СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Игра «Сапер» является одной из самых узнаваемых компьютерных игр, получившая свою известность благодаря простоте, увлекательности и глубине игрового процесса. Основная задача игрока – открыть все ячейки игрового поля, не попав на мину, используя подсказки в виде цифр, указывающих количество мин в соседних клетках.

Несмотря на кажущуюся простоту, разработка игры «Сапер» затрагивает ряд важных аспектов программирования, включая алгоритмы генерации случайных игровых полей, вычисления подсказок, обработку пользовательского ввода и реализацию интуитивно понятного графического интерфейса.

Цель курсовой работы: получение практических навыков программирования в области разработки игровых приложений под систему Аврора, путем создания реализованной и отлаженной игры «Сапер».

Задачи, необходимые для достижения поставленной цели:

1. Рассмотреть методы и алгоритмы программирования, подходящие для разработки программы.
2. Реализовать приложение.
3. Протестировать и отладить программу.

# ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

## Анализ существующих аналогов

Благ

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Введение

Составленное техническое задание по дисциплине «Разработка приложений на C++» является документом к курсовой работе, который отражает все этапы разработки программного продукта, а также процесс проектирования и выявление требований, предъявляемых конечному продукту.

### 1.1.1 Наименование программы

Название данного приложения «Сапер» будет напрямую связываться с темой курсовой работы «Игра “Сапер”». Сапер – это простая компьютерная игра, в которой игрок должен открыть все поля игровой области, избегая мин. Для этого у каждого игрового поля генерируется число – кол-во мин по соседству. На основе этих данных игрок должен определить какие клетки безопасны, а какие – нет. Каждую новую игру поле с минами генерируется случайным образом.

### 1.1.2 Краткая характеристика области применения программы

Программа является головоломкой для проведения досуга. Приложение будет полезно тем, кто хочет потренировать свой мозг с помощью мобильного устройства.

## 1.2 Основание для разработки

Основанием для разработки является курсовая работа по дисциплине Разработка приложений на C++», предусмотренная учебным планом направления подготовки 09.03.01 «Информатика и вычислительная техника» профиля «Цифровые комплексы, системы и сети».

## 1.3 Назначение разработки

Приложение предназначено как полезная игра для времяпрепровождения за мобильным устройством.

## 1.4 Требования, предъявляемые к программе

### 1.4.1 Требования к функциональным характеристикам программы

В приложении должны быть реализованы следующие функции:

* Начало новой игры со случайно сгенерированным полем
* Вывод чисел на открытых полях, соответствующих кол-ву соседних мин
* Открытие всех соседних пустых полей при нажатии на поле без мины.
* Вывод сообщений об выигрыше и проигрыше игрока
* Начало новой игры при проигрыше или выигрыше.

### 1.4.2 Требования к техническим средствам, используемым при работе программы

Персональный компьютер пользователя должен быть оснащён графическим адаптером, также должна быть установлена ОС Auora.

### 1.4.3 Требования к языкам программы и среде разработки программы

Для разработки используются языки программирования C++ и QML, в качестве среды разработки выступает Auora IDE, набор инструментов – Auora SDK.

## 1.5 Требования к программной документации

1. Пояснительная записка оформляется в соответствии с ЛНА РТУ МИРЭА.
2. Проектная документация, составленная в соответствии с ГОСТ.

В процессе создания приложения вся проделанная работа документируется, должны быть сохранены все детали разработки, а также трудности, с которыми пришлось столкнуться. Всё вышеперечисленное должно быть отражено в пояснительной записке, которая прилагается к работе.

## 1.6 Этапы разработки

28.04.2025 – Составление технического задания.

# 2 ОБЗОР СПОСОБОВ ОРГАНИЗАЦИИ ДАННЫХ И ОБОСНОВАНИЕ ВЫБОРА СТРУКТУРЫ ДАННЫХ ДЛЯ ЭФФЕКТИВНОГО ВЫПОЛНЕНИЯ ОПЕРАЦИЙ

Данные могут быть организованы различными способами. Тип структуры данных в программе оказывает большое влияние на ее производительность. Для того чтобы выбрать наиболее простой и эффективный способ организации данных в программе рассмотрим несколько типов структур данных.

## 2.1 Minesweeper

Minesweeper – модуль, хранящий все данные, выполняющий все функции игры, и обрабатывающий логику.

## 2.2 Kurs\_m

Модуль, отвечающий за внешний вид. В нем описано как должна выглядеть игра и обработчик сигналов модуля Minesweeper

## 2.6 Выбор структур данных.

В ходе написания программы в основном использовалось 3 типа – Qstring для описания текстовых значений, bool для логических и int для числовых.

