|  |  |
| --- | --- |
| *voenmeh* | Министерство науки и высшего образования РФ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | И |  | Информационные и управляющие системы |
|  |  | шифр |  | Наименование |
| Кафедра |  | И5 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | Наименование |
| Дисциплина |  | Программирование на языке высокого уровня | | |

КУРСОВАЯ РАБОТА

на тему

|  |
| --- |
| Объектно-ориентированная разработка |
| программ с графическим пользовательским |
| интерфейсом «снизу-вверх». |
| Игра «Судоку» |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И594 |
| Иванова А. А. | | | | | |
| Фамилия И.О. | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | |
| Ковалев Р.Е. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | |  |  |
| «\_\_\_\_\_» |  | | |  | 2021 г. |

САНКТ-ПЕТЕРБУРГ

2021г.

СОДЕРЖАНИЕ

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 3](#_Toc60427617)

[ВВЕДЕНИЕ 4](#_Toc60427618)

[1 Постановка задачи 5](#_Toc60427619)

[2 Правила игры 6](#_Toc60427620)

[3 Основные компоненты программы 7](#_Toc60427621)

[3.1 Функция «splashScreen» 8](#_Toc60427622)

[3.2 Функция «showMenu» 8](#_Toc60427623)

[3.3 Функция «backButton» 8](#_Toc60427624)

[3.4 Структура «CellData» 8](#_Toc60427625)

[3.5 Перечисляемый тип «StepType» 9](#_Toc60427626)

[3.6 Перечисляемый тип «LevelType» 9](#_Toc60427627)

[3.7 Перечисляемый тип «AreaType» 9](#_Toc60427628)

[3.8 Абстрактный класс «Menu» 10](#_Toc60427629)

[3.8.1 Класс «Rules» 10](#_Toc60427630)

[3.8.2 Класс «Control» 10](#_Toc60427631)

[3.8.3 Класс «Background» 10](#_Toc60427632)

[3.8.4 Класс «Level» 10](#_Toc60427633)

[3.8.5 Класс «Victory» 10](#_Toc60427634)

[3.9 Класс «Game» 11](#_Toc60427635)

[3.10 Класс «Sudoku» 11](#_Toc60427636)

[3.11 Класс «Area» 12](#_Toc60427637)

[4 Руководство пользователя 13](#_Toc60427638)

[ЗАКЛЮЧЕНИЕ 18](#_Toc60427639)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc60427640)

[ПРИЛОЖЕНИЕ А 20](#_Toc60427641)

# **ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

SDL(англ. Simple DirectMedia Layer)— это свободная кроссплатформенная мультимедийная [библиотека](https://ru.wikipedia.org/wiki/Библиотека_(программирование)), реализующая единый программный интерфейс к графической подсистеме, звуковым устройствам и средствам ввода для широкого спектра платформ [1].

SDL\_Delay – функция, которая позволяет установить задержу перед исполнением кода на определенное количество миллисекунд [1].

# **ВВЕДЕНИЕ**

В связи с развитием технологий резко возрос интерес к программированию. Оно играет большую роль во всех сферах человеческой жизни: существует множество программ для офисной работы; работы с графикой, видео и звуком; для решения различных вычислительных и многих других задач.

Еще одной интересной областью применения программирования является создание игр. Многие современные игры представляют собой платформу, реализующую работу с графикой, физикой игрового мира, взаимодействием объектов – очевидно, что создание игр является непростой задачей.

Но существуют также и мини-игры, не требующие глубокой разработки, но позволяющие весело провести свободное время. Классическим примером такой игры является «Судоку», реализация которого и будет рассмотрена в этой работе. Программа состоит из набора классов и функций, реализующих задумку игры.

## **1 Постановка задачи**

Целью курсовой работы является разработка компьютерной игры «Судоку».

На первом этапе необходимо ознакомится с поставленной задачей, сформулировать ограничения в реализации, выявить объекты, описать их свойства и поведение.

На втором этапе необходимо спроектировать реализацию игры, создать базовый каркас приложения [2], разработать дизайн приложения, подобрать основные цвета, шрифты, изображения.

На третьем этапе необходимо написать логику игры, каркас обработчиков событий на SDL[1].

На четвертом этапе необходимо организовать взаимодействие с пользователем на основе меню и реализовать все подпункты.

## **2 Правила игры**

В судоку играют на игровом поле, состоящем из 81 маленькой клетки. Внутри игрового поля находится 9 больших квадратов, каждый из которых состоит из 9 маленьких клеток. Каждая горизонтальная строка, вертикальный столбец и большой квадрат должны заполняться цифрами от 1 до 9, не повторяя других цифр в строке, столбце или квадрате. Цель игры – правильно заполнить игровое поле [3].

