

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Лабораторна робота № 5

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

на тему: «Впровадження тестів, що виконуються автоматично.»

Виконав:

ст. гр. КІ-410

Іванюк О.О.

Прийняв:

Кіцера А.О.

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

Вариант 7:

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

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.

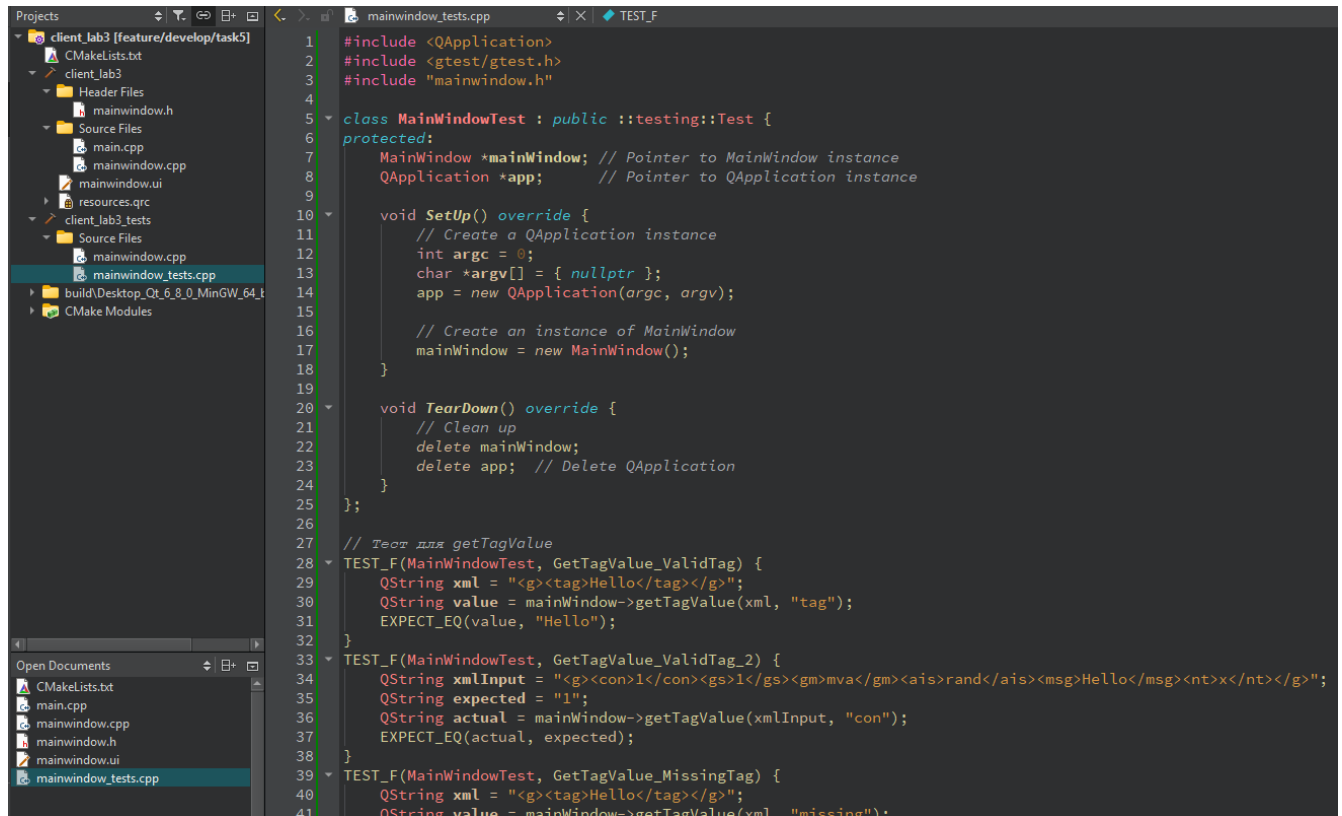
Для зроблення тестування більш систематичним та автоматизованим, розробники часто використовують спеціальні фреймворки для тестування, такі як `Google Test` або `Catch`. Ці фреймворки дозволяють створювати розширені тестові сценарії, вести звіти про результати тестування, та автоматизувати процес виявлення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

