**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

**Институт прикладной математики, физики и информатики**

кафедра физики и прикладной математики

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**

**ПО ДИСЦИПЛИНЕ «Разработка мобильных приложений»**

**на тему «Разработка Android-приложения «Лабиринты»»**

Студентов Владимировой Марины Алексеевны,

Петрова Максима Михайловича

Направления подготовки бакалавров 02.03.02 «Фундаментальная информатика и информационные технологии»

Группы ИТу-119

*Руководитель*:

доцент кафедры ФиПМ к.ф.-м. н Лексин А.Ю.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Владимир 2021

# Аннотация

Целью курсовой работы является разработка Android-приложения «Лабиринты». В процессе работы в соответствии с заданием проведен анализ предметной области – изучены требования к Android -приложению, определен его функционал, обоснован выбор программных и технических средств, спроектирована база данных приложения. В результате курсовой работы было разработано Android-приложение «Лабиринты», которое выполняет все функции, определенные в требованиях.

Пояснительная записка к курсовой работе содержит 70 с., 11 рис., 9 источников, 2 приложения.

# Summary

The purpose of the course work is to develop an Android application "Maze". During the work according to the task, an analysis of the subject area was carried out - the requirements for the Android application were studied, the list of functions to be automated was determined, the choice of software and hardware was justified, the application database was designed. As a result of the course work was developed Android-application " Maze", which performs all the functions defined in the requirements.

Explanatory note to the course work contains 70 p., 11 pictures, 9 sources, 2 applications.

# Список исполнителей

Владимирова М.А.­ – ответственный за проектирования модулей приложения, разработку и создание интерфейса приложения, проектирование базы данных и реализацию программного кода игрового приложения.

Петров М.М. – ответственный за разработку и создание интерфейса приложения, проектирование и создания базы данных, реализацию программного кода игрового приложения, написание тест-кейсов и выполнение тестирования приложения на наличие ошибок.

Содержание

[ВВЕДЕНИЕ 6](#_Toc92368486)

[1 АНАЛИЗ ЗАДАНИЯ 8](#_Toc92368487)

[2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 10](#_Toc92368488)

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 14](#_Toc92368489)

[3.1 Выбор средств разработки 14](#_Toc92368490)

[3.2 Описание ресурсов 15](#_Toc92368491)

[3.3 Описание классов и методов 17](#_Toc92368492)

[3.4 Тестирование 21](#_Toc92368493)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 24](#_Toc92368494)

[ЗАКЛЮЧЕНИЕ 31](#_Toc92368495)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc92368496)

[ПРИЛОЖЕНИЕ А 34](#_Toc92368497)

[ПРИЛОЖЕНИЕ Б 45](#_Toc92368498)

# ВВЕДЕНИЕ

 В настоящее время мобильные игры являются неотъемлемой частью в жизни человека. Почти у каждого человека, начиная с детей и до пенсионеров есть смартфон, на котором можно поиграть в свою любимую игру. Рынок мобильных игр и приложений стремительно растет, о чем заявляют аналитические компании.

Игры становятся главной причиной роста трат пользователей в магазинах, потому что мобильный гейминг растет быстрее всех остальных типов компьютерных игр. Основной причиной роста такого темпа роста мобильных игр становится быстрый рост числа Android-смартфонов [1].

Работа многих игровых приложений невозможна без базы данных. База данных – хранилище, в котором находятся и систематизируются все данные, которые собирает приложение. Использование баз данных для систематизации полученной информации значительно упрощается дальнейшее использование, мониторинг, хранение данных.

Целью курсовой работы является игрового приложение «Лабиринты» для работы на мобильных устройствах под управлением ОС Android. Игровые мобильные приложения являются самым популярным видом в разработке приложений, потому что именно игры в смартфонах приносят своим владельцам самую большую прибыль в сравнении с прочими видами мобильных приложений. Большая половина дохода всего рынка мобильных приложений приходится именно на игровые. Таким образом, разработка мобильного игрового приложения «Лабиринты» является очень актуальной на сегодняшний день.

Для выполнения данной цели были поставлены следующие задачи:

* спроектировать схему модулей Android-приложения;
* описать функционал приложения;
* спроектировать интерфейс;
* создать структуру базы данных;
* реализовать игровой алгоритм;
* разработать игровое приложение «Лабиринты»;
* протестировать разработанное Android-приложение.

# АНАЛИЗ ЗАДАНИЯ

Потенциальными пользователями Android-приложения «Лабиринты» являются люди, которые заинтересованы такой категорией игр-головоломок как лабиринты. Особенностью приложения является наличие четырех уровней сложности игры, от выбора которых зависит размер и сложность лабиринта. Для того, чтобы после прохождения всех уровней сложности у пользователей было желание вернутся к игре и пройти её еще раз, улучшив свои результаты, в игре реализована возможность сохранения времени, затраченного на прохождение каждого уровня сложности.

Для того, чтобы заинтересовать пользователя в игровом процессе, были продуманы следующие основные моменты:

* интуитивно понятный интерфейс приложения;
* удобное управление в игре с помощью свайпов по экрану;
* возможность сохранения своих достижений.

В рамках реализуемого Android-приложения «Лабиринты» должен быть предусмотрен следующий функционал:

1. выбрать уровень сложности игры;
2. пройти выбранный уровень сложности:
3. найти выход из трех лабиринтов на выбранном уровне;
4. после успешного прохождения одного лабиринта генерируется следующий;
5. все лабиринты одного уровня сложности имеют одинаковый размер, но имеют различные пути.
6. сохранить свое достижение:
7. дату и время прохождения лабиринта;
8. уровень сложности;
9. время, потраченное на прохождение уровня.
10. просмотреть список сохраненных достижений;
11. ознакомиться с правилами игры.

Системным требованием для запуска и работы приложения является Android версии 6.0 и выше.

# ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

Схема модулей Android-приложения «Лабиринты» представлена на рисунке 2.1 [2].

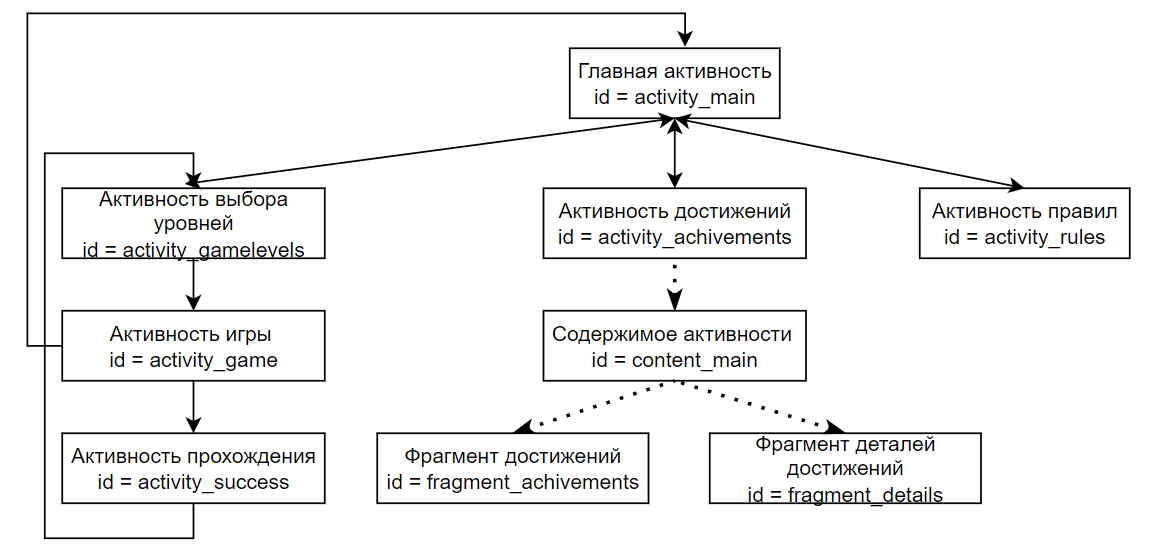


Рисунок 2.1 – Схема модулей приложения «Лабиринты»

«Главная активность» – главная страница Android-приложения, на которой расположены логотип и кнопки меню. На странице реализован функционал перехода на другие активности, такие как: «Активность выбора уровней», «Активность достижений» и «Активность правил» [3].

«Активность выбора уровней» представляет собой страницу со списком уровней сложности игры. На этой странице реализован функционал выбора уровня сложности игры и перехода на «Активность игры», а также возвращение на «Главную активность».

«Активность игры» – страница, на которой отображается игровое поле лабиринта. Функционал активности: перемещение игрока по лабиринту, прохождение уровня, которое приводит к переходу на «Активность прохождения», и переход на «Главную активность».

«Активность прохождения» представляет собой страницу с сообщением об успешном прохождении уровня. В сообощении отображается время, затраченное на прохождение уровня. Функционал составляют: возможность перехода на «Главную активность» с сохранением своего достижения и переход без сохранения.

«Активность достижений» представляет собой страницу, со списком всех сохранённых достижениях пользователя. Функционал активности: выбор достижения для просмотра его деталей и переход на «Главную активность».

«Активность правил» – страница с логотипом приложения и списком его правил. На данной странице реализована возможность перехода на «Главную активность»

После разработки схемы модулей приложения был спроектирован интерфейс приложения, рисунок интерфейса представлен на рисунке 2.2.

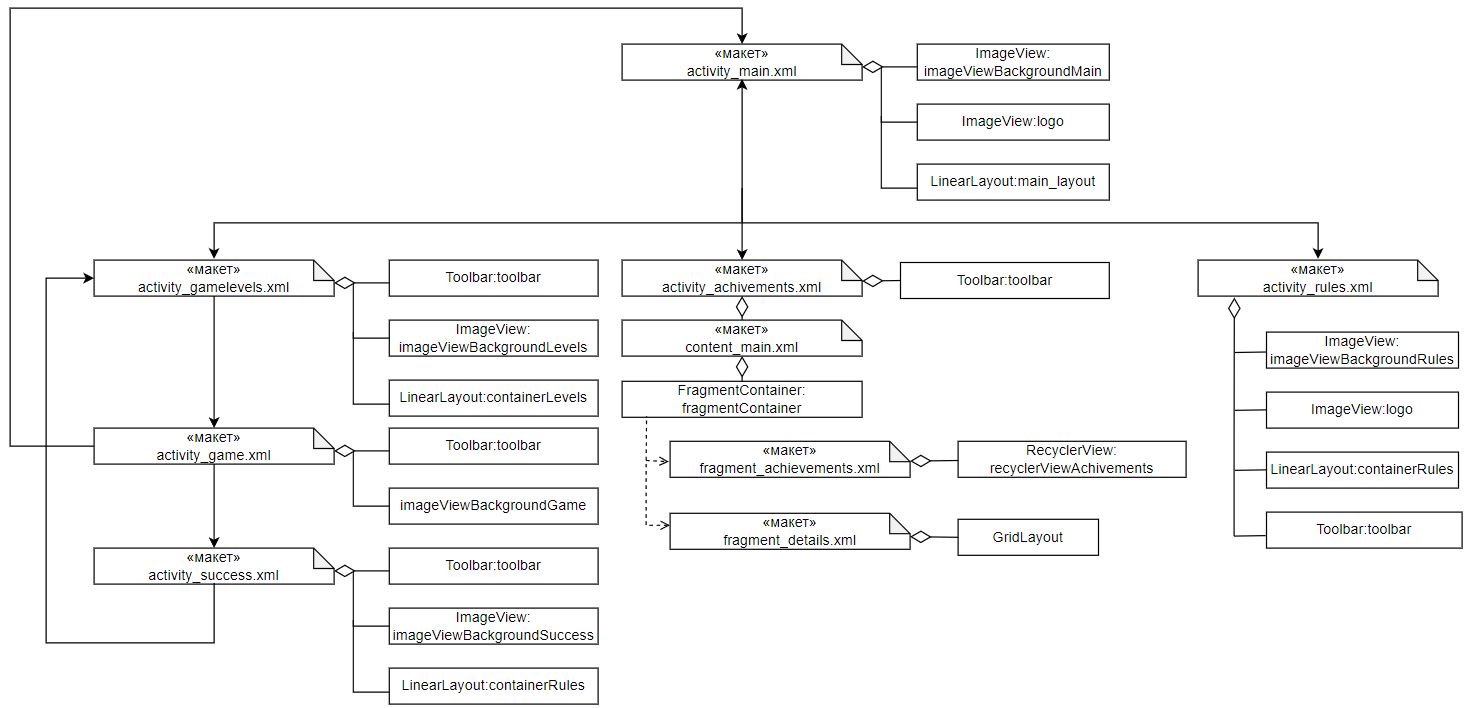


Рисунок 2.2 – Интерфейс приложения «Лабиринты»

На всех страницах приложения есть панель инструментов, в панели инструментов реализована кнопка возврата к «Главной активности».

На главной странице расположен задний фон и логотип приложения «Лабиринты». Также на странице есть меню, которое состоит из трёх кнопок: «Играть», «Достижения» и «Правила игры».

Страница выбора сложности состоит из сообщения, о необходимости выбора уровня сложности и четырёх кнопок для выбора уровня сложности: лёгкая, нормальная, сложная и трудная.

На странице игры находится игровое поле, которое состоит из лабиринта, квадрата красного цвета, представляющего игрока, и квадрата зелёного цвета, представляющего выход из лабиринта.

На странице прохождения находятся: сообщение об успешном прохождении, в котором выводится время, затраченное на прохождение уровня, кнопка «Сохранить и играть», кнопка «Сохранить».

Страница достижений состоит из списка дат сохранённых пользователем достижений.

На странице деталей достижения расположены следующие поля сохранённого достижения: дата прохождения, сложность лабиринта и время, затраченное на прохождение, в секундах.

Страница правил состоит из заднего фона, логотипа приложения «Лабиринты» и списка правил приложения.

Для сохранения достижений пользователя в Android-приложении «Лабиринты» была спроектирована база данных. Схема базы данных представлена на рисунке 2.3.

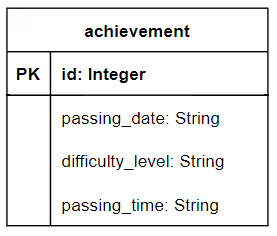


Рисунок 2.3 – Схема базы данных приложения «Лабиринты»

База данных состоит из одной таблицы «achievement». Таблица содержит в себе следующие столбцы: id, passing\_date, difficulty\_level и passing\_time.

Столбец id – первичный ключ таблицы, является идентификатором записи о достижении.

Столбец passing\_date предназначен для хранения информации о дате и времени успешного завершения уровня сложности.

Столбец difficulty предназначен для хранения информации о сложности успешно пройденного уровня сложности.

Столбец passing\_time хранит затраченное пользователем время на успешное прохождение уровня сложности.

# ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

## Выбор средств разработки

Для разработки Android-приложения «Лабиринты» использовались следующие средства разработки: Java, Android Studio, SQLite.

Java – строго типизированный объектно-ориентированный язык программирования общего назначения. Основной причина выбора этого языка является его крупное сообщество и большое количество документации для работы с этим языком [4 ,5].

Android Studio является официальной интегрированной средой разработки для операционной системы Android под управлением Google. Эта программа была выбрана из-за приятного пользовательского интерфейса, удобного XML редактора, встроенного эмулятора устройств, возможности использования облачного сервиса от Google – Firebase Test Lab для тестирования приложения на эмуляторах и реальных устройствах [6].

SQLite – это система управления реляционными базами данных, похожая на Oracle, MySQL, PostgreSQL и SQL Server. Она реализует большую часть стандарта SQL, но в отличие от четырех упомянутых выше СУБД она не поддерживает модель «клиент-сервер». Скорее, она встроена в конечную программу. Это означает, что можно связать базу данных SQLite с приложением и получить доступ ко всем возможностям БД в своем приложении [7].

## Описание ресурсов

В Android-приложении «Лабиринты» представлены следующие типы ресурсов:

* drawable;
* layout;
* mipmap;
* values.

В папке drawable находятся ресурсы изображений и фигур приложения. Ресурсы изображений – background.png и logo.png, представляют собой фон приложения и логотип соответственно. Ресурсы фигур содержат в себе описание стилей кнопок приложения. Button\_style.xml и button\_style\_pressed.xml содержат описание стилей кнопок – закругленные границы фиолетового цвета. В нажатом состоянии кнопка залита белым цветом. Ресурс button\_press\_unpressed.xml регулирует применение стилей для нажатой и не нажатой кнопки. Textview\_border.xml содержит описание границ для текстовых полей – границы серого цвета с закругленными краями.

Ресурсы layout – ресурсы разметки, которые отвечают за внешний вид приложения. Activity\_main.xml определяет графический интерфейс главной страницы приложения «Лабиринты». Содержит основной компонент RelativeLayout, в который вложены компоненты: фон приложения, логотип и LinearLayout с главным меню приложения. Главное меню приложения состоит из трех компонентов Button: buttonPlay, buttonAchievements, buttonRules.

Activity\_rules.xml – интерфейс страницы приложения с правилами. Основной элемент страницы – RelativeLayout. В него вложены следующие компоненты: Toolbar, два ImageView, определяющие фон и логотип, LinearLayout с вложенными компонентами TextView, в которых содержатся строковые ресурсы правил приложения. Toolbar – панель инструментов, в которой содержится название приложения и кнопка «Назад».

Activity\_gamelevels.xml определяет графический интерфейс страницы выбора уровней. В RelativeLayout вложены следующие компоненты: Toolbar, ImageView с фоном приложения, LinearLayout с вложенными компонентами TextView и контейнерами LinearLayout с кнопками выбора уровней.

Activity\_game.xml – графический интерфейс страницы с игровым полем лабиринта. Основной элемент страницы – RelativeLayout. В него вложены следующие компоненты: Toolbar и ImageView, определяющий фон страницы.

Activity\_success.xml определяет интерфейс страницы успешного прохождения уровня сложности. В RelativeLayout вложены следующие компоненты: Toolbar, ImageView с фоном приложения, LinearLayout с вложенными компонентами TextView и кнопками Button выбора способа дальнейшего прохождения игры с сохранением или без. В компонентах TextView содержатся строковые ресурсы с информацией об успешном прохождении уровня сложности и затраченном времени на прохождение.

Activity\_achievements.xml содержит компонент CoordinatorLayout с панелью приложения Toolbar. Также включает в себя ресурс Content\_main.xml.

Content\_main.xml определяет часть графического интерфейса activity\_achievements.xml. Содержит в себе компонент FrameLayout. В зависимости от выбранной пользователем страницы в него будет загружен ресурс fragment\_achievements.xml или fragment\_details.xml.

Fragment\_achievements.xml определяет интерфейс страницы со списком дат сохраненных достижений пользователя. Основной компонент – RecyclerView, в который из базы данных загружаются даты сохраненных пользователем достижений.

Fragment\_details.xml – графический интерфейс страницы с полным описанием сохраненного достижения. Компонент GridLayout содержит в себе компоненты TextView, определяющие поля для вывода информации о достижении и описание этих полей.

В папке mipmap хранится логотип приложения в нескольких разрешениях.

Группа ресурсов values содержит в себе: strings – строковые ресурсы, colors – цветовые ресурсы, dimens – ресурсы размеров, используемые для компонентов графического интерфейса, themes – темы приложения.

Для строковых ресурсов была выполнена интернационализация. Строковые ресурсы с их значениями на английском и русском языках представлены на рисунке 3.1.



Рисунок 3.1 – Строковые ресурсы приложения

Полный код файлов с описанием ресурсов приведен в приложении A.

## Описание классов и методов

В основном пакете приложения были созданы следующие классы:

* MainActivity – субкласс AppCompatActivity, реализует главное меню игры, предоставляя возможность выбора перехода к отображению классов: GameLevels, Achievements и Rules.
* Rules – субкласс AppCompatActivity, используется для отображения правил игры.
* Success – субкласс AppCompatActivity, предоставляет возможность сохранения результата прохождения уровня сложности и возврата к выбору уровня сложности лабиринта.
* ItemDivider – субкласс RecyclerView.ItemDecoration, используется компонентом RecyclerView класса AchievementFragment для рисования горизонтальной линии между элементами списка.
* GameLevels – субкласс AppCompatActivity, передает классу Game информацию о выбранном пользователем уровне сложности лабиринта.
* Game – субкласс AppCompatActivity, используется для отображения страницы с игровым полем, создания и инициализации объекта класса GameManager.
* DetailFragment – субкласс androidx.fragment.app.Fragment, отображающий детальную информацию одного достижения.
* AchievementFragment – субкласс androidx.fragment.app.Fragment, отображает список сохраненных пользователем достижений.
* AchievementAdapter – субкласс RecyclerView.Adapter, предоставляет данные компоненту RecyclerView класса AchievementFragment.
* AchievementActivity – субкласс AppCompatActivity, управляет фрагментами достижения и реализут их методы обратного вызова (вывод достижений, выбор одного достижения).

Класс GameLevels. В классе созданы следующие поля:

* private final String LOG\_TAG = "GameLevels\_logs". Определяет строку для отметки класса при использовании логирования.
* private int level. Определяет уровень сложности лабиринта, по умолчанию имеет значение 0.
* private int[] levels. Массив, который хранит в себе соответствующие уровням сложности лабиринтов значения для генерирования игрового поля. Имеет значения: 15, 20, 30, 40.

В классе GameLevels реализованы следующие методы:

* protected void onCreate(Bundle savedInstanceState){}. Задает начальную установку параметров при инициализации активности. Определяет слушателей кнопок с выбором уровня сложности. При выборе уровня сложности управление будет передано классу Game с параметрами, соответствущими выбранному уровню сложности лабиринта.
* public boolean onOptionsItemSelected(MenuItem item) {}. Указывает, что при нажатии на кнопку «назад» на панели инструментов будет передано управление классу MainActivity.
* public void onBackPressed() {}. Переопределяет работу системной кнопки «назад», которая также будет передавать управление классу MainActivity.

В пакете data были созданы классы, относящиеся к работе с базой данных SQLite:

* DatabaseDescription – содержит описание таблицы achievement базы данных.
* AchievementDatabaseHelper – субкласс SQLiteOpenHelper, создает базу данных и используется для работы с ней.
* AchievementContentProvider – субкласс ContentProvider, определяет методы для работы с базой данных.

Класс DatabaseDescription. В классе созданы следующие поля:

* public static final String AUTHORY = "com.example.maze.data". Определяет имя ContentProvider.
* private static final Uri BASE\_CONTENT\_UTI = Uri.parse("content://" + AUTHORY). Определяет базовый URI для взаимодействия с ContentProvider.
* public static final class Achievement implements BaseColumns{}. Вложенный класс, определяющий содержимое таблицы achievement.

Во вложенном классе Achievement созданы следующие поля:

* public static final String TABLE\_NAME = "achievement". Задает название таблицы в базе данных.
* public static final Uri CONTENT\_URI. Определяет объект URI для таблицы achievement.
* public static final String COLUMN\_PASSING\_DATE = "passing\_date", public static final String COLUMN\_DIFFICULTY\_LEVEL = "difficulty\_level", public static final String COLUMN\_PASSING\_TIME = "passing\_time". Определяют имена столбцов таблицы achievement.

Во вложенном классе Achievement реализован метод:

* public static Uri buildAchievementUri(long id){}. Используется для создания URI для конкретного достижения.

В пакете game были созданы следующие классы, определяющие интерфейс игрового поля:

* Dot – реализует интерфейс Drawable, используется для создания визуального представления выхода из лабиринта и игрока на игровом поле лабиринта.
* Drawable – интерфейс, определяет метод для рисования объекта.
* Exit – субкласс Dot, используется для отрисовки выхода из лабиринта на игровом поле.
* GameManager – субкласс GestureDetector.SimpleOnGestureListener, содержит методы, которые реагируют на перемещение игрока по лабиринту.
* Maze – реализует интерфейс Drawable, содержит методы для генерирования и отображения игрового поля.
* MazeView – субкласс View, используется для рисования представления игрового поля лабиринта.
* Player – субкласс Dot, содержит методы для отрисовки и перемещения игрока на игровом поле.

Класс Dot. В классе созданы следующие поля:

* private int size. Определяет размер точки на игровом поле.
* private Point point. Определяет координаты отрисовки точки в лабиринте.
* private Paint paint. Определяет свойства для отображения точки выхода и игрока на поле лабиринта.

В классе Dot реализованы методы:

* public Point getPoint(){}. Возвращает координаты отрисовки точки на игровом поле.
* public void draw(Canvas canvas, Rect rect){}. Используется для отрисовки точки на игровом поле.

Полный код описания классов приложения «Лабиринты» с комментариями представлен в приложении Б.

