

Федеральное агентство связи  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет  
телекоммуникаций и информатики»

**Лабораторная работа по теме:**  
**«Жуки»**

Выполнили:  
студентки 3 курса  
ИВТ, гр. ИП-712  
Гервас А.В.  
Онищенко А.В.

Новосибирск 2020

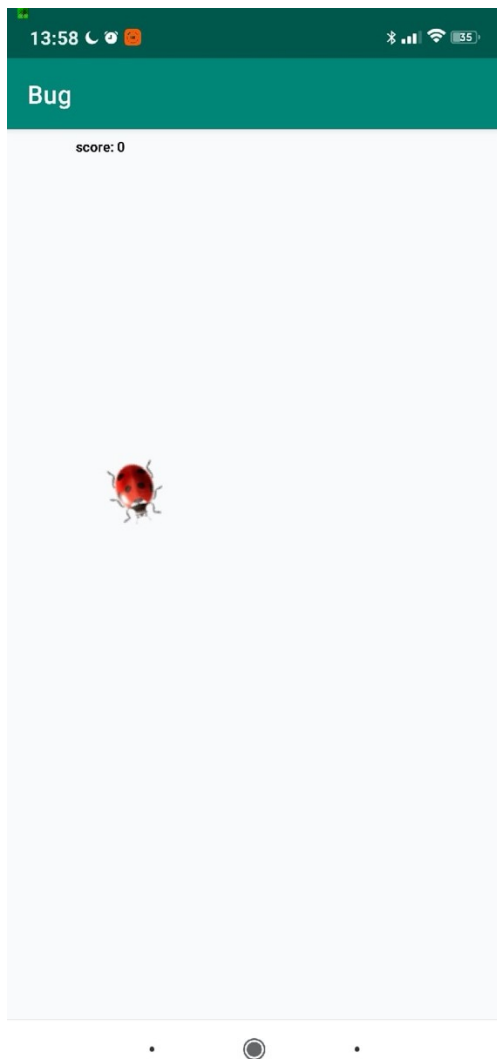
## Оглавление

<i>Задание.....</i>	<b>3</b>
<i>Скриншоты.....</i>	<b>3</b>
<i>Листинг кода.....</i>	<b>4</b>

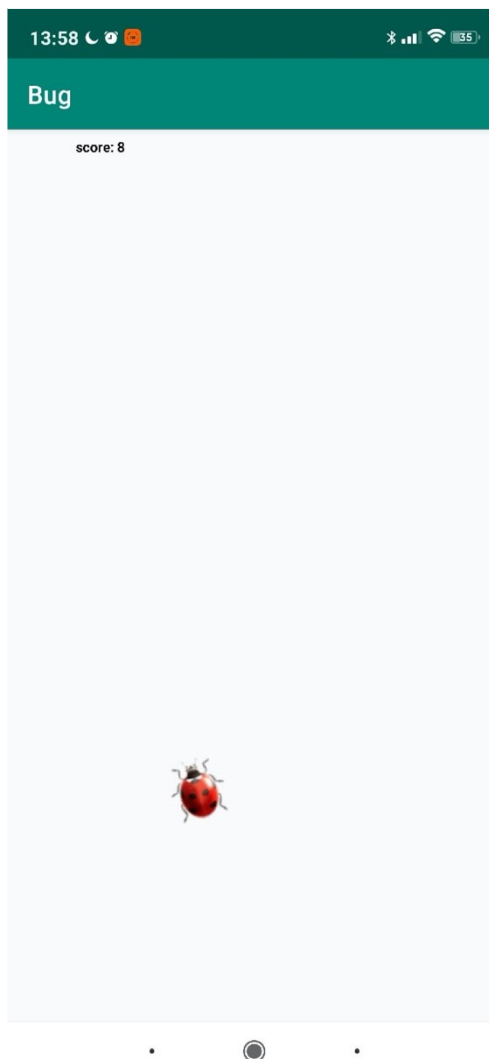
## Задание

Создайте игру "ЖУК". Жуки бегают по экрану. Игроку предлагается при помощи touchScreen-а уничтожить как можно большее число жуков. Обработка отдельного жука должна производиться в отдельном потоке. За каждый промах игроку начисляется штраф. Предусмотреть несколько видов насекомых. Попадание и промах должны иметь звуковое сопровождение. По окончании игры выводятся результаты.

