**Проект “Wordle Game”**

**Документация**

**Обзор**

* Название проекта: Wordle Game
* Дата: 11.12.2023
* Scrum team:
* Гацевич Кирилл – Scrum master, логика программы
* Жук Дмитрий – дизайнер
* Туцкая Татьяна – тестировщик, документация

**Введение**

Документация описывает проект “Wordle Game”, который представляет собой игру, основанную на угадывании слов. В игре предлагается угадать секретное слово, состоящее из семи букв. Игрок вводит буквы в текстовые поля (чтобы из них получилось существующее слово), а затем игра проверяет правильность введённого слова и предоставляет обратную связь. Цель игры – угадать секретное слово за 5 и менее попыток.

**Список технологий:**

* C++, Framework.net 5.0;
* IDE – Visual Studio

**Сроки выпуска:**

* Дата анонса: 04.12.2023
* Дата выхода: 11.12.2023

**Репозиторий кода:**

* Git-репозиторий: <https://github.com/KiriGal/Project_Lab_8>

**Структура проекта**

Проект состоит из следующих файлов:

* wordle\_form.cpp - основной исходный код игры, содержащий логику и интерфейс пользователя.
* wordle\_form.h - заголовочный файл для wordle\_form.cpp, содержащий объявления классов и функций.
* russian.txt - текстовый файл, содержащий словарь русских слов, из которого выбирается секретное слово.
* Другие неуказанные файлы - возможные зависимости, необходимые для сборки и запуска проекта.

**Зависимости**

Проект использует следующие зависимости:

* Windows.h - заголовочный файл для работы с операционной системой Windows.
* fstream - заголовочный файл для работы с файлами.
* string - заголовочный файл для работы со строками.
* msclr\marshal\_cppstd.h - заголовочный файл для преобразования между строками .NET и стандартными строками C++.
* cctype - заголовочный файл для работы с символами и классификацией символов.
* map - заголовочный файл для работы с ассоциативными контейнерами std::map.
* vector - заголовочный файл для работы с контейнерами std::vector.
* iostream - заголовочный файл для ввода/вывода данных.
* ctime - заголовочный файл для работы со временем.

**Основные функции и классы:**

**Класс wordle\_form** представляет главную форму игры и содержит логику игры и обработку событий пользовательского интерфейса.

*Основные методы класса wordle\_form:*

* initializeVirtualCodeToLetterMap(int numKey) - инициализирует отображение между виртуальными кодами клавиш и соответствующими буквами русского алфавита.
* convertToSystemString(const std::string& str) - преобразует стандартную строку C++ в строку .NET (System::String).
* ConvertSystemStringToStdString(String^ s, std::string& os) - преобразует строку .NET (System::String) в стандартную строку C++.
* InsertIntoSystemString(System::String^ original, System::String^ toInsert, int position) - вставляет строку toInsert в original на указанную позицию.
* getMasWords() - загружает слова из файла "russian.txt" в массив masWords.
* clearTextBox(bool flag) - очищает текстовые поля на форме. Если flag равен true, сбрасывает текущую позицию и ключевую клавишу.
* clearString(int line) - очищает строку (набор текстовых полей) на форме по указанному номеру строки.
* getWordForLine(int line) - получает слово из строки (набор текстовых полей) на форме по указанному номеру строки.
* endWinGame() - завершает игру с победой.
* endLoseGame() - завершает игру с поражением.
* isWordExist(int line) -Продолжение документации к проекту "Wordle Game"
* isWordExist(int line) - проверяет существование слова в заданной строке (наборе текстовых полей).
* checkWord(int line) - проверяет правильность введенного слова в заданной строке (наборе текстовых полей).
* checkStatus() - проверяет текущий статус игры (победа или поражение).
* updateTextBox(char keyChar) - обновляет текстовое поле на форме после ввода символа с клавиатуры.
* removeTextBox(int startPos) - удаляет текстовые поля на форме начиная с заданной позиции.
* wordle\_form\_KeyDown(Object^ sender, KeyEventArgs^ e) - обработчик события нажатия клавиши на форме.
* wordle\_form\_Load(Object^ sender, EventArgs^ e) - обработчик события загрузки формы.
* buttonSubmit\_Click(Object^ sender, EventArgs^ e) - обработчик события нажатия кнопки "Submit".

**Глобальные переменные**

* masWords - массив строк, содержащий загруженные слова из файла "russian.txt".
* virtualCodeToLetterMap - ассоциативный контейнер, связывающий виртуальные коды клавиш с соответствующими буквами русского алфавита.
* currentPos - текущая позиция на форме.
* flagure - флаг для определения состояния игры.
* stopGame - флаг для определения остановки игры.
* statusResult - флаг для определения статуса игры (победа или поражение).
* keyInt - виртуальный код ключевой клавиши.
* trueWord - секретное слово, которое нужно угадать.
* keyboardHook - хук клавиатуры.

**Использование**

1. Запустите программу "Wordle Game".

На форме отобразится набор текстовых полей. Каждое поле представляет одну букву.

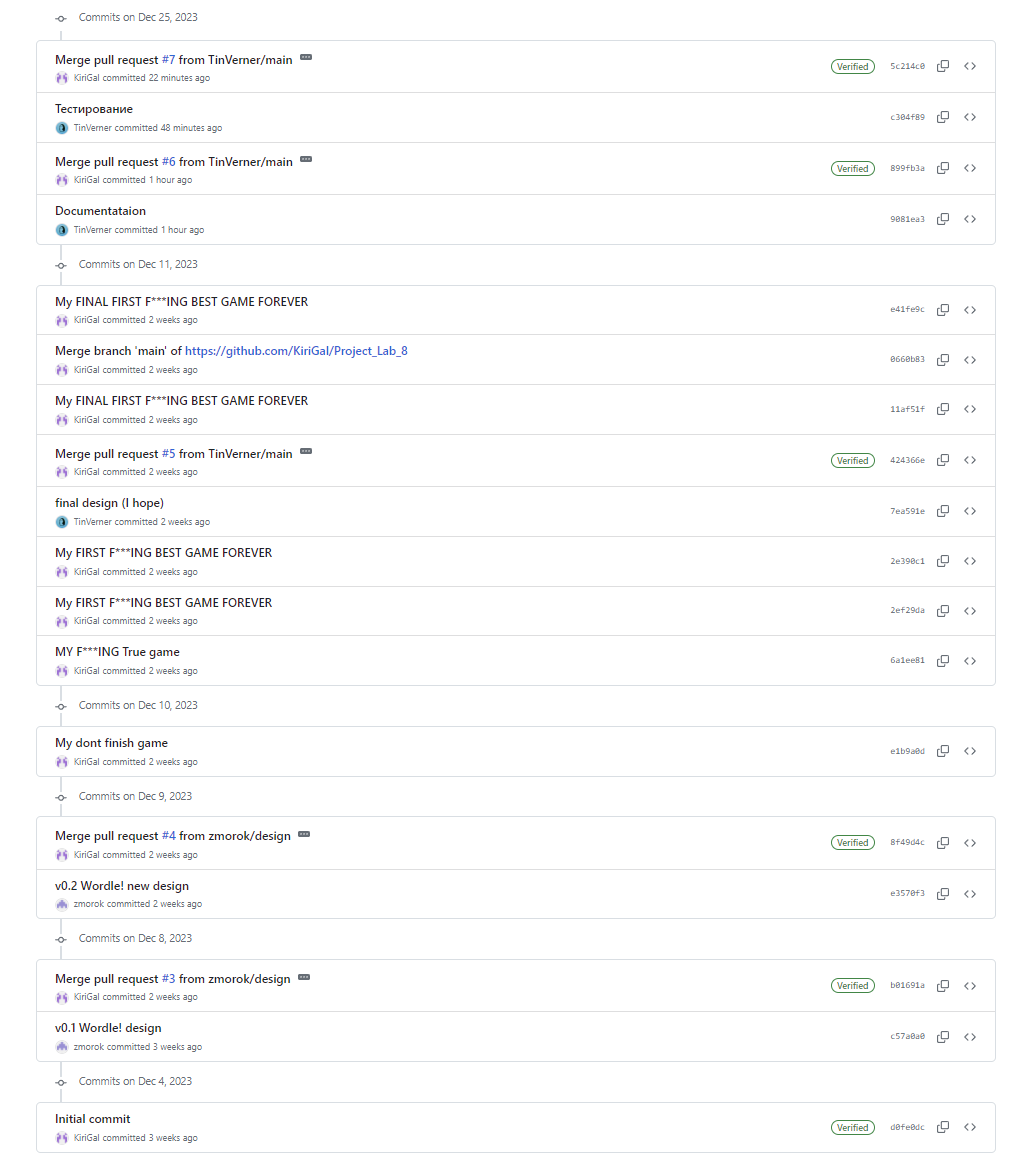
1. Введите буквы в текстовые поля, чтобы угадать секретное слово.

* Если введенная буква присутствует в секретном слове и находится на правильной позиции, поле станет зеленым.
* Если введенная буква присутствует в секретном слове, но находится на неправильной позиции, поле станет желтым.
* Если введенная буква не присутствует в секретном слове, поле останется серым.

1. Продолжайте вводить буквы и анализировать обратную связь, чтобы угадать секретное слово.
2. Если вы угадали секретное слово, игра завершится с победой. Если вы исчерпали все попытки и не угадали слово, игра завершится с поражением.

**История коммитов:**

1. Initial commit
2. Wordle design: v0.1
3. Wordle design: v0.2
4. The main code: *not finished*
5. The main code: v0.1
6. The main code: v0.2 *correction of defects*
7. Wordle design: v0.3 *final*
8. The main code: v0.4 *final*
9. Added documentation
10. Added testing
11. Added history of commits



**Заключение**

Документация описывает основные аспекты проекта "Wordle Game" и предоставляет информацию о структуре, функциях и классах, а также о принципе использования игры.