**КУРСОВАЯ РАБОТА**

**По дисциплине:** /МДК.01.03 Разработка мобильных приложений

**Специальность:** 09.02.07 Информационные системы и программирование

**Тема:** «Разработка мобильного приложения «Сапёр»

**Группа:** ИСП311, **Курс:** 3

**Выполнил студент** \_\_\_\_\_\_\_\_\_\_ /Прохоркина Н.С./

подпись ФИО

**Руководитель** \_\_\_\_\_\_\_\_\_\_\_/Коротков Д.В/

подпись ФИО

**Оценка** \_\_\_\_\_\_\_\_\_\_

**Дата** \_\_\_\_\_\_\_\_\_\_

Москва, 2023

ОГЛАВЛЕНИЕ

[**ВВЕДЕНИЕ** 3](#_Toc135001699)

[**1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 5](#_Toc135001700)

[**1.1 Рынок мобильных игр** 5](#_Toc135001701)

[**1.2 Использование ООП в игре "Сапер"** 5](#_Toc135001702)

[**1.3 Телефоны Android** 5](#_Toc135001703)

[**1.4 Проектирование приложения** 5](#_Toc135001704)

[**2. ПРАКТИЧЕСКАЯ ЧАСТЬ** 7](#_Toc135001715)

[**2.1 Создание проекта в Android Studio** 7](#_Toc135001716)

[**2.2 Создание пользовательского интерфейса** 7](#_Toc135001717)

[**2.3 Реализация логики игры** 9](#_Toc135001718)

[**2.4 Реализация логики игры** 10](#_Toc135001718)

[**2.5 Реализация логики игры** 11](#_Toc135001718)

[**2.6 Реализация логики игры** 13](#_Toc135001718)

[**2.7 Реализация логики игры** 16](#_Toc135001718)

[**3. СПИСОК ЛИТЕРАТУРЫ** 23](#_Toc135001715)

# **ВВЕДЕНИЕ**

В настоящее время мобильные приложения являются неотъемлемой частью нашей жизни. Большинство людей используют мобильные приложения каждый день для различных целей: коммуникации, общения в социальных сетях, чтения новостей, музыки и развлечений. Одним из самых популярных типов мобильных приложений являются игры. В данной курсовой работе рассматривается разработка игры "Сапер" на языке программирования Java в среде разработки Android Studio.

Цель данной курсовой работы - создание полнофункционального приложения игры "Сапер". В ходе работы будут рассмотрены основные элементы разработки мобильного приложения на примере игры "Сапер". Будут изучены концепции программирования, разработки пользовательского интерфейса и обработки пользовательского ввода в контексте мобильных приложений.

Для разработки игры "Сапер" была выбрана среда разработки Android Studio, которая является одной из самых популярных и мощных сред разработки мобильных приложений. Android Studio предоставляет широкий спектр инструментов для разработки, от создания пользовательского интерфейса до отладки приложения.

Язык программирования, используемый в этой курсовой работе, - Java. Java является одним из самых популярных языков программирования в мире и широко используется для разработки мобильных приложений. Он обладает множеством преимуществ, таких как простота синтаксиса, высокая скорость выполнения кода и широкое сообщество разработчиков.

Игра "Сапер" была выбрана для разработки в этой курсовой работе, так как она является одной из самых популярных игр в мире и обладает интересной логикой. Игра представляет собой головоломку, где игрок должен расставить флажки на клетках поля, которые скрывают в себе мину или же открыть клетки, которые не скрывают в себе мину. Цель игры - открыть все клетки, которые не скрывают в себе мину, и не открыть ни одной клетки, которая содержит мину. Игра "Сапер" является интересной и захватывающей и может быть развивающей для мозга.

**Цель:** Создание мобильного приложения «Сапёр» на платформе Android Studio.

**Задачи:**

1. Составить техническое задание к курсовой работе;
2. Поэтапно спланировать выполнение практической части работы;
3. Обозначить примерные временные рамки для выполнения каждого этапа практической части;
4. Выбрать язык и среду разработки;
5. Изучить необходимую литературу для корректного написания программы;
6. Написать необходимую документацию для курсовой работы.

# **1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1 Рынок мобильных игр**

Мобильные игры являются одним из самых быстрорастущих сегментов рынка развлечений. В 2020 году, по данным статистического портала Statista, мировой рынок мобильных игр оценивался в 77,2 миллиарда долларов США. В России, по данным компании App Annie, доходы от мобильных игр в 2020 году составили 113 миллионов долларов США. Эти данные показывают, что мобильные игры являются очень перспективным направлением для разработчиков.