## Тестирование

Для тестирования Android-приложения «Лабиринты» были написаны следующие тест-кейсы [8]:

Тест-кейс №1

Название: Возможность выбора уровня сложности

Шаги:

Открыть приложение «Лабиринты»

Нажать «Играть»

Выбрать любой уровень сложности

Ожидаемый результат:

Отображается лабиринт, в лабиринте есть игрок и выход, размер лабиринта соответствует выбранному уровню сложности.

Тест-кейс №2

Название: Лабиринты всех уровней сложности можно пройти

Шаги:

Открыть приложение «Лабиринты»

Нажать «Играть»

Выбрать любой уровень сложности

Ожидаемый результат:

На каждом уровне сложности имеется возможность дойти то конца лабиринта.

Тест-кейс №3

Название: Страница с сообщением об успешном прохождении уровня

Шаги:

Открыть приложение «Лабиринты»

Нажать «Играть»

Выбрать любой уровень сложности

Пройти все лабиринты на уровне

Ожидаемый результат:

Отображается страница с сообщением об успешном прохождении уровня, в сообщение указано время прохождения.

Тест-кейс №4

Название: Сохранение результата прохождения уровня

Шаги:

Открыть приложение «Лабиринты»

Нажать «Играть»

Выбрать любой уровень сложности

Пройти все лабиринты на уровне

Нажать кнопку «Сохранить и играть»

Ожидаемый результат:

Результат прохождения уровня сохранён, пользователь попал на экран выбора сложности игры.

Тест-кейс №5

Название: Просмотр достижений

Предусловия: Выполнить тест-кейс №4

Шаги:

Открыть приложение «Лабиринты»

Нажать «Достижения»

Ожидаемый результат:

Отображается список записей со всеми сохранёнными прохождениями уровней, название записи соответствует дате и времени прохождения уровня.

Тест-кейс №6

Название: Просмотр достижений

Предусловия: Выполнить тест-кейс №4

Шаги:

Открыть приложение «Лабиринты»

Нажать «Достижения»

Нажать на запись в списке достижений

Ожидаемый результат:

На странице отображаются поля с датой прохождения уровня сложности, сложность пройденного лабиринта и затраченное на прохождение время.

Тест-кейс №7

Название: Отображение правил игры

Шаги:

Открыть приложение «Лабиринты»

Нажать «Правила игры»

Ожидаемый результат:

На странице отображается текст с правилами, текст отображается полностью и не выходит за рамки экрана.

После прохождения написанных тест-кейсов фактический результат совпал с ожидаемым результатом. Все тест-кейсы были пройдены успешно.

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В Android-приложении «Лабиринты» реализована поддержка двух языков: Английского и Русского.

Для установки Android-приложения «Лабиринты» пользователю необходимо иметь устройство с версией Android не ниже шестой. Чтобы установить приложение пользователь должен запустить установочный файл Maze формата .APK [9].

После запуска приложения пользователь попадает на главную страницу, представленную на рисунке 4.1. С этой страницы пользователь может перейти к игре, к странице со своими достижениями и к странице с правилами игры.

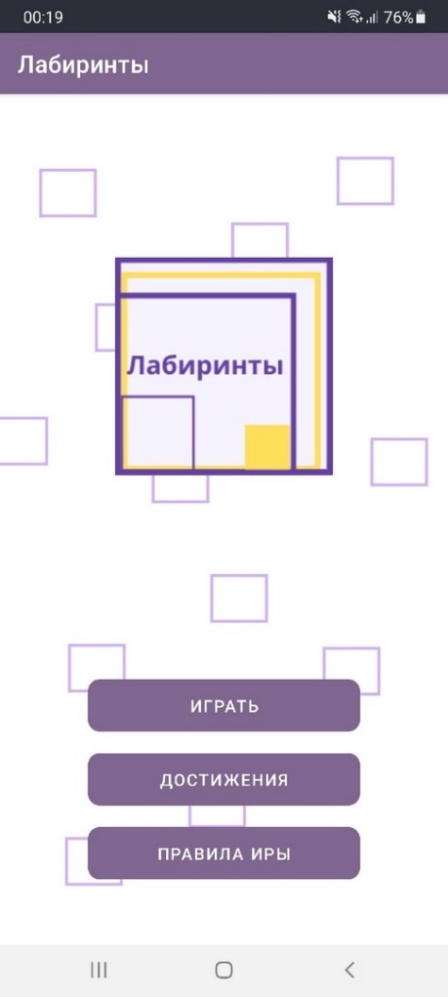


Рисунок 4.1 – Главная страница приложения

На каждой странице приложения, кроме главной, в верхней части экрана есть кнопка в виде стрелочки, нажатие на которую ведёт на главную страницу.

Нажав кнопку «Играть», пользователь видит страницу с возможностью выбора сложности лабиринта (рисунок 4.2), сложность лабиринта влияет на его размер. Нажав на интересующую его сложность, пользователь видит игровое поле, представленное на рисунке 4.3.



Рисунок 4.2 – Страница выбора сложности

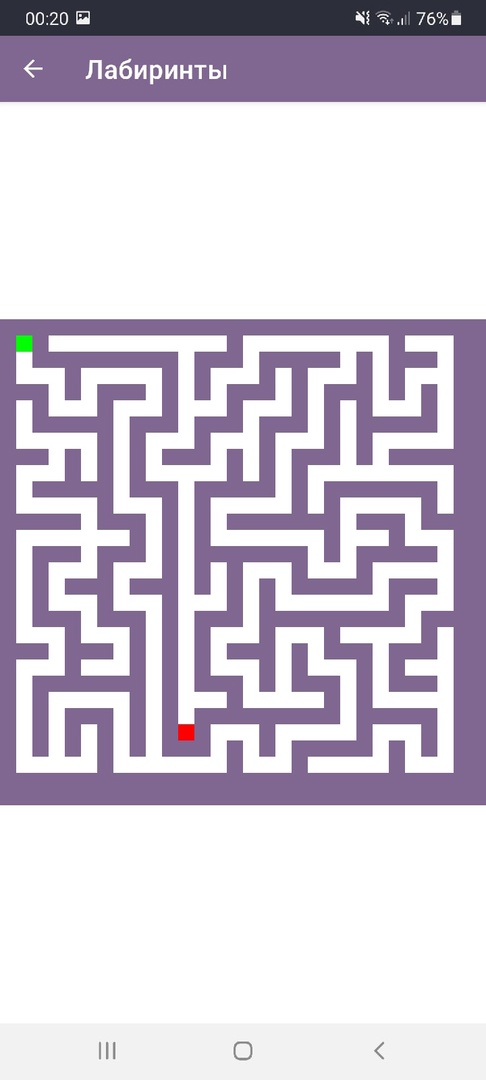


Рисунок 4.3 – Страница с игровым полем

Игровое поле состоит из лабиринта, квадрата красного цвета и квадрата зелёного цвета. Целью игры является прохождение лабиринта, для этого пользователю необходимо провести красный квадрат, представляющего игрока к выходу, который представлен зелёным квадратом.

Для перемещения игрока по лабиринту пользователю необходимо провести свайп (жест, когда пользователь кладет палец на экран мобильного устройства и ведет его в нужном направлении по экрану) в нужном ему направлении, так, чтобы переместить игрока на право, пользователю необходимо сделать свайп из левой части экрана в правую.

После того, как пользователь проведёт красный квадрат через весь лабиринт и доберётся до выхода из лабиринта, на странице появится новый лабиринт, чтобы полностью пройти уровень, пользователю необходимо таким образом пройти три лабиринта подряд. После прохождения уровня пользователю будет представлена страница с сообщением об успешном прохождении, которая показана на рисунке 4.4. На этой странице пользователь может ознакомится с затраченным им на прохождение уровня временем, сохранить свой результат и перейти на страницу с выбором уровня сложности, либо перейти на страницу с выбором уровня сложности без сохранения результата прохождения.

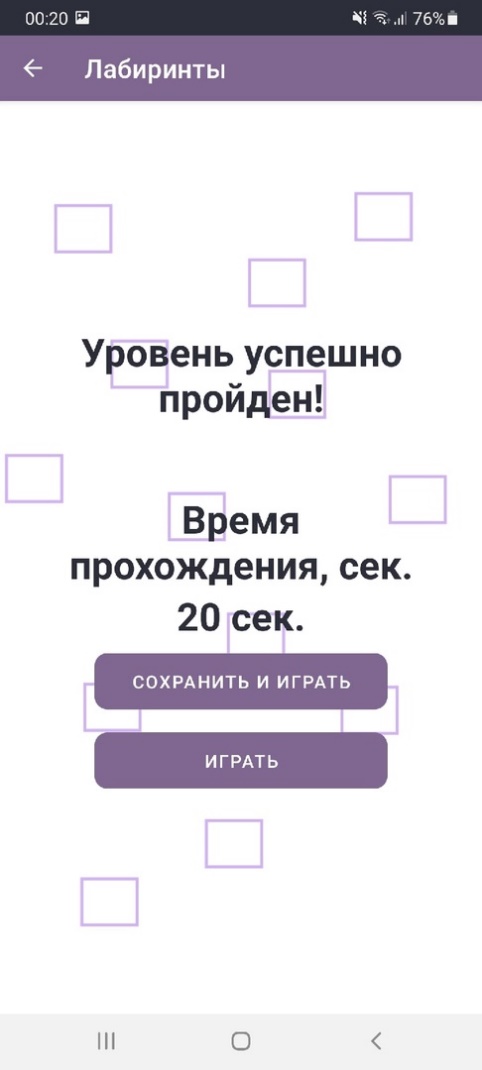


Рисунок 4.4 – Страница с сообщением об успешном прохождении уровня

Свои сохранённые результаты пользователь может посмотреть, нажав на кнопку «Достижения» на главной странице. На странице с достижениями пользователю представлен список всех сохранённых им достижений, состоящий из записей, названия которых соответствуют дате и времени прохождения уровня, страница представлена на рисунке 4.5.

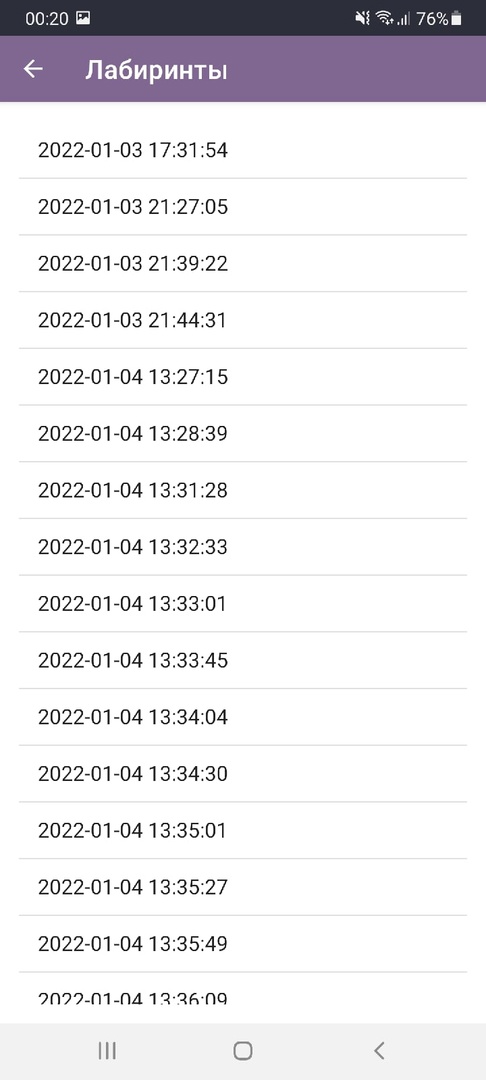


Рисунок 4.5 – Страница достижений

Для подробного просмотра достижения пользователь может нажать по интересующей его записи, после чего откроется страница с подробным описанием достижения, которая представлена на рисунке 4.6. На странице пользователю представлены три поля и соответствующие им подписи, а именно, поле, в котором отображается дата прохождения уровня, поле, в котором отображается сложность пройденного уровня и поле, в котором отображается время прохождения уровня.

