Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

Игра «Память»

Выполнил:

Студент 4 курса

Группы АС-56

Волк Н.А.

Проверил:

Козинский А.А.

Брест 2022

Лабораторная работа №1

Цель работы: реализовать интерфейс приложения для последовательного открытия пар карточек для запоминания.

Необходимо реализовать интерфейс приложения для последовательного открытия пар карточек для запоминания. Приложение в минимальном представлении должно:

* отображать игровое поле минимального размера 4х4
* в каждый момент времени на экране отображается не более одной картинки. При открытии каждой следующей картинки происходит парное удаление с игрового поля только в случае совпадения текущей и предыдущей картинки.
* предоставлять возможность перезапустить игру

Бонусы (то, что способствует оценке выше 4)

– Возможность настраивать установки игрового поля (переменный размер игрового поля, цвета игры, наборы картинок и т.п.)

– Возможность присутствия на игровом поле не только ПАР, но и ТРОЕК картинок

– Запись информации (пользователи, рекорды, время игры и т.п.)

– Собственное оформление игры (отображение информации для пользователя время, число оставшихся пар-троек и т.п.)

**Код программы:**

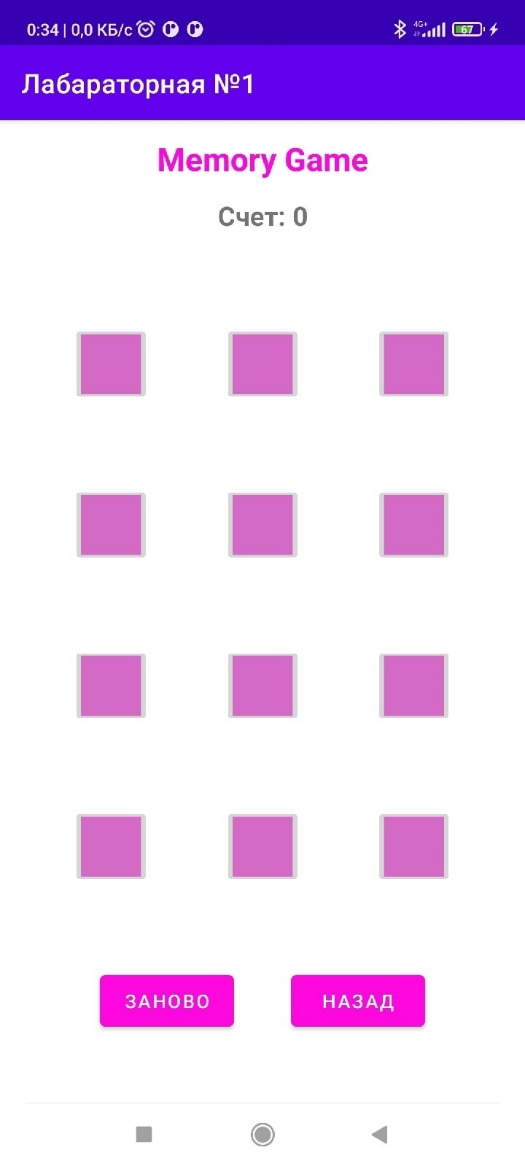
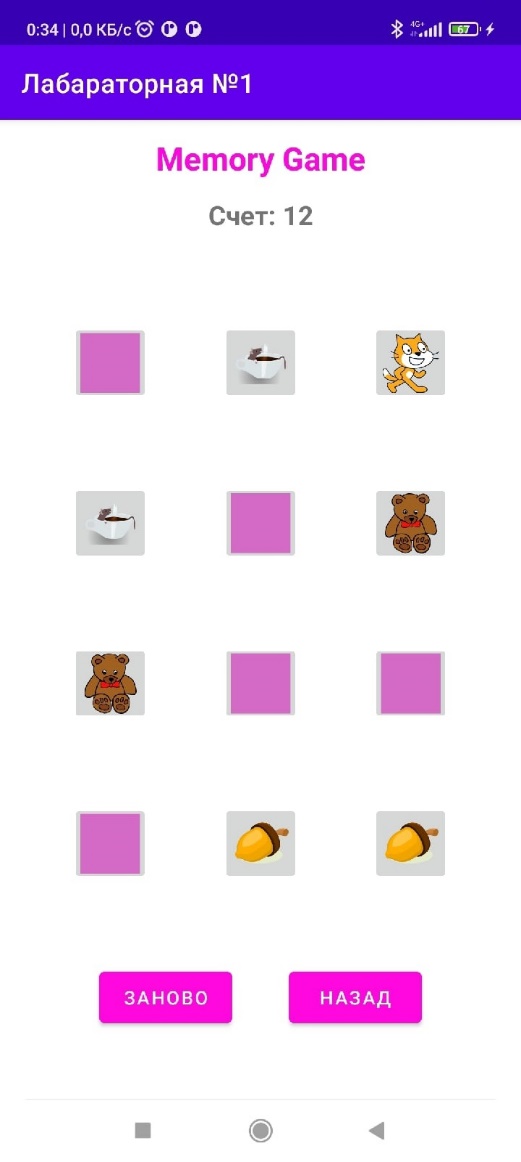
MainActivity.java