Управление производится с помощью мыши (для меню) и клавиатуры (для игрового процесса) с помощью клавиш управления курсором, соответствующих 4 направлениям движения [4].

## **3 Основные компоненты программы**

Экранная заставка при запуске игры реализуется с помощью функции «splashScreen» [4].

Главное меню реализовано с помощью функции «showMenu».

Кнопка возврата в главное меню реализуется с помощью функции «backButton».

Каждый из экранов пунктов меню реализован соответствующим классом. Каждый такой класс наследуется от интерфейса [5] «Menu» с методами «handleEvent» и «renderText»:

* «Rules»;
* «Control»;
* «Background»;
* «Level»;
* «Victory».

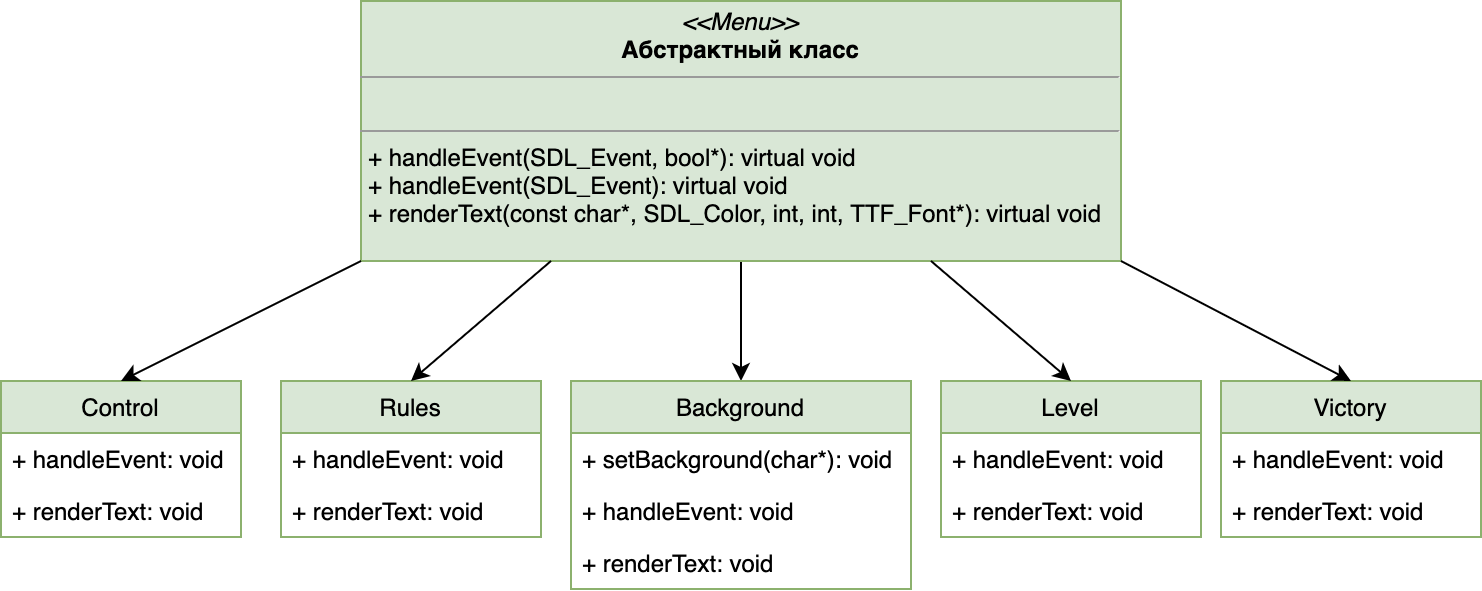
Иерархия представлена на рисунке 1.

Рисунок 1 - Иерархия классов, наследуемых от «Menu»

Для хранения информации о каждой клетке игрового поля используется структура «CellData».

Также используются следующие перечисляемые типы:

* «StepType»;
* «LevelType»;
* «AreaType».

В реализации игрового процесса программы также участвуют следующие классы:

* «Game»;
* «Sudoku»;
* «Area».