**1.2 Использование ООП в игре "Сапер"**

ООП (объектно-ориентированное программирование) - это подход к программированию, который основан на объектах, которые включают в себя данные и методы для работы с этими данными. В игре "Сапер" можно использовать ООП для создания объектов, таких как игровое поле, клетки поля, минные поля и т.д. Каждый объект будет содержать свои собственные данные и методы для обработки данных.

Например, класс "Клетка поля" может содержать следующие свойства: координаты клетки, содержимое клетки (минное поле или число мин вокруг клетки) и флаг, указывающий, была ли клетка открыта. Методы класса могут быть использованы для обработки событий, таких как открытие клетки или установка флага на клетку.

Использование ООП в игре "Сапер" позволяет создавать более гибкий и модульный код, который легче поддерживать и расширять. Кроме того, ООП позволяет лучше организовать код и разбить его на более мелкие и понятные части.

**1.3 Телефоны Android**

Android - это операционная система для мобильных устройств, которая разрабатывается компанией Google. Android является одной из самых популярных операционных систем для мобильных устройств, и на ней работает большое количество мобильных приложений, в том числе и игр.

Для разработки игр для Android используется язык программирования Java и среда разработки Android Studio. Android Studio предоставляет широкий спектр инструментов для разработки мобильных приложений, включая инструменты для создания пользовательского интерфейса.

**1.4 Проектирование приложения**

Предложение в задумке представляло из себя классическую игру «Сапёр», в которую были бы добавлены какие-либо новые функции, в данном случае введена функция сохранения результатов. Также планировалось изменить интерфейс.

# **2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1 Создание проекта в Android Studio**

Для начала разработки проекта "Сапер" в Android Studio необходимо создать новый проект. Для этого нужно выбрать пункт "Start a new Android Studio project" на стартовом экране.

Затем нужно указать настройки проекта, такие как название приложения, пакет приложения, версию Android SDK и другие. После указания настроек можно перейти к созданию основных элементов пользовательского интерфейса и реализации логики игры "Сапер".

**2.2 Создание пользовательского интерфейса**

Для создания пользовательского интерфейса приложения "Сапер" необходимо использовать Android XML-разметку. Это позволяет создавать интерфейс, используя графические элементы, такие как кнопки, текстовые поля и изображения.

Для создания основного экрана игры можно использовать LinearLayout или RelativeLayout. Для отображения игрового поля можно использовать GridView, который позволяет создавать таблицы с элементами. В каждой ячейке таблицы можно разместить кнопку, которая будет отображать состояние клетки поля.

Для моего приложения нужно всего три Layout:

1. Главное меню приложения, на котором можно выбрать уровень сложности и посмотреть, за какое время были сыграны предыдущие партии.

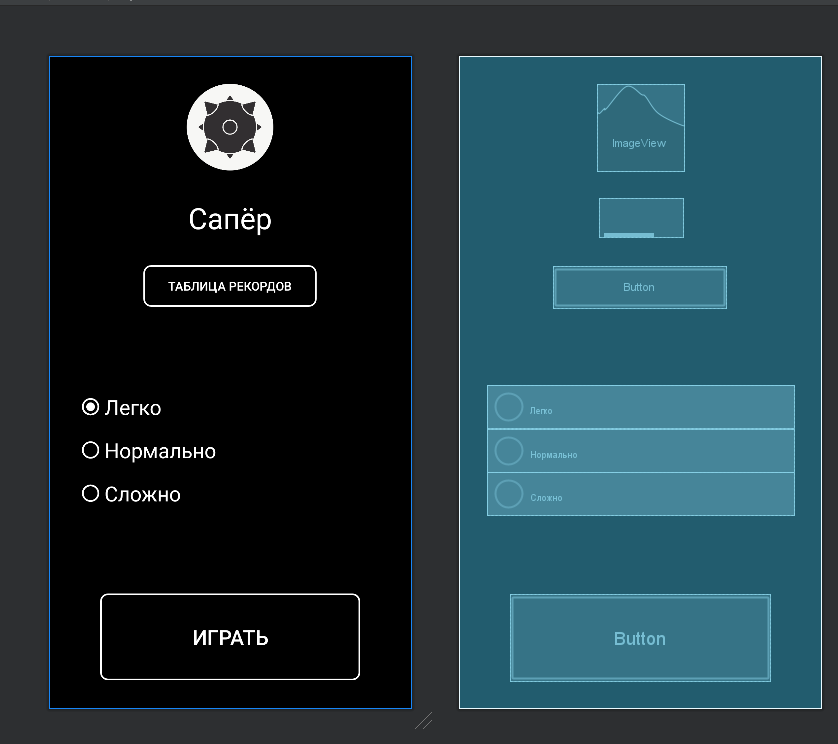


Рис.1 – Главное меню

1. На этом Layout находится поле игры, таймер, кнопки для перезапуска игры и для возвращения в главное меню.