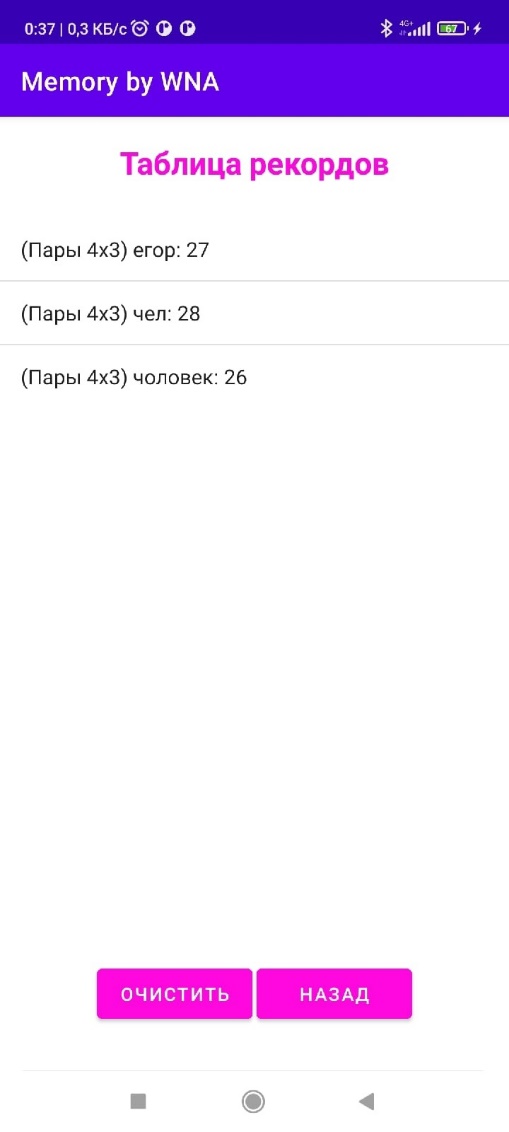
package com.example.lab1;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.Html;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
import java.util.Objects;  
  
import static com.example.lab1.lab1SettingsActivity.*field*;  
import static com.example.lab1.lab1SettingsActivity.*gamemode*;  
  
public class lab1GameActivity extends AppCompatActivity implements View.OnClickListener {  
 Button btnBack;  
 Button btnAgain;  
 TextView title, score;  
 ArrayList<ImageButton> buttons = new ArrayList<>();  
 ArrayList<Integer> pics = new ArrayList<>();  
 private int endGame = 0, pic1 = -1, pic2 = -1, pic3 = -1;  
 private String recordName = "";  
 int recordCounting = 0;  
 private List<Record> records;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 if (*field* == 12) setContentView(R.layout.*lab1\_field4x3*);  
 else if (*field* == 16) setContentView(R.layout.*lab1\_field4x4*);  
 else if (*field* == 20) setContentView(R.layout.*lab1\_field4x5*);  
 else setContentView(R.layout.*lab1\_field4x6*);  
 setTitle("Лабараторная №1");  
  
 records = new ArrayList<>();  
  
 title = (TextView) findViewById(R.id.*tvTitle*);  
 score = (TextView) findViewById(R.id.*tvScore*);  
 score.setText("Счет: 0");  
  
 pics.add(R.drawable.*ic\_1*); pics.add(R.drawable.*ic\_2*); pics.add(R.drawable.*ic\_3*); pics.add(R.drawable.*ic\_4*); pics.add(R.drawable.*ic\_5*); pics.add(R.drawable.*ic\_6*); pics.add(R.drawable.*ic\_7*); pics.add(R.drawable.*ic\_8*); pics.add(R.drawable.*ic\_9*); pics.add(R.drawable.*ic\_10*); pics.add(R.drawable.*ic\_11*); pics.add(R.drawable.*ic\_12*);  
 Collections.*shuffle*(pics);  
 if (!*gamemode*) {  
 pics.subList(*field*/2, pics.size()).clear();  
 pics.addAll(pics);  
 } else {  
 pics.subList(*field*/3, pics.size()).clear();  
 pics.addAll(pics);  
 pics.addAll(pics);  
 }  
  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn1*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn2*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn3*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn4*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn5*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn6*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn7*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn8*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn9*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn10*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn11*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn12*));  
 if (*field* != 12){  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn13*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn14*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn15*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn16*));  
 if (*field* != 16){  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn17*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn18*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn19*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn20*));  
 if (*field* != 20) {  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn21*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn22*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn23*));  
 buttons.add((ImageButton)findViewById(R.id.*gameBtn24*));  
 }  
 }  
 }  
  