## Скриншоты



Стартовая активность



Сама игра, когда уже убил несколько жуков

## Листинг кода

### MainActivity.java

```
package ru.lab3.bug;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;

import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    private BugView view;
    private Handler handler;
    private final static int interval = 50;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // setContentView(R.layout.activity_main);

        view = new BugView(this);
        setContentView(view);

        handler = new Handler();
        Timer timer = new Timer();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        view.invalidate();
                    }
                });
            }
        }, 0, interval);
    }

    /*@Override
    public void onDraw(Canvas c) {
        Paint p1 = new Paint();// Создаем кисть для рисования
        p1.setColor(Color.rgb(100, 0, 255)); // Меняем цвет
        p1.setTextSize(50); // Задаем размер текста
        c.drawText("Жмяк:", 100, 100, p1); // Выводим текст
    }*/
}
```

### Bug.java

```

package ru.lab3.bug;

import android.graphics.Bitmap;
import android.graphics.Point;

class Bug {
    public Bitmap texture;
    public Float x, y, stepX, stepY, destX, destY;
    public Integer p;

    private Point point;

    public Bug() {
        x = 0f;
        y = 0f;
        p = 0;
        destX = 0f;
        destY = 0f;
        p = 0;
    }

    public boolean checkCol(float Sx, float Sy) {
        return true;
    }
}

```

## BugView.java

```

package ru.lab3.bug;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Matrix;
import android.graphics.Paint;
import android.graphics.Typeface;
import android.graphics.drawable.BitmapDrawable;
import android.util.DisplayMetrics;
import android.view.MotionEvent;
import android.view.View;

public class BugView extends View {

    private Bitmap background;
    private Bug bug = new Bug();
    private Paint score;
    private Matrix matrix;
    private boolean reached;
    private Integer sc = 0;

    public BugView(Context context) {
        super(context);
        bug.texture = BitmapFactory.decodeResource(context.getResources(), R.drawable.images);

        /*background = BitmapFactory.decodeResource(getResources(), R.drawable.grass);
        BitmapDrawable bitmapDrawable = new BitmapDrawable(background);
        bitmapDrawable.setTileModeXY(android.graphics.Shader.TileMode.REPEAT,
        android.graphics.Shader.TileMode.REPEAT); // говорим объекту как рисовать (у меня это повторяющийся фон)
        layout.setBackgroundDrawable(bitmapDrawable); */

        score = new Paint();
        score.setColor(Color.BLACK);
        score.setTextSize(30);
        score.setTypeface(Typeface.DEFAULT_BOLD);
        score.setAntiAlias(true);
    }
}

```

```

matrix = new Matrix();
reached = true;
DisplayMetrics metrics = getResources().getDisplayMetrics();
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    //canvas.drawBitmap(background, 0, 0, null);
    canvas.drawText("score: " + sc, 150, 50, score);
    //matrix.postScale(3.0f, 3.0f); // Масштабируем
    if (reached) {
        bug.destX = (float) Math.random() * getWidth();
        bug.destY = (float) Math.random() * getHeight();
        bug.stepX = (bug.destX - bug.x) / 50;
        bug.stepY = (bug.destY - bug.y) / 50;
        Integer tp;
        if (bug.x <= bug.destX && bug.y >= bug.destY)
            tp = (int) Math.floor(Math.toDegrees(Math.atan(Math.abs(bug.x - bug.destX) / Math.abs(bug.y - bug.destY))));
        else if (bug.x <= bug.destX && bug.y <= bug.destY)
            tp = 90 + (int) Math.floor(Math.toDegrees(Math.atan(Math.abs(bug.y - bug.destY) / Math.abs(bug.x - bug.destX))));
        else if (bug.x >= bug.destX && bug.y <= bug.destY)
            tp = 180 + (int) Math.floor(Math.toDegrees(Math.atan(Math.abs(bug.x - bug.destX) / Math.abs(bug.y - bug.destY))));
        else
            tp = 270 + (int) Math.floor(Math.toDegrees(Math.atan(Math.abs(bug.y - bug.destY) / Math.abs(bug.x - bug.destX))));
        matrix.preRotate(tp - bug.p, bug.texture.getWidth() / 2, bug.texture.getHeight() / 2);
        bug.p = tp;
        reached = false;
    } else {
        if (Math.abs(bug.x - bug.destX) < 0.1 &&
            Math.abs(bug.y - bug.destY) < 0.1)
            reached = true;

        matrix.postTranslate(bug.stepX, bug.stepY);
        bug.x += bug.stepX;
        bug.y += bug.stepY;
    }
    canvas.drawBitmap(bug.texture, matrix, null);
}

@Override
public boolean onTouchEvent(MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        if (Math.abs(bug.x - event.getX() + 30) < 100 &&
            Math.abs(bug.y - event.getY() + 30) < 120) {
            matrix.setRotate(0, bug.texture.getWidth() / 2, bug.texture.getHeight() / 2);
            matrix.reset();
            bug.p = 0;
            ++sc;
            reached = true;
            float ty, tx;
            int temp = (int) Math.floor(Math.random() * 4);
            switch (temp) {
                case 0:
                    ty = (float) Math.random() * getHeight();
                    bug.x = 0f;
                    bug.y = ty;
                    break;
                case 1:
                    ty = (float) Math.random() * getHeight();
                    bug.x = (float) getWidth();
                    bug.y = ty;
                    break;
                case 2:
                    tx = (float) Math.random() * getWidth();

```

```
        bug.x = tx;
        bug.y = 0f;
        break;
    case 3:
        tx = (float) Math.random() * getWidth();
        bug.x = tx;
        bug.y = (float) getHeight();
        break;
    }
    matrix.postTranslate(bug.x, bug.y);
}
}
return true;
}
```