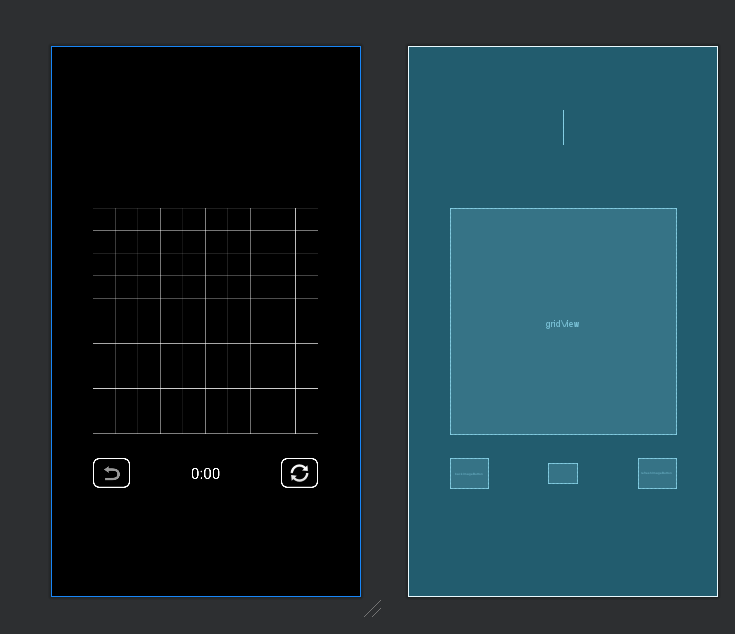


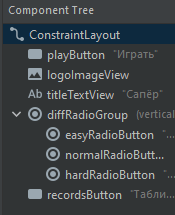
Рис.2 – Игровое поле

1. Этот интерфейс будет выводиться при победе. Здесь будет указано время, за которое вы прошли игру.



Рис.3 – Конец игры

View, которые добавлены в интерфейс.



**2.3 Реализация логики игры**

Для реализации логики игры "Сапер" необходимо создать Java-класс, который будет управлять игровым процессом. В этом классе можно определить методы для инициализации игры, обработки нажатий на клетки поля, определения победы или поражения.

Класс также должен содержать методы для создания игрового поля и расстановки мин на поле. Можно использовать двумерный массив для представления поля, где каждая ячейка массива будет представлять одну клетку поля.

Каждая кнопка, которая представляет клетку поля, должна иметь свой собственный обработчик нажатий. При нажатии на кнопку нужно определить ее положение на поле и обработать событие в зависимости от ее содержимого. Если клетка содержит мину, игрок проигрывает. Если клетка не содержит мину, нужно отобразить число мин вокруг клетки или открыть все соседние пустые клетки.

**2.4 Класс GridView.**

Этот код определяет пользовательский класс View, который реализует сетку (grid view) на экране. Он имеет четыре конструктора, которые вызывают метод init(), который инициализирует объект Paint. Метод onDraw() используется для отрисовки сетки на экране, используя метод drawLine() объекта Canvas. Метод setDimension() устанавливает количество столбцов и строк в сетке.

