МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

Направление подготовки «Фундаментальная информатика и информационные технологии»

**ОТЧЕТ**

по лабораторной работе

**Проект по предмету “Технологии программирования мобильных систем”**

**Выполнил:** студент группы

381906-2 Углинский Б.С

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Подпись

**Проверил:** Карчков Д.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2022

Оглавление

[1. Цели работы. 3](#_Toc121155350)

[2. Постановка задачи. 4](#_Toc121155351)

[3. Решение задачи. 5](#_Toc121155352)

[4. Руководство пользователя. 8](#_Toc121155353)

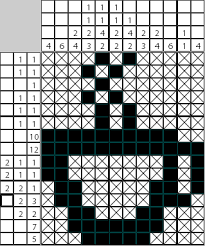
[Приложение. 11](#_Toc121155354)

# 1. Цели работы.

Целью данного проекта является закрепление всех ранее изученных материалов, а также освоение их комбинирования без использования чёткого технического задания и заранее оговоренного макета приложения.

# 2. Постановка задачи.

В качестве итогового приложения было выбрано создание вариации японского кроссворда-нонограммы размером 10 на 10 клеток. Это головоломка, в которой, в отличие от обычных кроссвордов закодированы не слова, а изображение. Изображения закодированы числами, расположенными слева от строк, а также сверху над столбцами. Количество чисел показывает, сколько групп цветных клеток находится в соответствующей строке или столбце, а сами числа – сколько слитных клеток содержит каждая из этих групп. Группы должны быть разделены, как минимум, одной пустой клеткой.

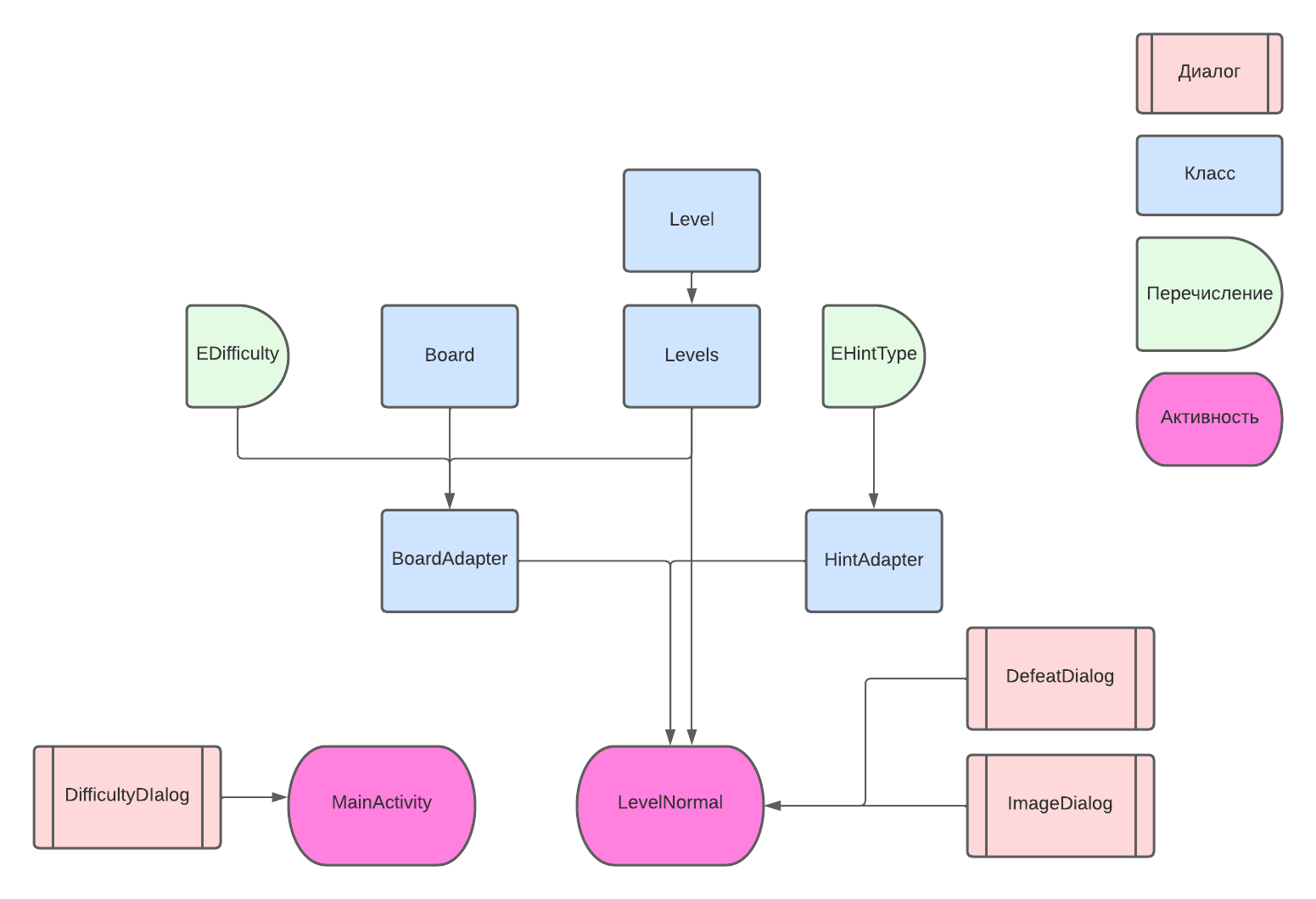


Для создания данного приложения были поставлены следующие задачи:

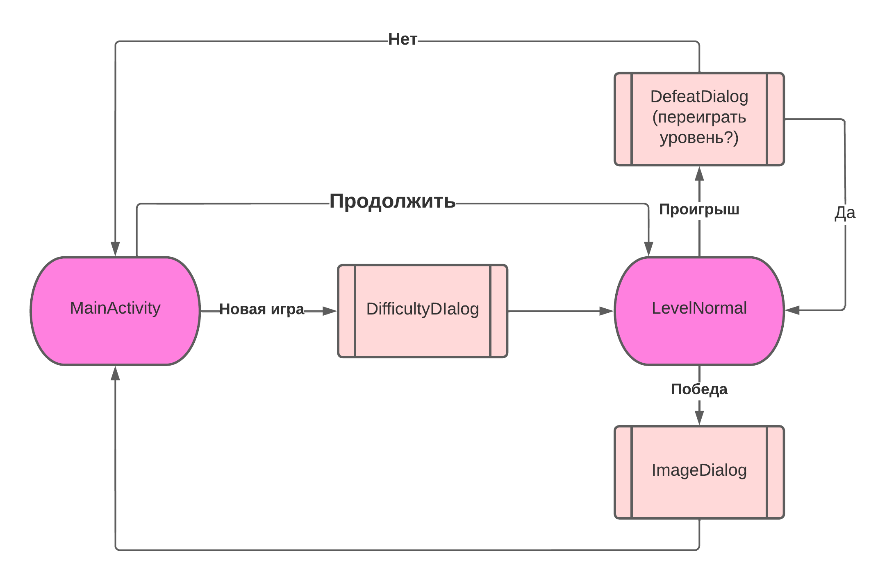
1. Создать главную страницу приложения. На ней должна быть возможность начать новую игру, а также продолжить предыдущую, если она не была завершена. Также, на странице должна присутствовать информация сколько игрок одержал побед всего и как долго длится его серия побед.
2. При запуске новой игры пользователь должен иметь выбор уровня сложности.
3. Страница с уровнем должна состоять из следующих компонент:
   1. Игровое поле.
   2. Подсказки слева и сверху от поля.
   3. Количество текущих жизней (от 0 до 3 включительно).
   4. Кнопка для возврата на главный экран.
   5. Переключатель между режимами закрашивания.
4. При выигрыше пользователю должна показываться картинка с итоговым изображением, а также кнопка для возврата на главное меню.
5. Если пользователь проиграл, то у него должна быть возможность, как переиграть уровень заново, так и вернуться в главное меню. Серия побед при этом должна обрываться.

