

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

Ковальчук А.О.

Прийняв:

к.т.н. асит. каф. СКС

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
- 4.

#### **Варіант 8:**

Student number	Game	config format
8	rock paper scissors	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.

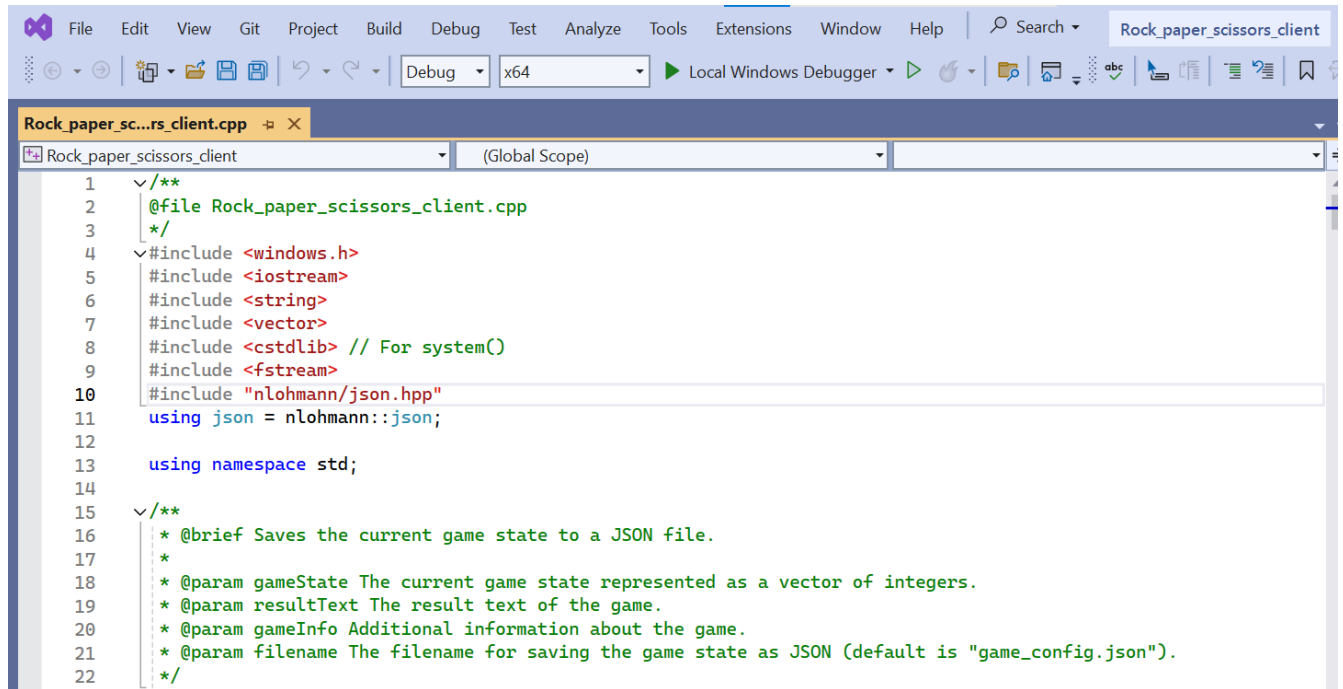
Doxygen аналізує ваші вихідні файли та коментарі, перетворюючи їх у структуровану документацію. Ваші слова стають ключовими, а код — зрозумілим інструментом для всіх, хто працює з вашим проектом.

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

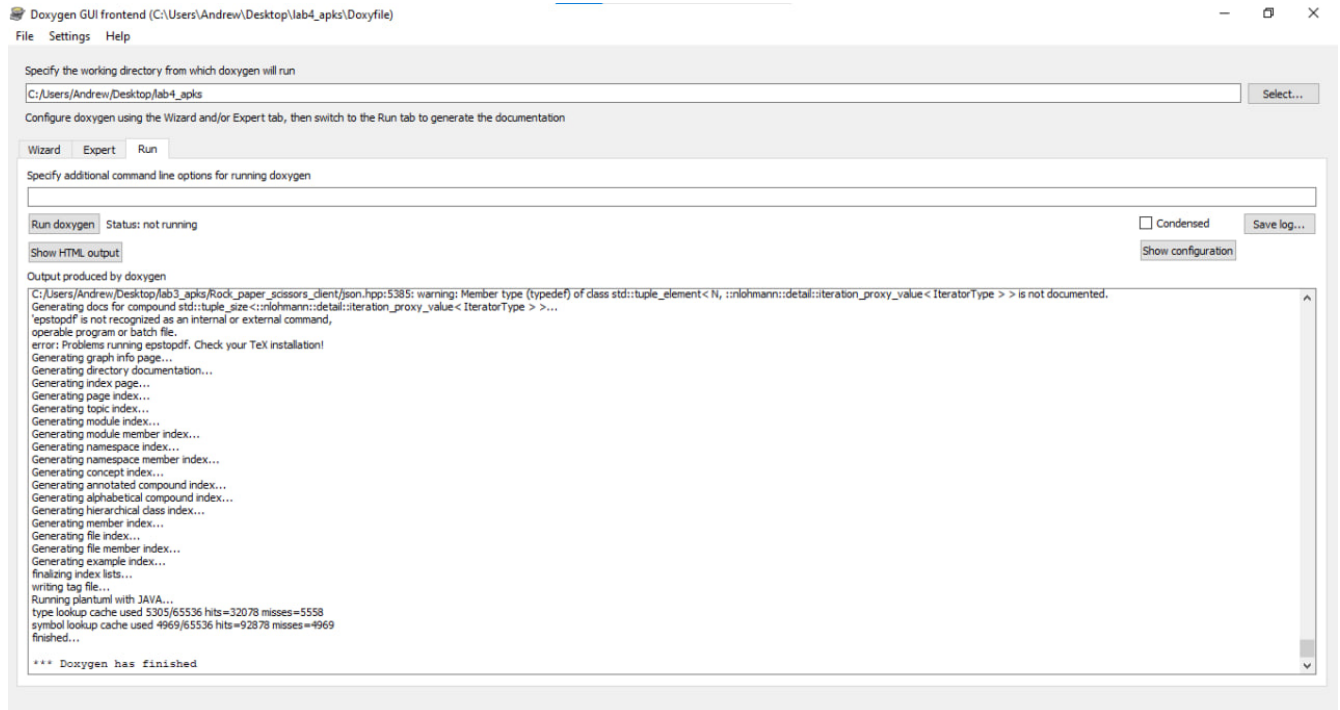


```
1  /**
2   * @file Rock_paper_scissors_client.cpp
3   */
4  #include <windows.h>
5  #include <iostream>
6  #include <string>
7  #include <vector>
8  #include <cstdlib> // For system()
9  #include <fstream>
10 #include "nlohmann/json.hpp"
11 using json = nlohmann::json;
12
13 using namespace std;
14
15 /**
16  * @brief Saves the current game state to a JSON file.
17  *
18  * @param gameState The current game state represented as a vector of integers.
19  * @param resultText The result text of the game.
20  * @param gameInfo Additional information about the game.
21  * @param filename The filename for saving the game state as JSON (default is "game_config.json").
22  */
```