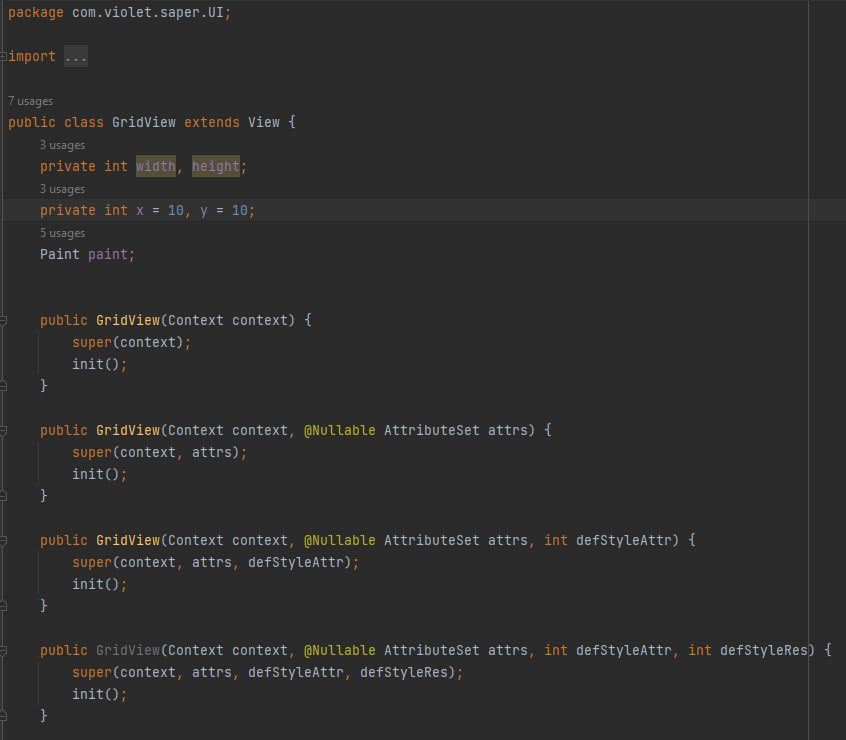
****

Рис.4.1 – Сетка

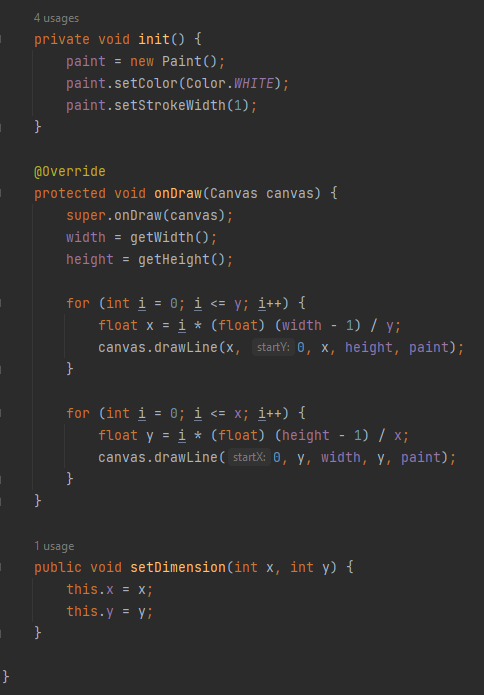
****

Рис.4.2 – Сетка

**2.5 Класс MainActivity.**

Данный класс является процессом. В нем находятся основные функции, которые применяются в кнопках в интерфейсе игры, такие как возвращение в главное меню, смена сложности и другие.

Этот метод создает игровое поле, которое уже включает в себя различные компоненты, необходимые для структуры Сапёра.

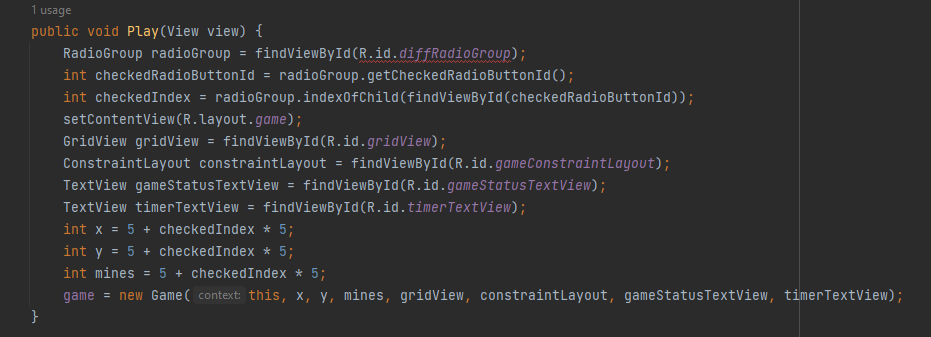


Рис 5 – Создание игрового поля

Эта функция позволяет заново начать игру, не выходя в главное меню.



Рис 6 – Рестарт игры

Данный метод даёт возможность вернуться в главное меню.



Рис 7 – Возвращение в главное меню

Этот код работает с кнопкой «Таблица рекордов».