 Collections.*shuffle*(buttons);  
 for(ImageButton i : buttons)  
 i.setOnClickListener(this);  
  
 btnBack = (Button) findViewById(R.id.*btnBack*);  
 btnBack.setOnClickListener(this);  
 btnAgain = (Button) findViewById(R.id.*btnAgain*);  
 btnAgain.setOnClickListener(this);  
 }  
  
 @SuppressLint("SetTextI18n")  
 public void gamemodePairs(View view){  
 int i = buttons.indexOf(view);  
 buttons.get(i).setImageResource(pics.get(i));  
 if (pic1 == -1){  
 pic1 = i;  
 buttons.get(pic1).setClickable(false);  
 buttons.get(pic1).setEnabled(false);  
 }  
 else if (pic2 == -1 && endGame != *field*-2){  
 pic2 = i;  
 buttons.get(pic2).setClickable(false);  
 buttons.get(pic2).setEnabled(false);  
 }  
 else {  
 if (endGame == *field*-2) pic2 = i;  
 if (Objects.*equals*(pics.get(pic1), pics.get(pic2))){  
 endGame += 2;  
 recordCounting += 6;  
 score.setText("Счет: " + recordCounting);  
 if (endGame == *field*) {  
 title.setText("Победа!");  
 buttons.get(pic2).setClickable(false);  
 buttons.get(pic2).setEnabled(false);  
 recordBox(recordCounting);  
 }  
 else {  
 recordCounting--;  
 score.setText("Счет: " + recordCounting);  
  
 pic1 = i;  
 buttons.get(pic1).setClickable(false);  
 buttons.get(pic1).setEnabled(false);  
 pic2 = -1;  
 }  
 }  
 else {  
 recordCounting--;  
 score.setText("Счет: " + recordCounting);  
  
 buttons.get(pic1).setClickable(true);  
 buttons.get(pic1).setEnabled(true);  
 buttons.get(pic2).setClickable(true);  
 buttons.get(pic2).setEnabled(true);  
 buttons.get(i).setClickable(false);  
 buttons.get(i).setEnabled(false);  
 buttons.get(pic1).setImageResource(R.drawable.*ic\_filled*);  
 buttons.get(pic2).setImageResource(R.drawable.*ic\_filled*);  
 pic1 = i;  
 pic2 = -1;  
 }  
 }  
 }  
  
 @SuppressLint("SetTextI18n")  
 public void gamemodeTrio(View view){  
 int i = buttons.indexOf(view);  
 buttons.get(i).setImageResource(pics.get(i));  
 if (pic1 == -1){  
 pic1 = i;  
 buttons.get(pic1).setClickable(false);  
 buttons.get(pic1).setEnabled(false);  
 }  
 else if (pic2 == -1){  
 pic2 = i;  
 buttons.get(pic2).setClickable(false);  
 buttons.get(pic2).setEnabled(false);  
 }  
 else if (pic3 == -1 && endGame != *field*-3){  
 if (Objects.*equals*(pics.get(pic1), pics.get(pic2))){  
 pic3 = i;  
 buttons.get(pic3).setClickable(false);  
 buttons.get(pic3).setEnabled(false);  
 } else {  
 recordCounting--;  
 score.setText("Счет: " + recordCounting);  
  
 buttons.get(pic1).setClickable(true);  
 buttons.get(pic1).setEnabled(true);  
 buttons.get(pic2).setClickable(true);  
 buttons.get(pic2).setEnabled(true);  
 buttons.get(i).setClickable(false);  
 buttons.get(i).setEnabled(false);  
 buttons.get(pic1).setImageResource(R.drawable.*ic\_filled*);  
 buttons.get(pic2).setImageResource(R.drawable.*ic\_filled*);  
 pic1 = i;  
 pic2 = -1;  
 }  
 }  
 else {  
 if (endGame == *field*-3) pic3 = i;  
 if (Objects.*equals*(pics.get(pic1), pics.get(pic2)) && Objects.*equals*(pics.get(pic1), pics.get(pic3))){  
 endGame += 3;  
 recordCounting += 6;  
 score.setText("Счет: " + recordCounting);  
  