# 3. Решение задачи.

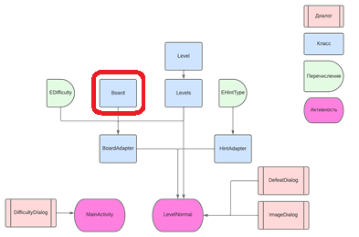
1. Структура приложения выглядит следующим образом:



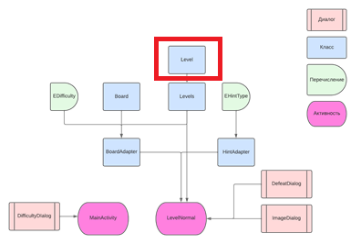
1. Ниже представлена схема переходов:



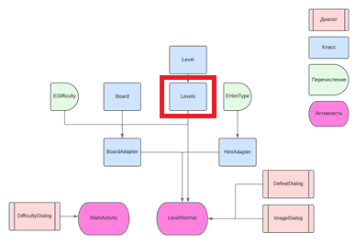
1. На первом этапе необходимо реализовать сердце программы – игральную доску. Для этого был создан класс Board. Доска состоит из двумерного массива типа boolean, где каждый элемент означает, закрашена эта клетка(true) или нет(false). Также это класс позволяет узнать сколько блоков и какой длинны содержится в строке/столбце (GetNumberoOfFilledTilesRow() / GetNumberoOfFilledTilesColumn() ).



1. Для того, чтобы хранить данные об уровне был создан класс Level, который содержит в себе информацию об игральной доске, а также id той картинки, которая должна быть показана пользователю, при победе.



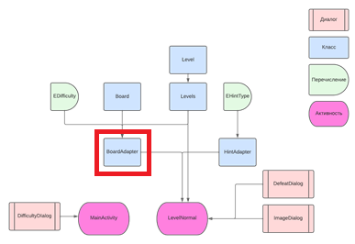
1. В качестве импровизированной базы данных выступает статический класс Levels, который хранит в себе все уровни доступные в игре. Также этот класс имеет возможность выбора уровня, в том числе и случайного. В игре содержится 4 уровня(3 обычных + 1 тестовый). Помимо прочего, класс ведёт учёт здоровья игрока и сбрасывает его при открытии нового уровня. Для реализации функционала продолжения уровня был создан двумерный массив, содержащий в себе информацию о действиях игрока, он состоит из трёх состояний: игрок не посетил клетку, игрок нажал на клетку и там оказался крестик (не закрашенная клетка), игрок нажал на клетку, и она оказалась закрашенной. Это необходимо для восстановления состояния уровня на момент его закрытия.



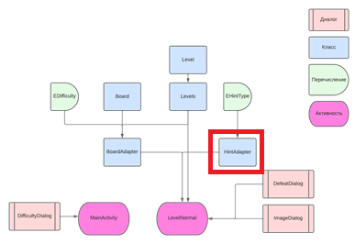
1. Для визуализации доски используется GridView. Чтобы воспользоваться GridView необходимо передать в него Adapter, который будет описывать построение и поведение таблицы. Для этого был создан класс BoardAdapter – наследник BaseAdapter. В качестве одной клетки выступает объект ImageView, которому добавлен обработчик нажатий (setOnTouchListener()). Он описывает поведение программы при нажатии на ту или иную клетку поля. При каждом нажатии происходит несколько проверок, а именно:
   1. Победил ли игрок. Если победил, то показывается диалог с победным экраном
   2. Не осталось ли больше закрашенных клеток в столбце/строке. Если не осталось, то все оставшиеся клетки в столбце/строке закрашиваются крестиками, т.к. в них точно не может быть элемента картинки.
   3. Совпадает ли режим закрашивания с тем, что содержится в клетке. Если не совпадает, то у игрока отнимается здоровье.
   4. Проигрыш игрока. Если после уменьшения здоровья на единицу у игрока осталось 0 здоровья, то необходимо показать игроку диалог с возможностью переиграть уровень.

На этом же этапе происходит отрисовка прогресса уровня, в случае, если было выбрано продолжить ранее начатый уровень.

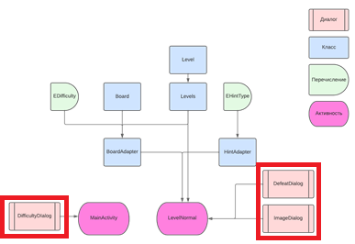
Если была выбрана лёгкая сложность, то пустые клетки показываются, с шансом 20%

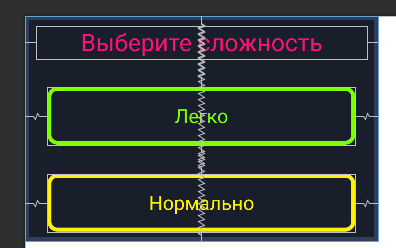
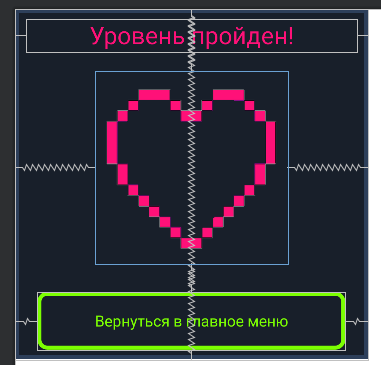
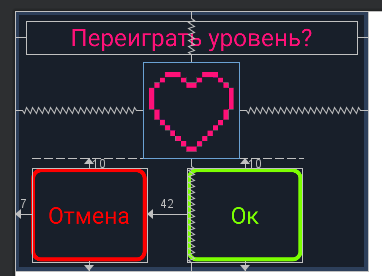


1. Чтобы подсказки были выровнены с основной доской, используются точно такие же объекты GridView. При этом, в горизонтальной подсказке настройки такие же, как и в основной таблице, а в вертикальной количество столбцов выставлено равное 1. Для связи данных с отображением был создан класс HintAdapter (наследник BaseAdapter), который выводит содержимое подсказок в соответствующие GridView. При создании доски происходит распарсивание её содержимого, с последующей передачей результатов в созданные адаптеры. В зависимости от количества цифр в подсказке происходит динамическое изменение размера текста, чтобы все цифры располагались чётко внутри подсказки.



