

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа № 3
по дисциплине «Программирование для мобильных устройств»

Выполнил:

студент группы ИП-811

Разумов Д.Б.

Работу проверил:

Павлова У.В.

Новосибирск 2021

Содержание

Содержание.....	2
Задание.....	3
Листинг программы.....	4
Скриншоты работы программы.....	9

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.

Листинг программы

Файл MainActivity.java

```
package com.example.bu;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.DialogInterface;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Matrix;
import android.media.AudioManager;
import android.media.SoundPool;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewDebug;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

import java.util.Random;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

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

        this.setVolumeControlStream(AudioManager.STREAM_MUSIC);
    }

    Panel panel;
    int n = 20;

    class Panel extends View {

        Panel.Bugs[] bugs;
        Thread[] thread;

        SoundPool soundPool;
        int hit, miss;
        int points;

        @SuppressLint("ClickableViewAccessibility")
```

```

public Panel(Context context) {
    super(context);

    points = 0;
    time_game();
    soundPool = new SoundPool(10, AudioManager.STREAM_MUSIC,0);
    hit = soundPool.load(context, R.raw.hit, 0);
    miss = soundPool.load(context, R.raw.miss, 1);
    this.setOnTouchListener(new OnTouchListener() {

        @Override
        public boolean onTouch(View v, MotionEvent event) {
            boolean touch_flag = false;
            float touchx = -1, touchy = -1;
            touchx = event.getX();
            touchy = event.getY();
            for(int i = 0; i < n; i++){
                if(bugs[i].kill(touchx,touchy)){
                    soundPool.play(hit, 1.0f, 1.0f, 0, 0, 1.5f);
                    touch_flag = true;
                    points++;
                    break;
                }
            }
            if(!touch_flag){
                soundPool.play(miss, 1.0f, 1.0f, 0, 0, 1.5f);
                points--;
            }
            return false;
        }
    });
}

public void onDraw(Canvas canvas){
    for(int i = 0; i < n; ++i)
    {
        bugs[i].draw(canvas);
    }
    invalidate();
}

public void time_game(){
    new CountDownTimer(30000,1000){
        @Override
        public void onTick(long millisUntilFinished) {

        }

        @Override
        public void onFinish() {
            AlertDialog.Builder builder = new
AlertDialog.Builder(getContext());
            builder.setTitle("Время игры вышло!")
                .setMessage("Очки: " + points)
                .setCancelable(false)
                .setNegativeButton("OK",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog,
int id) {

                            finish();
                            System.exit(0);
                            dialog.cancel();
                        }
                    }
                );
        }
    }.start();
}

```

```

        });
        AlertDialog alert = builder.create();
        alert.show();
    }
    }.start();
}

public void start() {

    bugs = new Panel.Bugs[n];
    for(int i = 0 ; i < n; ++i)
    {
        bugs[i] = new Panel.Bugs();
    }
    thread = new Thread[n];
    for(int i = 0; i < n; ++i)
    {
        final int index = i;
        thread[i] = new Thread(new Runnable() {
            @Override
            public void run() {
                while (true)
                {
                    if(bugs[index].moveflag == 0) {
                        bugs[index].move();
                        try {
                            Thread.sleep(350);
                        } catch (InterruptedException e) {
                            e.printStackTrace();
                        }
                    }
                }
            }
        });
        thread[i].start();
    }

}

class Bugs {

    Bitmap b;
    Matrix matrix;
    float x, y;
    int moveflag;
    public Bugs() {
        moveflag = 0;
        final Random random = new Random();
        int r = random.nextInt(3)+1;
        switch (r){

            case 1:
                b = BitmapFactory.decodeResource(getResources(),
R.drawable.b1);
                break;
            case 2:
                b = BitmapFactory.decodeResource(getResources(),
R.drawable.b2);
                break;
            case 3:
                b = BitmapFactory.decodeResource(getResources(),