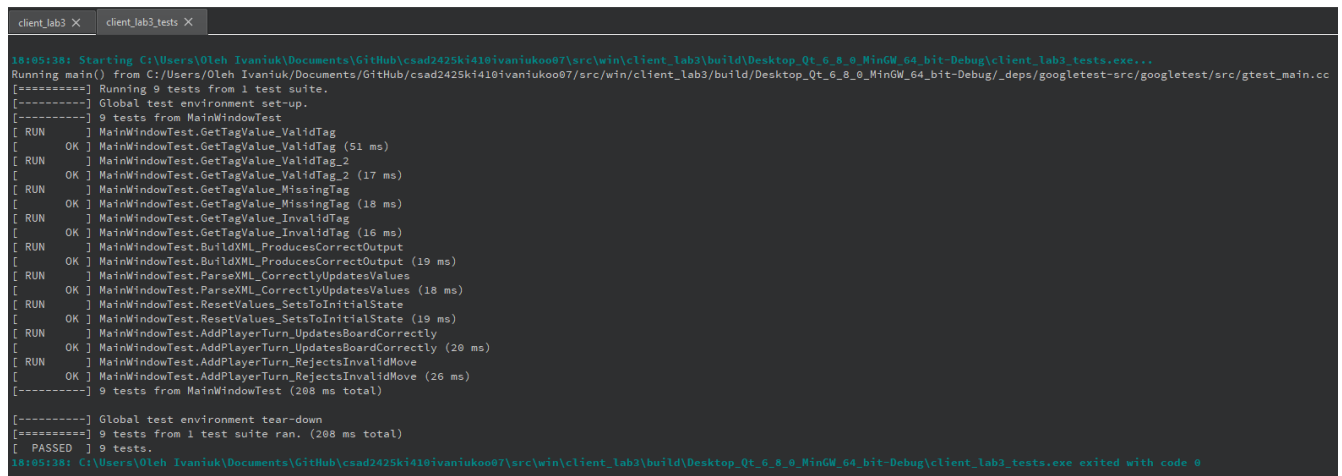
Хід роботи

1. Вніс тестові сценарії до файлу програмного коду для клієнтської частини.



```
1 #include <QApplication>
2 #include <gtest/gtest.h>
3 #include "mainwindow.h"
4
5 class MainWindowTest : public ::testing::Test {
6 protected:
7     MainWindow *mainWindow; // Pointer to MainWindow instance
8     QApplication *app;      // Pointer to QApplication instance
9
10 void SetUp() override {
11     // Create a QApplication instance
12     int argc = 0;
13     char *argv[] = { nullptr };
14     app = new QApplication(argc, argv);
15
16     // Create an instance of MainWindow
17     mainWindow = new MainWindow();
18 }
19
20 void TearDown() override {
21     // Clean up
22     delete mainWindow;
23     delete app; // Delete QApplication
24 }
25 };
26
27 // Тест для getTagValue
28 TEST_F(MainWindowTest, GetTagValue_ValidTag) {
29     QString xml = "<g><tag>Hello</tag></g>";
30     QString value = mainWindow->getTagValue(xml, "tag");
31     EXPECT_EQ(value, "Hello");
32 }
33
34 TEST_F(MainWindowTest, GetTagValue_ValidTag_2) {
35     QString xmlInput = "<g><con>1</con><gs>1</gs><gm>mva</gm><ais>rand</ais><msg>Hello</msg><nt>x</nt></g>";
36     QString expected = "1";
37     QString actual = mainWindow->getTagValue(xmlInput, "con");
38     EXPECT_EQ(actual, expected);
39 }
40
41 TEST_F(MainWindowTest, GetTagValue_MissingTag) {
42     QString xml = "<g><tag>Hello</tag></g>";
43     QString value = mainWindow->getTagValue(xml, "missing");
```

