

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Лабораторна робота № 4

з дисципліни: «Автоматизоване проектування комп'ютерних систем»,

на тему: «Створення документації за допомогою Doxygen.»

Виконав:

ст. гр. КІ-410

Іванюк О.О.

Прийняв:

Кіцера А. О.

Львів – 2024

Task 4. Create doxygen documentation:

1. Add doxygen comments for all public functions, classes, properties, fields...
2. Generate documentation based on doxygen comments
3. Required steps

Варіант 7:

7	tik-tac-toe 3x3	XML
---	-----------------	-----

Теоретичні відомості

Doxygen - це інструмент для автоматизації процесу створення документації для програмного забезпечення. Він призначений для проектів, написаних мовами програмування C++, C, Java, Objective-C, Python, IDL (Corba, Microsoft, і так далі), Fortran, VHDL, PHP, C#, і має підтримку інших мов.

За допомогою Doxygen можна створювати HTML-документацію, а також документацію у форматах PDF, LaTeX і інших. Він може генерувати звіти про класи, залежності між класами, графи виклику функцій та інші корисні візуалізації.

Doxygen аналізує ваші вихідні файли та коментарі, перетворюючи їх у структуровану документацію. Ваші слова стають ключовими, а код — зрозумілим інструментом для всіх, хто працює з вашим проектом.

Коментарі для Doxygen можуть починатися з `/**`, `/*!`, `///`, або `///
Використовуйте теги, такі як @param, @return, @see, для деталей функцій, параметрів, та інших елементів коду.`

Doxygen легко інтегрується з іншими інструментами, дозволяючи легко працювати з системами контролю версій та тестуванням коду. Використовує теги Markdown або HTML, щоб розширити можливості форматування документації.