Рис 8 – Кнопка таблицы рекордов

**2.6 Класс Field.**

В этом классе работа идет над игровым полем приложения.  
Данная функция используется для определения координаты клетки (данная функция пригодится чуть позже).

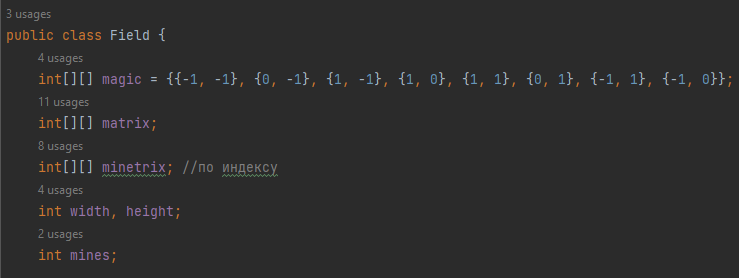


Рис 9 – Координаты клеток

В этом методе происходит очистка поля.

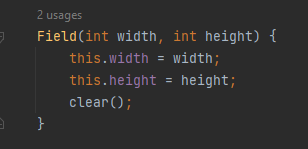


Рис 10 – Очистка поля

В данном методе производится случайное распределение мин на игровом поле.

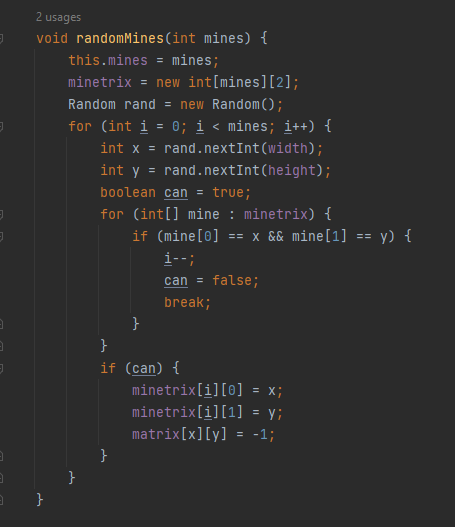


Рис 11 – Расставление бомб

Этот код находит все пустые ячейки на игровом поле и ячейки, которые окружены числами, начиная с заданных координат x и y. Результаты поиска возвращаются в виде списка массивов int[]. Метод getNulls использует рекурсию для поиска всех пустых ячеек, а метод getNullsWithOthers находит все пустые ячейки, окруженные числами, используя метод getNulls и проверяя ячейки вокруг каждой найденной пустой ячейки.

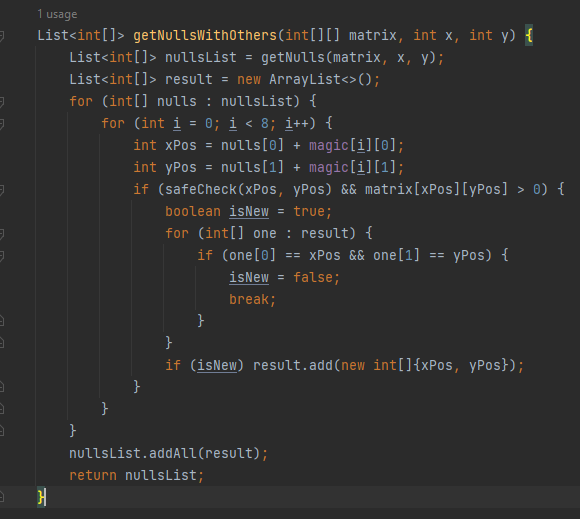


Рис 12 – Поиск пустых ячеек

Этот код ищет все пустые ячейки на игровом поле, начиная с заданных координат x и y. Он использует рекурсию для поиска всех пустых ячеек и возвращает результаты в виде списка массивов int[]. Код проверяет, что заданные координаты x и y находятся в пределах игрового поля и что ячейка пуста. Если это не так, то метод возвращает пустой список.

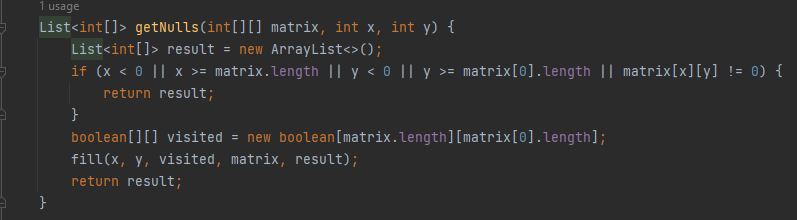


Рис 13 – Рекурсия

Этот метод возвращает результаты в виде списка массивов int[]. Рекурсивная функция fill проверяет, что заданные координаты находятся в пределах игрового поля, что ячейка пуста и что она еще не была посещена. Если это так, то функция добавляет координаты в список результатов и продолжает поиск в соседних ячейках, вызывая саму себя для каждой из них. Если какая-то из соседних ячеек не удовлетворяет условиям, то функция просто возвращает управление.