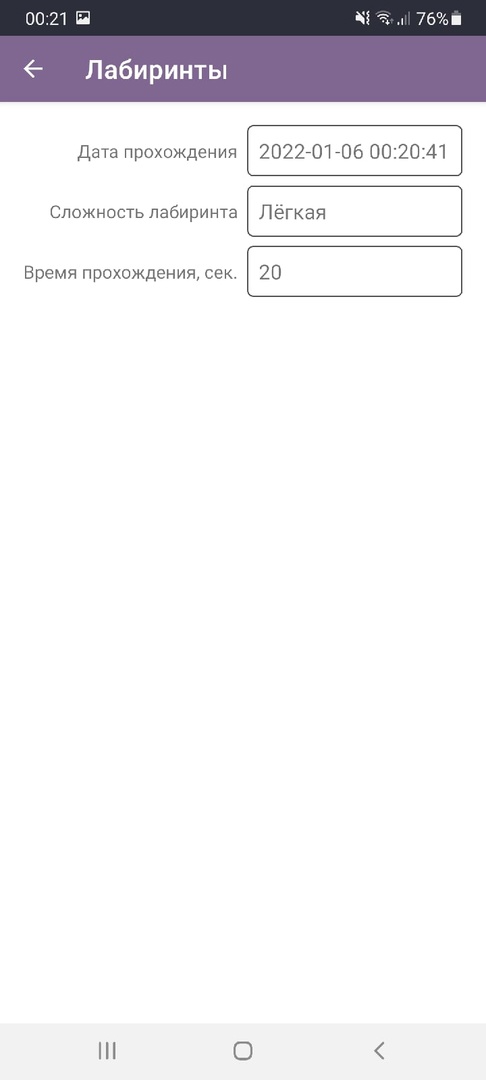


Рисунок 4.6 – Страница с подробным описанием достижения

Также в приложении указаны правила игры в кратком виде, которые доступны пользователю при нажатии по кнопке «Правила игры» на главном экране приложения. Страница с правилами игры представлена на рисунке 4.7.

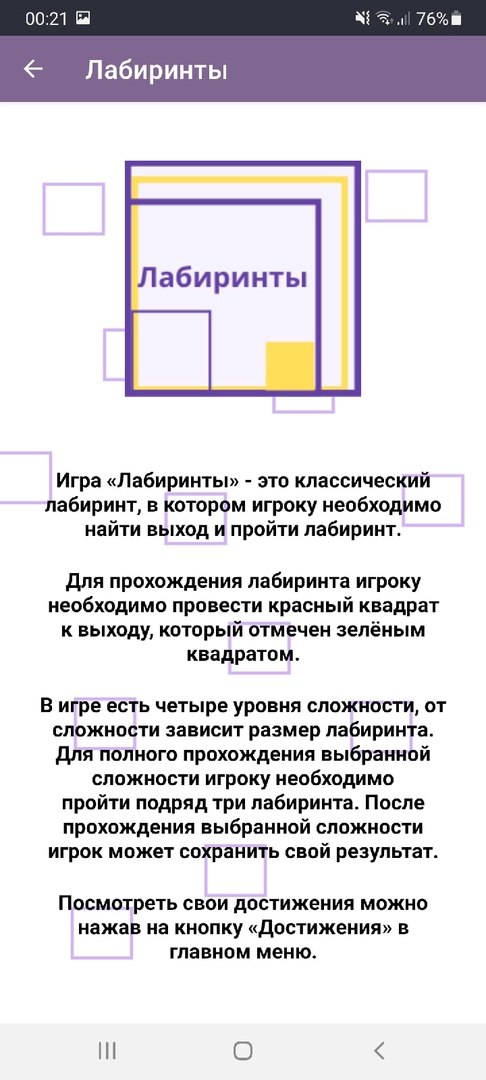


Рисунок 4.7 – Страница с правилами игры

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы было разработано игровое приложение «Лабиринты» для работы на мобильных устройствах под управлением ОС Android. Спроектированное мобильное приложение отвечает всем требованиям предметной области.

В игровом приложении «Лабиринты» были продуманы следующие основные моменты:

* интуитивно понятный интерфейс приложения;
* удобное управление в игре с помощью свайпов по экрану;
* возможность сохранения своих достижений.

В рамках реализуемого Android-приложения «Лабиринты» был разработан следующий функционал:

1. выбор уровня сложности игры;
2. прохождение выбранного уровня сложности игры:
3. поиск выхода из трех лабиринтов на выбранном уровне;
4. после успешного прохождения одного лабиринта происходит генерация следующего;
5. все лабиринты одного уровня сложности имеют одинаковый размер, но имеют различные пути.
6. возможность сохранения своего достижения:
7. дату и время прохождения лабиринта;
8. уровень сложности;
9. время, потраченное на прохождение уровня.
10. просмотр списка сохраненных достижений;
11. ознакомление с правилами игры.

Системным требованием для запуска и работы приложения является Android версии 6.0 и выше.

К достоинствам разработанного игрового приложения «Лабиринты» можно отнести следующие:

* интуитивно понятный интерфейс приложения;
* возможность сохранения и просмотра своих игровых достижений;
* простое и удобное управление пользователем на игровом поле лабиринта с помощью свайпов;
* возможность выбора уровня сложности лабиринта;
* случайная генерация игрового поля, каждый лабиринт не похож на предыдущий.

Недостатком Android-приложения можно считать отсутствие в игровом поле лабиринта дополнительных игровых объектов, таких как дополнительные препятствий и бонусов.

В перспективах дальнейшего развития игрового приложения «Лабиринты» планируется устранение выявленных недостатков и добавление нового функционала, предоставляющего возможность выбора цветовой темы игрового поля лабиринта.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* 1. Тренды мобильных игр в 2021 году — обзор индустрии мобильных игр // vc.ru [Электронный ресурс] – https://vc.ru/flood/273356-trendy-mobilnyh-igr-v-2021-godu-obzor-industrii-mobilnyh-igr
  2. Модули Android приложения // Java-online [Электронный ресурс] – https://java-online.ru/android-module.xhtml
  3. Активность и её жизненный цикл // Android Tools [Электронный ресурс] – https://android-tools.ru/coding/aktivnost-i-eyo-zhiznennyj-cikl/
  4. Программирование под Андроид на Java // metanit.com [Электронный ресурс] – https://metanit.com/java/android/
  5. Как работает Android. Введение для Java-разработчиков// JAVARUSH [Электронный ресурс] – https://javarush.ru/groups/posts/481-kak-rabotaet-android-vvedenie-dlja-java-razrabotchikov
  6. Основные преимущества Android Studio // Mobile [Электронный ресурс] – http://mobile.incredibleart.ru/blog/preimuschestva-android-studio/
  7. SQLite — замечательная встраиваемая БД // Хабр [Электронный ресурс] – https://habr.com/ru/post/149356/
  8. How to Write Test Cases for Software // PARASOFT [Электронный ресурс] – https://www.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/
  9. Extracting an APK file from Android Studio // educative [Электронный ресурс] – https://www.educative.io/edpresso/extracting-an-apk-file-from-android-studio

# ПРИЛОЖЕНИЕ А

Описание ресурсов

Ресурсы drawable:

Код ресурса button\_press\_unpressed.xml

<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:state\_pressed="true" android:drawable="@drawable/button\_style\_pressed"/>  
 <item android:state\_pressed="false" android:drawable="@drawable/button\_style"/>  
</selector>

Код ресурса button\_style.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <stroke android:width="@dimen/button\_width\_2dp" android:color="@color/purple"/>  
 <corners android:radius="@dimen/button\_radius\_10dp"/>  
</shape>

Код ресурса textview\_border.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
 <corners android:radius="5dp"/>  
 <stroke  
 android:width="1dp"  
 android:color="#555" />  
 <padding  
 android:bottom="10dp"  
 android:left="10dp"  
 android:right="10dp"  
 android:top="10dp" />  
  
</shape>

Код ресурса button\_style\_pressed.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
  
  
 <stroke android:width="@dimen/button\_width\_2dp" android:color="@color/purple"/>  
 <corners android:radius="@dimen/button\_radius\_10dp"/>  
 <solid android:color="@color/white"/>  
</shape>

Ресурсы layout:

Код ресурса activity\_achivements.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout 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:id="@+id/coordinatorLayot"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 </com.google.android.material.appbar.AppBarLayout>  
  
 <include layout="@layout/content\_main" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

Код ресурса activity\_game.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"  
 android:orientation="vertical">  
  
 <Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <ImageView  
 android:id="@+id/imageViewBackgroundGame"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:contentDescription="@string/background\_contentDescription"  
 app:srcCompat="@drawable/background" />  
  
</RelativeLayout>

Код ресурса activity\_gamelevels.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"  
 android:orientation="vertical">  
  
 <Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <ImageView  
 android:id="@+id/imageViewBackgroundLevels"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:contentDescription="@string/background\_contentDescription"  
 android:scaleType="centerCrop"  
 android:src="@drawable/background" />  
  
 <LinearLayout  
 android:id="@+id/containerLevels"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/textViewChooseLevel"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="@dimen/text\_size\_messages"  
 android:text="@string/choose\_level"  
 android:textAlignment="center"  
 android:textColor="@color/dark\_brown"  
 android:textSize="@dimen/text\_size\_messages"  
 android:textStyle="bold" />  
  
 <LinearLayout  
 android:id="@+id/buttonLevelsContainer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <LinearLayout  
 android:id="@+id/lineFirst"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/buttonEasy"  
 android:layout\_width="@dimen/button\_width\_height\_150dp"  
 android:layout\_height="@dimen/button\_width\_height\_150dp"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:layout\_weight="1"  
 android:text="@string/difficulty\_easy" />  
  
 <Button  
 android:id="@+id/buttonNormal"  
 android:layout\_width="@dimen/button\_width\_height\_150dp"  
 android:layout\_height="@dimen/button\_width\_height\_150dp"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:layout\_weight="1"  
 android:text="@string/difficulty\_normal" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/lineSecond"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/buttonHard"  
 android:layout\_width="@dimen/button\_width\_height\_150dp"  
 android:layout\_height="@dimen/button\_width\_height\_150dp"  
 android:layout\_weight="1"  
 android:text="@string/difficulty\_hard"  
 android:layout\_margin="@dimen/button\_margin\_10dp" />  
  
 <Button  
 android:id="@+id/buttonVeryHard"  
 android:layout\_width="@dimen/button\_width\_height\_150dp"  
 android:layout\_height="@dimen/button\_width\_height\_150dp"  
 android:layout\_weight="1"  
 android:text="@string/difficulty\_very\_hard"  
 android:layout\_margin="@dimen/button\_margin\_10dp" />  
 </LinearLayout>  
  
 </LinearLayout>  
 </LinearLayout>  
</RelativeLayout>

Код ресурса activity\_main.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"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 >  
  
 <ImageView  
 android:id="@+id/imageViewBackgroundMain"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:contentDescription="@string/background\_contentDescription"  
 android:scaleType="centerCrop"  
 android:src="@drawable/background" />  
  
 <ImageView  
 android:id="@+id/logo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="@dimen/dimension\_logo"  
 android:contentDescription="@string/logo\_contentDescription"  
 android:src="@drawable/logo" />  
  
 <LinearLayout  
 android:id="@+id/main\_layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:padding="@dimen/button\_padding\_50dp"  
 android:orientation="vertical">  
  
 <Button  
 android:id="@+id/buttonPlay"  
 android:layout\_width="@dimen/button\_width\_250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:background="@drawable/button\_press\_unpressed"  
 android:text="@string/play" />  
  
 <Button  
 android:id="@+id/buttonAchievements"  
 android:layout\_width="@dimen/button\_width\_250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:background="@drawable/button\_press\_unpressed"  
 android:text="@string/achievements" />  
  
 <Button  
 android:id="@+id/buttonRules"  
 android:layout\_width="@dimen/button\_width\_250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:background="@drawable/button\_press\_unpressed"  
 android:text="@string/rules" />  
 </LinearLayout>  
  
</RelativeLayout>

Код ресурса activity\_rules.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"  
 android:orientation="vertical">  
  
 <Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <ImageView  
 android:id="@+id/imageViewBackgroundLevels"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:contentDescription="@string/background\_contentDescription"  
 android:scaleType="centerCrop"  
 android:src="@drawable/background" />  
  
 <ImageView  
 android:id="@+id/logo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="@dimen/dimension\_logo\_rules"  
 android:contentDescription="@string/logo\_contentDescription"  
 android:src="@drawable/logo" />  
  
 <LinearLayout  
 android:id="@+id/containerLevels"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginBottom="@dimen/button\_padding\_50dp"  
 android:paddingRight="@dimen/padding\_left\_right\_30dp"  
 android:paddingLeft="@dimen/padding\_left\_right\_30dp">  
  