```

```

R.drawable.b3);
                break;
            }

            matrix = new Matrix();
            spawn();
            matrix.postTranslate(x, y);
        }

        public void draw(Canvas c) {
            matrix.setTranslate(x,y);
            c.drawBitmap(b, matrix, null);
        }

        public void spawn() {
            final Random random = new Random();

            x = random.nextInt(1000)+1;
            y = random.nextInt(2000)+1;
        }

        public boolean kill(float touchx, float touchy)
        {
            if(touchx < x + 70 && touchx > x - 20 && touchy < y + 100 &&
touchy > y - 20) {
                b = BitmapFactory.decodeResource(getResources(),
R.drawable.b0);

                moveflag = 1;
                //soundPool.play(hit, 1.0f, 1.0f, 0, 0, 1.5f);
                return true;
            }
            else return false;
        }

        public void move() {
            if(x > 1062 || y > 2062 || x < 0 || y < 0)
            {
                spawn();
            }
            else
            {
                final Random random = new Random();
                int ra = random.nextInt(2)+1;
                int nd = random.nextInt(2)+1;
                if(ra == 1)
                {
                    x+=50;
                }
                else x-=50;
                if(nd == 1)
                {
                    y+=50;
                }
                else y-=50;
                matrix.setTranslate(x,y);
            }
        }
    }
}

```

```
public void startGame(View view)
{
    Button button = (Button) findViewById(R.id.startbutton);
    ViewGroup layout = (ViewGroup) button.getParent();
    if(null!=layout){
        layout.removeAllViews();
    }
    panel = new Panel(this);
    setContentView(panel);
    panel.start();
}
}
```