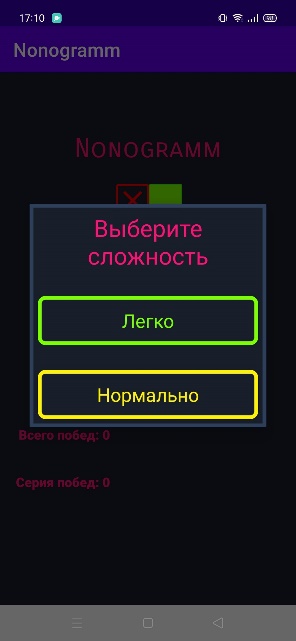
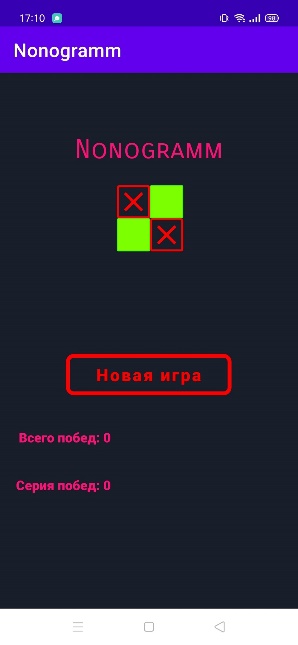
1. В качестве взаимодействия пользователя с игрой в некоторых моментах используются диалоговые окна, а именно: DifficultyDialog для выбора сложности, ImageDialog в случае победы игрока и DefeatDialog при его поражении. Создание диалога требует от программиста создания под каждый отдельный диалог отдельного layout. Поэтому были созданы difficulty\_dialog.xml, image\_dialog.xml, defeat\_dialog.xml.



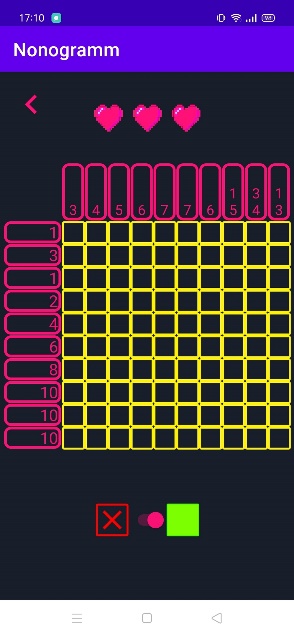
  

# 4. Руководство пользователя.

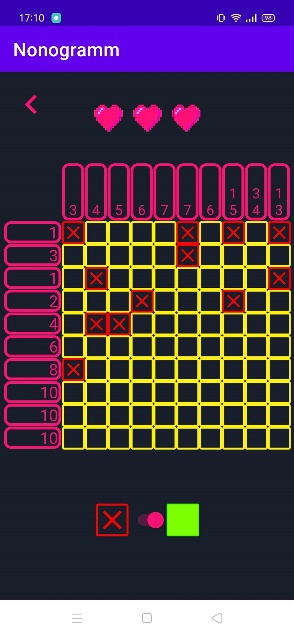
На главной странице приложения располагается информация об общем количестве побед игрока, а также о его текущей серии побед. Если игрок до этого не начинал новую игру, то на экране будет находится только кнопка для создания новой игры. При нажатии на кнопку откроется диалог с выбором сложность (Легко/Нормально).



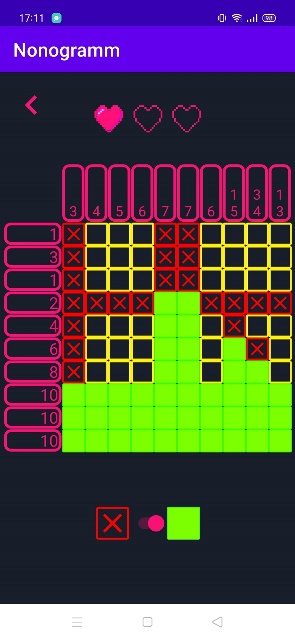
После выбора сложности игрока перекинет на игральное поле.



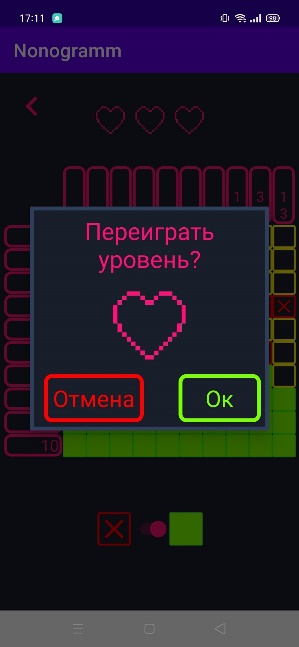
Если была выбрана лёгкая сложность, то часть пустых клеток будет раскрыта.



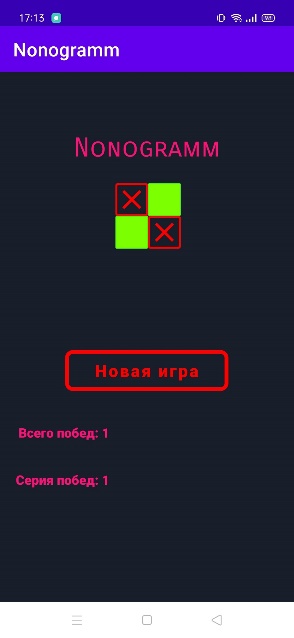
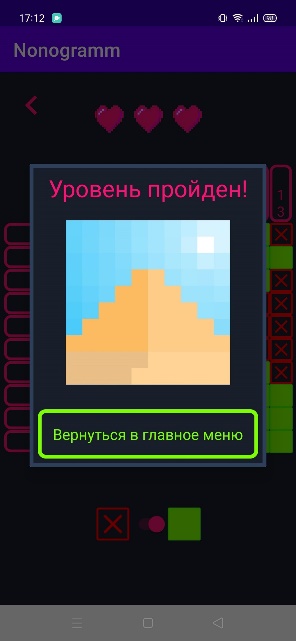
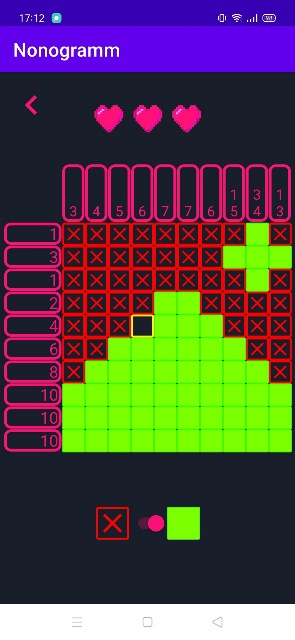
При нажатии на клетку она будет показывать, что за ней скрывается, в независимости от выбранного режима, однако если режим закрашивания не согласован с содержимым клетки, то у игрока отнимется одно из 3-х очков здоровья.



Если пользователь ошибётся 3 раза, то откроется диалог с возможностью переиграть уровень или вернуться в главное меню. В независимости от выбора серия побед обнуляется.



При успешном прохождении уровня пользователя ожидает диалог с тем, изображением, которое было закодировано, а также кнопкой для возврата на главный экран. При этом происходит увеличения общего количества побед и серии побед на 1.



