Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

**Лабораторная работа №12**

По дисциплине «Основы программной инженерии»

Исполнители

Студенты 1 курса

Корнелюк В. В.

Кивлинас О. Л.

Коваль И. И.

Минск 2022

**Содержание:**

**1.Введение1**

1.1. Цели1

1.2. Задачи1

1.3. План разработки приложения1

**2.Инструкция к игре**2

2.1 Меню2

2.2 Игровой процесс2

**3.Техническая документация к игре3**

3.1.Модуль main\_2048.3

3.2. Модуль display4

3.3. Модуль game\_process4

1. **Введение**
   1. **Цель**

Создание консольной игры на языке C++. Цель – объединить клетки с одинаковым номиналом, таким образом растет их «вес», так нужно дойти до числа 2048. При создании данного проекта мы использовали навыки, полученные во время обучения.

**Задачи**

-Разработать игру в консоли.

-Создать игровой алгоритм.

- При разработке использовать известные для нас команды ЯП С/С++;

- При необходимости, дополнительно изучить команды, которые помогут реализовать приложение.

* 1. **План разработки приложения**

-Создать примитивный консольный интерфейс к игре и оформление.

-Разработать:

* Правила игры
* Графическое представление в консоли
* Логику приложения
* Выход из приложения, в зависимости от результата.

-Написать инструкцию к эксплуатации.

1. **Инструкция к игре**
   1. **Меню**

Меню представляет собой стандартную консоль windows с изменённой цветовой палитрой.

**2.2 Игровой процесс**

При запуске игры перед нами появляется консоль. Для начала игры у нас просят ввести любой символ, или предоставляется возможность еще до начала игры завершить работу приложения.   
 После нажатия любой кнопки для начала игры, перед игроком появляются короткий свод правил и игровое поле 4х4 клеточки.

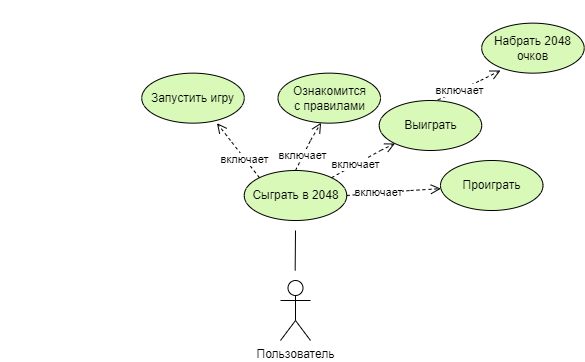
Правила игры:

1. В каждом раунде появляется плитка номинала «2» (с вероятностью 90%) или «4» (с вероятностью 10%).
2. Нажатием кнопок управления игрок может скинуть все плитки игрового поля в одну из 4 сторон. Если при сбрасывании две плитки одинакового номинала «налетают» одна на другую, то они превращаются в одну, номинал которой равен сумме соединившихся плиток. После каждого хода на свободной секции поля появляется новая плитка номиналом «2» или «4». Если при нажатии кнопки местоположение плиток или их номинал не изменится, то ход не совершается.
3. За каждое соединение игровые очки увеличиваются на номинал получившейся плитки.
4. Игра заканчивается победой, если игрок получает плитку номиналом «2048» или, если после очередного хода невозможно совершить действие.
5. **Техническая документация к игре.**

Пользовательская история для проекта:

Как **пользователь** я хочу **использовать «2048\_game»**, чтобы развивать логическое мышление.

Диаграмма вариантов использования для проекта:



**3.1 Модуль main\_2048.**

Пользовательская история для main\_2048:

Как **пользователь** я хочу, чтобы при входе в приложение **было меню** входа/выхода из игры, чтобы можно было управлять приложением независимо от чего-либо.

Диаграмма вариантов использования для main\_2048:

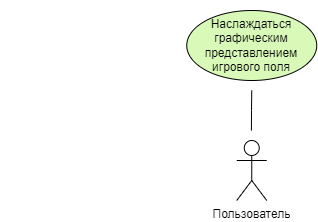


**3.2** **Модуль display.**

Пользовательская история для display:

Как **пользователь** я хочу, чтобы **было графическое представление игрового поля**, для того чтобы визуально приятно использовать приложение.

Диаграмма вариантов использования для display:



**3.3 Модуль game\_process.**

Пользовательские истории для game\_process:

Как **пользователь** я хочу, чтобы **было легко осуществляемое управление** игровым процессом, чтобы легко управлять игровым процессом.

Как **пользователь** я хочу, чтобы **при случайном нажатии неправильной кнопки результат игрового процесса не менялся**, чтобы можно было беспроблемно продолжить игру.

Как **пользователь** я хочу, чтобы **была возможность завершения игры**, чтобы при необходимости досрочно завершить игру.

Диаграмма вариантов использования для game\_process:

