Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине: «РПО»

Тема: “Игра “Память” на Android”

Выполнил:

Студент 3 курса

Группы ПО-7

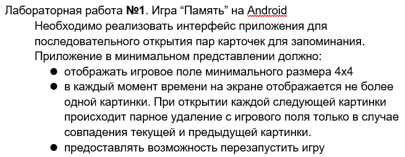
Комиссаров А.Е.

Проверил:

Козинский А.А.

2023

**Цель работы:** получить знания о написании программ с графическим интерфейсом на Android путём создания игры.



**Ход работы.**

В ходе лабораторной работы были реализованы функции:

* Главное меню с кнопками запуска игры, выхода, а также перехода в настройки.
* Можно настроить тип картинок, размеры поля, громкость игровых звуков, откат к стандартным настройкам.
* Игровое поле с кнопками рестарта, паузой, выходом в меню.
* Отображение игрового времени (останавливается с паузой), количества попыток, оставшихся плиток.
* Анимация плиток.
* Звуки нажатия на плитки.

**Файл MainActivity.java**

**package com.example.memorygame;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.util.DisplayMetrics;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.LinearLayout;**

**public class MainActivity extends Activity {**

**int SETTINGS\_REQUEST = 1;**

**int GAME\_REQUEST = 2;**

**@Override**

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

**super.onCreate(savedInstanceState);**

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

**DisplayMetrics displayMetrics = new DisplayMetrics();**

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

**int height = displayMetrics.heightPixels;**

**int width = displayMetrics.widthPixels;**

**LinearLayout mainLayout = findViewById(R.id.mainLayout);**

**Button buttonStart = (Button) findViewById(R.id.mainButtonStart);**

**Button buttonSettings = (Button) findViewById(R.id.mainButtonSettings);**

**Button buttonExit = (Button) findViewById(R.id.mainButtonExit);**

**buttonStart.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**Intent myIntent = new Intent(getApplicationContext(), GameField.class);**

**MainActivity.this.startActivityForResult(myIntent,GAME\_REQUEST);**

**}**

**});**

**buttonSettings.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**Intent myIntent = new Intent(MainActivity.this, SettingsActivity.class);**

**MainActivity.this.startActivityForResult(myIntent,SETTINGS\_REQUEST);**

**}**

**});**

**buttonExit.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**finishAffinity();**

**}**

**});**

**}**

**}**

**Файл GameField.java**

**package com.example.memorygame;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.graphics.Color;**

**import android.os.Bundle;**

**import android.util.DisplayMetrics;**

**import android.view.View;**

**import android.widget.AdapterView;**

**import android.widget.Button;**

**import android.widget.GridView;**

**import android.widget.TextView;**

**import android.widget.Toast;**

**import androidx.preference.PreferenceManager;**

**import java.util.Timer;**

**import java.util.TimerTask;**

**public class GameField extends Activity {**

**int SETTINGS\_REQUEST = 1;**

**private GridAdapter mAdapter;**

**boolean counterReady = false;**

**private Timer timer = new Timer();**

**public TimerTask timerTask = new TimerTask() {**

**@Override**

**public void run() {**

**if(counterReady){**

**int time = mAdapter.gameTime;**

**int tiles = mAdapter.tilesLeft;**

**int tries = mAdapter.tries;**

**TextView timeText = findViewById(R.id.timeText);**

**if(mAdapter.paused){**

**timeText.setText("Time : " + time + " (paused)");**

**} else {**

**timeText.setText("Time : " + time);**

**}**

**TextView tileText = findViewById(R.id.tileText);**

**tileText.setText("Tiles Left : " + tiles);**

**TextView triesText = findViewById(R.id.triesText);**

**triesText.setText("Tries : " + tries);**

**}**

**}**

**};**

**@Override**

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

**int GRID\_WIDTH = PreferenceManager.getDefaultSharedPreferences(getApplicationContext()).getInt("gridWidth", 2) \* 2;**

**int GRID\_HEIGHT = PreferenceManager.getDefaultSharedPreferences(getApplicationContext()).getInt("gridHeight", 2) \* 2;**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.game\_field);**

**DisplayMetrics displayMetrics = new DisplayMetrics();**

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

**int height = displayMetrics.heightPixels/2;**

**GridView mGrid = findViewById(R.id.field);**

**mGrid.setNumColumns(GRID\_WIDTH);**

**mGrid.setVerticalSpacing(height/GRID\_HEIGHT);**

**mGrid.setEnabled(true);**

**mAdapter = new GridAdapter(getApplicationContext(), GRID\_WIDTH, GRID\_HEIGHT);**

**mGrid.setAdapter(mAdapter);**

**mGrid.setOnItemClickListener(new AdapterView.OnItemClickListener() {**

**@Override**

**public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {**

**mAdapter.onPress(position);**

**mAdapter.notifyDataSetChanged();**

**}**

**});**

**Button buttonRestart = (Button) findViewById(R.id.buttonRestart);**

**Button buttonPause = (Button) findViewById(R.id.buttonPause);**

**Button buttonExit = (Button) findViewById(R.id.buttonExit);**

**buttonRestart.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**mAdapter.resetCells();**

**mAdapter.notifyDataSetChanged();**

**buttonPause.setText("pause");**

**mGrid.setEnabled(true);**

**mGrid.setBackgroundColor(Color.rgb(59, 59, 59));**

**}**

**});**

**buttonPause.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**boolean paused = mAdapter.pauseGame();**

**if(paused){**

**Toast.makeText(getApplicationContext(), "paused", Toast.LENGTH\_SHORT).show();**

**buttonPause.setText("unpause");**

**mGrid.setEnabled(false);**

**mGrid.setBackgroundColor(Color.rgb(163, 64, 64));**

**} else {**

**Toast.makeText(getApplicationContext(), "unpaused", Toast.LENGTH\_SHORT).show();**

**buttonPause.setText("pause");**

**mGrid.setEnabled(true);**

**mGrid.setBackgroundColor(Color.rgb(59, 59, 59));**

**}**

**}**

**});**

**buttonExit.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**finish();**

**}**

**});**

**counterReady = true;**

**timer.schedule(timerTask, 0, 100);**

**}**

**@Override**

**public void onActivityResult(int requestCode, int resultCode, Intent data)**

**{**

**if(requestCode == SETTINGS\_REQUEST){**

**this.finish();**

**startActivity(this.getIntent());**

**}**

**}**

**}**

**Файл GridAdapter.java**

**package com.example.memorygame;**

**import android.content.Context;**

**import android.media.MediaPlayer;**

**import android.os.Handler;**

**import android.os.Looper;**

**import android.view.View;**

**import android.view.ViewGroup;**

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

**import android.widget.BaseAdapter;**

**import android.widget.ImageView;**

**import android.widget.Toast;**

**import androidx.preference.PreferenceManager;**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.Timer;**

**import java.util.TimerTask;**

**public class GridAdapter extends BaseAdapter {**

**private Context mContext;**

**private Integer mCols = 0, mRows = 0;**

**private MediaPlayer itemPressSound;**

**private MediaPlayer youWinSound;**

**private MediaPlayer successSound;**

**private MediaPlayer failedSound;**

**private MediaPlayer reloadSound;**

**private boolean soundsReady = false;**

**private enum Status {STARTING, OPENED, CLOSING, CLOSED, DYING, DEAD};**

**private ArrayList<Status> cellStats = new ArrayList<>();**

**private ArrayList<Integer> cellType = new ArrayList<>();**

**private Integer[] animals = { R.drawable.animal1, R.drawable.animal2, R.drawable.animal3, R.drawable.animal4, R.drawable.animal5, R.drawable.animal6, R.drawable.animal7, R.drawable.animal8, R.drawable.animal9, R.drawable.animal10, R.drawable.animal11, R.drawable.animal12, R.drawable.animal13, R.drawable.animal14, R.drawable.animal15, R.drawable.animal16, R.drawable.animal17, R.drawable.animal18};**

**private Integer[] shapes = {R.drawable.shape1,R.drawable.shape2,R.drawable.shape3,R.drawable.shape4,R.drawable.shape5,R.drawable.shape6,R.drawable.shape7,R.drawable.shape8,R.drawable.shape9,R.drawable.shape10,R.drawable.shape11,R.drawable.shape12,R.drawable.shape13,R.drawable.shape14,R.drawable.shape15,R.drawable.shape16,R.drawable.shape17,R.drawable.shape18};**