Для хранения поля в памяти использовался двумерный массив типа int/

# 3 ОПИСАНИЕ ПРОГРАММЫ

## 3.1 Общие сведения

В ходе выполнения курсовой работы была создана игра «Сапер» с пользовательским графическим интерфейсом под ОС Аврора. В ней выполняются все условия, обозначенные в техническом задании, и содержатся все необходимые компоненты, инструменты для корректной работы

### 3.1.1 Наименование программы

Название программы: «Сапер» или на английском языке «Minesweeper». Оно отражает предназначение и главную функцию созданного приложения.

### 3.1.2 Программное обеспечение, необходимое для функционирования программы

Для корректного функционирования данного программного продукта необходимо, чтобы на персональном компьютере или ноутбуке пользователя была установлена ОС АВРОРА. Также требуется наличие графического адаптера, чтобы устройство могло справляться с обработкой отображения приложения. Другие требования к устройству пользователя не предусмотрены.

### 3.1.2 Язык программирования, на котором написана программа

Для написания программы были использованы языки программирования С++ и QML, за их доступность, понятность и высокую производительность

## 3.2 Функциональное назначение программы (классы решаемых задач и функциональные ограничения на применения)

Данная игра написана для полезного проведения досуга за электронным устройством. Функциональные цели приложения включают операции с начало игры при выигрыше/проигрыше, генерацию случайного поля при начале новой игры, полный функционал самой игры “сапер”

## 3.3 Описание логической структуры программы

В программе используется Объектно-ориентированный подход реализации алгоритмов для упрощения данного приложения. Для хранения данных о поле и состояния игры используются встроенные структуры данных, такие как двумерные массивы и примитивные типы данных. Исходный код программы представлен в Приложении А.

### 3.3.1 Алгоритмы, используемые в программе

Для написания программы необходимо подключить библиотеку «rand» для работы с генератором случайных чисел, работа с векторами осуществляется средствами библиотеки «QVector», для нормальной генерации поля необходимо подключить библиотеку «ctime».

Основными алгоритмами для работы данной игры являются генерация минного поля, проверка ячейки поля и открытие всех свободных клеток поля.

### 3.3.2 Структура программы с описанием функций составных частей и связей между ними

Для грамотного структурирования программы внутри объекта Minesweeper были созданы несколько функций, которые вызываются при разных действиях пользователя:

Reset() – сброс игры в начальное состояние

Reveal(int row, int col) – открытие значения клетки

generateBoard() – случайная генерация поля

getCellValue(int row, int col) – получение значения на вывод минного поля (сколько мин рядом)

reveaEmptyCells(int row, int col) – Открытие всех минных полей с 0 мин соседствующих с нажатой клеткой.

## 3.4 Вызов программы

Изначальное поле:

Открытие свободных полей:

Проигрыш:

Новая игра:

## 3.6 Входные данные

В качестве входных данных программа принимает нажатия пользователя

## 3.7 Выходные данные

Выходные данные – графическое отображение минного поля.

ЗАКЛЮЧЕНИЕ

На протяжении всего процесса проектирования и создания программного продукта были получены практические навыки в разработке приложений с графическим интерфейсом на C++ и QML

Успешно выполнены поставленные задачи: создание игры “сапер”, реализация игрового цикла, генерация случайного поля, начало игры с начала после проигрыша или выигрыша

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лозовский В.В. Алгоритмические основы обработки данных: учебное пособие / Лозовский В.В., Платонова О.В., Штрекер Е.Н. — М.: МИРЭА – Российский технологический университет, 2022. – 337 с.

2. Платонова О.В. Алгоритмические основы обработки данных: методические указания / Платонова О.В., Асадова Ю.С., Расулов М.М. — М.: МИРЭА – Российский технологический университет, 2022. — 73 с.

3. Белик А.Г. Алгоритмы и структуры данных: учебное пособие / А.Г. Белик, В.Н. Цыганенко. — Омск: ОмГТУ, 2022. — 104 с. — ISBN 978-5-8149-3498-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/343688 (дата обращения: 03.09.2024)

4. Павлов Л.А. Структуры и алгоритмы обработки данных / Л.А. Павлов, Н.В. Первова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 256 с. — ISBN 978-5-507-44105-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/207563 (дата обращения: 03.09.2024)

5. Пантелеев Е.Р. Алгоритмы и структуры данных: учебное пособие / Е.Р. Пантелеев, А.Л. Алыкова. — Иваново: ИГЭУ, 2018. — 142 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/154576 (дата обращения: 03.09.2024)

Приложение А

Код программы приведен в листингах 4.1 4.2 и 4.3