# Приложение.

* MainActivity.java

package com.example.nonogramm;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 Button continueGame;  
 TextView victories, winStreak;  
  
 public void UpdateMenu() {  
 if (Levels.*currentLevel* != null)  
 continueGame.setVisibility(View.*VISIBLE*);  
 else  
 continueGame.setVisibility(View.*INVISIBLE*);  
 victories.setText("Всего побед: " + Levels.*victories*);  
 winStreak.setText("Серия побед: " + Levels.*winStreak*);  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 continueGame = findViewById(R.id.*btn\_continue*);  
 victories = findViewById(R.id.*victories*);  
 winStreak = findViewById(R.id.*winstreak*);  
  
 }  
  
 @Override  
 protected void onResume() {  
 UpdateMenu();  
 super.onResume();  
 }  
  
 public void NewGame(View v) {  
 DifficultyDialog dialog = new DifficultyDialog(this);  
 dialog.show(getFragmentManager(),  
 "MyCustomDialog");  
 }  
  
 public void ContinueGame(View v) {  
 Intent intent = new Intent(this, LevelNormal.class);  
 intent.putExtra("new game", false);  
 startActivity(intent);  
 }  
}

* activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/neonBack"  
 app:backgroundTint="@null"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/new\_game"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="340dp"  
 android:background="@drawable/button\_back"  
 android:fontFamily="sans-serif-black"  
 android:onClick="NewGame"  
 android:text="Новая игра"  
 android:textAllCaps="false"  
 android:textColor="@color/neonRed"  
 android:textSize="18sp"  
 app:backgroundTint="@null"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_continue"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:background="@drawable/button\_back"  
 android:fontFamily="sans-serif-black"  
 android:onClick="ContinueGame"  
 android:text="Продолжить"  
 android:textAllCaps="false"  
 android:textColor="@color/neonRed"  
 android:textSize="18sp"  
 app:backgroundTint="@null"  
 app:layout\_constraintBottom\_toTopOf="@+id/new\_game"  
 app:layout\_constraintEnd\_toEndOf="@+id/new\_game"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="@+id/new\_game"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView4"  
 app:layout\_constraintVertical\_bias="0.736" />  
  
 <TextView  
 android:id="@+id/victories"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:fontFamily="sans-serif-black"  
 android:gravity="center"  
 android:text="Всего побед:"  
 android:textColor="@color/neonPink"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.091"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/new\_game" />  
  
 <TextView  
 android:id="@+id/winstreak"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="36dp"  
 android:fontFamily="sans-serif-black"  
 android:gravity="center"  
 android:text="Серия побед:"  
 android:textColor="@color/neonPink"  
 app:layout\_constraintEnd\_toEndOf="@+id/victories"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="@+id/victories"  
 app:layout\_constraintTop\_toBottomOf="@+id/victories" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="sans-serif-smallcaps"  
 android:includeFontPadding="false"  
 android:text="Nonogramm"  
 android:textColor="@color/neonPink"  
 android:textSize="28sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.121" />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/fill"  
  
 app:layout\_constraintBottom\_toBottomOf="@+id/imageView3"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView3"  
 app:layout\_constraintTop\_toTopOf="@+id/imageView3" />  
  
 <ImageView  
  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/cross\_back"  
 app:layout\_constraintEnd\_toEndOf="@+id/imageView2"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView4"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"  
 app:srcCompat="@drawable/cross" />  
  
 <ImageView  
  
 android:id="@+id/imageView3"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="24dp"  
 android:background="@drawable/cross\_back"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.442"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"  
 app:srcCompat="@drawable/cross" />  
  
 <ImageView  
  
 android:id="@+id/imageView4"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@drawable/fill"  
 app:layout\_constraintEnd\_toStartOf="@+id/imageView2"  
 app:layout\_constraintStart\_toStartOf="@+id/imageView3"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView3" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

* Board.java

package com.example.nonogramm;  
  
import android.util.Log;  
  
import java.util.ArrayList;  
  
class Board {  
 public boolean[][] grid;  
 public Integer size;  
  
 Board(int size) {  
 this.size = size;  
 grid = new boolean[size][size];  
 }  
  
 Board(int size, boolean[][] booleans) {  
 this.size = size;  
 grid = booleans;  
 }  
  
 public static boolean[][] GenerateBoard(int size) {  
 boolean[][] grid = new boolean[size][size];  
 ;  
 for (int i = 0; i < size; i++)  
 for (int j = 0; j < size; j++) {  
 if (i % 2 == 0 &&j % 2 == 0)  
 grid[i][j] = true;  
 else  
 grid[i][j] = false;  
 }  
 return grid;  
 }  
  
 public ArrayList<Integer> GetNumberOfFilledTilesRow(int i) {  
 ArrayList<Integer> numbers = new ArrayList<>();  
 if (i < size) {  
 int cur = 0;  
 for (int l = 0; l < size; l++) {  
 while ( l < size &&grid[i][l] == true ) {  
 cur++;  
 l++;  
 }  
 if (cur > 0) {  
 numbers.add(cur);  
 }  
 cur = 0;  
 }  
 }  
 return numbers;  
 }  
  
 public ArrayList<Integer> GetNumberOfFilledTilesColumn(int j) {  
 ArrayList<Integer> numbers = new ArrayList<>();  
 if (j < size) {  
 int cur = 0;  
 for (int l = 0; l < size; l++) {  
 while (l < size &&grid[l][j] == true ) {  
 cur++;  
 l++;  
 }  
 if (cur > 0) {  
 numbers.add(cur);  
 }  
 cur = 0;  
 }  
 }  
 if (numbers.isEmpty())  
 numbers.add(0);  
 return numbers;  
 }  
  
}

* Level.java

package com.example.nonogramm;  
  
public class Level {  
  
  
 public String[][] grid;  
 public int size = 10;  
 public int background\_id;  
  
 public Level(int size, int id, String[][] grid) {  
 this.size = size;  
 this.grid = grid;  
 background\_id = id;  
 }  
  
 public boolean[][] ParseGrid() {  
 boolean b\_grid[][] = new boolean[size][size];  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 if (grid[i][j] == Levels.*CROSS*)  
 b\_grid[i][j] = false;  
 else  
 b\_grid[i][j] = true;  
 }  
 }  
 return b\_grid;  
 }  
  
}

* Levels.java

package com.example.nonogramm;  
  
import android.app.Activity;  
import android.util.Log;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import java.util.Random;  
  
enum EDifficulty {  
 *NORMAL*,  
 *EASY*}  
  
public class Levels {  
 public static int *winStreak* = 0;  
 public static int *victories* = 0;  
  
 public static String *CROSS* = "X";  
 public static String *FILL* = "O";  
 public static String *NOT\_CLICKED* = ".";  
 public static int *Lives* = 3;  
 public static boolean *mode* = true;  
 public static ImageView[] *livesIcons* = new ImageView[3];  
 public static Activity *currentActivity*;  
  
 public static TextView *victory*;  
  
