**ОТЧЁТ ПО ИНДИВИДУАЛЬНОМУ ЗАДАНИЮ**

**ПРИЛОЖЕНИЕ “СУДОКУ”**

**Цель работы:** создать приложение “Судоку” для мобильных устройств

Напишем код, реализующий главное меню приложения с возможностью перехода в другие активности

Код MainActivity.java:

package com.example.myapplication;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.graphics.Typeface;  
import android.os.Bundle;  
  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 Button aboutBTN;  
 Button exitBTN;  
 Button newGameBTN;  
 private static Bundle *mSavedInstanceState*;  
  
  
 TextView title;  
 private long mTimeRemainingInMillis;  
 private TextView mTimerTextView;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 title = (TextView)findViewById(R.id.*mainTitle*);  
 aboutBTN = (Button) findViewById(R.id.*btnAbout*);  
 exitBTN = (Button) findViewById(R.id.*btnExit*);  
 newGameBTN = (Button) findViewById(R.id.*btnNewGame*);  
  
  
  
  
 aboutBTN.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 showAbout();  
 }  
 });  
  
 exitBTN.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 finish();  
 }  
 });  
  
 newGameBTN.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startGame();  
 }  
 });  
  
 Typeface type = Typeface.*createFromAsset*(getAssets(), "my-font.ttf");  
 title.setTypeface(type);  
 aboutBTN.setTypeface(type);  
 exitBTN.setTypeface(type);  
 newGameBTN.setTypeface(type);  
   
  
 if (savedInstanceState != null) {  
 *mSavedInstanceState* = savedInstanceState;  
 }  
  
  
  
 }  
  
  
 private void startGame() {  
 // mGame.generateNewField();  
 Intent intent = new Intent(this, GameActivity.class);  
 startActivity(intent);  
 }  
  
 private void showAbout() {  
 Intent intent = new Intent(this, About.class);  
 startActivity(intent);  
 }  
  
  
  
}

Код activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:padding="30dp"  
 android:orientation="vertical"  
 android:background="#8A2BE2">  
  
 <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="40sp"  
 android:text="@string/mainTitle"  
 android:gravity="center"  
 android:layout\_marginBottom="5dip"  
 android:id="@+id/mainTitle"  
 android:textColor="#EFEBF3"/>  
   
 <Button  
 android:id="@+id/btnNewGame"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="100dp"  
 android:onClick="newGame"  
 android:text="@string/new\_game"  
 android:textSize="45dp" />  
  
 <Button  
 android:id="@+id/btnAbout"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="100dp"  
 android:onClick="aboutGame"  
 android:text="@string/about"  
 android:textSize="45dp" />  
  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/btnExit"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="100dp"  
 android:onClick="exitGame"  
 android:text="@string/exitGame"  
 android:textSize="45dp" />  
  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="900px"  
 android:layout\_height="900px"  
 android:layout\_centerInParent="true"  
 android:src="@drawable/sudoku1"  
 android:layout\_marginTop="25dp"/>  
  
  
</LinearLayout>

В файле GameActivity.java напишем код, реализующий игровое поле Судоку, а также возможность проверки на повторение числа в строке/столбце/квадрате 3x3, возможность вывода экрана победы с временем прохождения при правильно заполненном поле и т.д.

Код GameActivity.java:

package com.example.myapplication;  
  
import android.app.Activity;  
import android.content.DialogInterface;  
import android.graphics.Typeface;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import android.view.View;  
import android.os.SystemClock;  
import android.widget.Chronometer;  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.view.View.OnClickListener;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
  
public class GameActivity extends Activity implements View.OnClickListener {  
  
 private GridView mGridField;  
 private Button b1, b2, b3, b4, b5, b6, b7, b8, b9;  
 private String selectedButton = "n1";  
 private Game mGame;  
 TextView title;  
 private Chronometer chronometer;  
 private boolean isFirstRun = true;  
 private long baseTime;  
 private static final String *GAME\_PREFERENCES* = "GamePreferences";  
 private static final String *KEY\_BASE\_TIME* = "baseTime";  
 private static final String *KEY\_GAME\_STATE* = "gameState";  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*game*);  
  