 <TextView  
 android:id="@+id/textViewRulesRule1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:paddingBottom="@dimen/padding\_inner"  
 android:text="@string/rules\_list\_rule\_1"  
 android:textColor="#000000"  
 android:textSize="@dimen/text\_size\_rules"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textViewRulesRule2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:paddingBottom="@dimen/padding\_inner"  
 android:text="@string/rules\_list\_rule\_2"  
 android:textColor="#000000"  
 android:textSize="@dimen/text\_size\_rules"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textViewRulesRule3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:paddingBottom="@dimen/padding\_inner"  
 android:text="@string/rules\_list\_rule\_3"  
 android:textColor="#000000"  
 android:textSize="@dimen/text\_size\_rules"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textViewRulesRule4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:text="@string/rules\_list\_rule\_4"  
 android:textColor="#000000"  
 android:textSize="@dimen/text\_size\_rules"  
 android:textStyle="bold" />  
  
 </LinearLayout>  
</RelativeLayout>

Код ресурса activity\_success.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"  
 android:orientation="vertical">  
  
 <Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <ImageView  
 android:id="@+id/imageViewBackgroundSuccess"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:contentDescription="@string/background\_contentDescription"  
 android:src="@drawable/background" />  
  
 <LinearLayout  
 android:id="@+id/containerLevels"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="@dimen/button\_padding\_50dp">  
  
 <TextView  
 android:id="@+id/textViewRulesRule1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="@dimen/button\_radius\_10dp"  
 android:text="@string/success"  
 android:textAlignment="center"  
 android:textColor="@color/dark\_brown"  
 android:textSize="@dimen/text\_size\_messages"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textViewPassingTimeMessage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="@dimen/button\_padding\_50dp"  
 android:text="@string/label\_passing\_time"  
 android:textAlignment="center"  
 android:textColor="@color/dark\_brown"  
 android:textSize="@dimen/text\_size\_messages"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textViewPassingTimeValue"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:textColor="@color/dark\_brown"  
 android:textSize="@dimen/text\_size\_messages"  
 android:textStyle="bold" />  
  
 <Button  
 android:id="@+id/buttonSaveAndPlay"  
 android:layout\_width="@dimen/button\_width\_250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:background="@drawable/button\_press\_unpressed"  
 android:text="@string/saveAndPlay" />  
  
 <Button  
 android:id="@+id/buttonPlay"  
 android:layout\_width="@dimen/button\_width\_250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/button\_margin\_10dp"  
 android:background="@drawable/button\_press\_unpressed"  
 android:text="@string/play" />  
  
 </LinearLayout>  
</RelativeLayout>

Код ресурса content\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout 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:id="@+id/fragmentContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context=".MainActivity" />

Код ресурса fragment\_achivements.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/fragmentContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerViewAchivements"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" >  
  
 </androidx.recyclerview.widget.RecyclerView>  
  
</FrameLayout>

Код ресурса fragment\_details.xml

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent">  
  
 <GridLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:columnCount="2"  
 android:useDefaultMargins="true">  
  
 <TextView  
 android:id="@+id/passingDateLabelTextView"  
 style="@style/MazeLabelTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_row="0"  
 android:layout\_column="0"  
 android:text="@string/label\_passing\_date" />  
  
 <TextView  
 android:id="@+id/passingDateTextView"  
 style="@style/MazeTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_row="0"  
 android:layout\_columnWeight="1"/>  
  
 <TextView  
 android:id="@+id/levelLabelTextView"  
 style="@style/MazeLabelTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_row="1"  
 android:layout\_column="0"  
 android:text="@string/label\_level" />  
  
 <TextView  
 android:id="@+id/levelTextView"  
 style="@style/MazeTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_row="1"  
 android:layout\_columnWeight="1"  
 />  
  
 <TextView  
 android:id="@+id/passingTimeLabelTextView"  
 style="@style/MazeLabelTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_row="2"  
 android:layout\_column="0"  
 android:text="@string/label\_passing\_time" />  
  
 <TextView  
 android:id="@+id/passingTimeTextView"  
 style="@style/MazeTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_row="2"  
 android:layout\_column="1" />  
  
 </GridLayout>  
  
  
</ScrollView>

Ресурсы values:

Код ресурса colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="purple\_200">#8972B8</color>  
 <color name="purple\_500">#5E35B1</color>  
 <color name="purple\_700">#5424B5</color>  
 <color name="teal\_200">#43B5C3</color>  
 <color name="teal\_700">#039BE5</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="dark\_white">#f3f3f5</color>  
 <color name="dark\_brown">#2c2e39</color>  
 <color name="orange">#e48826</color>  
 <color name="purple">#7f6791</color>  
 <color name="purpleStr">#7f6799</color>  
 <color name="pink">#ecc8c9</color>  
 <color name="light\_brown">#c6a78f</color>  
 <color name="light\_blue">#a0b3a8</color>  
</resources>

Код ресурса dimens.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <dimen name="dimension\_logo">150dp</dimen>  
 <dimen name="dimension\_logo\_rules">50dp</dimen>  
 <dimen name="button\_radius\_10dp">10dp</dimen>  
 <dimen name="button\_width\_2dp">2dp</dimen>  
 <dimen name="button\_padding\_50dp">50dp</dimen>  
 <dimen name="padding\_left\_right\_30dp">30dp</dimen>  
 <dimen name="padding\_inner">20dp</dimen>  
 <dimen name="button\_margin\_10dp">10dp</dimen>  
 <dimen name="button\_width\_250dp">250dp</dimen>  
 <dimen name="button\_width\_height\_150dp">150dp</dimen>  
 <dimen name="text\_size\_messages">30sp</dimen>  
 <dimen name="text\_size\_rules">16sp</dimen>  
 <dimen name="fab\_margin">16dp</dimen>  
 <dimen name="activity\_horizontal\_margin">16dp</dimen>  
 <dimen name="activity\_vertical\_margin">16dp</dimen>  
 <dimen name="divider\_margin">16dp</dimen>  
</resources>

Код ресурса strings.xml

<resources>  
 <string name="app\_name">Maze</string>  
 <string name="play">Play</string>  
 <string name="saveAndPlay">Save and play</string>  
 <string name="achievements">Achievements</string>  
 <string name="rules">Rules</string>  
 <string name="difficulty\_easy">Easy</string>  
 <string name="difficulty\_normal">Normal</string>  
 <string name="difficulty\_hard">Hard</string>  
 <string name="difficulty\_very\_hard">Very hard</string>  
 <string name="rules\_list\_rule\_1">The game «Maze» is a classic maze game in which the player must find a way out and pass the maze.</string>  
 <string name="rules\_list\_rule\_2">To pass the labyrinth, the player must lead the red square to the exit, which is marked by a green square.</string>  
 <string name="rules\_list\_rule\_3">The game has four levels of difficulty, the difficulty depends on the size of the maze. To fully pass the selected difficulty, the player must pass three mazes in a row. After passing the selected difficulty, the player can save his result.</string>  
 <string name="rules\_list\_rule\_4">To view achievements, click the «Achievements» button on the main menu.</string>  
 <string name="success">Level successfully completed!</string>  
 <string name="label\_passing\_time">Passing time, sec.</string>  
 <string name="background\_contentDescription" translatable="false">Background</string>  
 <string name="logo\_contentDescription" translatable="false">Logo</string>  
 <string name="choose\_level">Choose a difficulty of the level</string>  
 <string name="seconds"> sec. </string>  
 <string name="label\_passing\_date">Passing date</string>  
 <string name="label\_level">Difficulty of the level</string>  
 <string name="invalid\_query\_uri">Incorrect request</string>  
 <string name="insert\_failed">Insertion error</string>  
</resources>

Код ресурса локализации strings.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string name="app\_name">Лабиринты</string>  
 <string name="play">Играть</string>  
 <string name="saveAndPlay">Сохранить и играть</string>  
 <string name="achievements">Достижения</string>  
 <string name="rules">Правила иры</string>  
 <string name="difficulty\_easy">Лёгкая</string>  
 <string name="difficulty\_normal">Средняя</string>  
 <string name="difficulty\_hard">Сложная</string>  
 <string name="difficulty\_very\_hard">Трудная</string>  
 <string name="success">Уровень успешно пройден!</string>  
 <string name="label\_passing\_time">Время прохождения, сек.</string>  
 <string name="choose\_level">Выберите сложность лабиринта</string>  
 <string name="seconds">сек.</string>  
 <string name="label\_passing\_date">Дата прохождения</string>  
 <string name="label\_level">Сложность лабиринта</string>  
 <string name="invalid\_query\_uri">Неверный запрос</string>  
 <string name="insert\_failed">Ошибка вставки</string>  
 <string name="rules\_list\_rule\_1">Игра «Лабиринты» - это классический лабиринт, в котором игроку необходимо найти выход и пройти лабиринт.</string>  
 <string name="rules\_list\_rule\_2">Для прохождения лабиринта игроку необходимо провести красный квадрат к выходу, который отмечен зелёным квадратом.</string>  
 <string name="rules\_list\_rule\_3">В игре есть четыре уровня сложности, от сложности зависит размер лабиринта. Для полного прохождения выбранной сложности игроку необходимо пройти подряд три лабиринта. После прохождения выбранной сложности игрок может сохранить свой результат.</string>  
 <string name="rules\_list\_rule\_4">Посмотреть свои достижения можно нажав на кнопку «Достижения» в главном меню.</string>  
</resources>

# ПРИЛОЖЕНИЕ Б

Программный код приложения

Основной пакет приложения содержит следующие классы:

Код класса AchievementActivity.java

package com.example.maze;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.widget.Toolbar;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.FragmentTransaction;  
import java.io.Serializable;  
  
public class AchievementActivity extends AppCompatActivity implements AchievementFragment.AchivementsFragmentListener,  
 DetailFragment.DetailFragmentListener,  
 Serializable {  
  
 //Ключ для сохранения Uri достижения в переданном объекте Bundle  
 public static final String *ACHIEVEMENT\_URI* = "achivement\_uri";  
 private AchievementFragment achievementFragment; //Вывод списка достижений  
  
 private void setSupportActionBar(Toolbar toolbar) {  
 }  
  
 //кнопка "назад" - возврат в MainActivity  
 public boolean onOptionsItemSelected(MenuItem item){  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 return true;  
 }  
 catch (Exception e){  
 return false;  
 }  
 }  
  
 //Отображает AchievementsFragment при первой загрузке MainActivity  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_achivements*);  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
  
 //Создание AchivementsFragment  
 achievementFragment = new AchievementFragment();  
  
 //Добавление фрагмента в FrameLayot  
 FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  
 transaction.add(R.id.*fragmentContainer*, achievementFragment);  
 transaction.commit(); //Вывод AchivementsFragment  
 }  
  
 //Отображение DetailFragment для выбранного достижения  
 @Override  
 public void onAchievementSelected(Uri achivementsUri) {  
 displayAchivement(achivementsUri, R.id.*fragmentContainer*);  
 }  
  
 @Override  
 public void onAddAchievement() {  
  
 }  
  
 //Отображение информации о достижении  
 private void displayAchivement(Uri achievementsUri, int viewID) {  
 DetailFragment detailFragment = new DetailFragment();  
  
 //Передача Uri достижения в аргументе DetailFragment  
 Bundle arguments = new Bundle();  
 arguments.putParcelable(*ACHIEVEMENT\_URI*, achievementsUri);  
 detailFragment.setArguments(arguments);  
  
 //Использование FragmentTransaction для отображения  
 FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  
 transaction.replace(viewID, detailFragment);  
 transaction.addToBackStack(null);  
 transaction.commit(); //Приводит к отображению DetailFragment  
 }  
}

Код класса AchievementAdapter.java

package com.example.maze;  
  
import android.database.Cursor;  
import android.net.Uri;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.recyclerview.widget.RecyclerView;  
import com.example.maze.data.DatabaseDescription.\*;  
  
public class AchievementAdapter extends RecyclerView.Adapter<AchievementAdapter.ViewHolder> {  
  
 //Переменные экземпляров AchievementsAdapter  
 private Cursor cursor = null;  
 private final AchievementClickListener clickListener;  
  
 public AchievementAdapter(AchievementClickListener clickListener) {  
 this.clickListener = clickListener;  
 }  
  
 //Интерфейс реализуется AchievementsFragment для обработки прикосновения к элементу в списке RecyclerView  
 public interface AchievementClickListener {  
 void onClick(Uri achievementsUri);  
 }  
  
 //Вложенный субкласс RecyclerView.ViewHolder используется для реализации  
 //паттерна View-Holder в контексте RecyclerView  
 public class ViewHolder extends RecyclerView.ViewHolder {  
 public final TextView textView;  
 private long rowID;  
  