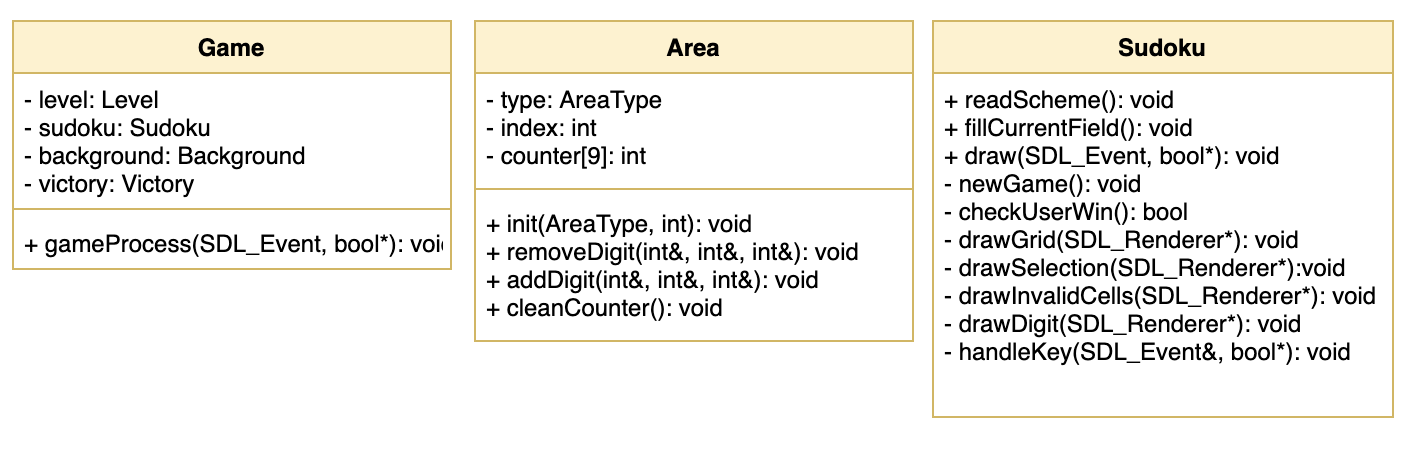
Схемы этих классов представлены на рисунке 2.

Рисунок 2 – Схемы классов «Game», «Sudoku», «Area»

### 3.1 Функция «splashScreen»

Не принимает входных параметров, ничего не возвращает. С помощью функции «renderText» создаются поверхности с текстом, накладываемые на основную поверхность. С помощь функции SDL\_Delay[1] происходит задержка заставки на экране пользователя.

### 3.2 Функция «showMenu»

Не принимает входных параметров, ничего не возвращает. С помощью функции «renderText» создаются поверхности с названиями пунктов меню. В зависимости от состояния пункта изменяется его текст.

Переход в пункт меню или завершение программы реализуется с помощью обработчика событий. Для выхода из пункта меню используется указатель, который в случае необходимости изменяется из пункта по ссылке.

### 3.3 Функция «backButton»

Не принимает входных параметров, ничего не возвращает. Создает поверхность с кнопкой возврата. Вызывается внутри функции «handleEvent», где отслеживается клик на кнопку возврата.

### 3.4 Структура «CellData»

Содержит поля:

* digit – тип int, текущая цифра в клетке (для пустой клетки – 0);
* validateCount – тип int, поле для проверки введенной цифры на валидность (если цифра введена неверно, ее значение превышает 0);
* noEdit – тип bool, с помощью этого поля устанавливается запрет на редактирование считанных из файла ячеек игрового поля.

Для инициализации игрового поля используется двумерный массив initialField, для хранения информации о текущем состоянии игрового поля – двумерный массив currentField.

### 3.5 Перечисляемый тип «StepType»

Отвечает за переключение между этапами игры.

Содержит идентификаторы:

* BACKGROUND – этап выбора фона;
* LEVEL – этап выбора уровня сложности;
* SUDOKU – этап игрового процесса;
* VICTORY – этап наступления победы.

### 3.6 Перечисляемый тип «LevelType»

Изменяет значение уровня сложности игры в зависимости от выбора пользователя.

Содержит идентификаторы:

* LOW – низкий уровень сложности;
* MIDDLE – средний уровень сложности;
* HIGH – высокий уровень сложности.

### 3.7 Перечисляемый тип «AreaType»

Определяет тип площади на игровом поле при отрисовке или удалении с него цифр.

Содержит идентификаторы:

* ROW – строка;
* COL – столбец;
* BOX – большой квадрат.

### 3.8 Абстрактный класс «Menu»

Используется конструктор по умолчанию.

Метод «renderText» принимает строку для рендеринга, цвет шрифта, координаты начальной точки отображения по х и у, размер шрифта; ничего не возвращает. Создает текстуру с текстом, переданным в нее в качестве параметра.

Метод «handleEvent» принимает событие для обработки, в одном из случаев – указатель на bool, с помощью которого осуществляется переход в главное меню. Обрабатывает событие, передаваемое из функции «showMenu».

#### 3.8.1 Класс «Rules»

Дополнительных полей и методов не содержит.

С помощью метода «renderText» добавляет текст в методе «handleEvent» (принимает 2 параметра).

Использует функцию «backButton», позволяющую вернуться в главное меню.

#### 3.8.2 Класс «Control»

Дополнительных полей и методов не содержит.

