

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ «МИСИС»
Институт информационных технологий и компьютерных наук
Кафедра инженерной кибернетики

Курсовая работа
по дисциплине «Технологии программирования»
на тему «Игра «Тренажер кратковременной памяти»»

Выполнила:

Студентка 3-го курса, группы БПМ-20-2 Ларичева А.Ю.

Проверил:

Доцент кафедры ИК, к.т.н. Полевой Д. В.

Москва 2022

Оглавление

Описание задачи.....	3
Пользовательское описание	3
Техническое описание	5
Документация	7
Руководство по сборке.....	7
Ссылки.....	9

Описание задачи

Написать пользовательское приложение с игрой «Тренажер кратковременной памяти». Игра рассчитана на одного игрока и имеет два режима: тренировка памяти на числа и тренировка на запоминание текстовой информации.

Реализовать графическое приложение для развития кратковременной памяти на языке программирования C++ 17 версии. Для написания приложения использовать графическую библиотеку Qt.

Пользовательское описание

При запуске игры перед пользователем появляется начальный экран, где можно с помощью кнопок выбрать один из двух режимов: Number Memory или Text Memory.

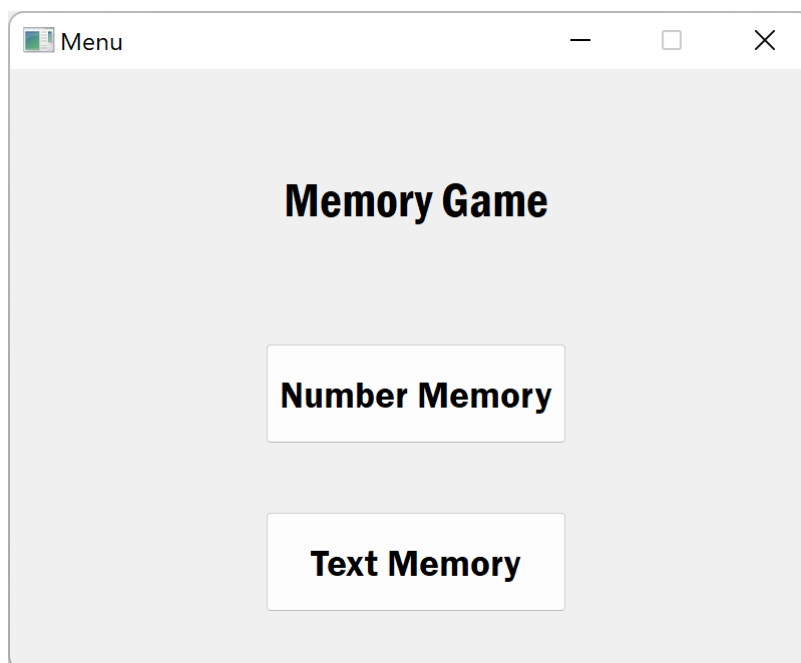


Рисунок 1. Начальный экран игры

При выборе режима Number Memory открывается окно с игрой. После нажатия кнопки “Start” перед пользователем на несколько секунд появляется число. Затем число пропадает и появляется поле для ввода текста. Задача пользователя: запомнить, показанное ему число и верно ввести его. С каждым уровнем число для запоминания становится все длиннее.