 //Найстройка объекта ViewHolder элемента RecyclerView  
 public ViewHolder(View itemView) {  
 super(itemView);  
 textView = (TextView) itemView.findViewById(android.R.id.text1);  
 //Присоединение слушаетля к itemView  
 itemView.setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 clickListener.onClick(Achievement.buildAchievementUri(rowID));  
 }  
 }  
 );  
 }  
  
 //Идентификатор записи базы данных для достижения в ViewHolder  
 public void setRowID(long rowID) {  
 this.rowID = rowID;  
 }  
 }  
  
 //Подготовка нового элемента списка и его объекта ViewHolder  
 @Override  
 public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 //заполнение макета android.R.layout.simple\_list\_item\_1  
 View view = LayoutInflater.from(parent.getContext()).inflate(android.R.layout.simple\_list\_item\_1, parent, false);  
 return new ViewHolder(view); //ViewHolder текущего элемента  
 }  
  
 //Назначает текст элемента списка  
 @Override  
 public void onBindViewHolder(ViewHolder holder, int position) {  
 cursor.moveToPosition(position);  
 int index = cursor.getColumnIndex(Achievement.\_ID);  
 holder.setRowID(cursor.getLong(index));  
 int text = cursor.getColumnIndex(Achievement.COLUMN\_PASSING\_DATE);  
 holder.textView.setText(cursor.getString(text));  
 }  
  
 //Возвращает количество элементов, предоставляемых адаптером  
 @Override  
 public int getItemCount() {  
 return (cursor != null) ? cursor.getCount() : 0;  
 }  
  
 //Текущий объект Cursor адаптера заменяется новым  
 public void swapCursor(Cursor cursor) {  
 this.cursor = cursor;  
 notifyDataSetChanged();  
 }  
}

Код класса AchievementFragment.java

package com.example.maze;  
  
import android.content.Context;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import androidx.fragment.app.Fragment;  
import androidx.loader.app.LoaderManager;  
import androidx.loader.content.CursorLoader;  
import androidx.loader.content.Loader;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import com.example.maze.data.DatabaseDescription.Achievement;  
  
public class AchievementFragment extends Fragment implements LoaderManager.LoaderCallbacks<Cursor> {  
  
 //Идентификатор Loader  
 private static final int ACHIVEMENTS\_LOADER = 0;  
  
 //Сообщает MainActivity о выборе достижения  
 private AchivementsFragmentListener listener;  
  
 //Адаптер для recyclerView  
 private AchievementAdapter achievementAdapter;  
  
 @Override  
 public View onCreateView(  
 LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState  
 ) {  
 super.onCreateView(inflater, container, savedInstanceState);  
 setHasOptionsMenu(true); //У фрагмента есть команды меню  
  
 //Заполнение GUI и получение ссылки на RecyclerView  
 View view = inflater.inflate(R.layout.fragment\_achivements, container, false);  
 RecyclerView recyclerView = (RecyclerView) view.findViewById(R.id.recyclerViewAchivements);  
  
 //recyclerView выводит элементы в вертикальном списке  
 recyclerView.setLayoutManager(new LinearLayoutManager(getActivity().getBaseContext()));  
  
 //создание адаптера recyclerView и слушателя щелчков на элементах  
 achievementAdapter = new AchievementAdapter(new AchievementAdapter.AchievementClickListener() {  
 @Override  
 public void onClick(Uri achivementsUri) {  
 listener.onAchievementSelected(achivementsUri);  
 }  
 }  
 );  
  
 recyclerView.setAdapter(achievementAdapter); //Назначение адаптера  
  
 //Присоединение ItemDecorator для вывода разделителей  
 recyclerView.addItemDecoration(new ItemDivider(getContext()));  
  
 //Улучшает быстродействие, если размер макета RecyclerView не изменяется  
 recyclerView.setHasFixedSize(true);  
  
 return view;  
 }  
  
 //Присваивание AchivementsFragment при присоединении фрагмента  
 @Override  
 public void onAttach(Context context) {  
 super.onAttach(context);  
 listener = (AchivementsFragmentListener) context;  
 }  
  
 //Удаление AchivementsFragment при отсоединении фрагмента  
 @Override  
 public void onDetach() {  
 super.onDetach();  
 listener = null;  
 }  
  
 //Инициализация Loader при создании активности этого фрагмента  
 @Override  
 public void onActivityCreated(Bundle savedInstanceState) {  
 super.onActivityCreated(savedInstanceState);  
 getLoaderManager().initLoader(ACHIVEMENTS\_LOADER, null, this);  
 }  
  
 //Вызывается из MainActivity при обновлении базы данных другим фрагментом  
 public void updateAchivementsList() {  
 achievementAdapter.notifyDataSetChanged();  
 }  
  
 //Вызывается LoaderManager для создания Loader  
 @Override  
 public Loader<Cursor> onCreateLoader(int id, Bundle args) {  
 //Созданиек CursorLoader на основании аргумента id;  
 //в этом фрагменте только один объект Loader, и команда switch не нужна  
 switch (id) {  
 case ACHIVEMENTS\_LOADER:  
 return new CursorLoader(getActivity(),  
 Achievement.CONTENT\_URI, //Uri таблицы booking  
 null,  
 null,  
 null,  
 Achievement.COLUMN\_PASSING\_DATE + " COLLATE NOCASE ASC "  
 ); //сортировка  
 default:  
 return null;  
 }  
 }  
  
 //Вызывается LoaderManager при завершении загрузки  
 @Override  
 public void onLoadFinished(Loader<Cursor> loader, Cursor data) {  
 achievementAdapter.swapCursor(data);  
 }  
  
 //Вызывается LoaderManager при сбросе Loader  
 @Override  
 public void onLoaderReset(Loader<Cursor> loader) {  
 achievementAdapter.swapCursor(null);  
 }  
  
 //Метод обратного вызова, реализуемый MainActivity  
 public interface AchivementsFragmentListener {  
 //Вызывается при выборе достижения  
 void onAchievementSelected(Uri achivementsUri);  
  
 //Вызывается при нажатии кнопки добавления  
 void onAddAchievement();  
 }  
}

Код класса DetailFragment.java

package com.example.maze;  
  
import android.content.Context;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.fragment.app.Fragment;  
import androidx.loader.app.LoaderManager;  
import androidx.loader.content.CursorLoader;  
import androidx.loader.content.Loader;  
import com.example.maze.data.DatabaseDescription.\*;  
  
public class DetailFragment extends Fragment implements LoaderManager.LoaderCallbacks<Cursor> {  
  
 private static final int ACHIVEMENTS\_LOADER = 0; //Идентифицирует Loader  
 private DetailFragmentListener listener; //MainActivity  
 private Uri achivementsUri; //Uri выбранного достижения  
  
 private TextView passingDateTextView;  
 private TextView levelTextView;  
 private TextView passingTimeTextView;  
  
 //Назначение DetailFragmentListener при присоединении фрагмента  
 @Override  
 public void onAttach(Context context) {  
 super.onAttach(context);  
 listener = (DetailFragmentListener) context;  
 }  
  
 //Удаление DetailFragmentListener при отсоединении фрагмента  
 @Override  
 public void onDetach() {  
 super.onDetach();  
 listener = null;  
 }  
  
 //Методы обратного вызова, реализованные MainActivity  
 public interface DetailFragmentListener {  
 }  
  
 //Вызывается при создании представлений фрагмента  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 super.onCreateView(inflater, container, savedInstanceState);  
 //setHasOptionsMenu(true); //У фрагмента есть команды меню  
  
 //Получение объекта Bundle с аргументами и извлечение Uri  
 Bundle arguments = getArguments();  
 if (arguments != null)  
 achivementsUri = arguments.getParcelable(AchievementActivity.ACHIEVEMENT\_URI);  
  
 //Заполнение макета DetailFragment  
 View view = inflater.inflate(R.layout.fragment\_details, container, false);  
  
 //Получение компонентов EditText  
 passingDateTextView = (TextView) view.findViewById(R.id.passingDateTextView);  
 levelTextView = (TextView) view.findViewById(R.id.levelTextView);  
 passingTimeTextView = (TextView) view.findViewById(R.id.passingTimeTextView);  
  
 //Загрузка достижения  
 getLoaderManager().initLoader(ACHIVEMENTS\_LOADER, null, this);  
 return view;  
 }  
  
 // Вызывается LoaderManager для создания Loader  
 @Override  
 public Loader<Cursor> onCreateLoader(int id, Bundle args) {  
 // Создание CursorLoader на основании аргумента id; в этом  
 // фрагменте только один объект Loader, и команда switch не нужна  
 CursorLoader cursorLoader;  
 switch (id) {  
 case ACHIVEMENTS\_LOADER:  
 cursorLoader = new CursorLoader(getActivity(),  
 achivementsUri, //Uri отображаемого достижения  
 null, //Все столбцы  
 null, // Все записи  
 null, //Без аргументов  
 null); //Порядок сортировки  
 break;  
 default:  
 cursorLoader = null;  
 break;  
 }  
 return cursorLoader;  
 }  
  
 //Вызывается LoaderManager при завершении загрузки  
 @Override  
 public void onLoadFinished(Loader<Cursor> loader, Cursor data) {  
 //Если достижение существует в бд, вывести его информацию  
 if (data != null && data.moveToFirst()) {  
 //Полоучение индекса столбца для каждого элемента данных  
 int passingDateIndex = data.getColumnIndex(Achievement.COLUMN\_PASSING\_DATE);  
 int levelIndex = data.getColumnIndex(Achievement.COLUMN\_DIFFICULTY\_LEVEL);  
 int passingTimeIndex = data.getColumnIndex(String.valueOf(Achievement.COLUMN\_PASSING\_TIME));  
  
 //Заполненение TextView полученными данными  
 passingDateTextView.setText(data.getString(passingDateIndex));  
 levelTextView.setText(data.getString(levelIndex));  
 passingTimeTextView.setText(data.getString(passingTimeIndex));  
 }  
 }  
  
 //Вызывается LoaderManager при сбросе Loader  
 @Override  
 public void onLoaderReset(Loader<Cursor> loader) {  
 }  
}

Код класса Game.java

package com.example.maze;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.GestureDetector;  
import android.view.MenuItem;  
import android.view.MotionEvent;  
import android.widget.Toolbar;  
  
import androidx.appcompat.app.AppCompatActivity;  
import com.example.maze.game.GameManager;  
import com.example.maze.game.MazeView;  
  
public class Game extends AppCompatActivity {  
  
 private final String LOG\_TAG = "Game\_logs";  
 private MazeView view;  
 private GestureDetector gestureDetector;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 int gameLevel = getIntent().getIntExtra("level", 15);  
 String gameLevelString = getIntent().getStringExtra("levelString");  
  
 GameManager gameManager = new GameManager(this, gameLevel, gameLevelString);  
 view = new MazeView(this, gameManager);  
 setContentView(view);  
 gestureDetector = new GestureDetector(this, gameManager);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
 }  
  
 private void setSupportActionBar(Toolbar toolbar) {  
 }  
  
 //кнопка "назад" - возврат в MainActivity  
 public boolean onOptionsItemSelected(MenuItem item) {  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 return true;  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 return false;  
 }  
 }  
  
 //Системаня кнопка назад (работает также)  
 @Override  
 public void onBackPressed() {  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
  
 @Override  
 public boolean onTouchEvent(MotionEvent event) {  
 return gestureDetector.onTouchEvent(event);  
 }  
}

Код класса GameLevels.java

package com.example.maze;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toolbar;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class GameLevels extends AppCompatActivity {  
  
 private final String LOG\_TAG = "GameLevels\_logs";  
 private int level = 0;  
 private int[] levels = {15, 20, 30, 40};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_gamelevels);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
  