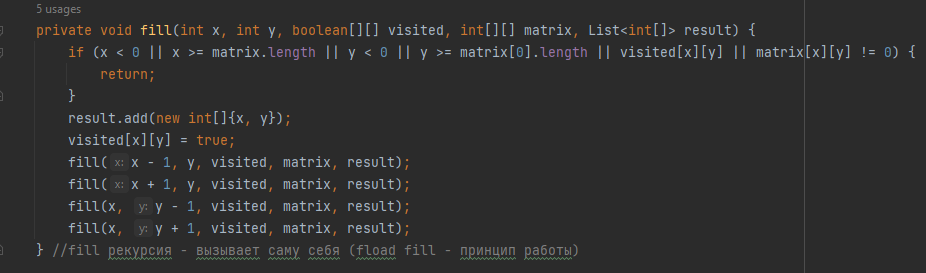


Рис 14 - Fill

Метод safeCheck проверяет, находятся ли координаты (x,y) в пределах игрового поля, где width и height - ширина и высота игрового поля соответственно. Если координаты находятся в пределах игрового поля, метод возвращает true, иначе - false.

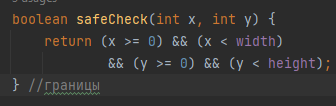


Рис 15 - Метод safeCheck

Эта функция добавляет мину в ячейку с координатами (x,y) на игровом поле, если эта ячейка не содержит уже мины (-1). Перед добавлением мины происходит проверка, находятся ли координаты в пределах игрового поля с помощью функции safeCheck. Если координаты находятся в пределах поля и ячейка не содержит мины, то значение в этой ячейке увеличивается на 1 (метод matrix[x][y]++).

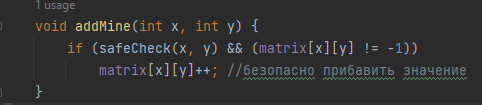


Рис 16 – Добавление мины

**2.7 Класс Game.**

Конструктор класса Game принимает объект Context в качестве параметра и устанавливает значение переменной context. Он также получает массив целых чисел с именем "records" из SharedPreferences и записывает его в переменную records класса Game.

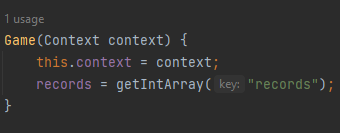
****

Рис 17 – Конструктор 1

Здесь создан еще один конструктор, в котором описан интерфейс игры.

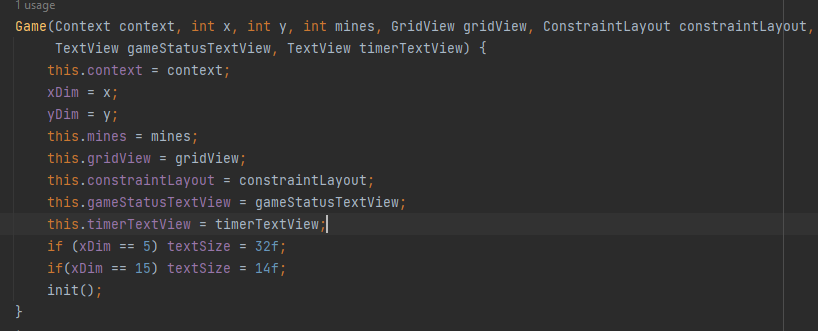


Рис 18 – Конструктор 2

Инициализация игрового поля и интерфейса.

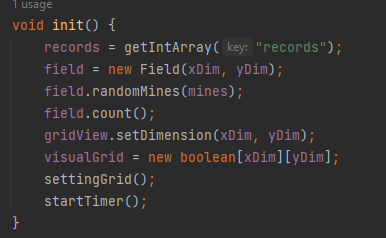


Рис 19 - Инициализация игрового поля и интерфейса

Этот метод работает с таймером, который засекает время прохождения игры.



Рис 20 - Таймер

По завершению игры результаты прохождения переносятся в «Таблицу рекордов».

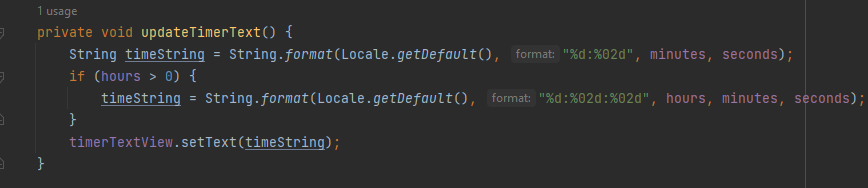


Рис 21 – Сохранение результатов

Функция перезапуска игры



Рис 22 – Перезапуск игры

Сохранение данных на память телефона в виде типа String.

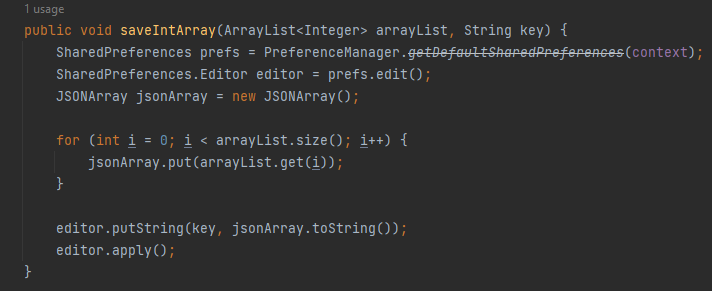


Рис 23 - Сохранение данных на память телефона

Получение данных с памяти телефона, которые ранее были сохранены.

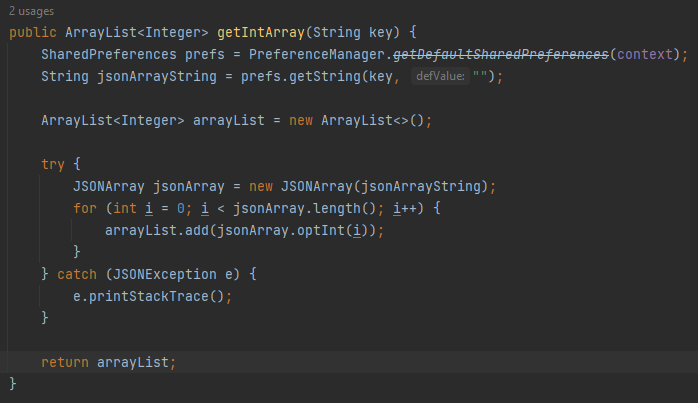


Рис 24 - Получение данных с памяти телефона

Этот код настраивает сетку для игрового поля, определяя ширину и высоту сетки, а также ее координаты на экране. Затем он вычисляет шаги по оси X и Y, необходимые для правильного размещения элементов на сетке.

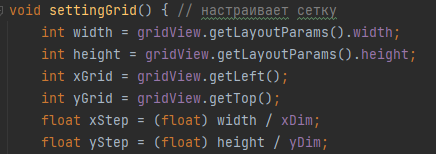


Рис 25 – Настройка сетки игрового поля

Этот код отслеживает касание пользователя на игровом поле и вычисляет координаты касания в соответствии с размерами сетки. Координаты X и Y делятся на ширину и высоту сетки соответственно, а затем умножаются на количество ячеек по соответствующей оси, чтобы определить позицию касания внутри сетки.

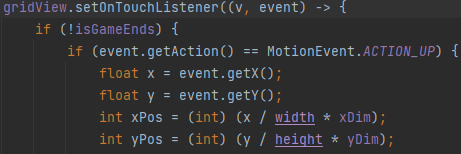


Рис 26 – Отслеживание касаний

Проверяет безопасность поля с помощью метода safeCheck(), передавая ему координаты xPos и yPos. Условие возвращает true, если проверка прошла успешно.  
Если ячейка visualGrid[xPos][yPos] равна false (т.е. не была ранее посещена), выполняется следующий блок кода.

Устанавливает значение true для ячейки visualGrid[xPos][yPos], чтобы отметить ее как посещенную.

Создает объект RelativeLayout.LayoutParams с размерами (int) xStep - 1 и (int) yStep - 1.

Проверяет значение ячейки field.matrix[xPos][yPos]. Если оно равно 0, выполняется следующий блок кода.

