y + rectangle.height() / 2);**

**}**

**@Override**

**public void draw(Canvas canvas, Paint paint) {**

**paint.setColor(color);**

**canvas.drawRect(rectangle, paint);**

**}**

**public Rect getRectangle() {**

**return rectangle;**

**}**

**}**

**Файл PlayerNameActivity.java**

**package com.example.lab10;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import androidx.appcompat.app.AppCompatActivity;**

**public class PlayerNameActivity extends AppCompatActivity {**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_player\_name);**

**Intent intent = getIntent();**

**Button button = findViewById(R.id.okButton);**

**EditText nameField = findViewById(R.id.playerNameField);**

**button.setOnClickListener(v -> {**

**Intent nextIntent = new Intent(getBaseContext(), ScoreboardActivity.class);**

**nextIntent.putExtra(ScoreboardActivity.EXTRA\_NAME, nameField.getText().toString());**

**nextIntent.putExtra(ScoreboardActivity.EXTRA\_SCORE, intent.getIntExtra(ScoreboardActivity.EXTRA\_SCORE, -1));**

**startActivity(nextIntent);**

**});**

**}**

**}**

**Файл Score.java**

**package com.example.lab10;**

**import android.annotation.SuppressLint;**

**import android.content.ContentValues;**

**import android.database.Cursor;**

**import android.database.sqlite.SQLiteDatabase;**

**public class Score {**

**public static final String TABLE\_NAME = "score";**

**public static final String COLUMN\_ID = "\_id";**

**public static final String COLUMN\_NAME = "name";**

**public static final String COLUMN\_SCORE = "score";**

**public static final String[] COLUMNS = {COLUMN\_ID, COLUMN\_NAME, COLUMN\_SCORE};**

**public int id;**

**public String name;**

**public int score;**

**public Score() {**

**this.id = 0;**

**this.name = null;**

**this.score = 0;**

**}**

**@SuppressLint("Range")**

**public Score(Cursor c) {**

**this.id = c.getInt(c.getColumnIndex(COLUMN\_ID));**

**this.name = c.getString(c.getColumnIndex(COLUMN\_NAME));**