 //Кнопка уровень простой  
 Button buttonEasy = (Button) findViewById(R.id.buttonEasy);  
 buttonEasy.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentGame = new Intent(getApplicationContext(), Game.class);  
 level = levels[0];  
 intentGame.putExtra("level", level);  
 intentGame.putExtra("levelString", getApplicationContext().getResources().getString(R.string.difficulty\_easy));  
  
 startActivity(intentGame);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
  
 //Кнопка уровень стандартный  
 Button buttonNormal = (Button) findViewById(R.id.buttonNormal);  
 buttonNormal.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentGame = new Intent(getApplicationContext(), Game.class);  
 level = levels[1];  
 intentGame.putExtra("level", level);  
 intentGame.putExtra("levelString", getApplicationContext().getResources().getString(R.string.difficulty\_normal));  
  
 startActivity(intentGame);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
  
 //Кнопка уровень сложный  
 Button buttonHard = (Button) findViewById(R.id.buttonHard);  
 buttonHard.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentGame = new Intent(getApplicationContext(), Game.class);  
 level = levels[2];  
 intentGame.putExtra("level", level);  
 intentGame.putExtra("levelString", getApplicationContext().getResources().getString(R.string.difficulty\_hard));  
  
 startActivity(intentGame);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
  
 //Кнопка уровень очень сложный  
 Button buttonVeryHard = (Button) findViewById(R.id.buttonVeryHard);  
 buttonVeryHard.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentGame = new Intent(getApplicationContext(), Game.class);  
 level = levels[3];  
 intentGame.putExtra("level", level);  
 intentGame.putExtra("levelString", getApplicationContext().getResources().getString(R.string.difficulty\_very\_hard));  
  
 startActivity(intentGame);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
 }  
  
 private void setSupportActionBar(Toolbar toolbar) {  
 }  
  
 //кнопка "назад" - возврат в MainActivity  
 public boolean onOptionsItemSelected(MenuItem item) {  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 return true;  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 return false;  
 }  
 }  
  
 //Системаня кнопка назад (работает также)  
 @Override  
 public void onBackPressed() {  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
}

Код класса ItemDivider.java

package com.example.maze;  
  
import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.drawable.Drawable;  
import android.view.View;  
import androidx.recyclerview.widget.RecyclerView;  
  
public class ItemDivider extends RecyclerView.ItemDecoration {  
 private final Drawable divider;  
  
 //Конструктор загружает встроенный разделитель элементов списка  
 public ItemDivider(Context context) {  
 int[] attrs = {android.R.attr.listDivider};  
 divider = context.obtainStyledAttributes(attrs).getDrawable(0);  
 }  
  
 //Рисование разделителей элементов списка в RecyclerView  
 @Override  
 public void onDrawOver(Canvas c, RecyclerView parent, RecyclerView.State state) {  
 super.onDrawOver(c, parent, state);  
  
 //Вычисление координат x для всех разделителей  
 int left = parent.getPaddingLeft();  
 int right = parent.getWidth() - parent.getPaddingRight();  
  
 //Для каждого элемента, кроме последнего, нарисовать линию  
 for (int i = 0; i < parent.getChildCount() - 1; ++i) {  
 View item = parent.getChildAt(i); //Получить i-тый элемент списка  
  
 //Вычисление координат y (игрек) текущего разделителя  
 int top = item.getBottom() + ((RecyclerView.LayoutParams) item.getLayoutParams()).bottomMargin;  
 int bottom = top + divider.getIntrinsicHeight();  
  
 //Рисование разделителя с вычисленными границами  
 divider.setBounds(left, top, right, bottom);  
 divider.draw(c);  
 }  
 }  
}

Код класса MainActivity.java

package com.example.maze;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
  
 private final String LOG\_TAG = "MainActivity\_logs";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 //Кнопка правила  
 Button buttonRules = (Button)findViewById(R.id.buttonRules);  
 buttonRules.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentRules = new Intent(MainActivity.this, Rules.class);  
 startActivity(intentRules);  
 finish();  
 }catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
  
 //Кнопка играть  
 Button buttonPlay = (Button)findViewById(R.id.buttonPlay);  
 buttonPlay.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentPlay = new Intent(MainActivity.this, GameLevels.class);  
 startActivity(intentPlay);  
 finish();  
 }catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
  
 //Кнопка достижения  
 Button buttonAchivements = (Button)findViewById(R.id.buttonAchievements);  
 buttonAchivements.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentAchievements = new Intent(MainActivity.this, AchievementActivity.class);  
 startActivity(intentAchievements);  
 finish();  
 }catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
 }  
}

Код класса Rules.java

package com.example.maze;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.widget.Toolbar;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class Rules extends AppCompatActivity {  
  
 private final String LOG\_TAG = "Rules\_logs";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_rules*);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
 }  
  
 private void setSupportActionBar(Toolbar toolbar) {  
 }  
  
 //кнопка "назад" - возврат в MainActivity  
 public boolean onOptionsItemSelected(MenuItem item){  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 return true;  
 }  
 catch (Exception e){  
 Log.*v*(LOG\_TAG, e.toString());  
 return false;  
 }  
 }  
  
 //Системаня кнопка назад (работает также)  
 @Override  
 public void onBackPressed(){  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 }  
 catch (Exception e){  
 Log.*v*(LOG\_TAG, e.toString());  
 }  
 }  
}

Код класса Rules.java

package com.example.maze;  
  
import android.content.ContentValues;  
import android.content.Intent;  
import android.content.UriMatcher;  
import android.net.Uri;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toolbar;  
  
import android.database.sqlite.SQLiteDatabase;  
import androidx.appcompat.app.AppCompatActivity;  
import com.example.maze.data.AchievementDatabaseHelper;  
import com.example.maze.data.DatabaseDescription;  
  
public class Success extends AppCompatActivity {  
  
 private final String LOG\_TAG = "Success\_logs";  
 private AchievementDatabaseHelper dbHelper;  
 private String passingDate;  
 private String level;  
 private Integer passingTime;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_success);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 getSupportActionBar().setDisplayShowHomeEnabled(true);  
  
 passingTime = getIntent().getIntExtra("time", 1);  
 level = getIntent().getStringExtra("level");  
 passingDate = getIntent().getStringExtra("date");  
  
 TextView resultTime = findViewById(R.id.textViewPassingTimeValue);  
 resultTime.setText(String.valueOf(passingTime) + " " + getApplicationContext().getResources().getString(R.string.seconds));  
  
 //Кнопка играть  
 Button buttonPlay = (Button) findViewById(R.id.buttonPlay);  
 buttonPlay.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 Intent intentPlay = new Intent(Success.this, GameLevels.class);  
 startActivity(intentPlay);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
  
 //объект для работы с БД  
 dbHelper = new AchievementDatabaseHelper(this);  
 //Кнопка сохранить и играть  
 Button buttonSaveAndPlay = (Button) findViewById(R.id.buttonSaveAndPlay);  
 buttonSaveAndPlay.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
  
 ContentValues contentValues = new ContentValues();  
 contentValues.put("passing\_date", passingDate);  
 contentValues.put("difficulty\_level", level);  
 contentValues.put("passing\_time", passingTime);  
  
 //вставляем запись и получаем ее ID  
 long rowID = db.insert(DatabaseDescription.Achievement.TABLE\_NAME, null, contentValues);  
 Log.v(LOG\_TAG, "row inserted, ID = " + rowID);  
 dbHelper.close();  
  
 try {  
 Intent intentPlay = new Intent(Success.this, GameLevels.class);  
 startActivity(intentPlay);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
 });  
 }  
  
 private void setSupportActionBar(Toolbar toolbar) {  
 }  
  
 //кнопка "назад" - возврат в MainActivity  
 public boolean onOptionsItemSelected(MenuItem item) {  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 return true;  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 return false;  
 }  
 }  
  
 //Системаня кнопка назад (работает также)  
 @Override  
 public void onBackPressed() {  
 try {  
 Intent mainIntent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivityForResult(mainIntent, 0);  
 finish();  
 } catch (Exception e) {  
 Log.v(LOG\_TAG, e.toString());  
 }  
 }  
}

Пакет приложения «data» содержит следующие классы:

Код класса AchievementContentProvider.java

package com.example.maze.data;  
  
import android.content.ContentProvider;  
import android.content.ContentValues;  
import android.content.UriMatcher;  
import android.database.Cursor;  
import android.database.SQLException;  
import android.database.sqlite.SQLiteQueryBuilder;  
import android.net.Uri;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import com.example.maze.R;  
import com.example.maze.data.DatabaseDescription.Achievement;  
  
public class AchievementContentProvider extends ContentProvider {  
  
 //Используется для обращения к базе данных  
 private AchievementDatabaseHelper dbHelper;  
  
 //UriMatcher помогает ContentProvider определить выполняемую операцию  
 private static final UriMatcher uriMatcher = new UriMatcher(UriMatcher.NO\_MATCH);  
  
 //Константы, используемые для определения выполняемой операции  
 private static final int ONE\_ACHIEVEMENT = 1; //одно достижение  
 private static final int ACHIEVEMENTS = 2; //таблица достижений  
  
 //Статический блок для настройки UriMatcher объекта ContentProvider  
 static {  
 //Uri для достижения с заданным идентификатором  
 uriMatcher.addURI(DatabaseDescription.AUTHORY, Achievement.TABLE\_NAME + "/#", ONE\_ACHIEVEMENT);  
 //Uri для таблицы  
 uriMatcher.addURI(DatabaseDescription.AUTHORY, Achievement.TABLE\_NAME, ACHIEVEMENTS);  
 }  
  
 //Вызывается при создании AchievementContentProvider  
 @Override  
 public boolean onCreate() {  
 //Создание объекта AchievementDatabaseHelper  
 dbHelper = new AchievementDatabaseHelper(getContext());  
 return true; //Объект AchievementDatabaseHelper создан успешно  
 }  
  
 @Override  
 public String getType(Uri uri) {  
 return null;  
 }  
  
 //Получение информации из базы данных  
 @Override  
 public Cursor query(Uri uri, String[] projection, String selection,  
 String[] selectionArgs, String sortOrder) {  
  
 //Создание SQLiteQueryBuilder для запроса к таблице achievements  
 SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();  
 queryBuilder.setTables(Achievement.TABLE\_NAME);  
  
 switch (uriMatcher.match(uri)) {  
 case ONE\_ACHIEVEMENT: //Выбрать достижение с заданным идентификатором  
 queryBuilder.appendWhere(  
 Achievement.\_ID + "=" + uri.getLastPathSegment());  
 break;  
 case ACHIEVEMENTS: //Выбрать все достижения  
 break;  
 default:  
 throw new UnsupportedOperationException(  
 getContext().getString(R.string.invalid\_query\_uri) + uri);  
 }  
 //Выполнить запрос для получения одного или всех достижений  
 Cursor cursor = queryBuilder.query(dbHelper.getReadableDatabase(),  
 projection, selection, selectionArgs, null, null, sortOrder);  
 cursor.setNotificationUri(getContext().getContentResolver(), uri);  
  
 return cursor;  
 }  
  
 //Вставка нового достижения в бд  
 @Override  
 public Uri insert(Uri uri, ContentValues values) {  
 Uri newAchievementUri = null;  
 switch (uriMatcher.match(uri)) {  
 case ACHIEVEMENTS:  
 // При успехе возвращается идентификатор записи нового достижения  
 long rowId = dbHelper.getWritableDatabase().insert(  
 Achievement.TABLE\_NAME, null, values);  
 //Если достижение было добавлено, создать подходящий Uri; иначе выдать исключение  
 if (rowId > 0) {  
 newAchievementUri = Achievement.buildAchievementUri(rowId);  
 //Оповестить наблюдателей об изменениях в базе данных  
 getContext().getContentResolver().notifyChange(uri, null);  
 } else  
 throw new SQLException(  
 getContext().getString(R.string.insert\_failed) + uri);  
 break;  
 default:  
 throw new UnsupportedOperationException(  
 getContext().getString(R.string.insert\_failed) + uri);  
 }  
 return newAchievementUri;  
 }  
  
 @Override  
 public int delete(Uri uri, String selection, String[] selectionArgs) {  
 return 0;  
 }  
  
 @Override  
 public int update(@NonNull Uri uri, @Nullable ContentValues values, @Nullable String selection, @Nullable String[] selectionArgs) {  
 return 0;  
 }  
}

Код класса AchievementDatabaseHelper.java

package com.example.maze.data;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import com.example.maze.data.DatabaseDescription.Achievement;  
  
public class AchievementDatabaseHelper extends SQLiteOpenHelper {  
  
 private static final String DATABASE\_NAME = "Achievements.db";  
 private static final int DATABASE\_VERSION = 1;  
  
 //Конструктор  
 public AchievementDatabaseHelper(Context context) {  
 super(context, DATABASE\_NAME, null, DATABASE\_VERSION);  
 }  
  