Хід роботи

1. Додав коментарі до коду з попереднього завдання, які відповідають правилам документування doxygen.

```
mainwindow.cpp  portArduino -> QString

/**
 * @file mainwindow.cpp
 */

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QFile>
#include <QTextStream>
#include <windows.h>

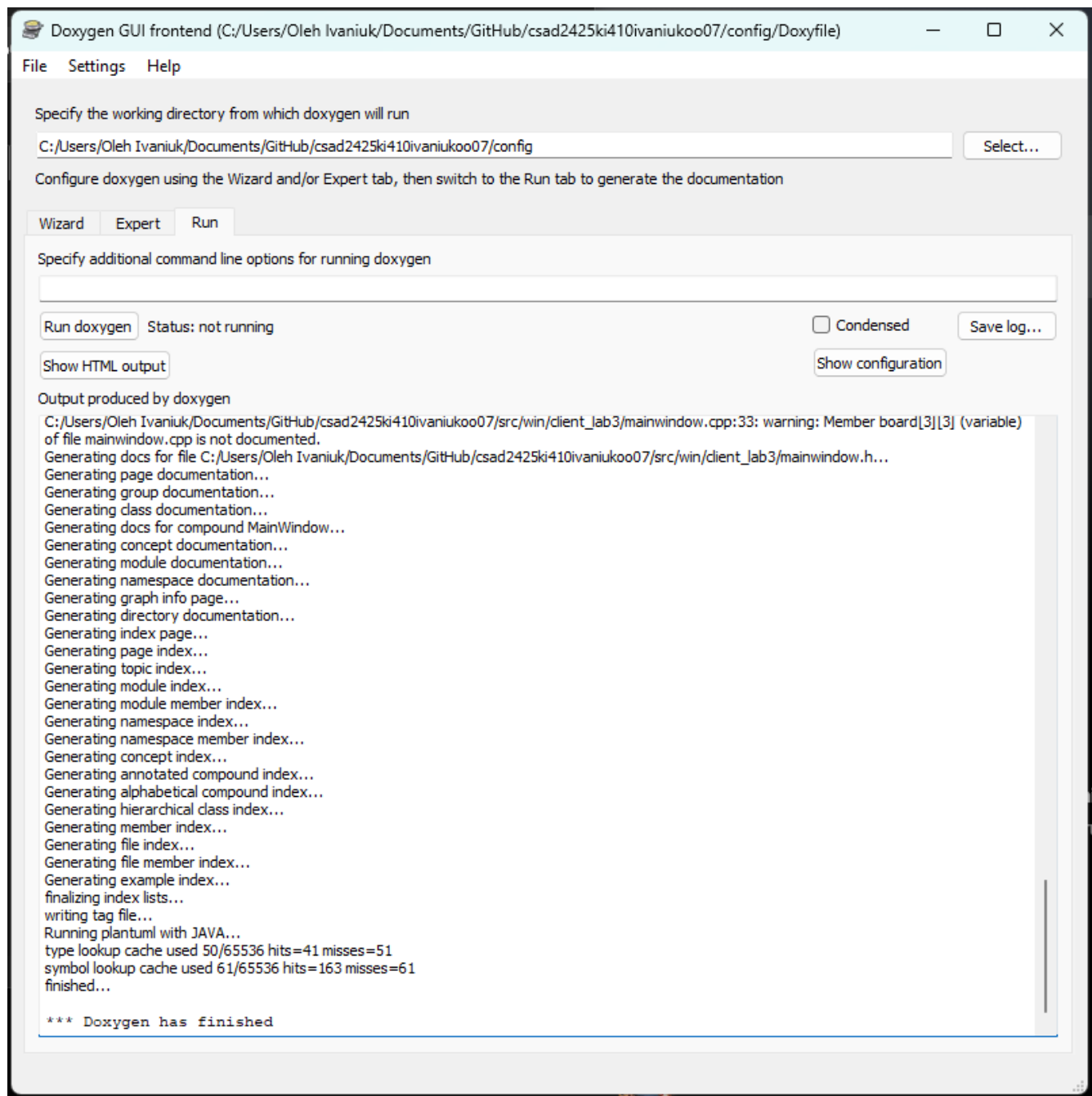
/**
 * @brief A QString variable to store the port information for Arduino.
 */
QString portArduino;

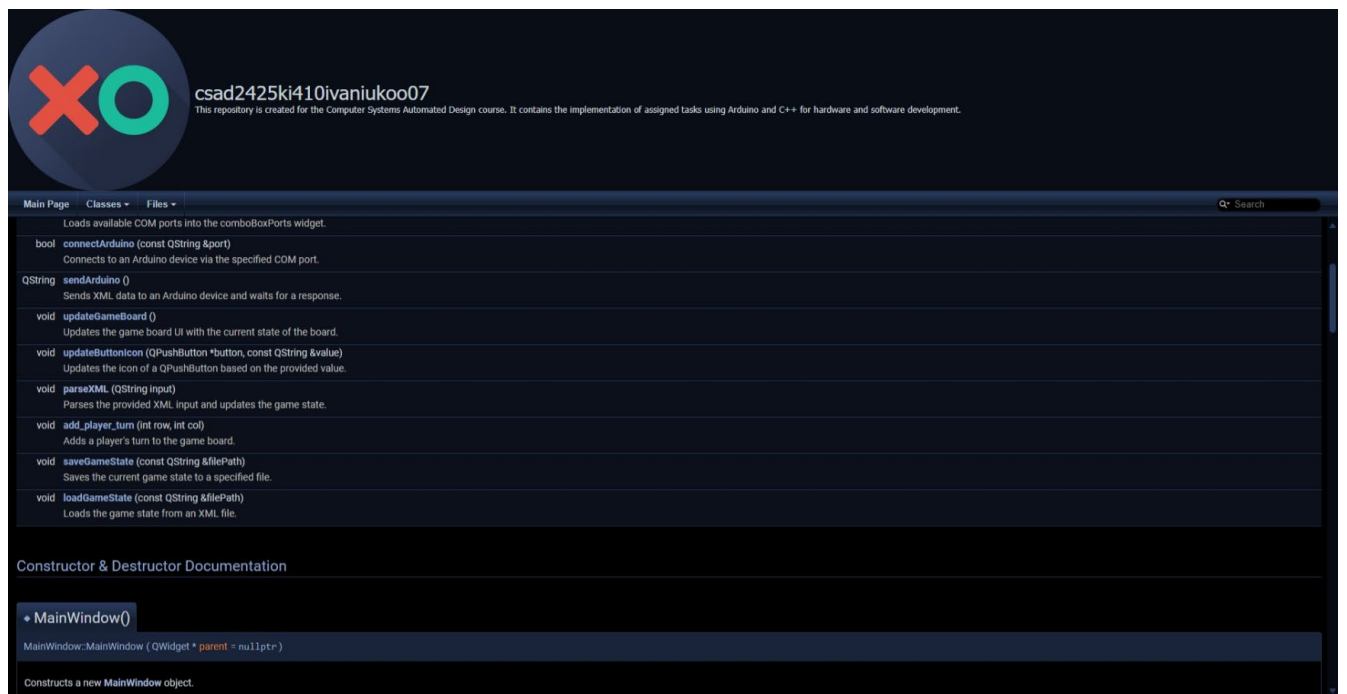
/**
 * @brief A handle to the serial port for communication with Arduino.
 */
HANDLE hSerial;

/**
 * @brief Variables used in the main window of the client application.
 *
 * @var connect_arduino Indicates the connection status with the Arduino device.
 * @var game_started Indicates whether the game has started.
 * @var game_mode Represents the current game mode.
 * @var ai_strategy Represents the AI strategy being used.
 * @var message Stores messages to be displayed or processed.
 * @var next_turn Indicates whose turn is next in the game.
 */
QString connect_arduino, game_started, game_mode, ai_strategy, message, next_turn;