С помощью метода «renderText» добавляет текст в методе «handleEvent» (принимает 2 параметра).

Использует функцию «backButton», позволяющую вернуться в главное меню.

#### 3.8.3 Класс «Background»

Метод «handleEvent» (1 параметр) вызывает метод добавления текста «renderText», вызывает метод установки фона «setBackground», изменяет значение игрового этапа - значение поля «step» изменяется на «LEVEL».

Метод «setBackground» принимает в качестве входного параметра имя файла, задающего задний фон, и создает текстуру с выбранным изображением; ничего не возвращает.

#### 3.8.4 Класс «Level»

Метод «handleEvent» (1 параметр) вызывает метод добавления текста «renderText», изменяет значение игрового этапа - значение поля «step» изменяется на «SUDOKU», изменяет значение уровня сложности - значение поля «levelDifficulty» изменяется по выбору пользователя.

#### 3.8.5 Класс «Victory»

Дополнительных полей и методов не содержит.

С помощью метода «renderText» добавляет текст в методе «handleEvent» (2 параметра).

### 3.9 Класс «Game»

Содержит поля:

* background – экземпляр класса «Background»;
* level – экземпляр класса «Level»;
* sudoku – экземпляр класса «Sudoku»;
* victory – экземпляр класса «Victory».

Метод «gameProcess» принимает обрабатываемое событие, указатель на bool для перехода в главное меню; ничего не возвращает. Отслеживает текущий этап игры (выбор фона, выбор уровня сложности, игровой процесс, победа), осуществляет переключение между этапами.

### 3.10 Класс «Sudoku»

Класс содержит в себе конструктор, создающий основу игрового поля – текстуру игрового поля, текстуры с цифрами, проводит инициализацию столбцов, строк и больших клеток игрового поля.

Метод «newGame» не принимает параметров, ничего не возвращает. Обнуляет начальное и текущее игровое поле (этот метод необходим в случае рестарта).

Метод «readScheme» не принимает параметров, ничего не возвращает. Считывает из файла выбранную согласно уровню сложности схему судоку в двумерный массив initialField.

Метод «fillCurrentField» не принимает параметров, ничего не возвращает. Заполняет двумерный массив currentField цифрами из initialField, исключая некоторое количество цифр из каждой строки в зависимости от выбранного уровня сложности.

Метод «drawGrid» принимает указатель на поверхность; ничего не возвращает. Отрисовывает сетку игрового поля.

Метод «drawSelection» принимает указатель на поверхность; ничего не возвращает. Отрисовывает клетку, в которой находится пользователь.

Метод «drawInvalidCells» принимает указатель на поверхность; ничего не возвращает. Отрисовывает все клетки, противоречащие решению судоку.

Метод «drawDigit» принимает указатель на поверхность; ничего не возвращает. Отрисовывает цифры, записанные в двумерный массив currentField.

Метод «handleKey» принимает событие по ссылке, указатель на bool; ничего не возвращает. Отлавливает перемещение пользователя по игровому полю или выход пользователя в главное меню. Вызывает метод «checkUserWin».

Метод «checkUserWin» не принимает параметров, возвращает true в результате наступления победы, иначе – false. Производит проверку соответствия текущего игрового поля начальному полю, в случае полного соответствия изменяет значение поля «step» на «VICTORY».

Метод «draw» принимает обрабатываемое событие, указатель на bool; ничего не возвращает. Производит последовательную отрисовку игрового поля, вызывает методы «handleEvent», «drawGrid», «drawSelection», «drawInvalidCells», «drawDigit».

Использует функцию «backButton», позволяющую вернуться в главное меню.

### 3.11 Класс «Area»

Содержит поля:

* type – тип AreaType, обозначает тип поля;
* index – тип int, индекс текущего типа поля;
* counter – тип int, массив, содержащий количество повторяющихся цифр в данном типе поля для последующей проверки на валидность.

Метод «init» принимает тип площади и ее индекс; ничего не возвращает. Присваивает созданному экземпляру переданный в него тип и индекс.

Метод «addDigit» принимает по ссылке введенную цифру и ее координаты на игровом поле; ничего не возвращает. Добавляет введенную цифру в поле, отслеживает наличие неверно введенных цифр.

Метод «removeDigit» принимает по ссылке введенную цифру и ее координаты на игровом поле; ничего не возвращает. Удаляет цифру из поля.

Метод «cleanCounter» ничего не принимает, ничего не возвращает. Обнуляет массив повторяющихся цифр при рестарте из главного меню.

## **4 Руководство пользователя**

При запуске программы в течение 2 секунд показывается заставка, представленная на рисунке 3.



Рисунок 3 – Заставка, показываемая при запуске игры

После заставки появляется главное меню. Пункт меню изменяет цвет при наведении. Внешний вид меню представлен на рисунке 4.

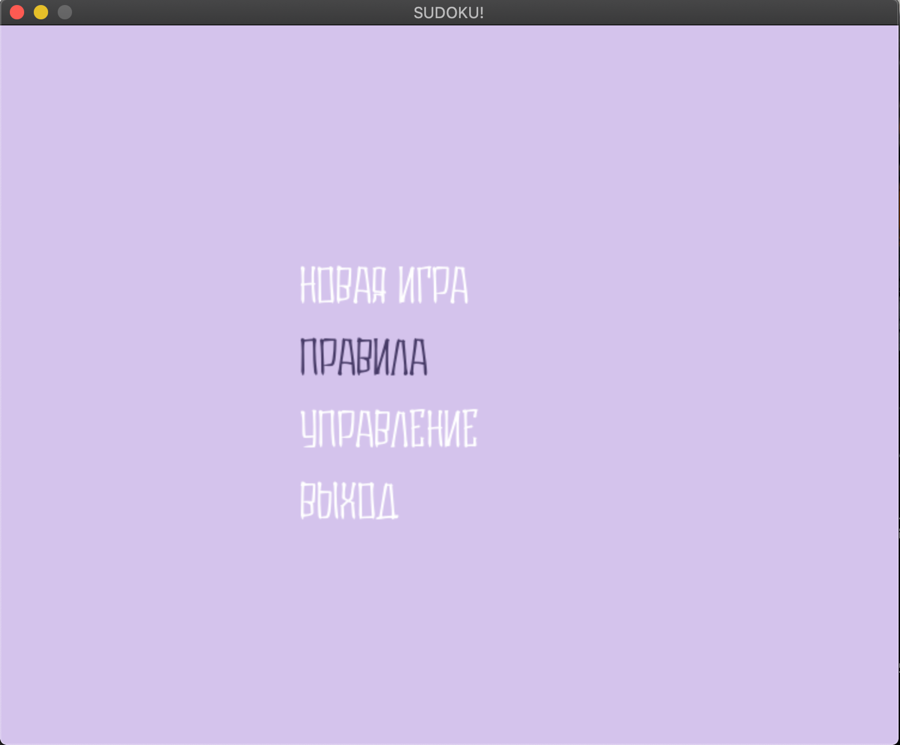


Рисунок 4 – Главное меню

При выборе пункта меню «Правила» пользователю предлагается ознакомиться с правилами игры. В левом верхнем углу расположена кнопка возврата в главное меню. На рисунке 5 представлен внешний вид этого раздела.

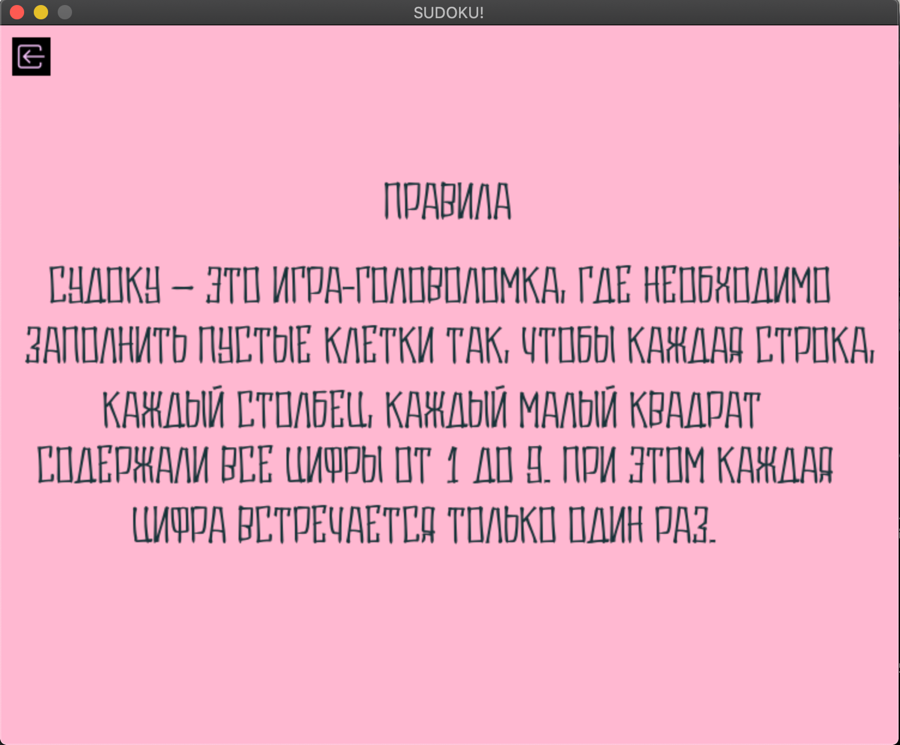


Рисунок 5 – Правила игры

При выборе пункта меню «Управление» пользователю предоставляется информация о клавишах управления. Рисунок 6 соответствует этому пункту меню.

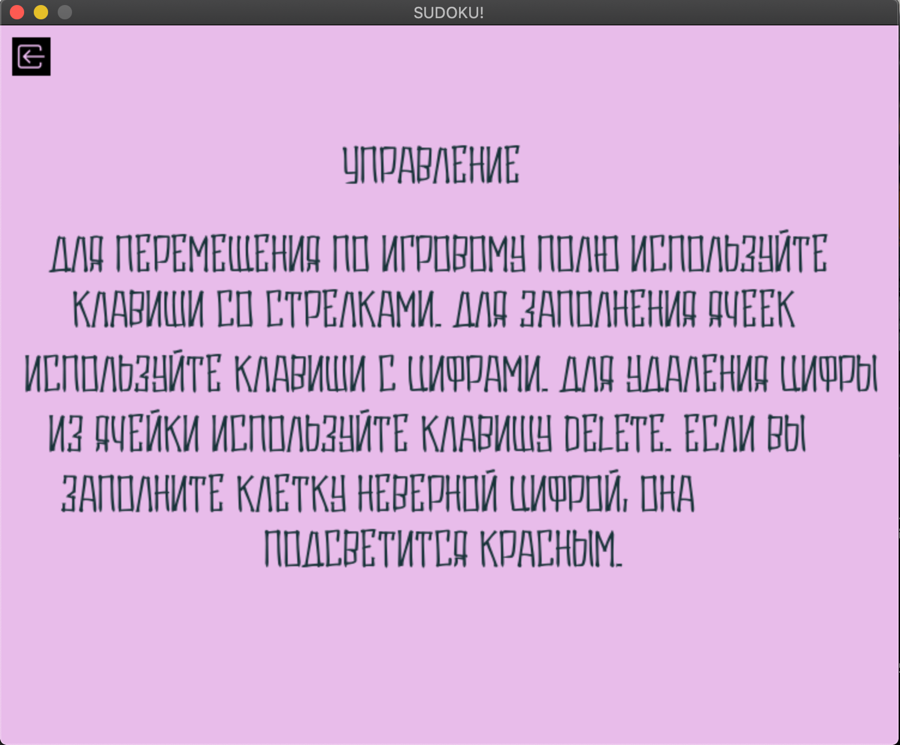


Рисунок 6 – Управление в игре

При выборе пункта меню «Начать игру» пользователю предоставляется возможность выбрать задний фон для игры. Этот пункт продемонстрирован на рисунке 7.

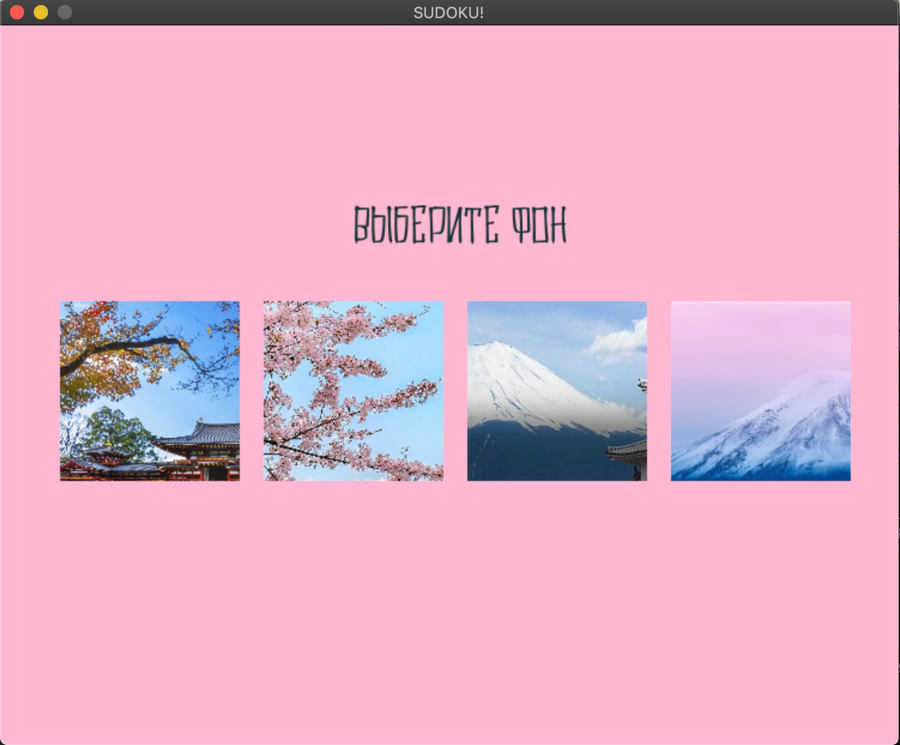


Рисунок 7 – Меню выбора заднего фона

После выбора заднего фона пользователю также предлагается выбрать уровень сложности игры. Внешний вид этого раздела представлен на рисунке 8.