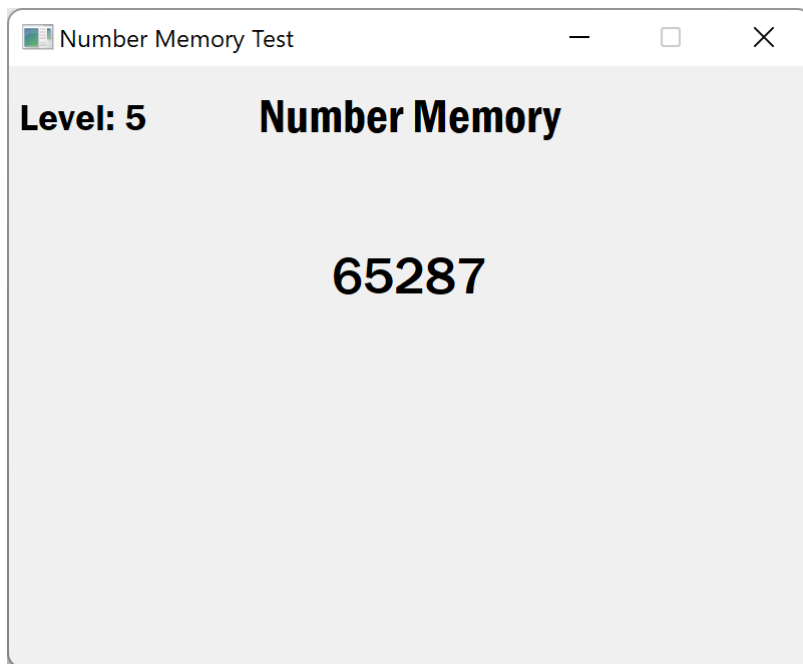


Рисунок 2. Игра в режиме Number Memory

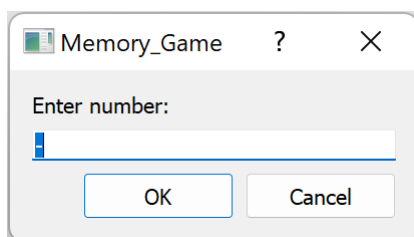


Рисунок 3. Поле для ввода

При выборе режима Text Memory также открывается окно с аналогичной игрой, но вместо числа пользователю нужно запомнить последовательность из букв латинского алфавита.

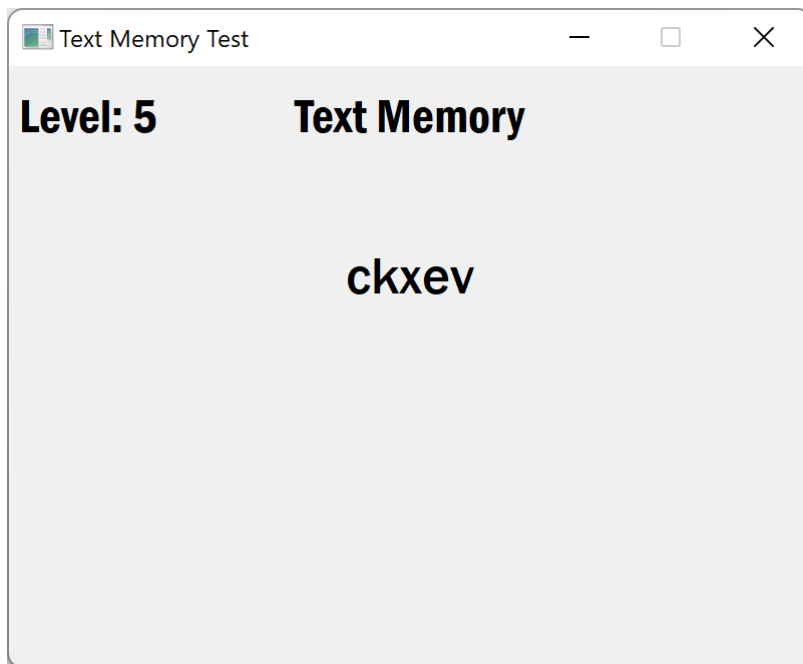


Рисунок 4. Игра в режиме Text Memory

Когда пользователь ошибается, игра заканчивается и на экран выводится уровень, которого он успел достичь и его ошибка.

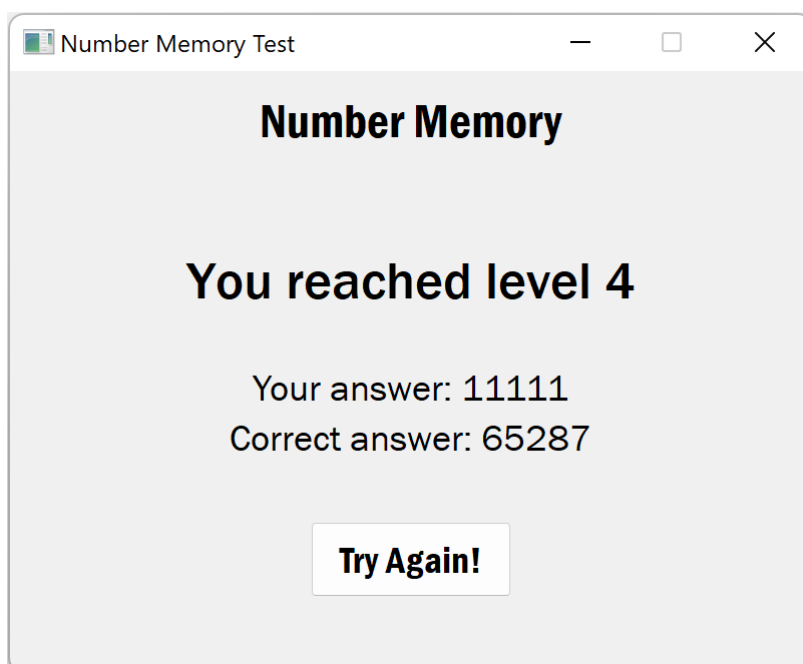


Рисунок 5. Завершение игры

Техническое описание

Приложение имеет три окна. Первое начальное меню открывается

автоматически при запуске приложения (рисунок 1). На нем с помощью кнопок можно открыть два других окна (рисунок 2 и 4). Они все имеют фиксированный размер.