**private Integer[] emojis = {R.drawable.emoji1,R.drawable.emoji2,R.drawable.emoji3,R.drawable.emoji4,R.drawable.emoji5,R.drawable.emoji6,R.drawable.emoji7,R.drawable.emoji8,R.drawable.emoji9,R.drawable.emoji10,R.drawable.emoji11,R.drawable.emoji12,R.drawable.emoji13,R.drawable.emoji14,R.drawable.emoji15,R.drawable.emoji16,R.drawable.emoji17,R.drawable.emoji18};**

**public boolean paused = false;**

**public int gameTime = 0;**

**public int tilesLeft = 0;**

**public int tries = 0;**

**private Timer gameTimeCounter = new Timer();**

**public TimerTask countTime = new TimerTask() {**

**@Override**

**public void run() {**

**if(!paused){**

**gameTime++;**

**}**

**tilesLeft=mCols\*mRows - countDeleted();**

**}**

**};**

**public boolean pauseGame(){**

**if(paused){**

**paused=false;**

**} else {**

**paused = true;**

**}**

**return paused;**

**}**

**public GridAdapter(Context context, int cols, int rows) {**

**mContext = context;**

**mCols = cols;**

**mRows = rows;**

**tilesLeft = mCols\*mRows;**

**setupSounds();**

**soundsReady=true;**

**setupCells();**

**gameTimeCounter.schedule(countTime, 0, 1000);**

**}**

**public void setupSounds(){**

**int soundVolumeSrc = PreferenceManager.getDefaultSharedPreferences(mContext.getApplicationContext()).getInt("soundVolume", 100);**

**boolean soundEnabled = PreferenceManager.getDefaultSharedPreferences(mContext.getApplicationContext()).getBoolean("soundEnabled", true);**

**float soundVolume = 1.f;**

**if(soundEnabled){**

**soundVolume = soundVolumeSrc/100.f;**

**} else {**

**soundVolume = 0;**

**}**

**if(!soundsReady){**

**itemPressSound = MediaPlayer.create(mContext.getApplicationContext(), R.raw.press);**

**youWinSound = MediaPlayer.create(mContext.getApplicationContext(), R.raw.levelup);**

**successSound = MediaPlayer.create(mContext.getApplicationContext(), R.raw.success);**

**failedSound = MediaPlayer.create(mContext.getApplicationContext(), R.raw.failed);**

**reloadSound = MediaPlayer.create(mContext.getApplicationContext(), R.raw.reload);**

**}**

**failedSound.setVolume(0.6f \* soundVolume,0.6f \* soundVolume);**

**itemPressSound.setVolume(soundVolume, soundVolume);**

**youWinSound.setVolume(soundVolume, soundVolume);**

**successSound.setVolume(soundVolume, soundVolume);**

**reloadSound.setVolume(soundVolume, soundVolume);**

**}**

**public int getPicId(int number){**

**String picType = PreferenceManager.getDefaultSharedPreferences(mContext.getApplicationContext()).getString("pictureTypeSetting", "1");**