sketch\_nov6a.ino

```
1  /**
2   * @file sketch_nov6a.ino
3   */
4
5  #include <ArduinoJson.h>
6
7  /// @brief The target score needed to win the game.
8  const int targetScore = 3;
9
10 void setup() {
11     /**
12      * @brief Initializes the Serial communication and the random number generator.
13      *
14      * Sets the baud rate to 9600 for Serial communication and seeds the random
15      * number generator using an analog read on pin 0 to ensure randomness.
16      */
17     Serial.begin(9600);
18     while (!Serial) { }
19     randomSeed(analogRead(0));
20 }
```

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



## Rock\_Paper\_Scissors\_Game

Main Page	Related Pages	Namespaces ▾	Classes ▾	Files ▾
lab3_apks	Rock_paper_scissors_client			

### Rock\_paper\_scissors\_client.cpp File Reference

```
#include <windows.h>
#include <iostream>
#include <string>
#include <vector>
#include <cstdlib>
#include <fstream>
#include "nlohmann/json.hpp"
```

### Typedefs

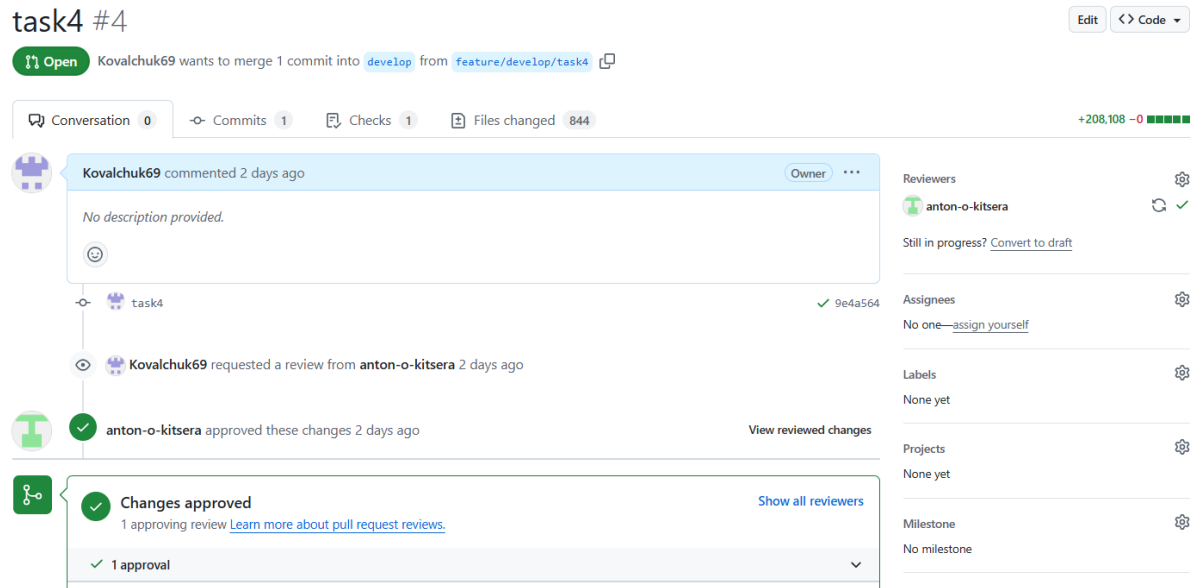
```
using json = nlohmann::json
```

### Functions

```
void saveGameState (const vector< int > &gameState, const string &resultText, const string &gameInfo, const string &filename="game_config.json")
    Saves the current game state to a JSON file.

bool loadGameState (vector< int > &gameState, string &resultText, string &gameInfo, const string &filename="game_config.json")
    Loads the game state from a JSON file.
```

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.