При переходе в режимы игры Number Memory и Text Memory появляется кнопка “Start”, по нажатию которой начинается игра. Изначально пользователь имеет уровень 1. Длина предоставляемого для запоминания символа равна уровню игрока, а время на запоминание – (уровень игрока + 1) секунд.

По истечении времени число пропадает с экрана и появляется диалоговое окно для ввода текста. Введенный пользователем текст сравнивается с правильным значением. В случае совпадения уровень игрока повышается на 1, и игра продолжается. Если введена неправильная последовательность символов на экран выводится сообщение о том, какого уровня достиг игрок, а также его ответ и ожидаемый ответ. На месте кнопки “Start” появляется кнопка “Try again”, по нажатию на которую игра начинается заново с первого уровня. Для завершения игры достаточно закрыть окно приложения.

Основными классами являются:

`mainwindow` – класс, отвечающий за начальное окно приложения и переход в режимы игры;

`numbermem` – класс, в котором реализован функционал игры в режиме Number Memory;

`textmem` – класс, в котором реализован функционал игры в режиме Text Memory.

Основными функциями являются:

`void next_level(int level)` – выводит на экран последовательность символов нужной длины, запускает таймер;

`void get_number()` и `void get_text()` – по истечении времени

таймера выводит диалоговое окно для ввода ответа, сравнивает ответ с правильным.

Документация

Документация, созданная с помощью Doxygen доступна в папке docs репозитория. В папке rft можно найти документацию в формате Rich Text Format.

Class Documentation

MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Member Functions

- **MainWindow** (QWidget *parent=nullptr)
The constructor that creates QWidget of main window.
- **~MainWindow** ()
The destructor.

The documentation for this class was generated from the following files:

- mainwindow.h
- mainwindow.cpp

NumberMem Class Reference

```
#include <numbermem.h>
```

Inheritance diagram for NumberMem:



Public Member Functions

- **NumberMem** (QWidget *parent=nullptr)
- **~NumberMem** ()
The destructor.

Public Attributes

- **QTimer * timer**
The instance of class QTimer.
- **QString user_num = "-"**
Field user_num variable by default "-".
- **QString number_str = ""**
Field number_str variable by default "".
- **int number = 0**
Field number variable by default 0.
- **int level = 1**
Field number variable by default 1.

Рисунок 6. Часть документации в формате rft

Руководство по сборке

Исходные файлы находятся по данному адресу:

https://github.com/lrchva/Memory_Game/tree/main/Memory_Game

Чтобы собрать проект локально, можно использовать CMake. Для этого в графе “source code” указываем путь до папки с проектом, а в графе “where to build” указываем {путь до папки с проектом}/build. Затем нажимаем «Configure». Необходимо выбрать папку в CMAKE_PREFIX_PATH, куда будет собираться готовый проект. После этого нажимаем «Generate» и «Open project».