  
  
 chronometer = findViewById(R.id.*Chronometer*);  
 chronometer.setBase(SystemClock.*elapsedRealtime*());  
 chronometer.start();  
  
 baseTime = chronometer.getBase();  
  
  
 mGridField = (GridView) findViewById(R.id.*field*);  
 mGridField.setNumColumns(9);  
 mGridField.setEnabled(true);  
  
  
  
 title = (TextView) findViewById(R.id.*gameTitle*);  
  
 mGame = new Game(this);  
 mGridField.setAdapter(mGame);  
 if (savedInstanceState != null) {  
 Game savedGame = savedInstanceState.getParcelable(*KEY\_GAME\_STATE*);  
 if (savedGame != null) {  
 mGame = savedGame;  
 if (isFirstRun || savedInstanceState == null) {  
 mGame = new Game(this);  
 isFirstRun = false;  
 }  
 mGridField.setAdapter(mGame);  
 }  
 }  
  
  
 mGridField.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {  
  
 mGame.setNumber(position, selectedButton);  
 if(mGame.checkRepeatedValues(selectedButton)){  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Число повторяется: " + selectedButton.split("n")[1], Toast.*LENGTH\_SHORT*);  
 toast.show();  
  
 }  
  
 if(mGame.checkWinner()) showWinnerDialog();  
 mGame.updateField();  
 }  
  
  
 });  
  
  
 b1 = (Button) findViewById(R.id.*btn1*);  
 b2 = (Button) findViewById(R.id.*btn2*);  
 b3 = (Button) findViewById(R.id.*btn3*);  
 b4 = (Button) findViewById(R.id.*btn4*);  
 b5 = (Button) findViewById(R.id.*btn5*);  
 b6 = (Button) findViewById(R.id.*btn6*);  
 b7 = (Button) findViewById(R.id.*btn7*);  
 b8 = (Button) findViewById(R.id.*btn8*);  
 b9 = (Button) findViewById(R.id.*btn9*);  
  
 b1.setOnClickListener(this);  
 b2.setOnClickListener(this);  
 b3.setOnClickListener(this);  
 b4.setOnClickListener(this);  
 b5.setOnClickListener(this);  
 b6.setOnClickListener(this);  
 b7.setOnClickListener(this);  
 b8.setOnClickListener(this);  
 b9.setOnClickListener(this);  
  
 Typeface type = Typeface.*createFromAsset*(getAssets(), "my-font.ttf");  
 title.setTypeface(type);  
  
  
  
 }  
  
 @Override  
 public void onClick(View view) {  
 if (view.getId() == R.id.*btn1*) {  
 selectedButton = "n1";  
 } else if (view.getId() == R.id.*btn2*) {  
 selectedButton = "n2";  
 } else if (view.getId() == R.id.*btn3*) {  
 selectedButton = "n3";  
 } else if (view.getId() == R.id.*btn4*) {  
 selectedButton = "n4";  
 } else if (view.getId() == R.id.*btn5*) {  
 selectedButton = "n5";  
 } else if (view.getId() == R.id.*btn6*) {  
 selectedButton = "n6";  
 } else if (view.getId() == R.id.*btn7*) {  
 selectedButton = "n7";  
 } else if (view.getId() == R.id.*btn8*) {  
 selectedButton = "n8";  
 } else if (view.getId() == R.id.*btn9*) {  
 selectedButton = "n9";  
 }  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 checkForVictory();  
 }  
 private void showWinnerDialog() {  
 final AlertDialog.Builder alertBox = new AlertDialog.Builder(this);  
 alertBox.setTitle("Победа");  
  
 long elapsedMillis = SystemClock.*elapsedRealtime*() - chronometer.getBase();  
 int hours = (int) (elapsedMillis / 3600000);  
 int minutes = (int) (elapsedMillis - hours \* 3600000) / 60000;  
 int seconds = (int) (elapsedMillis - hours \* 3600000 - minutes \* 60000) / 1000;  
  
 String TextToast = "Вы прошли судоку!\n" + String.*format*("Время: %02d:%02d", minutes, seconds);  
 chronometer.stop();  
  
 alertBox.setMessage(TextToast);  
 alertBox.setNeutralButton("Ok", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 finish();   
 }  
 });  
 alertBox.show();  
 }  
 private void checkForVictory() {  
 if (mGame.checkWinner()) {  
 showWinnerDialog();  
 }  
 }  
  
  
  
}

В файле Game.java напишем код, реализующий генерацию поля при старте, закрытие определённого количества ячеек, невозможность изменить число в автоматически заполненных ячейках и т.д.

