

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Лабораторна робота № 5

на тему: «Тестування гри Тіс Тас Тое»

Виконав:

ст. гр. КІ-404

Давида В.Р.

Прийняв:

Федак П.Р.

Львів – 2024

Task 5. Implement automated tests:

1. Implement or use existing test framework;
2. Create a set of automated tests;
3. Test report should contain number of all tests, passed tests, failed tests, coverage;
4. Coverage must be more than 80%
5. Required steps

Варіант 5:

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

Теоретичні відомості

Автоматизовані тести є невід'ємною частиною розробки програмного забезпечення, які виконуються автоматично для перевірки правильності роботи програм та виявлення можливих помилок. У мові програмування C++, для написання тестів, можна використовувати вбудовану бібліотеку `cassert` (`assert.h` у C), яка містить макрос `assert`. Цей макрос використовується для визначення виразів, які мають бути істинними. У випадку, якщо вираз виявиться хибним (`FALSE`), програма завершиться з повідомленням про помилку.

Для зроблення тестування більш систематичним та автоматизованим, розробники часто використовують спеціальні фреймворки для тестування, такі як Google Test або Catch. Ці фреймворки дозволяють створювати розширені тестові сценарії, вести звіти про результати тестування, та автоматизувати процес виявлення помилок.

Тести в розробці програмного забезпечення є важливою частиною процесу, оскільки вони дозволяють перевіряти правильність роботи коду, виявляти можливі помилки та забезпечувати стабільність програми. Тестування

може бути використане на різних етапах розробки, від написання нового коду до внесення змін у вже існуючий.

Тестування може бути виконане як вручну (manual testing), так і автоматично (automated testing). Автоматизовані тести мають численні переваги, такі як швидкість виконання, можливість запуску тестів в будь-який момент, ідентифікація помилок на ранніх етапах розробки та можливість повторного використання тестових сценаріїв.

Хід роботи

Відсоток покриття тестами клієнта:

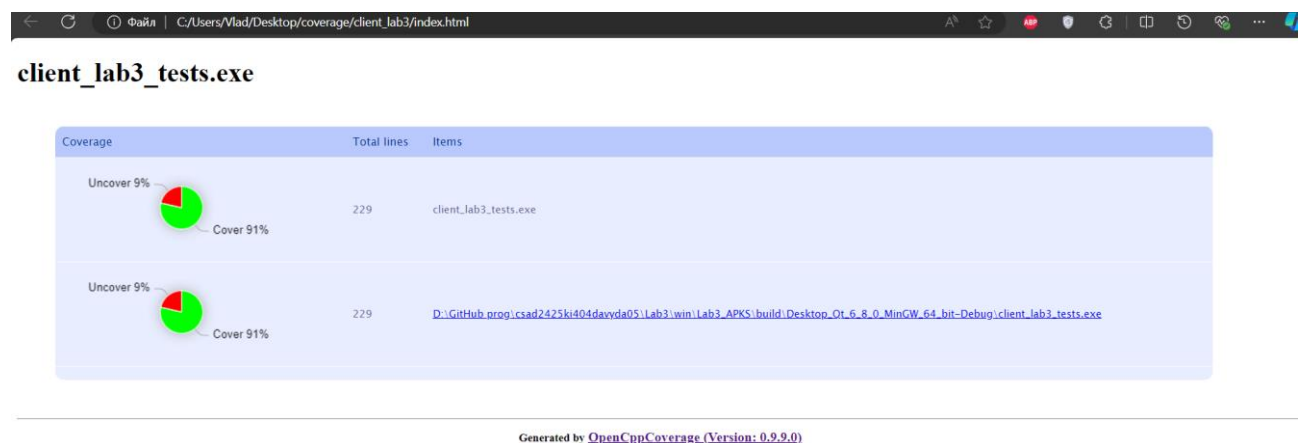


Рис.1

Відсоток покриття тестами сервера

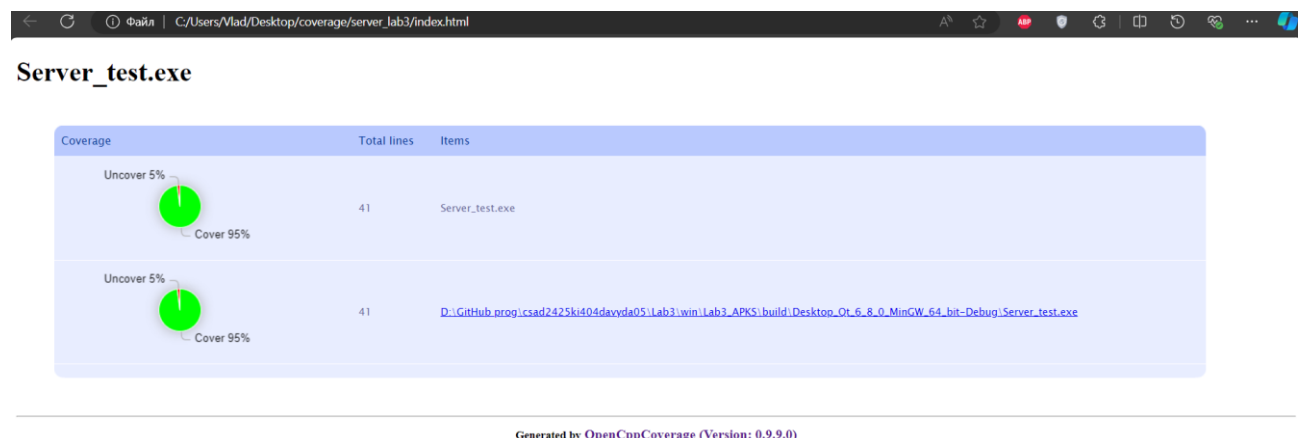
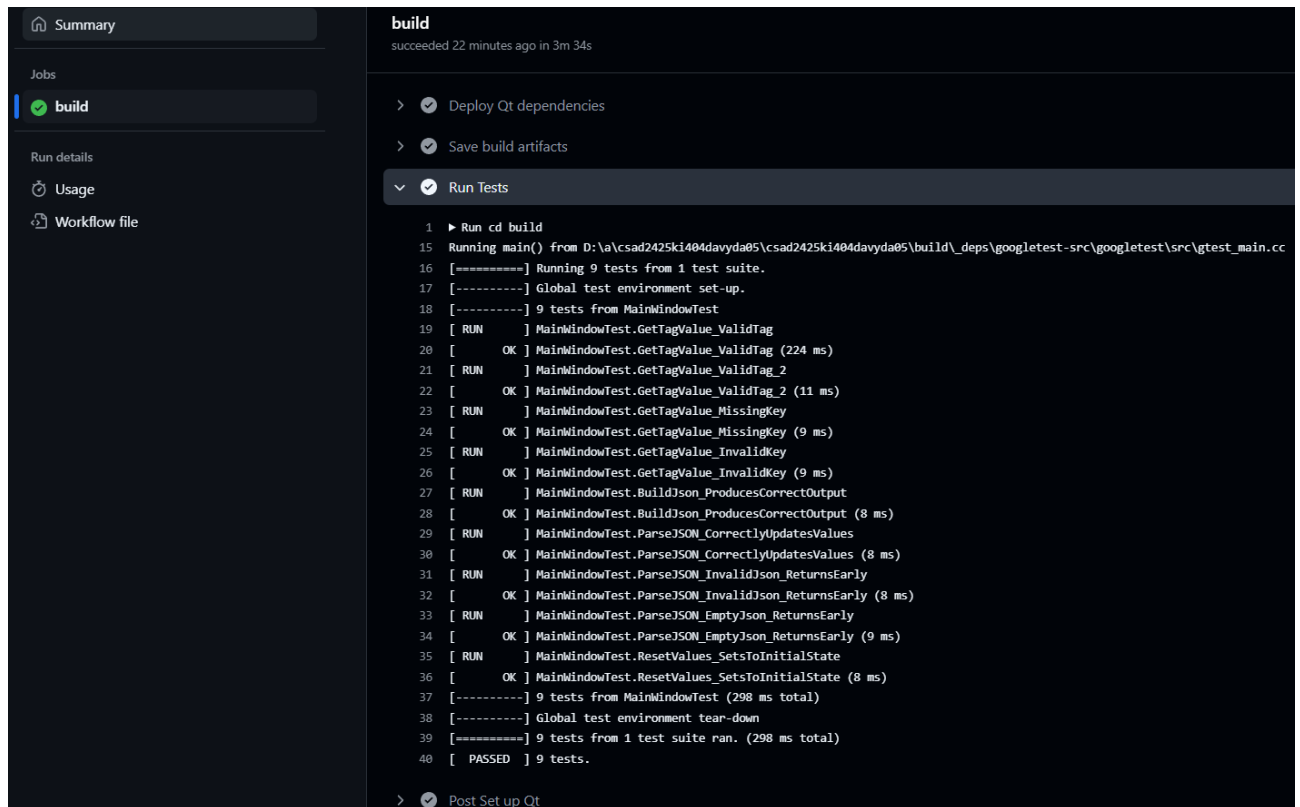


