

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Лабораторна робота № 4

на тему: «Ознайомлення з doxygen»

Виконав:

ст. гр. КІ-404

Давида В.Р.

Прийняв:

Федак П.Р.

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 5:

Student number	Game	config format
5	tik-tac-toe 3x3	JSON

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

```
mainwindow.cpp  hSerial -> HANDLE

/**
 * @file mainwindow.cpp
 */

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QObject>
#include <QJsonDocument>
#include <QFile>

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

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

/**
 * @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 then 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 = "";
}
```

```
1  /**
2  * @file server_lab3.ino
3  */
4  String connect_arduino, game_started, game_mode, ai_strategy, message, next_turn;
5  String board[3][3];
6
7  /**
8  * @brief Sets up the Arduino by initializing serial communication and random seed.
9  * Configures the serial port and prepares random number generation for AI moves.
10 */
11 void setup() {
12
13     Serial.begin(9600); //Configure serial communication at 9600 bps
14     randomSeed(analogRead(0)); //Initialization of the random number generator
15 }
16 /**
17 * @brief Main loop of the Arduino program.
18 * Processes serial input to control game flow and sends responses based on the game state.
19 */
20 void loop() {
21     if (Serial.available() > 0) {
22
23         String receivedMessage = Serial.readString(); //Read data from the serial port
24         parseJSON(receivedMessage);
25
26         if (connect_arduino == "0"){
27             connect_arduino = "1";
28         }
29
30
31         if (game_started == "1") {
32             if (game_mode == "mva") {
33                 if (ai_strategy == "rand") {
34                     makeRandomMove();
35                 }
36             }
37         }
38     }
39 }
```