Вызывает метод getNullsWithOthers() объекта field, передавая ему матрицу field.matrix и координаты xPos и yPos. Метод возвращает список nulls, содержащий нулевые значения из матрицы field.matrix, вместе с другими значениями из окружающих ячеек.

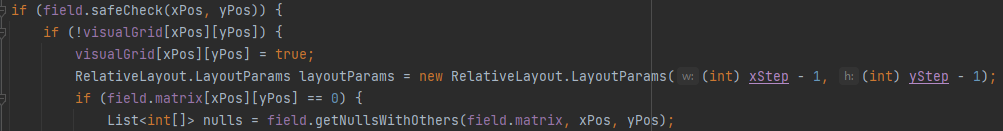


Рис 27 – Работа с игровым полем

Проходит по каждому элементу aNull в списке nulls с помощью цикла for-each.  
Извлекает значения xNull и yNull из текущего элемента aNull.  
Создает новый объект TextView с помощью конструктора, принимающего контекст context.

Устанавливает размер текста для textView с помощью метода setTextSize() и передает значение textSize.

Устанавливает полужирный шрифт для textView с помощью метода setTypeface() и передает null в качестве параметра для шрифта и константу Typeface.BOLD для полужирного стиля.

Осуществляет переключение на основе значения field.matrix[xNull][yNull] с помощью оператора switch.



Рис 28 – Работа с текстом, который будет на экране

Текст в виде числа добавляется на экран.

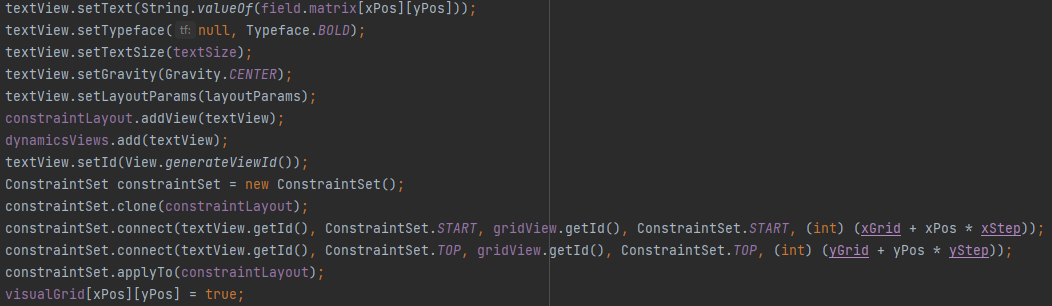


Рис 29 – Добавление текста на экран

Срабатывает при проигрыше.

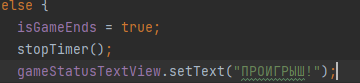


Рис 30 - Проигрыш

Этот фрагмент кода подсчитывает количество false значений в массиве visualGrid и сохраняет результат в переменной boolCounter.

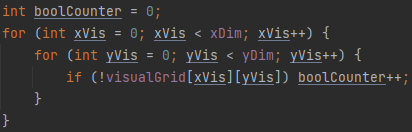


Рис 31 – Подсчет false значений и сохранение результов

Срабатывает при победе.

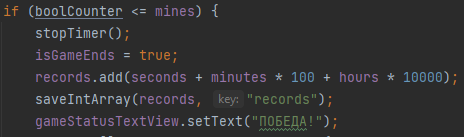


Рис 32 - Победа

# **3. СПИСОК ЛИТЕРАТУРЫ**

1. Авторы: Bill Phillips, Chris Stewart, Brian Hardy.

Android Programming: The Big Nerd Ranch Guide" (3rd Edition) by Bill Phillips, Chris Stewart, and Kristin Marsicano.

Год издания: 2020.

1. Автор: Michael Burton.

"Android App Development For Dummies" (3rd Edition) by Michael Burton.

Год издания: 2015.

1. Автор: Herbert Schildt

"Java: The Complete Reference" (11th Edition).

Год издания: 2018.

1. Авторы: Kathy Sierra, Bert Bates.

"Object-Oriented Programming in Java" (2nd Edition).

Год издания: 2019.

1. Авторы: Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra.

"Head First Design Patterns" (2nd Edition).

Год издания: 2020.

1. Автор: Joshua Bloch.

"Effective Java" (3rd Edition).

Год издания: 2022.

1. Автор: Robert C. Martin.

"Clean Code: A Handbook of Agile Software Craftsmanship".

Год издания: 2008.

1. Автор: Steve McConnell.

"Code Complete" (2nd Edition).

Год издания: 2009.