**this.score = c.getInt(c.getColumnIndex(COLUMN\_SCORE));**

**}**

**public void saveToDb(SQLiteDatabase db) throws Exception {**

**ContentValues values = new ContentValues();**

**values.put(COLUMN\_NAME, this.name);**

**values.put(COLUMN\_SCORE, this.score);**

**int id = (int)db.insert(TABLE\_NAME, null, values);**

**if (id == -1) {**

**throw new Exception("Exception while inserting score");**

**}**

**this.id = id;**

**}**

**}**

**Файл ScoreAdapter.java**

**package com.example.lab10;**

**import android.content.Context;**

**import android.database.Cursor;**

**import android.view.LayoutInflater;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.CursorAdapter;**

**import android.widget.TextView;**

**public class ScoreAdapter extends CursorAdapter {**

**public ScoreAdapter(Context context, Cursor c, int flags) {**

**super(context, c, flags);**

**}**

**@Override**

**public View newView(Context context, Cursor cursor, ViewGroup viewGroup) {**

**return LayoutInflater.from(context).inflate(android.R.layout.simple\_list\_item\_2, viewGroup, false);**

**}**

**@Override**

**public void bindView(View view, Context context, Cursor cursor) {**

**TextView playerName = view.findViewById(android.R.id.text1);**

**TextView scoreField = view.findViewById(android.R.id.text2);**

**Score score = new Score(cursor);**

**playerName.setText(score.name);**

**scoreField.setText("Score: " + score.score);**

**}**

**}**

**Файл ScoreboardActivity.java**

**package com.example.lab10;**

**import android.content.Intent;**

**import android.database.Cursor;**

**import android.database.sqlite.SQLiteDatabase;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.Menu;**

**import android.view.MenuInflater;**

**import android.view.MenuItem;**

**import android.widget.ListView;**

**import androidx.appcompat.app.AppCompatActivity;**

**import androidx.appcompat.widget.Toolbar;**

**import