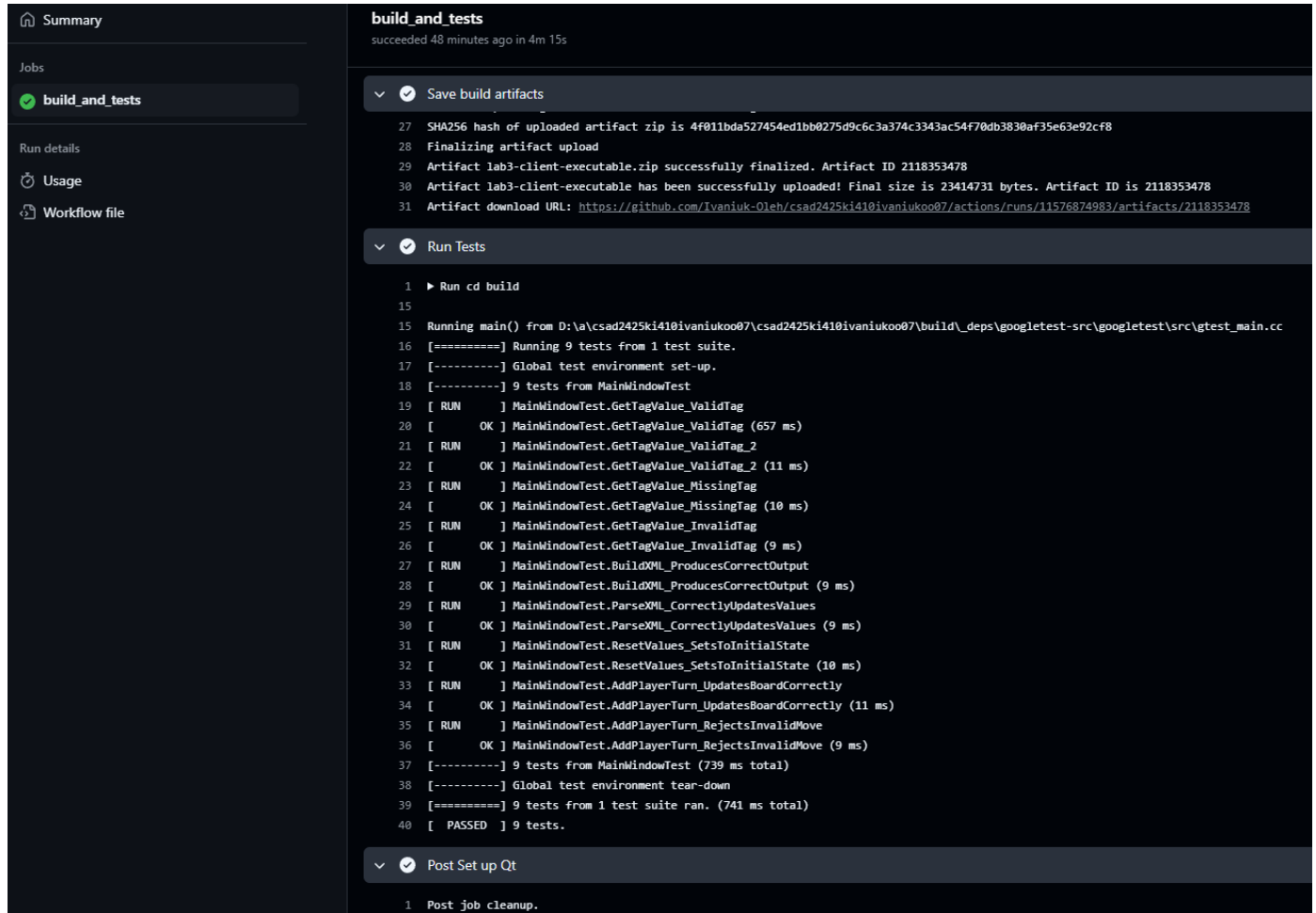
2. Здійснив виконання тестових сценаріїв.



```
18105138: Starting C:\Users\Oleh.Ivaniuk\Documents\GitHub\csad2425ki410ivaniukoo07\src\win\client_lab3\build\Desktop_Qt_6_8_0_MinGW_64_bit-Debug\client_lab3_tests.exe...
Running main() from C:\Users\Oleh.Ivaniuk\Documents\GitHub\csad2425ki410ivaniukoo07\src\win\client_lab3\build\Desktop_Qt_6_8_0_MinGW_64_bit-Debug\_deps\googletest-src\googletest\src\gtest_main.cc
[-----] Global test environment set-up.
[-----] 9 tests from MainWindowTest
[ RUN ] MainWindowTest.GetTagValue_ValidTag
[ OK ] MainWindowTest.GetTagValue_ValidTag (51 ms)
[ RUN ] MainWindowTest.GetTagValue_ValidTag_2
[ OK ] MainWindowTest.GetTagValue_ValidTag_2 (17 ms)
[ RUN ] MainWindowTest.GetTagValue_MissingTag
[ OK ] MainWindowTest.GetTagValue_MissingTag (18 ms)
[ RUN ] MainWindowTest.GetTagValue_InvalidTag
[ OK ] MainWindowTest.GetTagValue_InvalidTag (16 ms)
[ RUN ] MainWindowTest.BuildXML_ProducesCorrectOutput
[ OK ] MainWindowTest.BuildXML_ProducesCorrectOutput (19 ms)
[ RUN ] MainWindowTest.ParseXML_CorrectlyUpdatesValues
[ OK ] MainWindowTest.ParseXML_CorrectlyUpdatesValues (18 ms)
[ RUN ] MainWindowTest.ResetValues_SetsToInitialState
[ OK ] MainWindowTest.ResetValues_SetsToInitialState (19 ms)
[ RUN ] MainWindowTest.AddPlayerTurn_UpdatesBoardCorrectly
[ OK ] MainWindowTest.AddPlayerTurn_UpdatesBoardCorrectly (20 ms)
[ RUN ] MainWindowTest.AddPlayerTurn_RejectsInvalidMove
[ OK ] MainWindowTest.AddPlayerTurn_RejectsInvalidMove (26 ms)
[-----] 9 tests from MainWindowTest (208 ms total)

[-----] Global test environment tear-down
[-----] 9 tests from 1 test suite ran. (208 ms total)
[ PASSED ] 9 tests.
18105148: C:\Users\Oleh.Ivaniuk\Documents\GitHub\csad2425ki410ivaniukoo07\src\win\client_lab3\build\Desktop_Qt_6_8_0_MinGW_64_bit-Debug\client_lab3_tests.exe exited with code 0
```