 if (endGame == *field*) {  
 title.setText("Победа!");  
 buttons.get(pic3).setClickable(false);  
 buttons.get(pic3).setEnabled(false);  
 recordBox(recordCounting);  
 }  
 else {  
 recordCounting--;  
 score.setText("Счет: " + recordCounting);  
  
 pic1 = i;  
 buttons.get(pic1).setClickable(false);  
 buttons.get(pic1).setEnabled(false);  
 pic2 = -1;  
 pic3 = -1;  
 }  
 }  
 else {  
 recordCounting--;  
 score.setText("Счет: " + recordCounting);  
  
 buttons.get(pic1).setClickable(true);  
 buttons.get(pic1).setEnabled(true);  
 buttons.get(pic2).setClickable(true);  
 buttons.get(pic2).setEnabled(true);  
 buttons.get(pic3).setClickable(true);  
 buttons.get(pic3).setEnabled(true);  
 buttons.get(i).setClickable(false);  
 buttons.get(i).setEnabled(false);  
 buttons.get(pic1).setImageResource(R.drawable.*ic\_filled*);  
 buttons.get(pic2).setImageResource(R.drawable.*ic\_filled*);  
 buttons.get(pic3).setImageResource(R.drawable.*ic\_filled*);  
 pic1 = i;  
 pic2 = -1;  
 pic3 = -1;  
 }  
 }  
 }  
  
 public void recordBox(int record){  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
  
 final String string = "Введите имя:<br>Ваш результат: " + record;  
 builder.setTitle(Html.*fromHtml*("<font color='#6200ED'>"+ string +"</font>"));  
  
 final EditText nameInput = new EditText(this);  
 builder.setView(nameInput);  
  
 String additionalText = null;  
 if (!*gamemode*)  
 switch (*field*) {  
 case (12):  
 additionalText = "(Double 4x3) ";  
 case (16):  
 additionalText = "(Double 4x4) ";  
 case (20):  
 additionalText = "(Double 4x5) ";  
 case (24):  
 additionalText = "(Double 4x6) ";  
 }  
 else  
 switch (*field*) {  
 case (12):  
 additionalText = "(Triple 4x3) ";  
 case (24):  
 additionalText = "(Triple 4x6) ";  
 }  
  
 String finalAdditionalText = additionalText;  
 builder.setPositiveButton("Сохранить", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 recordName = finalAdditionalText + nameInput.getText().toString();  
 if (recordName.equals(finalAdditionalText))  
 recordName = finalAdditionalText + "unknown";  
  
 saveRecord(recordName, record);  
 }  
 });  
 builder.setNegativeButton("Отмена", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 dialog.cancel();  
 }  
 });  
  
 builder.show();  
 }  
  
 public void saveRecord(String recordName, int record){  
 Record rec = new Record(recordName, record);  
 records = JSONHelper.*importFromJSON*(this);  
 records.add(rec);  
 JSONHelper.*exportToJSON*(this, records);  
 }  
  
 @Override  
 public void onClick(View view) {  
 if (view.getId() == R.id.*btnBack*) {  
 Intent intent = new Intent(this, lab1SettingsActivity.class);  
 startActivity(intent);  
 } else if (view.getId() == R.id.*btnAgain*){  
 Intent intent = new Intent(this, lab1GameActivity.class);  
 startActivity(intent);  
 } else  
 if (!*gamemode*) gamemodePairs(view);  
 else gamemodeTrio(view);  
 }  
 }

Records.java

package com.example.lab1;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ListView;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class recordsActivity extends AppCompatActivity implements View.OnClickListener {  
  
 private ArrayAdapter<Record> adapter;  
 private List<Record> records;  
 ListView listView;  
  
 Button btnBack, btnClear;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_records*);  
  
 btnBack = (Button) findViewById(R.id.*btnBack*);  
 btnBack.setOnClickListener(this);  
 btnClear = (Button) findViewById(R.id.*btnClear*);  
 btnClear.setOnClickListener(this);  
  
 listView = findViewById(R.id.*list*);  
 records = JSONHelper.*importFromJSON*(this);  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, records);  
 listView.setAdapter(adapter);  
 }  
  
 @Override  
 public void onClick(View v){  
 switch (v.getId()){  
 case (R.id.*btnClear*):  
 records = new ArrayList<>();  
 JSONHelper.*exportToJSON*(this, records);  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, records);  
 listView.setAdapter(adapter);  
 break;  
  
 case (R.id.*btnBack*):  
 Intent intent = new Intent(this, lab1SettingsActivity.class);  
 startActivity(intent);  
 break;  
 }  
 }  
}

Пример выполнения программы:



***Вывод:*** реализовал интерфейс приложения для последовательного открытия пар карточек для запоминания.