 public static Level[] *levelsNormal* = new Level[]{  
 new Level(10, R.drawable.*bungalo*, //Level 1 - Bungalo  
 new String[][]{  
 // 1 2 3 4 5 6 7 8 9 10  
 {"X", "X", "X", "X", "O", "O", "X", "X", "X", "X"}, // 1  
 {"X", "X", "X", "O", "O", "O", "O", "X", "X", "X"}, // 2  
 {"X", "X", "O", "O", "O", "O", "O", "O", "X", "X"}, // 3  
 {"X", "O", "O", "O", "O", "O", "O", "O", "O", "X"}, // 4  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"}, // 5  
 {"X", "X", "O", "O", "O", "X", "X", "O", "X", "X"}, // 6  
 {"X", "X", "O", "O", "O", "X", "X", "O", "X", "X"}, // 7  
 {"X", "X", "O", "O", "O", "X", "X", "O", "X", "X"}, // 8  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"}, // 9  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"} // 10  
 }),  
 new Level(10, R.drawable.pyramid, //Level 2 - Pyramid  
 new String[][]{  
 // 1 2 3 4 5 6 7 8 9 10  
 {"X", "X", "X", "X", "X", "X", "X", "X", "O", "X"}, // 1  
 {"X", "X", "X", "X", "X", "X", "X", "O", "O", "O"}, // 2  
 {"X", "X", "X", "X", "X", "X", "X", "X", "O", "X"}, // 3  
 {"X", "X", "X", "X", "O", "O", "X", "X", "X", "X"}, // 4  
 {"X", "X", "X", "O", "O", "O", "O", "X", "X", "X"}, // 5  
 {"X", "X", "O", "O", "O", "O", "O", "O", "X", "X"}, // 6  
 {"X", "O", "O", "O", "O", "O", "O", "O", "O", "X"}, // 7  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"}, // 8  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"}, // 9  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"} // 10  
 }),  
 new Level(10, R.drawable.dawn, //Level 3 - Dawn  
 new String[][]{  
 // 1 2 3 4 5 6 7 8 9 10  
 {"O", "X", "O", "X", "O", "O", "X", "O", "X", "O"}, // 1  
 {"X", "O", "X", "O", "X", "O", "O", "X", "O", "X"}, // 2  
 {"X", "X", "O", "X", "O", "X", "O", "O", "X", "O"}, // 3  
 {"X", "X", "X", "O", "O", "O", "O", "X", "O", "X"}, // 4  
 {"X", "X", "X", "X", "X", "P", "O", "O", "X", "X"}, // 9  
 {"X", "X", "X", "X", "X", "X", "O", "X", "X", "X"}, // 9  
 {"X", "X", "X", "X", "X", "X", "O", "X", "X", "X"}, // 9  
 {"X", "X", "X", "X", "X", "X", "O", "X", "X", "X"}, // 9  
 {"X", "X", "X", "X", "X", "X", "O", "X", "X", "X"}, // 9  
 {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O"} // 10  
 }),  
 new Level(10, R.drawable.heart\_t, //Level 4 - test  
 new String[][]{  
 // 1 2 3 4 5 6 7 8 9 10  
 {"O", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 1  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 2  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 3  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 4  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 5  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 6  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 7  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 8  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 9  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"} // 10  
 })  
 };  
 public static Level currentLevel = null;  
 public static Level currentLevelProgress = null;  
 public static Level notClickedLevel = new Level(10, R.drawable.ic\_launcher\_background,  
 new String[][]{  
 // 1 2 3 4 5 6 7 8 9 10  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 1  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 2  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 3  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 4  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 5  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 6  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 7  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 8  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, // 9  
 {".", ".", ".", ".", ".", ".", ".", ".", ".", "."} // 10  
 });  
 public static int currentLvl = -1;  
 private static int sizeNormal = 10;  
 private static Level emptyLevel = new Level(10, R.drawable.ic\_launcher\_background,  
 new String[][]{  
 // 1 2 3 4 5 6 7 8 9 10  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 1  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 2  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 3  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 4  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 5  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 6  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 7  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 8  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}, // 9  
 {"X", "X", "X", "X", "X", "X", "X", "X", "X", "X"} // 10  
 });  
  
 public static void InitLives(ImageView life1, ImageView life2, ImageView life3) {  
 livesIcons[0] = life1;  
 livesIcons[1] = life2;  
 livesIcons[2] = life3;  
 }  
  
 public static boolean[][] ChooseLevel(int i, EDifficulty difficulty) {  
 Lives = 3;  
 mode = true;  
 currentLevelProgress = new Level(10, R.drawable.ic\_launcher\_background, notClickedLevel.grid);  
 for (int j = 0; j < 10; j++) {  
 for (int k = 0; k < 10; k++) {  
 currentLevelProgress.grid[j][k] = ".";  
 }  
 }  
 if (i < levelsNormal.length) {  
 currentLevel = new Level(10,levelsNormal[i].background\_id, levelsNormal[i].grid);  
 *currentLvl* = i;  
 return *levelsNormal*[i].ParseGrid();  
 } else {  
 *currentLevel* = new Level(10, R.drawable.*heart\_t*, *emptyLevel*.grid);  
 return *emptyLevel*.ParseGrid();  
 }  
 }  
  
 public static boolean[][] ChooseRandomLevel(EDifficulty difficulty) {  
 Random random = new Random();  
 return *ChooseLevel*(random.nextInt(*levelsNormal*.length), difficulty);  
 }  
  
 public static boolean[][] GetCurrentLevelProgress() {  
 return *currentLevelProgress*.ParseGrid();  
 }  
  
 public static boolean[][] getCurrentLevel() {  
 return *currentLevel*.ParseGrid();  
 }  
  
 public static void UpdateLives() {  
 Log.*d*("Lives", String.*valueOf*(Levels.*Lives*));  
 switch (Levels.*Lives*) {  
 case 3:  
 *livesIcons*[0].setImageResource(R.drawable.*heart\_t*);  
 *livesIcons*[1].setImageResource(R.drawable.*heart\_t*);  
 *livesIcons*[2].setImageResource(R.drawable.*heart\_t*);  
 break;  
 case 2:  
 *livesIcons*[0].setImageResource(R.drawable.*heart\_t*);  
 *livesIcons*[1].setImageResource(R.drawable.*heart\_t*);  
 *livesIcons*[2].setImageResource(R.drawable.*heart\_empy\_t*);  
 break;  
 case 1: {  
 *livesIcons*[0].setImageResource(R.drawable.*heart\_t*);  
 *livesIcons*[1].setImageResource(R.drawable.*heart\_empy\_t*);  
 *livesIcons*[2].setImageResource(R.drawable.*heart\_empy\_t*);  
 break;  
 }  
 case 0: {  
 *livesIcons*[0].setImageResource(R.drawable.*heart\_empy\_t*);  
 *livesIcons*[1].setImageResource(R.drawable.*heart\_empy\_t*);  
 *livesIcons*[2].setImageResource(R.drawable.*heart\_empy\_t*);  
 break;  
 }  
 default:  
 break;  
 }  
 }  
};

* BoardAdapter.java

package com.example.nonogramm;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.util.Log;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.Random;  
  
  
public class BoardAdapter extends BaseAdapter {  
 public Integer size;  
 private Context mContext;  
 private Board board;  
 private Board playerBoard;  
 private int difficulty;  
 private View[][] buttons;  
  
 public BoardAdapter(Context c, int size, Board board, int difficulty) {  
 mContext = c;  
 this.size = size;  
 this.board = board;  
 this.playerBoard = new Board(10);  
 this.difficulty = difficulty;  
 buttons = new View[size][size];  
 }  
  
 public boolean CheckWin() {  
 boolean win = true;  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 if (playerBoard.grid[i][j] != board.grid[i][j])  
 win = false;  
 }  
 }  
 return win;  
 }  
  
 public boolean CheckRow(int row) {  
 boolean full = true;  
 for (int j = 0; j < size; j++) {  
 if ((board.grid[row][j] == true) && (playerBoard.grid[row][j] == false))  
 full = false;  
 }  
 return full;  
 }  
  
 public boolean CheckColumn(int col) {  
 boolean full = true;  
 for (int j = 0; j < size; j++) {  
 if ((board.grid[j][col] == true) && (playerBoard.grid[j][col] == false))  
 full = false;  
 }  
 return full;  
 }  
  
 private void FillRow(int row) {  
 for (int j = 0; j < size; j++) {  
 if (Levels.currentLevel.grid[row][j] == "X") {  
 Levels.currentLevelProgress.grid[row][j] = "X";  
 ((ImageView) buttons[row][j]).setImageResource(R.drawable.cross);  
 buttons[row][j].setBackgroundResource(R.drawable.cross\_back);  
 buttons[row][j].setEnabled(false);  
 }  
 }  
 }  
  
 private void FillColumn(int col) {  
 for (int j = 0; j < size; j++) {  
 if (Levels.currentLevel.grid[j][col] == "X") {  
 Levels.currentLevelProgress.grid[j][col] = "X";  
 ((ImageView) buttons[j][col]).setImageResource(R.drawable.cross);  
 buttons[j][col].setBackgroundResource(R.drawable.cross\_back);  
 buttons[j][col].setEnabled(false);  
 }  
 }  
 }  
  
 public boolean PlayerAction(int i) {  
 playerBoard.grid[i / size][i % size] = true;  
 return CheckWin();  
 }  
  
 public Board getBoard() {  
 return board;  
 }  
  
 @Override  
 public int getCount() {  
 return size \* size;  
 }  
  
 @Override  
 public Boolean getItem(int i) {  
 return board.grid[i / size][i % size];  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return i;  
 }  
  