Скриншоты работы программы



Рисунок 1 – Запуск программы

Bu



Рисунок 2 – передвижение жуков

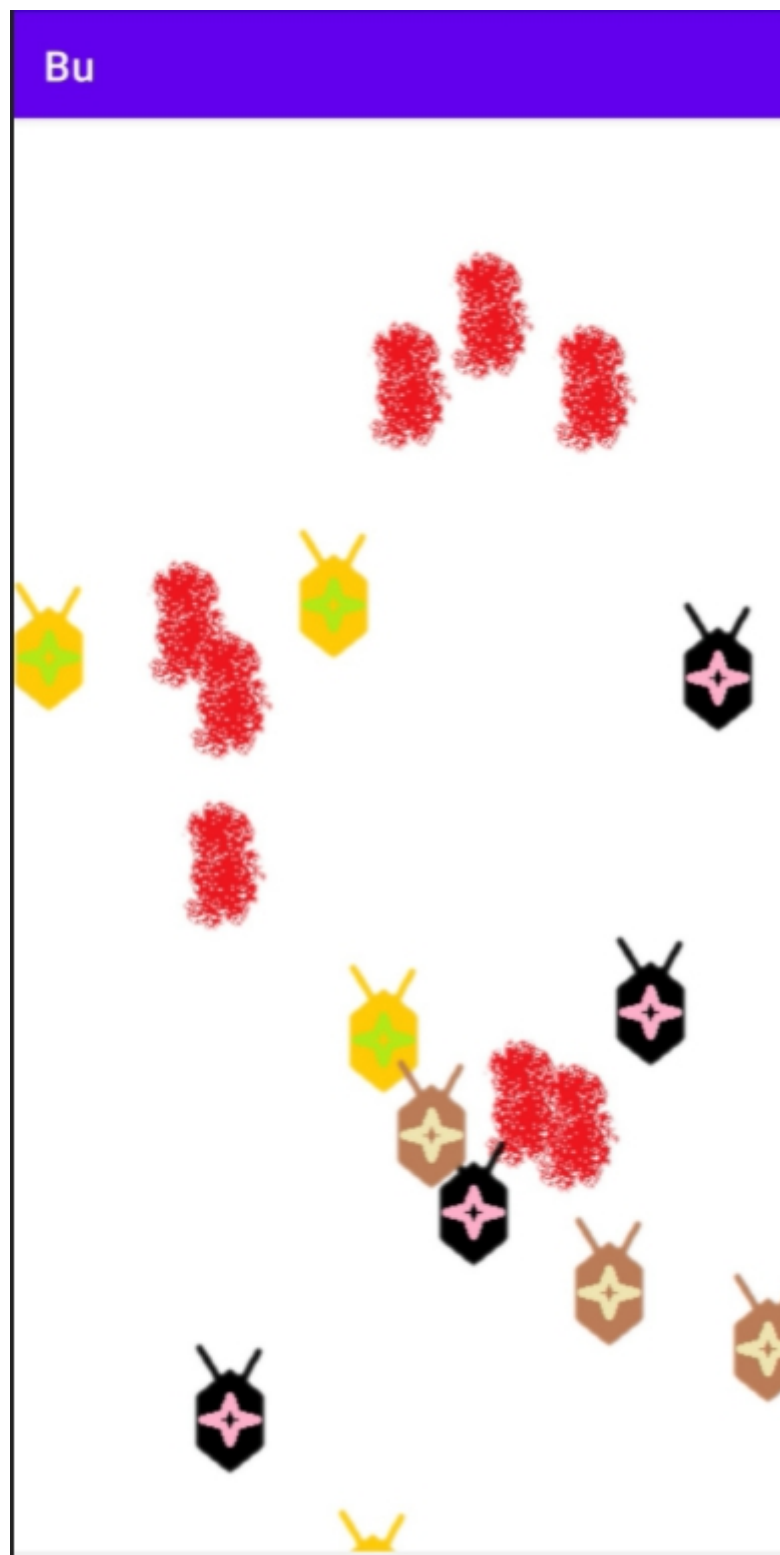


Рисунок 3 – уничтожение жуков

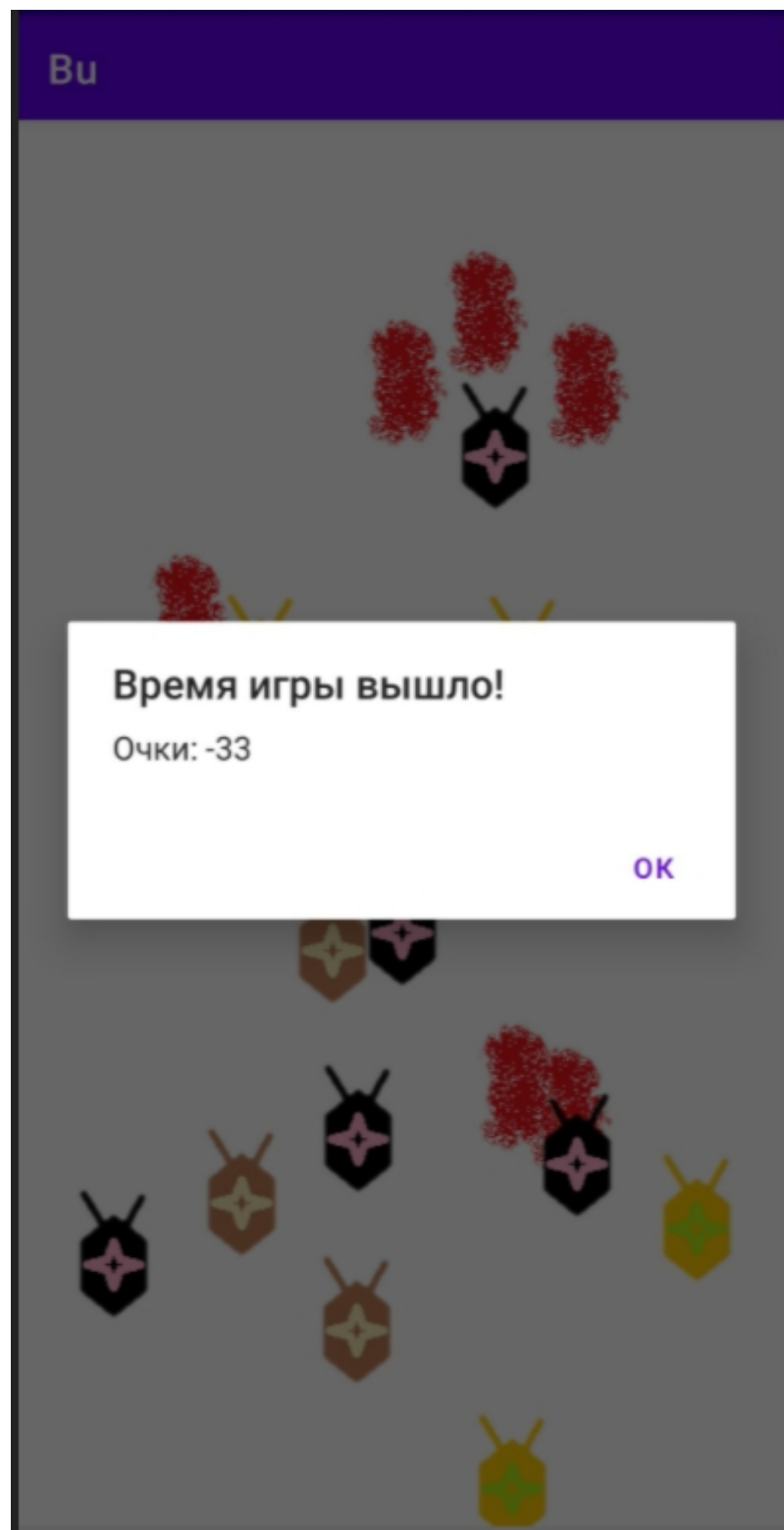


Рисунок 4 – итоговый счёт