com.google.android.material.floatingactionbutton.FloatingActionButton;**

**import java.util.Objects;**

**public class ScoreboardActivity extends AppCompatActivity {**

**public static final String TAG = "ScoreboardActivity";**

**public static final String EXTRA\_SCORE = "extra\_score";**

**public static final String EXTRA\_NAME = "extra\_name";**

**public static final String EXTRA\_ADDED = "extra\_added";**

**public ScoreDb dbHelper;**

**private ListView scoreListView;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_scoreboard);**

**Toolbar toolbar = findViewById(R.id.toolbar);**

**setSupportActionBar(toolbar);**

**dbHelper = new ScoreDb(this);**

**scoreListView = findViewById(R.id.scoreListView);**

**FloatingActionButton fab = findViewById(R.id.fab);**

**fab.setOnClickListener(view -> {**

**dbHelper.deleteEntries();**

**inflateList();**

**});**

**Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);**

**}**

**@Override**

**protected void onResume() {**

**addPlayerScore();**

**inflateList();**

**super.onResume();**

**}**

**private void addPlayerScore() {**

**Intent intent = getIntent();**

**if (intent.hasExtra(EXTRA\_SCORE) && !intent.hasExtra(EXTRA\_ADDED)) {**

**Score score = new Score();**

**score.name = intent.getStringExtra(EXTRA\_NAME);**

**score.score = intent.getIntExtra(EXTRA\_SCORE, 0);**

**try (SQLiteDatabase db = dbHelper.getWritableDatabase()) {**

**score.saveToDb(db);**

**} catch (Exception ex) {**

**Log.e(TAG, "Exception while saving score: " + ex.getMessage());**

**}**

**intent.putExtra(EXTRA\_ADDED, true);**

**}**

**}**

**private void inflateList() {**

**SQLiteDatabase db = dbHelper.getReadableDatabase();**

**Cursor c = ScoreDb.selectScores(db);**

**ScoreAdapter adapter = new ScoreAdapter(this, c, 0);**

**scoreListView.setAdapter(adapter);**

**}**

**@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

**MenuInflater menuInflater = getMenuInflater();**