Рисунок 8 – Меню выбора уровня сложности

После выбора уровня сложности пользователь переходит к игре. На этом этапе некоторые ячейки поля уже заполнены цифрами. Ячейка, в которой находится пользователь, подсвечивается голубым цветом. Пример игрового процесса представлен на рисунке 9.

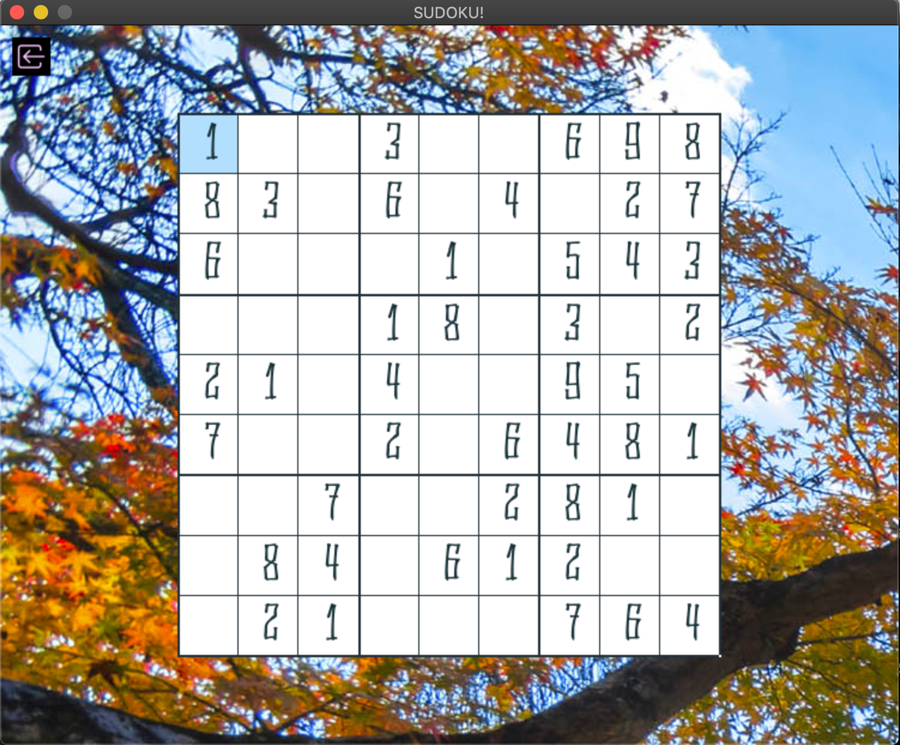


Рисунок 9 – Игровой процесс

При неправильном заполнении ячеек игрового поля все противоречащие решению клетки подсвечиваются красным цветом. Пример такого заполнения представлен на рисунке 10.