Код Game.java:

package com.example.myapplication;  
  
import android.content.Context;  
import android.content.res.Resources;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
import android.os.Parcel;  
  
import android.os.Parcelable;  
  
  
import java.util.ArrayList;  
import java.util.Random;  
  
  
public class Game extends BaseAdapter implements Parcelable {  
  
 private Context mContext;  
 private final Integer mRows = 9, mCols = 9;  
 private int numberArray[][] = new int[9][9];  
  
 private Resources mRes;  
 private ArrayList<String> arrPict;  
  
 int UnblockPositions[] = new int[mRows \* mCols];  
 int helperArray[][];  
  
 public Game(Context context) {  
 mContext = context;  
 arrPict = new ArrayList<>(mCols \* mRows);  
 mRes = mContext.getResources();  
  
 createField();  
 }  
  
 private void createField() {  
 initilizeArray(numberArray);  
  
 shiftNumbers(3, 1);  
 shiftNumbers(6, 2);  
  
 shiftNumbers(1, 3);  
 shiftNumbers(4, 4);  
 shiftNumbers(7, 5);  
  
 shiftNumbers(2, 6);  
 shiftNumbers(5, 7);  
 shiftNumbers(8, 8);  
  
 transposingArray(numberArray);  
 shakeRowInsideArea();  
 transposingArray(numberArray);  
  
 for (int i = 0; i < mRows; i++) {  
 for (int j = 0; j < mCols; j++) {  
 arrPict.add("n" + numberArray[i][j]);  
 }  
  
 }  
 helperArray = numberArray;  
 Random r = new Random();  
 int i = 0;  
 while (i < 60) {  
 int i1 = r.nextInt(80);  
 arrPict.set(i1, "nempty");  
 UnblockPositions[i] = i1;  
 helperArray[getRow(i1)][getCell(i1)] = -1;  
 i++;  
 }  
 notifyDataSetChanged();  
  
 }  
  
 private void initilizeArray(int array[][]) {  
 for (int i = 0; i < mRows; i++) {  
 for (int j = 0; j < mCols; j++) {  
 numberArray[i][j] = j + 1;  
 }  
 }  
 }  
  
 private void shiftNumbers(int count, int row) {  
 int index;  
 for (int j = 0; j < mCols; j++) {  
 index = (j + count) % 9 + 1;  
 numberArray[row][j] = index;  
 }  
 }  
  
 private void transposingArray(int array[][]) {  
 for (int i = 0; i < mRows; i++) {  
 for (int j = 0; j < i; j++) {  
 int tmp = array[i][j];  
 array[i][j] = array[j][i];  
 array[j][i] = tmp;  
 }  
 }  
 }  
  
 private void shakeRowInsideArea() {  
  
 int i = 0;  
 do {  
 int tempArray[] = numberArray[i];  
 int tempArray2[] = numberArray[i + 1];  
  
 numberArray[i] = numberArray[i + 2];  
 numberArray[i + 1] = tempArray;  
 numberArray[i + 2] = tempArray2;  
 i = i + 3;  
 } while (i < mRows);  
 }  
  
  
 @Override  
 public int getCount() {  
 return mCols \* mRows;  
 }  
  
 @Override  
 public Object getItem(int i) {  
 return null;  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return 0;  
 }  
  
 @Override  
 public View getView(int position, View view, ViewGroup viewGroup) {  
  
 ImageView imageView;  
  
 if (view == null)  
 imageView = new ImageView(mContext);  
 else  
 imageView = (ImageView) view;  
  