**if(picType.equals("1")){**

**return animals[number];**

**}**

**if(picType.equals("2")){**

**return shapes[number];**

**}**

**if(picType.equals("3")){**

**return emojis[number];**

**}**

**return animals[number];**

**}**

**@Override**

**public int getCount() {**

**return mCols \* mRows;**

**}**

**@Override**

**public Object getItem(int position) {**

**return null;**

**}**

**@Override**

**public long getItemId(int position) {**

**return 0;**

**}**

**@Override**

**public View getView(int position, View convertView, ViewGroup parent) {**

**ImageView view;**

**if (convertView == null)**

**view = new ImageView(mContext);**

**else**

**view = (ImageView) convertView;**

**switch(cellStats.get(position)){**

**case STARTING:**

**view.clearAnimation();**

**view.setImageResource(R.drawable.close);**

**view.startAnimation(AnimationUtils.loadAnimation(mContext.getApplicationContext(), R.anim.start));**

**final Handler handler3 = new Handler(Looper.getMainLooper());**

**handler3.postDelayed(new Runnable() {**

**@Override**

**public void run() {**

**cellStats.set(position, Status.CLOSED);**

**}**

**}, 300);**

**break;**

**case OPENED:**

**view.clearAnimation();**

**view.setImageResource(getPicId(cellType.get(position)));**

**view.startAnimation(AnimationUtils.loadAnimation(mContext.getApplicationContext(), R.anim.shake));**

**break;**

**case DYING:**

**view.clearAnimation();**

**view.setImageResource(getPicId(cellType.get(position)));**

**view.startAnimation(AnimationUtils.loadAnimation(mContext.getApplicationContext(), R.anim.shrink));**

**final Handler handler = new Handler(Looper.getMainLooper());**

**handler.postDelayed(new Runnable() {**

**@Override**

**public void run() {**

**cellStats.set(position, Status.DEAD);**

**notifyDataSetChanged();**

**}**

**}, 400);**

**break;**

**case CLOSING:**

**view.clearAnimation();**

**view.startAnimation(AnimationUtils.loadAnimation(mContext.getApplicationContext(), R.anim.spin));**

**view.setImageResource(getPicId(cellType.get(position)));**

**final Handler handler2 = new Handler(Looper.getMainLooper());**

**handler2.postDelayed(new Runnable() {**

**@Override**

**public void run() {**

**cellStats.set(position, Status.CLOSED);**

**notifyDataSetChanged();**

**}**

**}, 400);**

**break;**

**case CLOSED:**

**view.clearAnimation();**

**view.setImageResource(R.drawable.close);**

**break;**

**case DEAD:**

**view.clearAnimation();**

**view.setImageResource(R.drawable.none);**

**break;**

**}**

**return view;**

**}**

**public void setupCells(){**

**if(!cellStats.isEmpty()){**

**cellStats.clear();**

**}**

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

**cellStats.add(Status.STARTING);**

**cellType.add(i%(getCount()/2));**

**}**

**Collections.shuffle(cellType);**

**reloadSound.start();**

**}**

**public void resetCells(){**

**if(!cellStats.isEmpty()){**

**cellStats.clear();**

**}**

**if(!cellType.isEmpty()){**

**cellType.clear();**

**}**

**setupCells();**

**setupSounds();**

**gameTime = 0;**

**paused=false;**

**tries = 0;**

**}**

**public void onPress(int position){**

**if(!paused) {**

**int openedCellsCount = countOpen();**

**if (openedCellsCount < 2) {**

**itemPressSound.start();**

**switch (cellStats.get(position)) {**

**case CLOSED:**

**openedCellsCount++;**

**cellStats.set(position, Status.OPENED);**

**break;**

**case OPENED:**

**cellStats.set(position, Status.CLOSED);**

**break;**

**case DEAD:**

**break;**

**}**

**}**

**if (openedCellsCount >= 2) {**

**checkCells();**

**if (countDeleted() == getCount()) {**

**int seconds = gameTime;**

**int minutes = gameTime / 60;**

**seconds = seconds - (minutes \* 60);**

**youWinSound.start();**

**String winMessage = "You win! It only took you...";**

**if (gameTime > 60) {**

**winMessage = winMessage + minutes + " minutes, " + seconds + " seconds.";**

**} else {**

**winMessage = winMessage + seconds + " seconds.";**

**}**

**Toast.makeText(mContext.getApplicationContext(), winMessage, Toast.LENGTH\_LONG).show();**

**paused = true;**

**}**

**}**

**} else {**

**failedSound.start();**

**}**

**}**

**public void checkCells(){**

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

**for (int j = 0; j<getCount(); j++){**

**if(i != j){**

**if(cellStats.get(i) == Status.OPENED && cellStats.get(j) == Status.OPENED){**

**if(cellType.get(i) == cellType.get(j)){**

**cellStats.set(i, Status.DYING);**

**cellStats.set(j, Status.DYING);**

**successSound.start();**

**}**

**}**

**}**

**}**

**}**

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

**if(cellStats.get(i) == Status.OPENED){**

**cellStats.set(i, Status.CLOSING);**

**failedSound.start();**

**}**

**}**

**tries++;**

**}**

**public int countDeleted(){**

**int cells = 0;**

**for(Status cell : cellStats){**

**if(cell == Status.DEAD || cell == Status.DYING) cells++;**

**}**

**return cells;**

**}**

**public int countOpen(){**

**int cells = 0;**

**for(Status cell : cellStats){**

**if(cell == Status.OPENED) cells++;**

**}**

**return cells;**

**}**

**}**

**Файл SettingsActivity.java**

**package com.example.memorygame;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.view.View;**

**import android.widget.ImageButton;**

**import androidx.annotation.NonNull;**

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

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

**import androidx.preference.Preference;**

**import androidx.preference.PreferenceFragmentCompat;**

**import androidx.preference.PreferenceManager;**

**public class SettingsActivity extends AppCompatActivity {**

**@Override**

**public void onBackPressed() {**

**finish();**

**}**

**@Override**

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

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.settings\_activity);**

**ActionBar actionBar = getSupportActionBar();**

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

**actionBar.setDisplayHomeAsUpEnabled(true);**

**}**

**ImageButton buttonBack = (ImageButton) findViewById(R.id.settings\_buttonBack);**

**buttonBack.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**finish();**

**}**

**});**

**getSupportFragmentManager()**

**.beginTransaction()**

**.replace(R.id.settings\_fragment, new MySettings())**

**.commit();**

**}**

**public static class MySettings extends PreferenceFragmentCompat {**

**@Override**

**public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {**

**setPreferencesFromResource(R.xml.settings\_screen, rootKey);**

**Preference resetButton = findPreference("defaultReset");**

**assert resetButton != null;**

**resetButton.setOnPreferenceClickListener(new Preference.OnPreferenceClickListener() {**

**@Override**

**public boolean onPreferenceClick(@NonNull Preference preference) {**

**SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(getPreferenceManager().getContext());**

**SharedPreferences.Editor editor = preferences.edit();**

**editor.clear();**

**editor.apply();**

**getActivity().finish();**

**startActivity(getActivity().getIntent());**

**return true;**

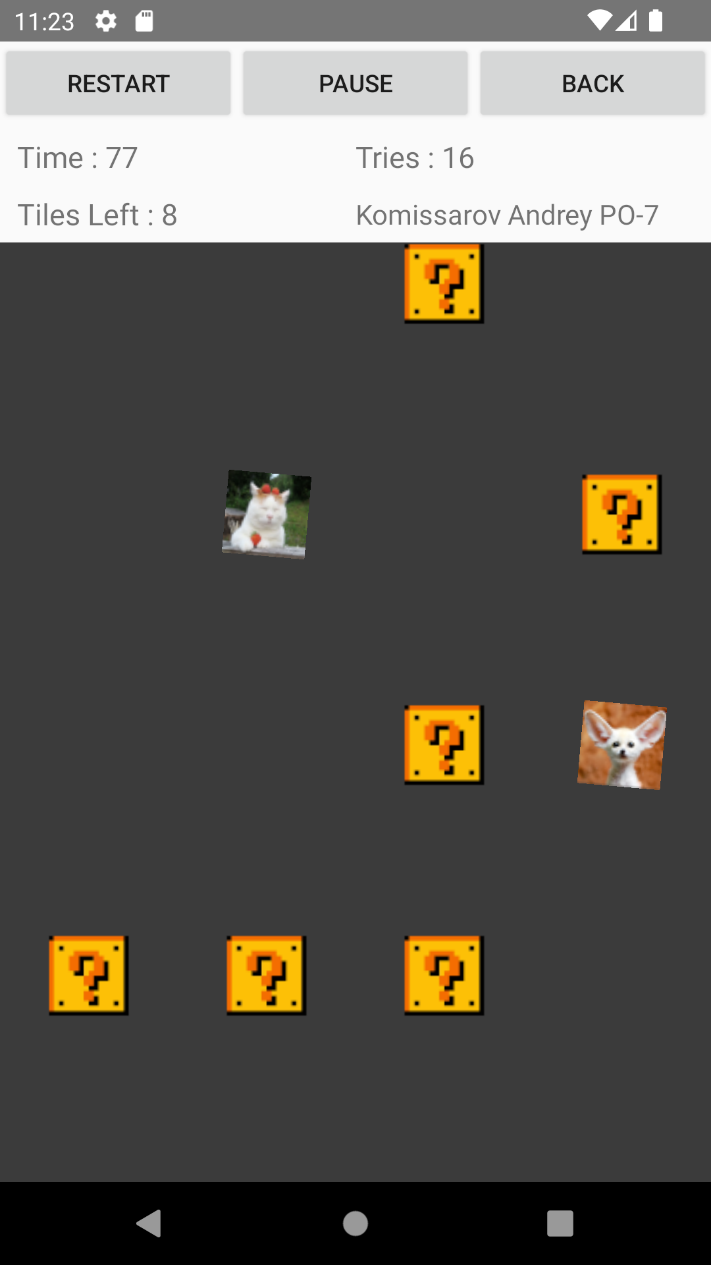
**}**

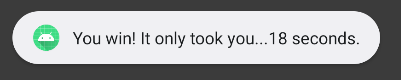
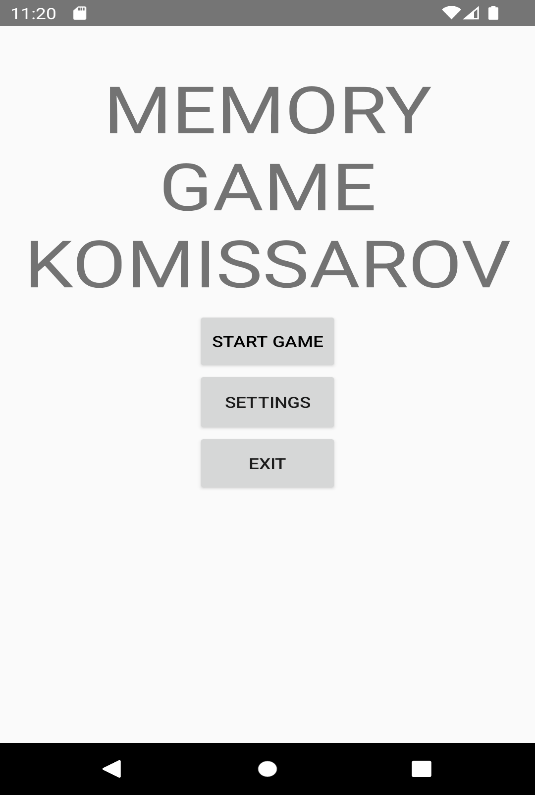
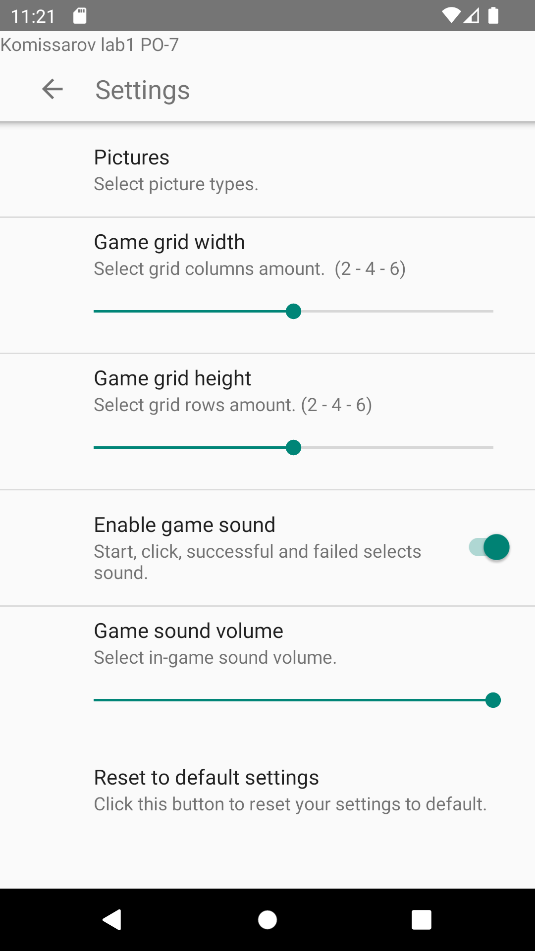
**});**

**}**

**}**

**}**

**Результат работы программы:**

****

**Вывод:** я получил знания о написании программ с графическим интерфейсом на Android путём создания игры.