 @Override  
 public View getView(int i, View view, ViewGroup viewGroup) {  
  
 ImageView button = new ImageView(mContext);  
  
 button.setPadding(1, 1, 1, 1);  
  
 button.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 @SuppressLint({"ResourceType"})  
 public boolean onTouch(View view, MotionEvent event) {  
 Log.*d*("Action", String.*valueOf*(event.getAction()));  
 switch (event.getAction()) {  
 case MotionEvent.*ACTION\_DOWN*: {  
 if (getItem(i) == true) {  
 view.setBackgroundResource(R.drawable.*fill*);  
  
 Levels.*currentLevelProgress*.grid[i / size][i % size] = "O";  
  
 boolean win = PlayerAction(i);  
 boolean row\_filled = CheckRow(i / size);  
 Log.*d*("Filled row", String.*valueOf*(row\_filled));  
 if (row\_filled)  
 FillRow(i / size);  
  
 boolean col\_filled = CheckColumn(i % size);  
 Log.*d*("Filled col", String.*valueOf*(col\_filled));  
 if (col\_filled)  
 FillColumn(i % size);  
 if (win) {  
  
 Levels.*winStreak*++;  
 Levels.*victories*++;  
 ImageDialog dialog = new ImageDialog(mContext);  
 dialog.show(((AppCompatActivity) mContext).getFragmentManager(),  
 "MyCustomDialog");  
  
 }  
  
  
 } else {  
 view.setBackgroundResource(R.drawable.cross\_back);  
 ((ImageView) view).setImageResource(R.drawable.cross);  
 Levels.currentLevelProgress.grid[i / size][i % size] = "X";  
 }  
 if (Levels.mode != getItem(i)) {  
 Levels.Lives--;  
 if (Levels.Lives <= 0) {  
 Levels.winStreak = 0;  
 DefeatDialog dialog = new DefeatDialog(mContext);  
 dialog.show(((AppCompatActivity) mContext).getFragmentManager(),  
 "MyCustomDialog");  
 }  
 Levels.UpdateLives();  
 }  
 button.setEnabled(false);  
 break;  
 }  
  
 }  
 return true;  
 }  
 });  
  
 if (Levels.currentLevelProgress.grid[i / size][i % size] == "O") {  
 button.setBackgroundResource(R.drawable.fill);  
 PlayerAction(i);  
 button.setEnabled(false);  
 } else if (Levels.currentLevelProgress.grid[i / size][i % size] == "X") {  
 button.setBackgroundResource(R.drawable.cross\_back);  
 button.setImageResource(R.drawable.cross);  
 button.setEnabled(false);  
 } else {  
 button.setBackgroundResource(R.drawable.nothing);  
 }  
  
 if (difficulty == EDifficulty.EASY.ordinal() && Levels.currentLevel.grid[i / size][i % size] == "X") {  
 Random random = new Random();  
 if (random.nextFloat() < 0.2) {  
 button.setBackgroundResource(R.drawable.cross\_back);  
 button.setImageResource(R.drawable.cross);  
 button.setEnabled(false);  
 Levels.currentLevelProgress.grid[i / size][i % size] = "X";  
 }  
 }  
 buttons[i / size][i % size] = button;  
 return button;  
 }  
}

* HintAdapter.java

package com.example.nonogramm;  
  
import android.content.Context;  
import android.graphics.Color;  
import android.view.Gravity;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
  
enum EHintType {  
 UPPER,  
 LEFT  
}  
  
public class HintAdapter extends BaseAdapter {  
 public ArrayList<String> hints;  
  
 private Context mContext;  
  
 private EHintType type;  
  
 public HintAdapter(Context c, ArrayList<String> hints, EHintType type) {  
 mContext = c;  
 this.hints = hints;  
 this.type = type;  
 }  
  
  
 @Override  
 public int getCount() {  
 return hints.size();  
 }  
  
 @Override  
 public String getItem(int i) {  
 return hints.get(i);  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return i;  
 }  
  
 @Override  
 public View getView(int i, View view, ViewGroup viewGroup) {  
 TextView hint;  
 hint = new TextView(mContext);  
  
 switch (type) {  
 case UPPER:  
 hint.setBackgroundResource(R.drawable.upper\_hint);  
 hint.setHeight(10);  
 hint.setGravity(Gravity.CENTER\_HORIZONTAL | Gravity.BOTTOM);  
 hint.setPadding(1, 1, 1, 5);  
 switch (hints.get(i).length()) {  
 case 5:  
 hint.setTextSize(12);  
 break;  
 case 7:  
 hint.setTextSize(11);  
 case 9:  
 hint.setTextSize(10);  
 break;  
 default:  
 hint.setTextSize(15);  
 break;  
 }  
 break;  
 case LEFT:  
 hint.setBackgroundResource(R.drawable.left\_hint);  
 hint.setWidth(100);  
 hint.setPadding(1, 1, 12, 1);  
 hint.setGravity(Gravity.END | Gravity.CENTER\_VERTICAL);  
 hint.setMaxLines(1);  
 switch (hints.get(i).length()) {  
 case 5:  
 hint.setTextSize(16);  
 break;  
 case 7:  
 hint.setTextSize(15);  
 case 9:  
 hint.setTextSize(14);  
 break;  
 default:  
 hint.setTextSize(17);  
 break;  
 }  
 break;  
 default:  
 break;  
 }  
  
 hint.setText(  
 // hints.get(i).length()+" "+  
 hints.get(i));  
  
 hint.setTextColor( Color.parseColor("#FF1178"));  
  
 return hint;  
 }  
}

* LevelNormal.java

package com.example.nonogramm;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.GridView;  
import android.widget.Switch;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.ArrayList;  
  
public class LevelNormal extends AppCompatActivity implements  
 AdapterView.OnItemSelectedListener {  
  
  
 BoardAdapter boardAdapter;  
 HintAdapter hintUpper;  
 HintAdapter hintLeft;  
 GridView gridView;  
 GridView upper;  
 GridView left;  
 Board board;  
 Switch aSwitch;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_level\_normal);  
  
  
 Intent intent = getIntent();  
 boolean newGame = intent.getBooleanExtra("new game", true);  
 boolean sameLevel = intent.getBooleanExtra("same level", false);  
 int difficulty = intent.getIntExtra("Difficulty", EDifficulty.NORMAL.ordinal());  
  