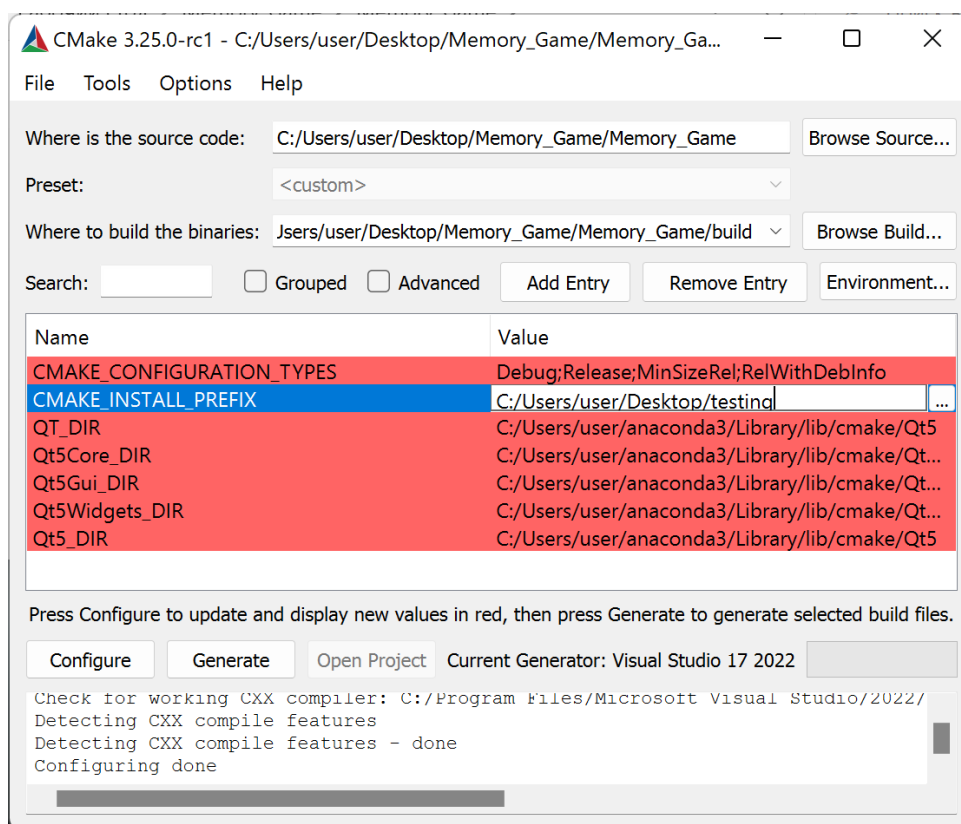


Рисунок 7. Пример заполнения полей CMake

Откроется компилятор (например, Visual Studio 17 2022). В нем необходимо начать сборку.

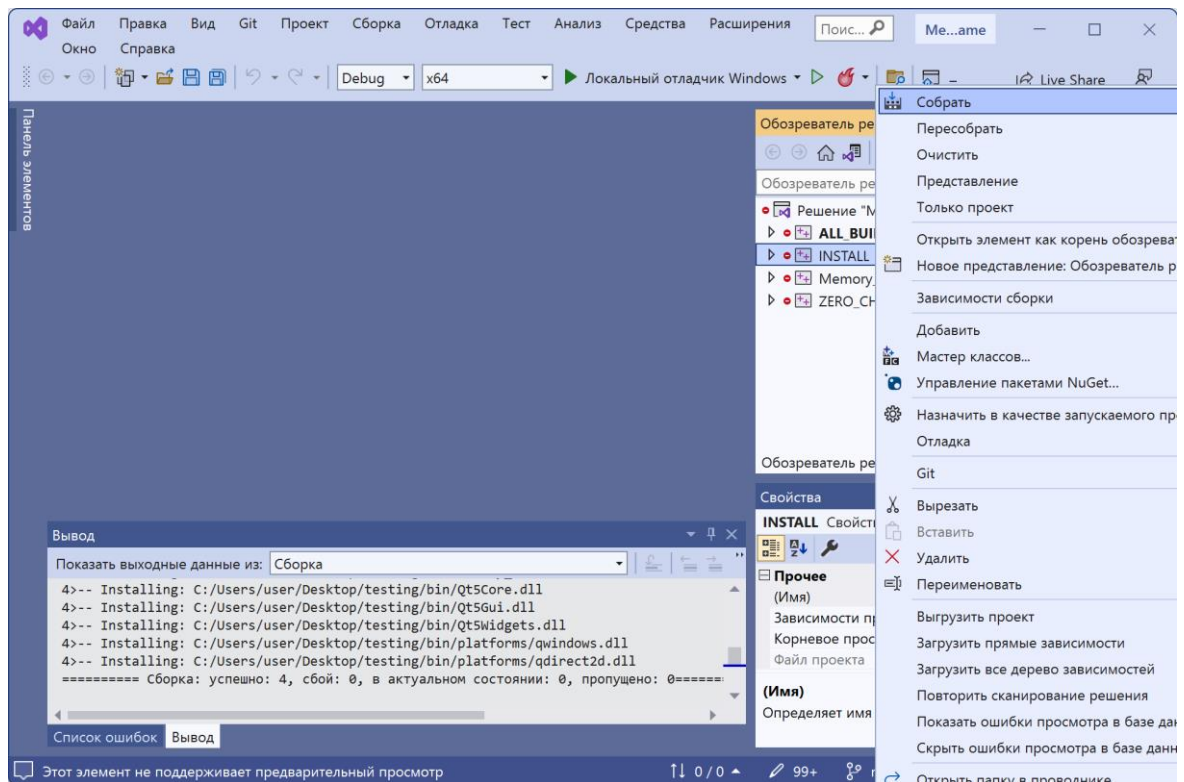


Рисунок 8. Сборка проекта в Visual Studio

После того, как сборка успешно завершена, в папке, указанной в `CMAKE_PREFIX_PATH`, появится файл приложения.

Ссылки

GitHub: https://github.com/lrchva/Memory_Game