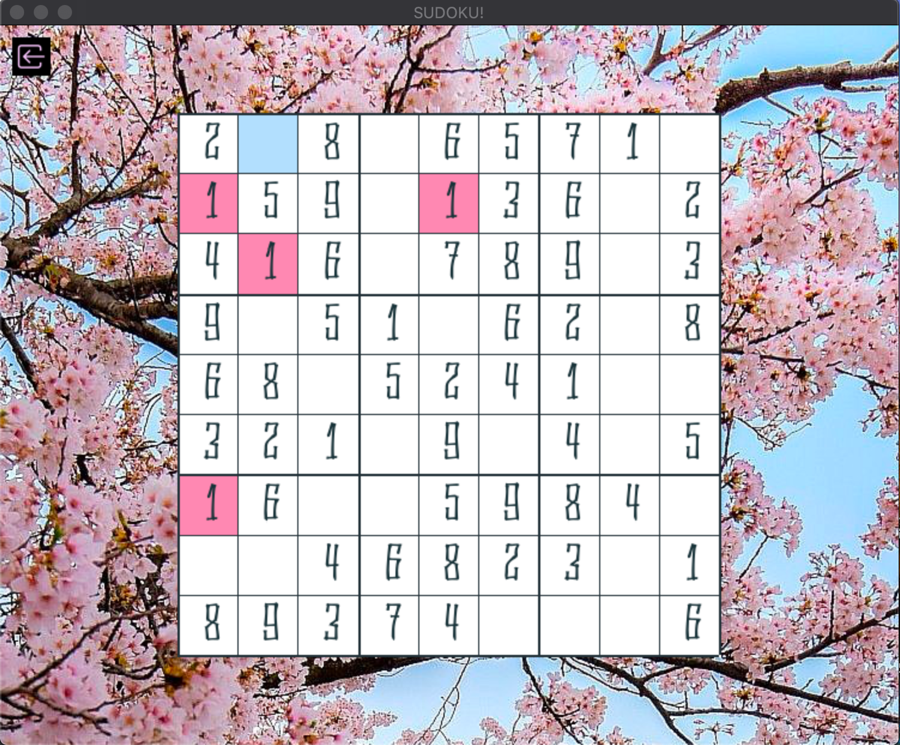


Рисунок 10 – Отображение неправильно заполненных клеток

Игра продолжается до победы игрока или до завершения ее игроком. Завершить игру можно при выходе в главное меню и выборе пункта «Выход» или по закрытию окна программы. В случае выигрыша пользователь получает сообщение о победе. На рисунке 11 продемонстрирован этот раздел.

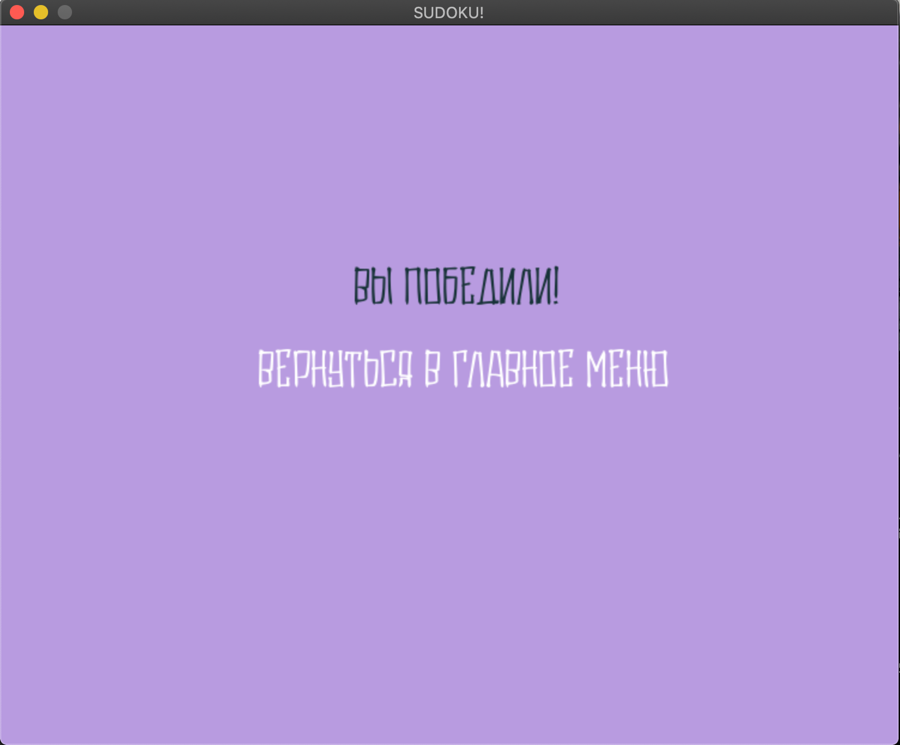


Рисунок 11 – Победа игрока

# **ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы была разработана программа, реализующая игру «Судоку». Все логические правила игры соблюдаются и соответствуют поставленной задаче.

Созданная игра может являться прототипом для дальнейшего улучшения организации игрового процесса.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. SDL Wiki [Электронный ресурс].  – URL: https://wiki.libsdl.org/ (дата обращения: 01.12.2020)
2. Арипова О.В. Программирование на языке высокого уровня: лаб. практикум / О.В. Арипова, А.Н. Гущин, О.А. Палехова; Балт. гос. техн. ун-т. – СПб., 2014. – 164 с.
3. Википедия. Судоку [Электронный ресурс]. – URL: https://en.wikipedia.org/wiki/Sudoku (дата обращения: 01.12.2020)
4. YouTube. Madsycode – Tutorial: Create 2D Game Engine using C++ and SDL [Электронный ресурс]. – URL: https://www.youtube.com/playlist?list=PL-K0viiuJ2RctP5nlJlqmHGeh66-GOZR\_ (дата обращения: 01.12.2020)
5. Страуструп Б. Программирование. Принципы и практика с использованием С++ (2е издание) 2016 – 135с.

# **ПРИЛОЖЕНИЕ А**

Приложение находится на компакт диске и содержит исходный текст программы.