QString board[3][3];

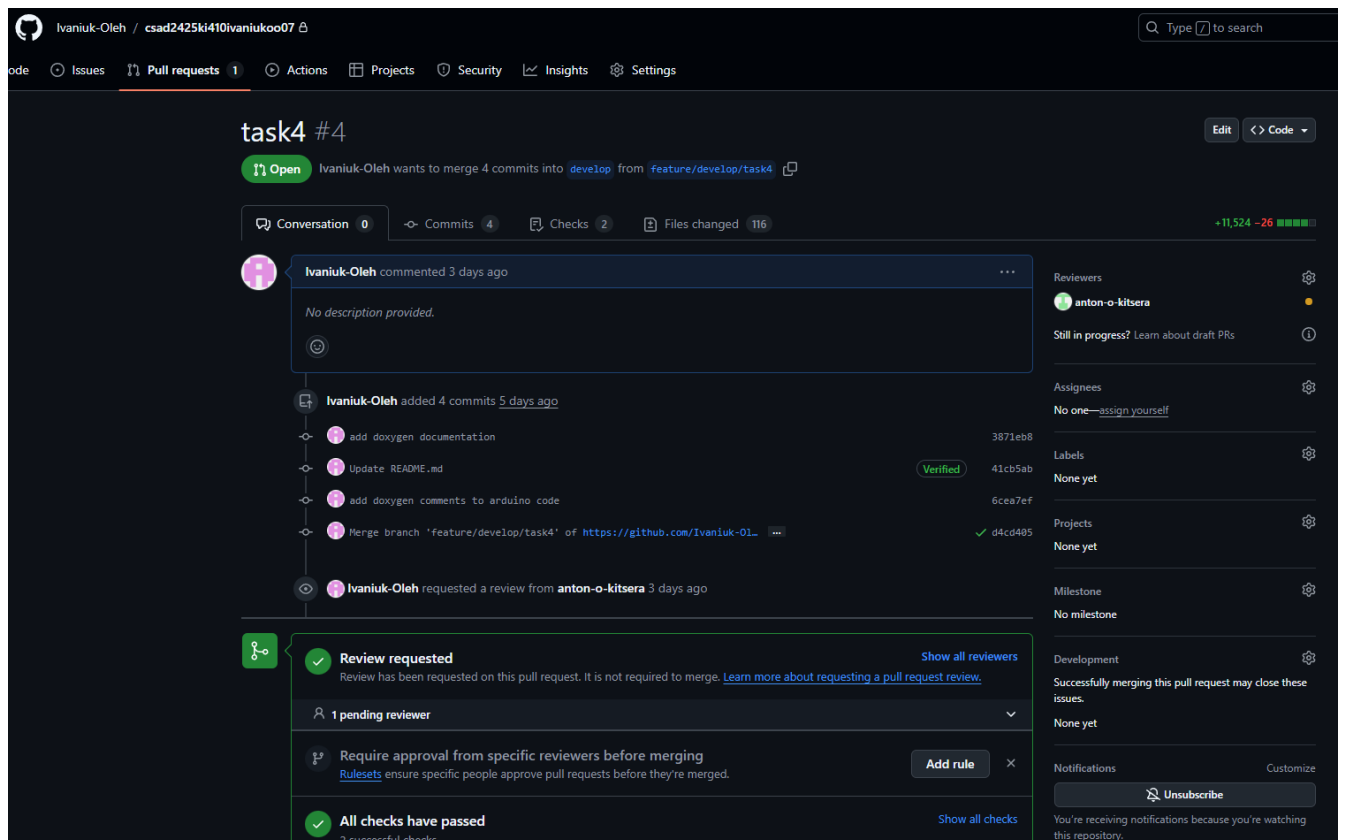
/**
 * @brief Resets the game values to their initial state.
 *
 * This function performs the following actions:
 * - Sets the game_started flag to "0".
 * - Clears the message string.
 * - Initializes the board array with the value "-".
 *
 * Note: Some variables such as connect_arduino, game_mode, and ai_strategy are commented out and not reset by this function.
 */
void resetValues() {
    // connect_arduino = "0";
    game_started = "0";
    // game_mode = "mva";
    // ai_strategy = "rand";
    message = "";
}
```

2. Використовую GUI версію Doxygen, створюю конфігураційний файл та запускаю генерацію документації.





3. Створив нову гілку `feature/develop/task4`. Створив Pull request для підтвердження змін в гілці `develop`, і надіслав запит на злиття викладачу.



Висновок:

В ході виконання лабораторної роботи було додано коментарі Doxygen до всіх публічних елементів коду, і створена з їх допомогою HTML-документація. Це полегшить розуміння коду та сприятиме ефективній комунікації в команді.