 Integer drawableId = mRes.getIdentifier(arrPict.get(position), "drawable", mContext.getPackageName());  
 imageView.setImageResource(drawableId);  
  
 return imageView;  
 }  
  
 public void setNumber(int position, String selectedButton) {  
 for (int i = 0; i < UnblockPositions.length; i++) {  
 if (UnblockPositions[i] == position) {  
 arrPict.set(position, selectedButton);  
 helperArray[getRow(position)][getCell(position)] = Integer.*parseInt*(selectedButton.split("n")[1]);  
 notifyDataSetChanged();  
 }  
 }  
  
 }  
  
 public boolean checkRepeatedValues(String selectedButton) {  
 int repeatedX = 0;  
 int repeatedY = 0;  
 int number = Integer.*parseInt*(selectedButton.split("n")[1]);  
  
 for (int i = 0; i < mRows; i++) {  
 for (int j = 0; j < mCols; j++) {  
 if (helperArray[i][j] == number) {  
  
 repeatedX++;  
 }  
 if (helperArray[j][i] == number) {  
 repeatedY++;  
 }  
 }  
  
 if (repeatedY >= 2 || repeatedX >= 2) {  
 return true;  
 }  
 repeatedX = 0;  
 repeatedY = 0;  
  
 }  
 return false;  
 }  
  
 public int getRow(int position) {  
  
 int row = 1;  
 if (position <= 8) {  
 return 0;  
 } else {  
 while (position >= 0 && position < 9) {  
 row++;  
 }  
 while (position >= 9) {  
 position = position - 9;  
 row++;  
 }  
 return row - 1;  
 }  
  
 }  
  
 public int getCell(int position) {  
 if (position <= 8) {  
 return position;  
 } else {  
 return position % 9;  
 }  
 }  
  
 public boolean checkWinner() {  
 int i1 = 0, i2 = 0, i3 = 0, i4 = 0, i5 = 0, i6 = 0, i7 = 0, i8 = 0, i9 = 0;  
 for (int i = 0; i < mCols; i++) {  
 for (int j = 0; j < mRows; j++) {  
 if (helperArray[i][j] == 1) i1++;  
 if (helperArray[i][j] == 2) i2++;  
 if (helperArray[i][j] == 3) i3++;  
 if (helperArray[i][j] == 4) i4++;  
 if (helperArray[i][j] == 5) i5++;  
 if (helperArray[i][j] == 6) i6++;  
 if (helperArray[i][j] == 7) i7++;  
 if (helperArray[i][j] == 8) i8++;  
 if (helperArray[i][j] == 9) i9++;  
 }  
 }  
  
 if (i1 == 9 && i2 == 9 && i3 == 9 && i4 == 9 && i5 == 9 && i6 == 9 && i7 == 9 && i8 == 9 && i9 == 9) {  
 return true;  
 }  
  
 return false;  
  
 }  
  
 protected Game(Parcel in) {  
 // Восстановление данных из Parcel  
 for (int i = 0; i < UnblockPositions.length; i++) {  
 UnblockPositions[i] = in.readInt();  
 }  
 for (int i = 0; i < helperArray.length; i++) {  
 for (int j = 0; j < helperArray[i].length; j++) {  
 helperArray[i][j] = in.readInt();  
 }  
 }  
 // ... (другие ваши данные)  
 }  
  
 public static final Parcelable.Creator<Game> *CREATOR* = new Parcelable.Creator<Game>() {  
 @Override  
 public Game createFromParcel(Parcel in) {  
 return new Game(in);  
 }  
  
 @Override  
 public Game[] newArray(int size) {  
 return new Game[size];  
 }  
 };  
  
  
 public int describeContents() {  
 return 0;  
 }  
  
 public void writeToParcel(Parcel dest, int flags) {  
 // Сохранение данных в Parcel  
 for (int i = 0; i < UnblockPositions.length; i++) {  
 dest.writeInt(UnblockPositions[i]);  
 }  
 for (int i = 0; i < helperArray.length; i++) {  
 for (int j = 0; j < helperArray[i].length; j++) {  
 dest.writeInt(helperArray[i][j]);  
 }  
 }  
 // ... (другие ваши данные)  
 }  
  
 public void updateField() {  
 notifyDataSetChanged();  
 }  
}

Код game.xml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"  
 android:background="#8A2BE2">  
  