Рис.2

Зробив тести автоматизованими з запуском при збірці бінарників за допомогою github actions.



The screenshot displays the GitHub Actions interface for a workflow run. On the left sidebar, the 'build' job is selected under the 'Jobs' section. The 'Run Tests' step is expanded, showing a detailed log of test execution. The log indicates that 9 tests from the 'MainWindowTest' suite were run successfully, with a total duration of 298 ms. The tests include 'GetTagValue_ValidTag', 'GetTagValue_ValidTag_2', 'GetTagValue_MissingKey', 'GetTagValue_InvalidKey', 'BuildJson_ProducesCorrectOutput', 'ParseJSON_CorrectlyUpdatesValues', 'ParseJSON_InvalidJson_ReturnsEarly', 'ParseJSON_EmptyJson_ReturnsEarly', and 'ResetValues_SetsToInitialState'.

```
1 ▶ Run cd build
15 Running main() from D:\a\csad2425ki404davyda05\csad2425ki404davyda05\build\_deps\googletest-src\googletest\src\gtest_main.cc
16 [=====] Running 9 tests from 1 test suite.
17 [-----] Global test environment set-up.
18 [-----] 9 tests from MainWindowTest
19 [ RUN      ] MainWindowTest.GetTagValue_ValidTag
20 [       OK ] MainWindowTest.GetTagValue_ValidTag (224 ms)
21 [ RUN      ] MainWindowTest.GetTagValue_ValidTag_2
22 [       OK ] MainWindowTest.GetTagValue_ValidTag_2 (11 ms)
23 [ RUN      ] MainWindowTest.GetTagValue_MissingKey
24 [       OK ] MainWindowTest.GetTagValue_MissingKey (9 ms)
25 [ RUN      ] MainWindowTest.GetTagValue_InvalidKey
26 [       OK ] MainWindowTest.GetTagValue_InvalidKey (9 ms)
27 [ RUN      ] MainWindowTest.BuildJson_ProducesCorrectOutput
28 [       OK ] MainWindowTest.BuildJson_ProducesCorrectOutput (8 ms)
29 [ RUN      ] MainWindowTest.ParseJSON_CorrectlyUpdatesValues
30 [       OK ] MainWindowTest.ParseJSON_CorrectlyUpdatesValues (8 ms)
31 [ RUN      ] MainWindowTest.ParseJSON_InvalidJson_ReturnsEarly
32 [       OK ] MainWindowTest.ParseJSON_InvalidJson_ReturnsEarly (8 ms)
33 [ RUN      ] MainWindowTest.ParseJSON_EmptyJson_ReturnsEarly
34 [       OK ] MainWindowTest.ParseJSON_EmptyJson_ReturnsEarly (9 ms)
35 [ RUN      ] MainWindowTest.ResetValues_SetsToInitialState
36 [       OK ] MainWindowTest.ResetValues_SetsToInitialState (8 ms)
37 [-----] 9 tests from MainWindowTest (298 ms total)
38 [-----] Global test environment tear-down
39 [=====] 9 tests from 1 test suite ran. (298 ms total)
40 [ PASSED  ] 9 tests.
```

Рис.3.

Висновок:

У ході виконання лабораторної роботи розроблено тести для клієнтської і серверної частини програмного коду. Усі тести пройшли успішно.