 Log.d("New?", String.valueOf(newGame));  
  
 if (Levels.currentActivity != null) {  
 Levels.currentActivity.finish();  
 Levels.currentActivity = null;  
 }  
  
 gridView = findViewById(R.id.grid);  
 upper = findViewById(R.id.upper);  
 left = findViewById(R.id.left);  
  
 Levels.InitLives(findViewById(R.id.life1), findViewById(R.id.life2), findViewById(R.id.life3));  
  
 aSwitch = findViewById(R.id.mode);  
 aSwitch.setChecked(true);  
  
  
 if (newGame) {  
 if (sameLevel)  
 board = new Board(10, Levels.ChooseLevel(Levels.currentLvl, EDifficulty.NORMAL));  
 else  
 board = new Board(10, Levels.ChooseRandomLevel(EDifficulty.NORMAL));  
 } else {  
 board = new Board(10, Levels.getCurrentLevel());  
 Levels.mode = true;  
 }  
  
 Levels.UpdateLives();  
  
 boardAdapter = new BoardAdapter(this, 10, board, difficulty);  
 gridView.setAdapter(boardAdapter);  
  
  
 hintUpper = new HintAdapter(this, ParseUpperHint(), EHintType.UPPER);  
 upper.setAdapter(hintUpper);  
  
 hintLeft = new HintAdapter(this, ParseLeftHint(), EHintType.LEFT);  
 left.setAdapter(hintLeft);  
  
  
 Levels.*currentActivity* = this;  
 }  
  
 private String ParseRow(int row) {  
 String res = "";  
 ArrayList<Integer> numbers = board.GetNumberOfFilledTilesRow(row);  
  
 for (int i = 0; i < numbers.size(); i++) {  
 if (i < numbers.size() - 1)  
 res += numbers.get(i) + " ";  
 else  
 res += numbers.get(i);  
 }  
 if (numbers.isEmpty())  
 res += "0";  
  
 return res;  
 }  
  
 private String ParseColumn(int col) {  
 String res = "";  
 ArrayList<Integer> numbers = board.GetNumberOfFilledTilesColumn(col);  
 for (int i = 0; i < numbers.size(); i++) {  
  
 if (i < numbers.size() - 1)  
 res += numbers.get(i).toString() + "\n";  
 else  
 res += numbers.get(i).toString();  
 }  
 if (numbers.isEmpty())  
 res += "0";  
  
 return res;  
 }  
  
 public void onSwitcherPressed(View v) {  
 if (Levels.*mode* == false) {  
 Levels.*mode* = true;  
 } else {  
 Levels.*mode* = false;  
 }  
 }  
  
 private ArrayList<String> ParseUpperHint() {  
 ArrayList<String> upperHints = new ArrayList<>();  
 for (int i = 0; i < 10; i++) {  
 String col = ParseColumn(i);  
 upperHints.add(col);  
 }  
  
 return upperHints;  
 }  
  
 private ArrayList<String> ParseLeftHint() {  
 ArrayList<String> leftHints = new ArrayList<>();  
 for (int i = 0; i < 10; i++) {  
 String row = ParseRow(i);  
 leftHints.add(row);  
 }  
  
 return leftHints;  
 }  
  
  
 public void BackToMenu(View v) {  
  
  
 startActivity(new Intent(this, MainActivity.class));  
 this.finish();  
 }  
  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View v, int position,  
 long id) {  
  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
  
 }  
}

* activity\_level\_normal.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/neonBack"  
 app:backgroundTint="@null"  
 tools:context=".LevelNormal">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="1dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:layout\_editor\_absoluteY="132dp">  
  
  
 <ImageView  
 android:id="@+id/life1"  
  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
  
 app:layout\_constraintBottom\_toBottomOf="@+id/life2"  
 app:layout\_constraintEnd\_toStartOf="@+id/life2"  
 app:layout\_constraintHorizontal\_bias="0.937"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/life2"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <ImageView  
 android:id="@+id/life2"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_centerHorizontal="true"  
 app:layout\_constraintBottom\_toTopOf="@+id/upper"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageView  
 android:id="@+id/life3"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/life2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.062"  
 app:layout\_constraintStart\_toEndOf="@+id/life2"  
 app:layout\_constraintTop\_toTopOf="@+id/life2"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <Switch  
 android:id="@+id/mode"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBottom="@id/grid"  
 android:layout\_centerHorizontal="true"  
 android:checked="true"  
 android:minHeight="48dp"  
 android:onClick="onSwitcherPressed"  
 android:thumbTint="@color/neonPink"  
 android:trackTint="@color/neonPink"  
  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/grid"  
 app:layout\_constraintVertical\_bias="0.455" />  
  
 <GridView  
 android:id="@+id/upper"  
 android:layout\_width="280dp"  
 android:layout\_height="70dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginBottom="1dp"  
  
 android:horizontalSpacing="1dp"  
 android:numColumns="10"  
 android:outlineProvider="bounds"  
 android:scaleX="1.0"  
 android:scaleY="1.0"  
 android:stretchMode="columnWidth"  
 android:verticalSpacing="1dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/grid"  
 app:layout\_constraintEnd\_toEndOf="@+id/grid"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="@+id/grid">  
  
 </GridView>  
  
 <GridView  
 android:id="@+id/left"  
 android:layout\_width="70dp"  
 android:layout\_height="280dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginEnd="1dp"  
  
 android:horizontalSpacing="1dp"  
 android:numColumns="1"  
 android:outlineProvider="bounds"  
 android:scaleX="1.0"  
 android:scaleY="1.0"  
  
 android:verticalSpacing="1dp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/grid"  
 app:layout\_constraintEnd\_toStartOf="@+id/grid"  
 app:layout\_constraintTop\_toTopOf="@+id/grid"  
 app:layout\_constraintVertical\_bias="0.0"></GridView>  
  
 <GridView  
 android:id="@+id/grid"  
 android:layout\_width="280dp"  
 android:layout\_height="280dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginEnd="4dp"  
  
 android:horizontalSpacing="0dp"  
 android:numColumns="10"  
 android:outlineProvider="bounds"  
 android:scaleX="1.0"  
 android:scaleY="1.0"  
 android:stretchMode="columnWidth"  
 android:verticalSpacing="0dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.498">  
  
 </GridView>  
  
 <ImageView  
 android:id="@+id/backToMenu"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:minWidth="48dp"  
 android:minHeight="48dp"  
 android:onClick="BackToMenu"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.043"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/back\_to\_menu" />  
  
 <ImageView  
  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/cross\_back"  
 android:foreground="@drawable/cross"  
 android:minWidth="40dp"  
 android:minHeight="40dp"  
  
 app:layout\_constraintBottom\_toBottomOf="@+id/mode"  
 app:layout\_constraintEnd\_toStartOf="@+id/mode"  
 app:layout\_constraintTop\_toTopOf="@+id/mode" />  
  
 <ImageView  
  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/fill"  
 android:minWidth="40dp"  
 android:minHeight="40dp"  
  
 app:layout\_constraintBottom\_toBottomOf="@+id/mode"  
 app:layout\_constraintStart\_toEndOf="@+id/mode"  
 app:layout\_constraintTop\_toTopOf="@+id/mode" />  
  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