 <TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/gameTitle"  
 android:text="@string/mainTitle"  
 android:gravity="center"  
 android:textSize="50dp" />  
  
 <GridView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:id="@+id/field"  
 android:background="@drawable/bg2" />  
  
 <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:background="#c5718079"  
 android:layout\_marginTop="20dp">  
 <Button  
 android:id="@+id/btn1"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n1" />  
 <Button  
 android:id="@+id/btn2"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n2" />  
 <Button  
 android:id="@+id/btn3"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n3" />  
 <Button  
 android:id="@+id/btn4"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n4" />  
 <Button  
 android:id="@+id/btn5"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n5" />  
 <Button  
 android:id="@+id/btn6"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n6" />  
 <Button  
 android:id="@+id/btn7"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n7" />  
 <Button  
 android:id="@+id/btn8"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n8" />  
 <Button  
 android:id="@+id/btn9"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/n9" />  
  
  
  
 </LinearLayout>  
 <Chronometer  
 android:id="@+id/Chronometer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="50sp"  
 android:gravity="center"  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="5dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:layout\_marginLeft="130dp"/>  
</LinearLayout>

В файле About.java напишем код, реализующий возможность предоставления пользователю информации о правилах игры.

Код About.java:

package com.example.myapplication;  
  
import android.app.Activity;  
import android.graphics.Typeface;  
import android.os.Bundle;  
import android.widget.TextView;  
  
  
public class About extends Activity {  
  
 TextView aboutInfo;  
 TextView aboutTitile;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*about*);  
  
 aboutInfo = (TextView) findViewById(R.id.*aboutTV*);  
 aboutTitile = (TextView) findViewById(R.id.*aboutTitle*);  
  
 Typeface type = Typeface.*createFromAsset*(getAssets(), "my-font.ttf");  
 aboutInfo.setTypeface(type);  
 aboutTitile.setTypeface(type);  
  
 }  
}

Код about.xml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="fill\_parent"  
 android:background="#8A2BE2" >  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/about\_game"  
 android:textSize="50dp"  
 android:id="@+id/aboutTitle"  
 android:gravity="center"  
 android:textColor="#EFEBF3"/>  
  
 <ScrollView  
 android:id="@+id/SCROLLER\_ID"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:scrollbars="vertical"  
 android:fillViewport="true">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/some\_information\_about\_game"  
 android:id="@+id/aboutTV"  
 android:textSize="40dp"  
 android:layout\_margin="20dp"  
 android:paddingBottom="30dp"  
 android:textColor="#EFEBF3"/>  
 </ScrollView>  
</LinearLayout>

Результат работы:



Рисунок 1 – Главное меню

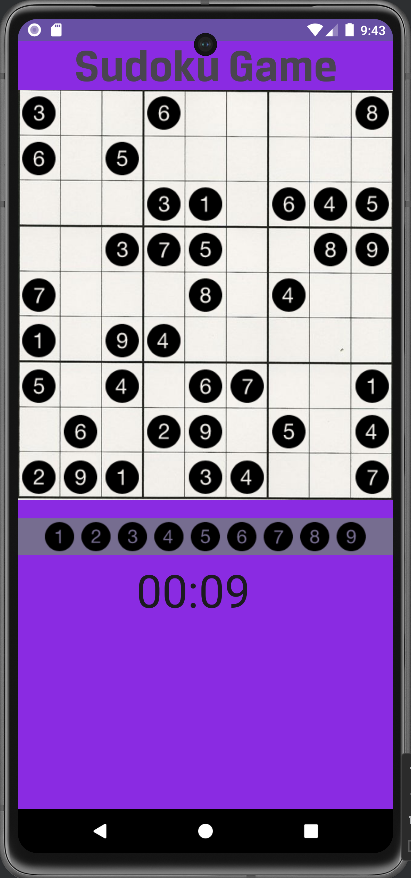


Рисунок 2 – Поле судоку

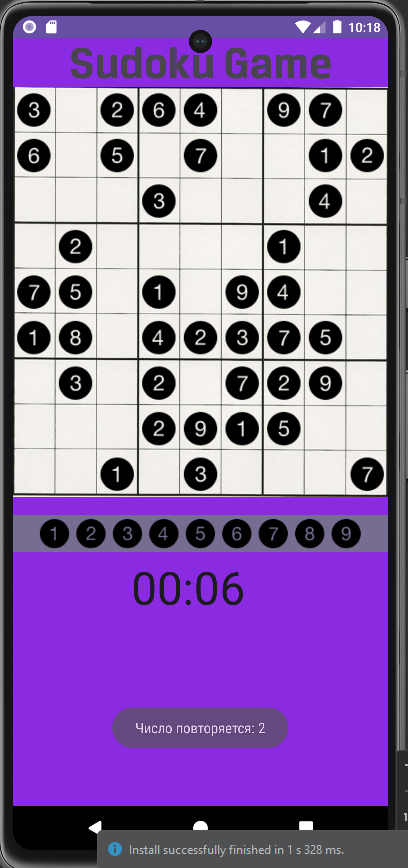


Рисунок 3 – Оповещением о повторяющемся числе

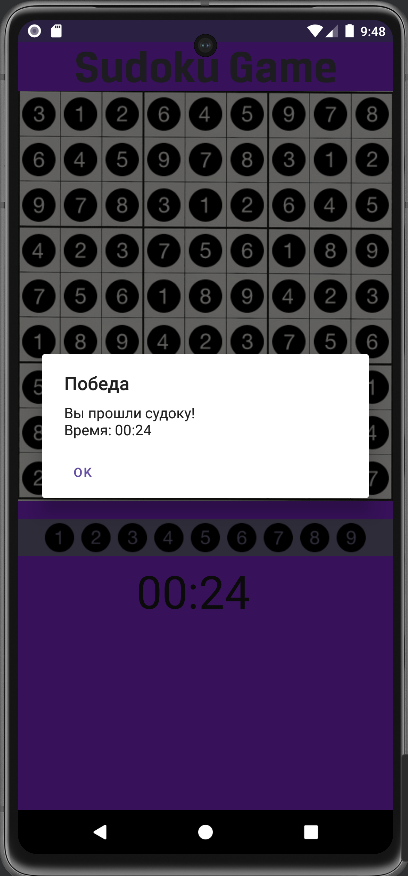


Рисунок 4 – Окно с оповещением о победе

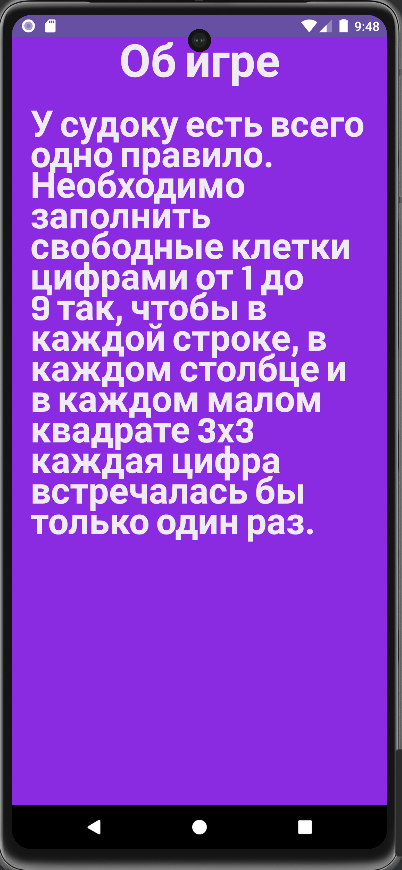


Рисунок 5 – Описание игры

**Вывод:** в ходе выполнения индивидуального задания было разработано приложение “Судоку” для мобильных устройств

|  |  |
| --- | --- |
| Выполнил | Семеняк В.А. |
| Проверил | Елкин Н.С. |