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



The screenshot displays the GitHub Actions interface for a workflow named **build_and_tests**. The workflow is shown as **succeeded 48 minutes ago in 4m 15s**. The left sidebar contains navigation links: **Summary**, **Jobs**, **Run details**, **Usage**, and **Workflow file**. Under the **Jobs** section, the **build_and_tests** job is highlighted with a green checkmark. The main panel shows the job's steps and their logs.

Jobs

- build_and_tests

Run details

- Usage
- Workflow file

build_and_tests
succeeded 48 minutes ago in 4m 15s

Save build artifacts

```
27 SHA256 hash of uploaded artifact zip is 4f011bda527454ed1bb0275d9c6c3a374c3343ac54f70db3830af35e63e92cf8
28 Finalizing artifact upload
29 Artifact lab3-client-executable.zip successfully finalized. Artifact ID 2118353478
30 Artifact lab3-client-executable has been successfully uploaded! Final size is 23414731 bytes. Artifact ID is 2118353478
31 Artifact download URL: https://github.com/Ivaniuk-Oleh/csad2425ki410ivaniukoo07/actions/runs/11576874983/artifacts/2118353478
```

Run Tests

```
1 ▶ Run cd build
15
15 Running main() from D:\a\csad2425ki410ivaniukoo07\csad2425ki410ivaniukoo07\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 (657 ms)
21 [ RUN ] MainWindowTest.GetTagValue_ValidTag_2
22 [ OK ] MainWindowTest.GetTagValue_ValidTag_2 (11 ms)
23 [ RUN ] MainWindowTest.GetTagValue_MissingTag
24 [ OK ] MainWindowTest.GetTagValue_MissingTag (10 ms)
25 [ RUN ] MainWindowTest.GetTagValue_InvalidTag
26 [ OK ] MainWindowTest.GetTagValue_InvalidTag (9 ms)
27 [ RUN ] MainWindowTest.BuildXML_ProducesCorrectOutput
28 [ OK ] MainWindowTest.BuildXML_ProducesCorrectOutput (9 ms)
29 [ RUN ] MainWindowTest.ParseXML_CorrectlyUpdatesValues
30 [ OK ] MainWindowTest.ParseXML_CorrectlyUpdatesValues (9 ms)
31 [ RUN ] MainWindowTest.ResetValues_SetsToInitialState
32 [ OK ] MainWindowTest.ResetValues_SetsToInitialState (10 ms)
33 [ RUN ] MainWindowTest.AddPlayerTurn_UpdatesBoardCorrectly
34 [ OK ] MainWindowTest.AddPlayerTurn_UpdatesBoardCorrectly (11 ms)
35 [ RUN ] MainWindowTest.AddPlayerTurn_RejectsInvalidMove
36 [ OK ] MainWindowTest.AddPlayerTurn_RejectsInvalidMove (9 ms)
37 [-----] 9 tests from MainWindowTest (739 ms total)
38 [-----] Global test environment tear-down
39 [=====] 9 tests from 1 test suite ran. (741 ms total)
40 [ PASSED ] 9 tests.
```

Post Set up Qt

```
1 Post job cleanup.
```

4. Створив нову гілку feature/develop/task5. Створив Pull request для підтвердження змін в гілці develop, і надіслав запит на злиття викладачу.

task5 #5

Ivaniuk-Oleh wants to merge 12 commits into develop from feature/develop/task5

Conversation 0 Commits 12 Checks 1 Files changed 121 +11,769 -42

Ivaniuk-Oleh commented now

No description provided.

Ivaniuk-Oleh added 12 commits 5 days ago

- add doxygen documentation 3871eb8
- Update README.md 41cb5ab **Verified**
- add doxygen comments to arduino code 6cea7ef
- Merge branch 'feature/develop/task4' of https://github.com/Ivaniuk-Ol... d4cd485
- add lab4 report b31e537
- add client tests 96edfb4
- Update qt_client_build.yml db0e040
- Update qt_client_build.yml 5ac2ec7
- Update qt_client_build.yml d856310
- run test after build 1817e88
- add additional tests 6b4f6f7
- Update qt_client_build.yml 05f03e6

Ivaniuk-Oleh requested a review from anton-o-kitsera now

Reviewers

anton-o-kitsera

Still in progress? Learn about draft PRs

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

Висновок:

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