 //Создание таблицы achievements при создании базы данных  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 //Команда SQL для создания таблицы achievements  
 final String CREATE\_ACHIEVEMENT\_TABLE =  
 "CREATE TABLE " + Achievement.TABLE\_NAME + "(" +  
 Achievement.\_ID + " integer primary key, " +  
 Achievement.COLUMN\_PASSING\_DATE + " TEXT, " +  
 Achievement.COLUMN\_DIFFICULTY\_LEVEL + " TEXT, " +  
 Achievement.COLUMN\_PASSING\_TIME + " TEXT);";  
 db.execSQL(CREATE\_ACHIEVEMENT\_TABLE); // Создание таблицы achievements  
 }  
  
 //Определяет способ обновления при изменения схемы бд  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 }  
}

Код класса DatabaseDescription.java

package com.example.maze.data;  
  
import android.content.ContentUris;  
import android.net.Uri;  
import android.provider.BaseColumns;  
  
public class DatabaseDescription {  
  
 //Имя ContentProvider обычно совпадает с именем пакета  
 public static final String AUTHORY = "com.example.maze.data";  
  
 //Базовый URI для взаимодействия с ContentProvider  
 private static final Uri BASE\_CONTENT\_UTI = Uri.parse("content://" + AUTHORY);  
  
 public static final class Achievement implements BaseColumns {  
 public static final String TABLE\_NAME = "achievement"; //Имя таблицы  
  
 //Объект Uri для таблицы achievements  
 public static final Uri CONTENT\_URI = BASE\_CONTENT\_UTI.buildUpon()  
 .appendPath(TABLE\_NAME).build();  
  
 //Имена столбцов таблиц  
 public static final String COLUMN\_PASSING\_DATE = "passing\_date";  
 public static final String COLUMN\_DIFFICULTY\_LEVEL = "difficulty\_level";  
 public static final String COLUMN\_PASSING\_TIME = "passing\_time";  
  
 //Создание Uri для конкретного достижения  
 public static Uri buildAchievementUri(long id) {  
 return ContentUris.withAppendedId(CONTENT\_URI, id);  
 }  
 }  
}

Пакет приложения «game» содержит следующие классы:

Код класса Dot.java

package com.example.maze.game;  
  
import android.graphics.Canvas;  
import android.graphics.Paint;  
import android.graphics.Point;  
import android.graphics.Rect;  
  
public class Dot implements Drawable {  
 private int size;  
 protected Point point;  
 protected Paint paint;  
  
 public Dot(Point point, Paint paint, int size) {  
 this.size = size;  
 this.point = point;  
 this.paint = paint;  
 }  
  
 public Point getPoint() {  
 return point;  
 }  
  
 @Override  
 public void draw(Canvas canvas, Rect rect) {  
 float cellSize = (float) (rect.right - rect.left) / size;  
 canvas.drawRect(  
 rect.left + point.x \* cellSize,  
 rect.top + point.y \* cellSize,  
 rect.left + point.x \* cellSize + cellSize,  
 rect.top + point.y \* cellSize + cellSize,  
 paint);  
 }  
}

Код класса Drawable.java

package com.example.maze.game;  
  
import android.graphics.Canvas;  
import android.graphics.Rect;  
  
public interface Drawable {  
 void draw(Canvas canvas, Rect rect);  
}

Код класса Exit.java

package com.example.maze.game;  
  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Point;  
  
public class Exit extends Dot {  
 public Exit(Point point, int size) {  
 super(point, getPaint(), size);  
 }  
  
 public static Paint getPaint(){  
 Paint paint = new Paint(Paint.ANTI\_ALIAS\_FLAG);  
 paint.setColor(Color.GREEN);  
 return paint;  
 }  
}

Код класса GameManager.java

package com.example.maze.game;  
  
import android.content.Intent;  
import android.graphics.Canvas;  
import android.graphics.Rect;  
import android.view.GestureDetector;  
import android.view.MotionEvent;  
import android.view.View;  
import com.example.maze.Game;  
import com.example.maze.Success;  
  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
import java.util.Locale;  
  
public class GameManager extends GestureDetector.SimpleOnGestureListener {  
 private List<Drawable> drawables = new ArrayList<>();  
 private View view;  
 private Exit exit;  
 private Player player;  
 private Maze maze;  
 private Rect rect = new Rect();  
 private int mazeSize;  
 private int screenSize = 0;  
  
 private Game gameActivity;  
 private Date timeStart;  
 private Date timeEnd;  
  
 private int levelComplete;  
 private String gameLevelString;  
  
 public GameManager(Game game, int mazeSize, String gameLevel) {  
 createGameManager(game, mazeSize);  
 levelComplete = 0;  
 timeStart = new Date();  
 this.gameLevelString=gameLevel;  
 }  
  
 private void createGameManager(Game game, int mazeSize) {  
 drawables.clear();  
  
 this.mazeSize = mazeSize;  
 maze = new Maze(mazeSize);  
 drawables.add(maze);  
  
 player = new Player(maze.getStart(), mazeSize);  
 drawables.add(player);  
  
 exit = new Exit(maze.getEnd(), mazeSize);  
 drawables.add(exit);  
  
 gameActivity = game;  
 }  
  
 @Override  
 public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {  
 int diffX = 0, diffY = 0;  
 diffX = Math.round(e2.getX() - e1.getX());  
 diffY = Math.round(e2.getY() - e1.getY());  
 if (Math.abs(diffX) > Math.abs(diffY)) {  
 diffX = diffX > 0 ? 1 : -1;  
 diffY = 0;  
 } else {  
 diffX = 0;  
 diffY = diffY > 0 ? 1 : -1;  
 }  
 int stepX = player.getX();  
 int stepY = player.getY();  
  
 while (maze.canPlayerMove(stepX + diffX, stepY + diffY)) {  
 stepX += diffX;  
 stepY += diffY;  
 if (diffX != 0) {  
 if (maze.canPlayerMove(stepX, stepY + 1) || maze.canPlayerMove(stepX, stepY - 1)) {  
 break;  
 }  
 }  
 if (diffY != 0) {  
 if (maze.canPlayerMove(stepX + 1, stepY) || maze.canPlayerMove(stepX - 1, stepY)) {  
 break;  
 }  
 }  
 }  
 player.moveTo(stepX, stepY);  
  
 //Уровень пройден  
 if (exit.getPoint().equals(player.getPoint())) {  
  
 if (++levelComplete != 3)  
 createGameManager(gameActivity, mazeSize);  
 else {  
 timeEnd = new Date();  
 int time = (int) ((timeEnd.getTime() - timeStart.getTime()) / 1000); //время в секундах  
  
 String date = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault()).format(new Date());  
  
 Intent successIntent = new Intent(gameActivity.getApplicationContext(), Success.class);  
 successIntent.putExtra("time", time);  
 successIntent.putExtra("level", gameLevelString);  
 successIntent.putExtra("date", date);  
 gameActivity.startActivityForResult(successIntent, 0);  
 gameActivity.finish();  
 }  
 }  
 view.invalidate(); //перерисовка  
 return super.onFling(e1, e2, velocityX, velocityY);  
 }  
  
 public void draw(Canvas canvas) {  
 for (Drawable drawableItem :  
 drawables) {  
 drawableItem.draw(canvas, rect);  
 }  
 }  
  
 public void setView(View view) {  
 this.view = view;  
 }  
  
 public void setScreenSize(int width, int height) {  
 screenSize = Math.min(width, height);  
 rect.set((width - screenSize) / 2,  
 (height - screenSize) / 2,  
 (width + screenSize) / 2,  
 (height + screenSize) / 2);  
 }  
}

Код класса Maze.java

package com.example.maze.game;  
  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Point;  
import android.graphics.Rect;  
  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Random;  
import java.util.Stack;  
  
public class Maze implements Drawable {  
  
 private Paint wallPaint;  
 private final boolean[][] array;  
 private final int size;  
 private final Point end = new Point(1, 1);  
 private int bestScore = 0;  
 private Point start;  
  
 public Maze(int size) {  
 wallPaint = new Paint(Paint.*ANTI\_ALIAS\_FLAG*);  
 wallPaint.setColor(Color.*parseColor*("#7f6791"));  
 this.size = size;  
 array = new boolean[size][size];  
 generateMaze();  
 }  
  
 public Point getStart() {  
 return start;  
 }  
 public Point getEnd() {  
 return end;  
 }  
  
 private void generateMaze() {  
 for (int i = 0; i < size; ++i) {  
 for (int j = 0; j < size; ++j) {  
 array[i][j] = i % 2 != 0 && j % 2 != 0 && i < size - 1 && j < size - 1;  
  
 }  
 }  
 Random random = new Random();  
 Stack<Point> stack = new Stack<>();  
 stack.push(end);  
 while (stack.size() > 0) {  
 Point current = stack.peek();  
 List<Point> unusedNeighbors = new LinkedList<>();  
 //проверка левого соседа  
 if (current.x > 2) {  
 if (!isUsedCell(current.x - 2, current.y)) {  
 unusedNeighbors.add(new Point(current.x - 2, current.y));  
 }  
 }  
 //проверка верхнего соседа  
 if (current.y > 2) {  
 if (!isUsedCell(current.x, current.y - 2)) {  
 unusedNeighbors.add(new Point(current.x, current.y - 2));  
 }  
 }  
 //проверка правого соседа  
 if (current.x < size - 2) {  
 if (!isUsedCell(current.x + 2, current.y)) {  
 unusedNeighbors.add(new Point(current.x + 2, current.y));  
 }  
 }  
 //проверка нижнего соседа  
 if (current.y < size - 2) {  
 if (!isUsedCell(current.x, current.y + 2)) {  
 unusedNeighbors.add(new Point(current.x, current.y + 2));  
 }  
 }  
 if (unusedNeighbors.size() > 0) {  
 int rnd = random.nextInt(unusedNeighbors.size());  
 Point direction = unusedNeighbors.get(rnd);  
 int diffX = (direction.x - current.x) / 2;  
 int diffY = (direction.y - current.y) / 2;  
 array[current.y + diffY][current.x + diffX] = true;  
 stack.push(direction);  
 } else {  
 if (bestScore < stack.size()) {  
 bestScore = stack.size();  
 start = current;  
 }  
 stack.pop();  
 }  
 }  
 }  
  
 public boolean isIntersection(int x, int y) {  
 return isUsedCell(x, y);  
 }  
  
 public boolean canPlayerMove(int x, int y) {  
 return array[y][x];  
 }  
  
 // Проверка, не стена ли клетка, и есть ли к ней доступ (вокруг неё не все стены)  
 private boolean isUsedCell(int x, int y) {  
 if (x < 0 || y < 0 || x >= size - 1 || y >= size - 1) {  
 return true;  
 }  
 return array[y - 1][x] //left  
 || array[y][x - 1] //top  
 || array[y + 1][x] //right  
 || array[y][x + 1]; //bottom  
 }  
  
 @Override  
 public void draw(Canvas canvas, Rect rect) {  
  
 float cellSize = (float) (rect.right - rect.left) / size;  
 for (int i = 0; i < size; ++i) {  
 for (int j = 0; j < size; ++j) {  
 if (!array[i][j]) {  
 float left = j \* cellSize + rect.left;  
 float top = i \* cellSize + rect.top;  
 canvas.drawRect(left, top, left + cellSize, top + cellSize, wallPaint);  
 }  
 }  
 }  
 }  
}

Код класса MazeView.java

package com.example.maze.game;  
  
import android.content.Context;  
import android.graphics.Canvas;  
import android.view.View;  
  
public class MazeView extends View {  
 private GameManager gameManager;  
  
 public MazeView(Context context, GameManager gameManager) {  
 super(context);  
 this.gameManager = gameManager;  
 gameManager.setView(this);  
 }  
  
 @Override  
 protected void onDraw(Canvas canvas) {  
 gameManager.draw(canvas);  
 }  
  
 @Override  
 protected void onSizeChanged(int w, int h, int oldw, int oldh) {  
 super.onSizeChanged(w, h, oldw, oldh);  
 gameManager.setScreenSize(w, h);  
 }  
}

Код класса Player.java

package com.example.maze.game;  
  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Point;  
  
public class Player extends Dot {  
  
 public Player(Point start, int size) {  
 super(start, getPaint(), size);  
 }  
  
 public static Paint getPaint(){  
 Paint paint = new Paint(Paint.ANTI\_ALIAS\_FLAG);  
 paint.setColor(Color.RED);  
 return paint;  
 }  
  
 public void moveTo(int x, int y) {  
 point.x = x;  
 point.y = y;  
 }  
  
 public int getX(){  
 return point.x;  
 }  
 public int getY(){  
 return point.y;  
 }  
  
}