**menuInflater.inflate(R.menu.menu\_scoreboard, menu);**

**return true;**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**if (item.getItemId() == R.id.aboutMenuItem) {**

**Intent intent = new Intent(this, AboutActivity.class);**

**startActivity(intent);**

**return true;**

**}**

**return false;**

**}**

**}**

**Файл ScoreDb.java**

**package com.example.lab10;**

**import android.content.Context;**

**import android.database.Cursor;**

**import android.database.sqlite.SQLiteDatabase;**

**import android.database.sqlite.SQLiteOpenHelper;**

**public class ScoreDb extends SQLiteOpenHelper {**

**public static final String DATABASE\_NAME = "score.db";**

**public static final int DATABASE\_VERSION = 1;**

**private static final String SQL\_SETUP =**

**"CREATE TABLE `score` (\n" +**

**"\t`\_id`\tINTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,\n" +**

**"\t`name`\tTEXT NOT NULL,\n" +**

**"\t`score`\tINTEGER NOT NULL\n" +**

**");\n" +**

**"CREATE INDEX `index\_score\_name` ON `score` (`name` ASC);\n" +**

**"CREATE INDEX `index\_score\_score` ON `score` (`score` DESC);";**

**private static final String SQL\_DELETE\_ENTRIES = "DELETE FROM `score`;";**

**public ScoreDb(Context context) {**

**super(context, DATABASE\_NAME, null, DATABASE\_VERSION);**

**}**

**@Override**

**public void onCreate(SQLiteDatabase db) {**

**db.execSQL(SQL\_SETUP);**

**}**

**@Override**

**public void onUpgrade(SQLiteDatabase db, int i, int i1) {**

**deleteEntries();**

**onCreate(db);**

**}**

**public static Cursor selectScores(SQLiteDatabase db) {**

**return db.query(Score.TABLE\_NAME, Score.COLUMNS, null, null, null, null, Score.COLUMN\_SCORE+" DESC", null);**

**}**

**public void deleteEntries() {**

**SQLiteDatabase db = this.getWritableDatabase();**

**db.execSQL(SQL\_DELETE\_ENTRIES);**

**db.close();**

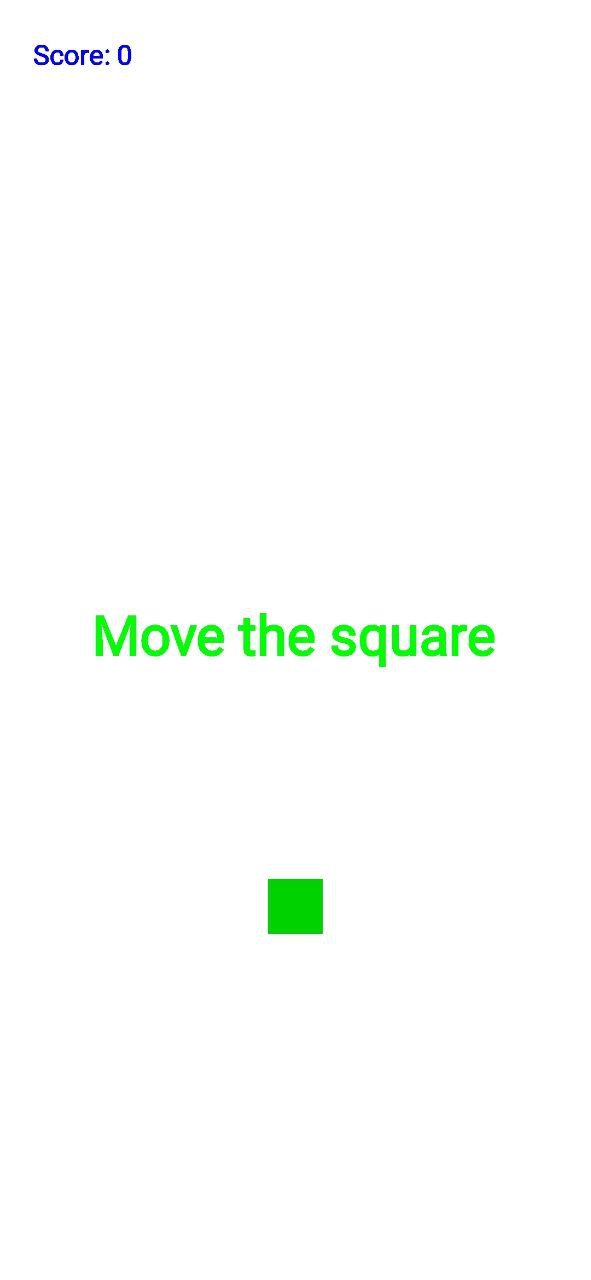
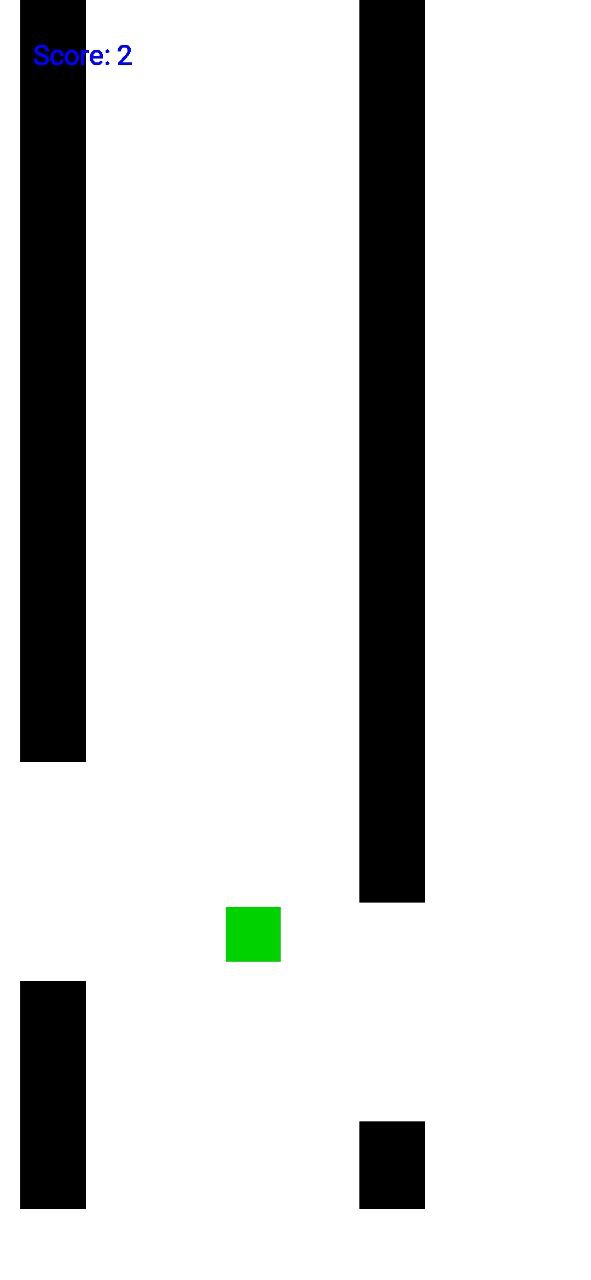
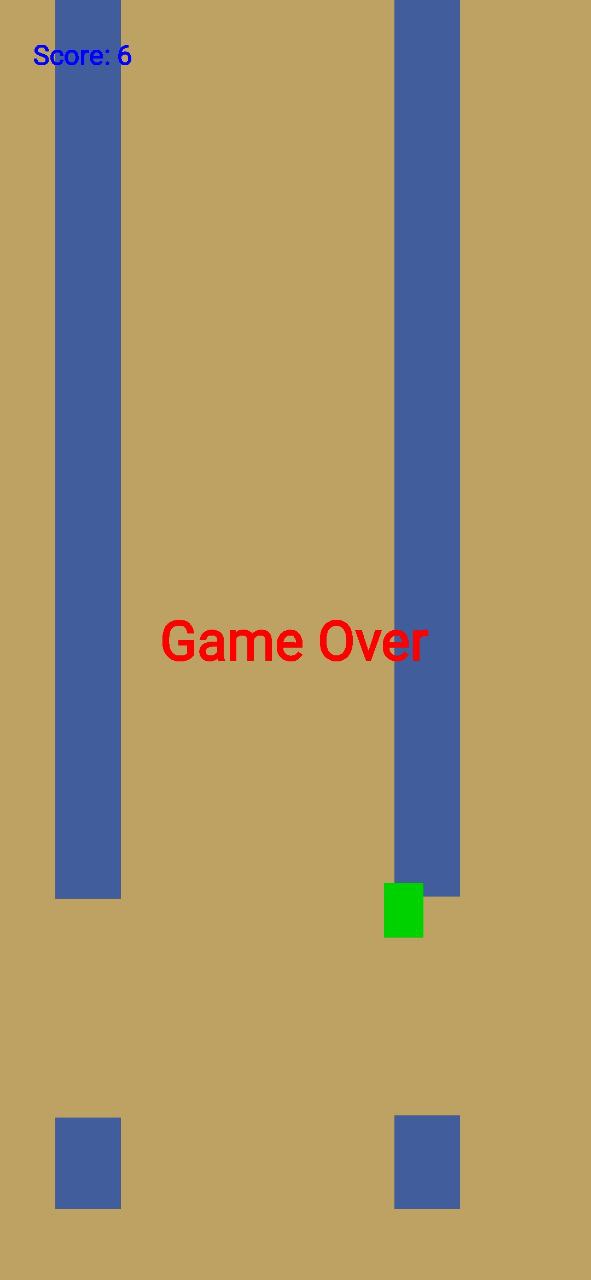
**}**