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

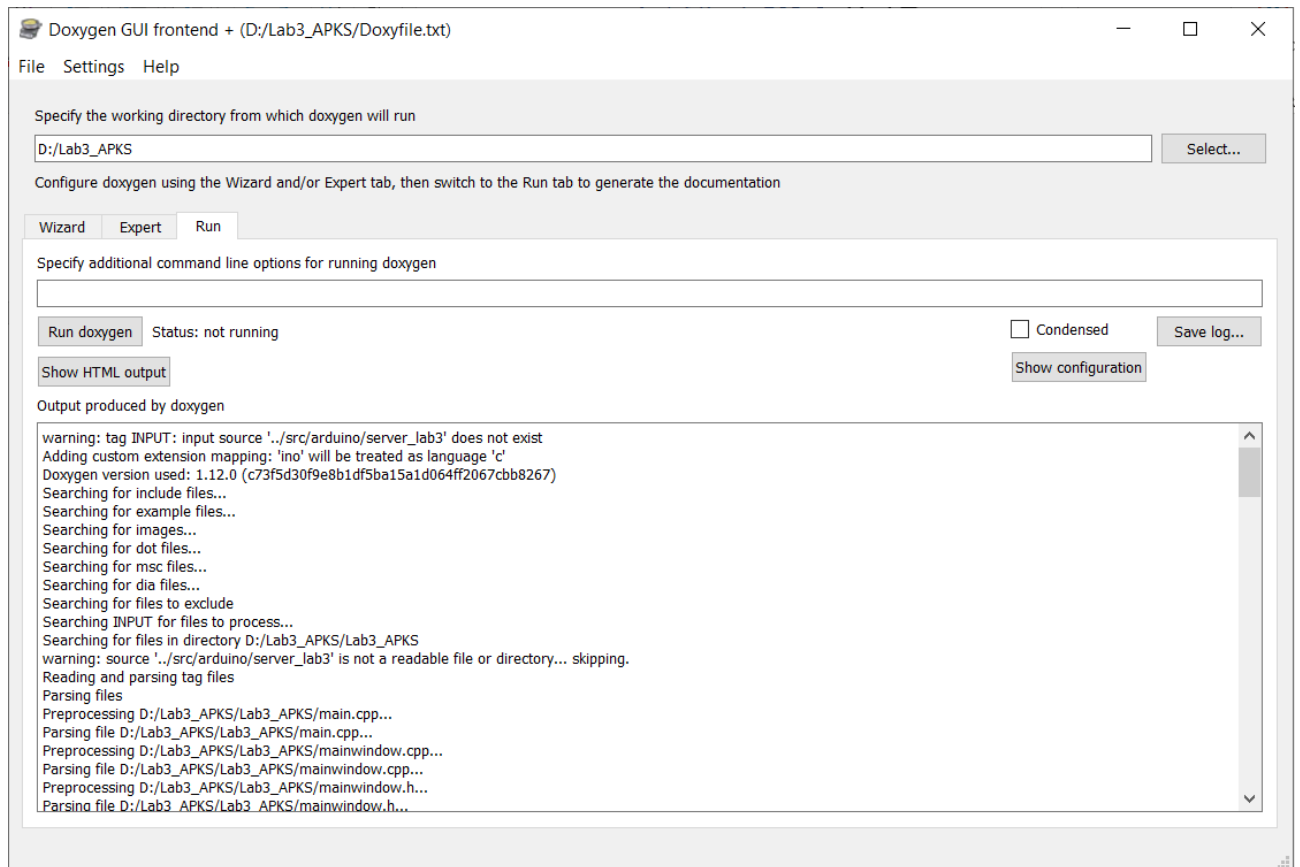


Рис.1. GUI версію Doxygen



Рис.2. Документація

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

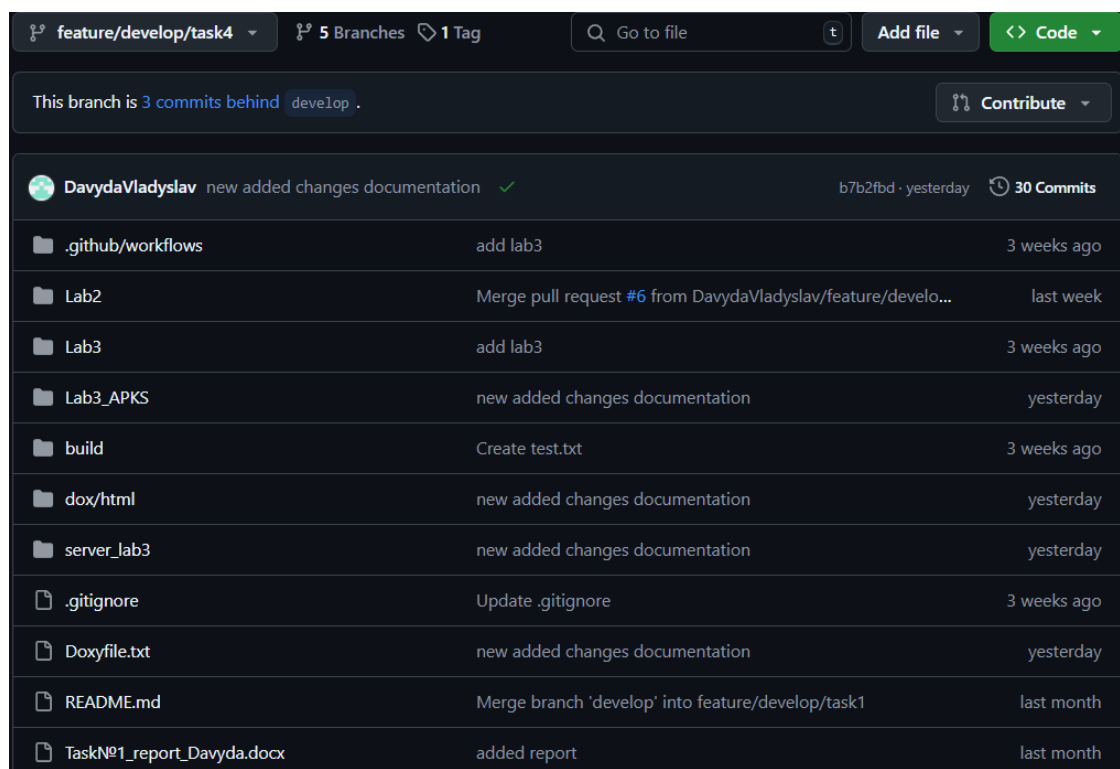


Рис.3.

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.