* DifficultyDialog.java

package com.example.nonogramm;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
  
public class DifficultyDialog  
 extends android.app.DialogFragment {  
  
 private static final String TAG = "DefeatDialog";  
 private TextView mNormal, mEasy;  
 private Context mContext;  
  
 public DifficultyDialog() {  
 }  
  
 @SuppressLint("ValidFragment")  
 public DifficultyDialog(Context context) {  
 mContext = context;  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater,  
 ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(  
 R.layout.difficulty\_dialog, container, false);  
  
  
 mNormal = view.findViewById(R.id.diff\_normal);  
 mEasy = view.findViewById(R.id.diff\_easy);  
  
 mNormal.setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 getDialog().dismiss();  
  
  
 Intent intent = new Intent(mContext, LevelNormal.class);  
 intent.putExtra("new game", true);  
 intent.putExtra("Difficulty", EDifficulty.NORMAL.ordinal());  
  
 mContext.startActivity(intent);  
 }  
 });  
  
 mEasy.setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 getDialog().dismiss();  
 Intent intent = new Intent(mContext, LevelNormal.class);  
 intent.putExtra("new game", true);  
 intent.putExtra("Difficulty", EDifficulty.EASY.ordinal());  
  
 mContext.startActivity(intent);  
  
 }  
 });  
  
 return view;  
 }  
  
  
}

* difficulty\_dialog.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <RelativeLayout  
  
 android:layout\_width="367dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/dialog\_back"  
  
 android:padding="10dp">  
  
 <TextView  
 android:id="@+id/heading"  
 android:layout\_width="344dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:gravity="center"  
 android:text="Выберите сложность"  
 android:textColor="@color/neonPink"  
 android:textSize="25sp" />  
  
  
 <TextView  
 android:id="@+id/diff\_easy"  
 android:layout\_width="320dp"  
 android:layout\_height="60dp"  
 android:layout\_below="@id/heading"  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="30dp"  
 android:background="@drawable/button\_back"  
 android:backgroundTint="@color/neonGreen"  
 android:gravity="center"  
 android:text="Легко"  
 android:textColor="@color/neonGreen"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/diff\_normal"  
 android:layout\_width="320dp"  
 android:layout\_height="60dp"  
 android:layout\_below="@id/diff\_easy"  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="30dp"  
 android:background="@drawable/button\_back"  
 android:backgroundTint="@color/neonYellow"  
 android:gravity="center"  
 android:text="Нормально"  
 android:textColor="@color/neonYellow"  
 android:textSize="20sp" />  
  
  
 </RelativeLayout>  
</RelativeLayout>

* ImageDialog.java

package com.example.nonogramm;  
  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
public class ImageDialog  
 extends android.app.DialogFragment {  
  
 private static final String TAG = "DialogFragment";  
 private TextView mActionOk;  
 private Context mContext;  
 private ImageView victoryImage;  
  
 public ImageDialog() {  
 }  
  
 @SuppressLint("ValidFragment")  
 public ImageDialog(Context context) {  
 mContext = context;  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater,  
 ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(  
 R.layout.image\_dialog, container, false);  
  
  
 mActionOk = view.findViewById(R.id.action\_ok);  
 victoryImage = view.findViewById(R.id.victoryImage);  
 int id = Levels.currentLevel.background\_id;  
 victoryImage.setImageResource(id);  
  
 Levels.currentLevel = null;  
 mActionOk.setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Log.d(TAG, "onClick: capturing input");  
 getDialog().dismiss();  
 Intent intent = new Intent(mContext, MainActivity.class);  
 mContext.startActivity(intent);  
 Levels.currentLevel = null;  
 Levels.currentActivity.finish();  
  
 }  
 });  
  
 return view;  
 }  
  
  
}

* image\_dialog.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <RelativeLayout  
  
 android:layout\_width="367dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/dialog\_back"  
  
 android:padding="10dp">  
  
 <TextView  
 android:id="@+id/heading"  
 android:layout\_width="344dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:gravity="center"  
 android:text="Уровень пройден!"  
 android:textColor="@color/neonPink"  
 android:textSize="25sp" />  
  
  
 <ImageView  
 android:id="@+id/victoryImage"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:layout\_below="@+id/heading"  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="20dp"  
 app:srcCompat="@drawable/heart\_empy\_t" />  
  
 <TextView  
 android:id="@+id/action\_ok"  
 android:layout\_width="320dp"  
 android:layout\_height="60dp"  
 android:layout\_below="@id/victoryImage"  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="30dp"  
 android:background="@drawable/button\_back"  
 android:backgroundTint="@color/neonGreen"  
  
 android:gravity="center"  
 android:text="Вернуться в главное меню"  
 android:textColor="@color/neonGreen"  
 android:textSize="16sp" />  
  
 </RelativeLayout>  
</RelativeLayout>

* DefeatDialog.java

package com.example.nonogramm;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
  
public class DefeatDialog  
 extends android.app.DialogFragment {  
  
 private static final String TAG = "DefeatDialog";  
 private TextView mActionOk, mActionCancel;  
 private Context mContext;  
  
 public DefeatDialog() {  
 }  
  
 @SuppressLint("ValidFragment")  
 public DefeatDialog(Context context) {  
 mContext = context;  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater,  
 ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(  
 R.layout.*defeat\_dialog*, container, false);  
  
  
 mActionOk = view.findViewById(R.id.*action\_ok\_def*);  
 mActionCancel = view.findViewById(R.id.*action\_cancel\_def*);  
  
  
 mActionOk.setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 getDialog().dismiss();  
  
 Intent intent = new Intent(mContext, LevelNormal.class);  
 intent.putExtra("new game", true);  
 intent.putExtra("same level", true);  
  
 mContext.startActivity(intent);  
  
 }  
 });  
  
 mActionCancel.setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 getDialog().dismiss();  
 Intent intent = new Intent(mContext, MainActivity.class);  
 Levels.*currentLevel* = null;  
 Levels.*currentActivity*.finish();  
 mContext.startActivity(intent);  
  
 }  
 });  
  
 return view;  
 }  
  
  
}

* defeat\_dialog.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_height="match\_parent">  
  
 <RelativeLayout  
  
 android:layout\_width="367dp"  
 android:layout\_height="270dp"  
 android:background="@drawable/dialog\_back"  
  
 android:padding="10dp">  
  
 <TextView  
 android:id="@+id/heading"  
 android:layout\_width="344dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:gravity="center"  
 android:text="Переиграть уровень?"  
 android:textColor="@color/neonPink"  
 android:textSize="25sp" />  
  
 <TextView  
 android:id="@+id/action\_cancel\_def"  
 android:layout\_width="120dp"  
 android:layout\_height="39dp"  
 android:layout\_below="@id/defeat\_icon"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginLeft="7dp"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/button\_back"  
 android:backgroundTint="@color/neonRed"  
 android:gravity="center"  
 android:text="Отмена"  
 android:textColor="@color/neonRed"  
 android:textSize="24sp" />  
  
 <ImageView  
 android:id="@+id/defeat\_icon"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_below="@+id/heading"  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="10dp"  
 app:srcCompat="@drawable/heart\_empy\_t" />  
  
 <TextView  
 android:id="@+id/action\_ok\_def"  
 android:layout\_width="120dp"  
 android:layout\_height="39dp"  
 android:layout\_below="@id/defeat\_icon"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginLeft="42dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_toRightOf="@+id/action\_cancel\_def"  
 android:background="@drawable/button\_back"  
 android:backgroundTint="@color/neonGreen"  
 android:gravity="center"  
 android:text="Ок"  
 android:textColor="@color/neonGreen"  
 android:textSize="24sp" />  
  
 </RelativeLayout>  
</RelativeLayout>