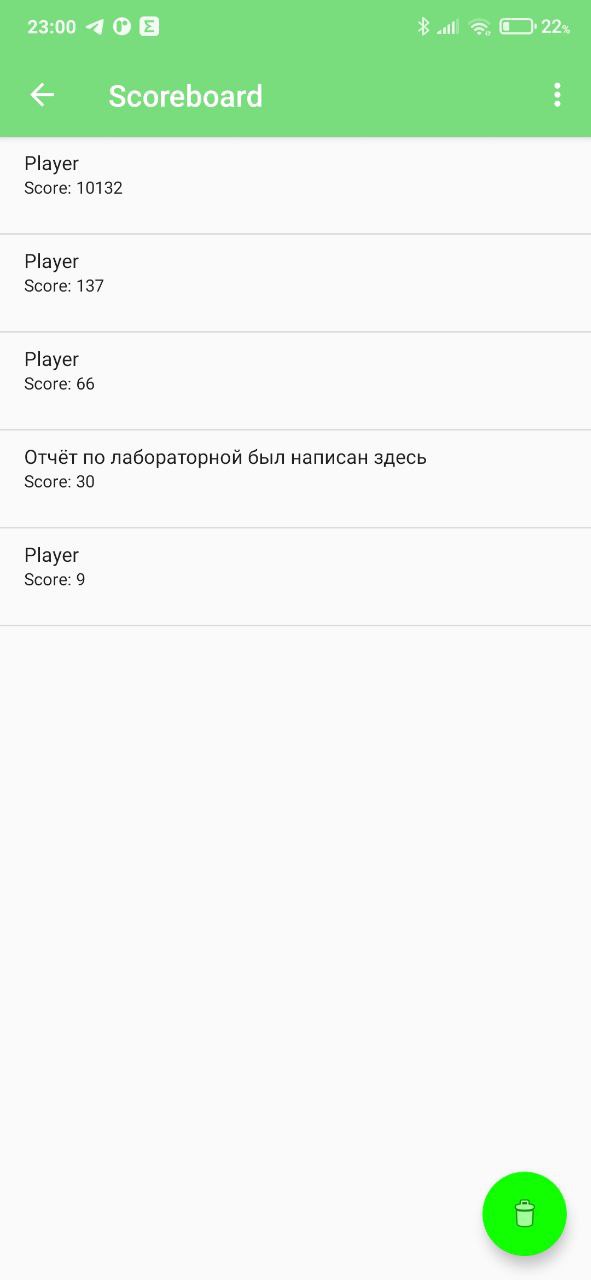
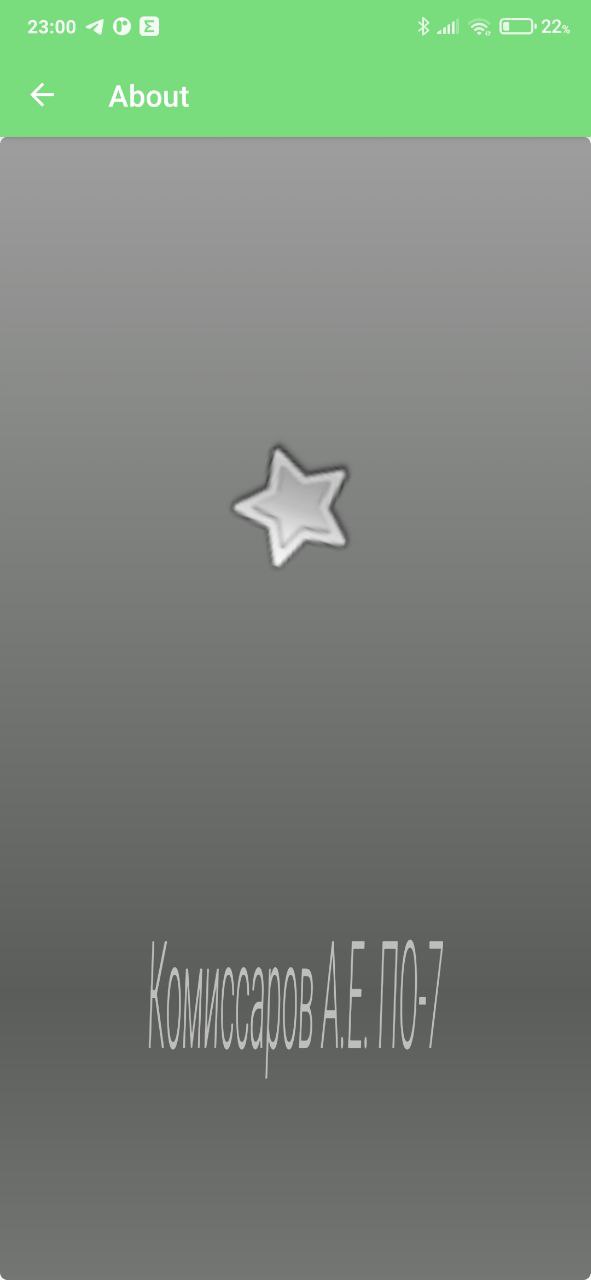
**}**

Сценарий:

* При запуске приложения появляется подсказка подвигать квадрат.
* При движении квадрата запускается игра, игра оканчивается как только игрок наводит квадрат на “трубу”. Игроку необходимо проходить между трубами, набирая тем самым очки.
* После окончания игры, игроку предлагается ввести своё имя для записи результата в таблицу лидеров. Таблицу можно очистить нажатием кнопки.
* Из таблицы лидеров можно вернуться обратно в игру, тогда она запускается заново. Также можно открыть окно с информацией о приложении.

**Результат работы программы:**

****

****

**Вывод:** я получил знания о создании игр на мобильных платформах на примере ОС Android.