

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЛЬВІВСЬКА ПОЛІТЕХНІКА**



**АВТОМАТИЗОВАНЕ  
ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ  
СИСТЕМ**

Лабораторна робота №3  
“Створення клієнт-серверної  
гри Камінь-Ножиці-Папір”

**Виконав:**

студент гр. КІ-404  
Попов М. Ю.

**Прийняв:**

аспірант  
Федак П. Р.

## Завдання:

**Implement Server (HW) and Client (SW) parts of game (FEF)**

1. Develop Server and Client.
2. Required steps.

### ***Required steps for all tasks:***

1. Add/modify UTs
2. Create/Modify the **ci** script to build, test and run project.
3. Create/Modify the **YML** file.
4. Commit changes to **feature/develop/<task number>**.
5. Update **README** with details:
  - how to build project;
  - how to run project;
  - Version number.
6. Create **GIT TAG**:  
  
    <PROJECT NAME>\_<task number>\_<VERSION>\_ww<YYWWD>
  - YY – current year;
  - WW – work week;
  - D – current day number of weak.
7. Create pull request with name <task number> and submit **lecturer** as reviewer.
8. After the reviewer approved – merge into the develop branch.

# Виконання

## 1) Створення клієнтської частини програми:

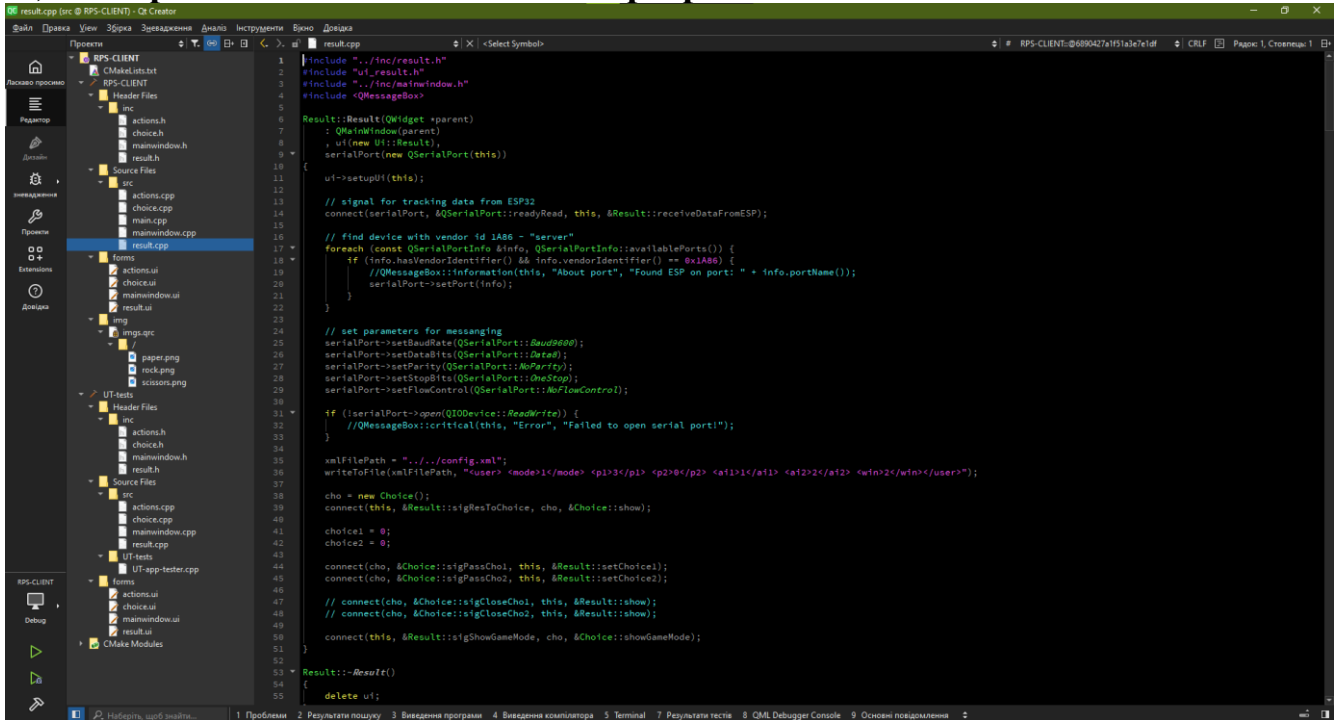
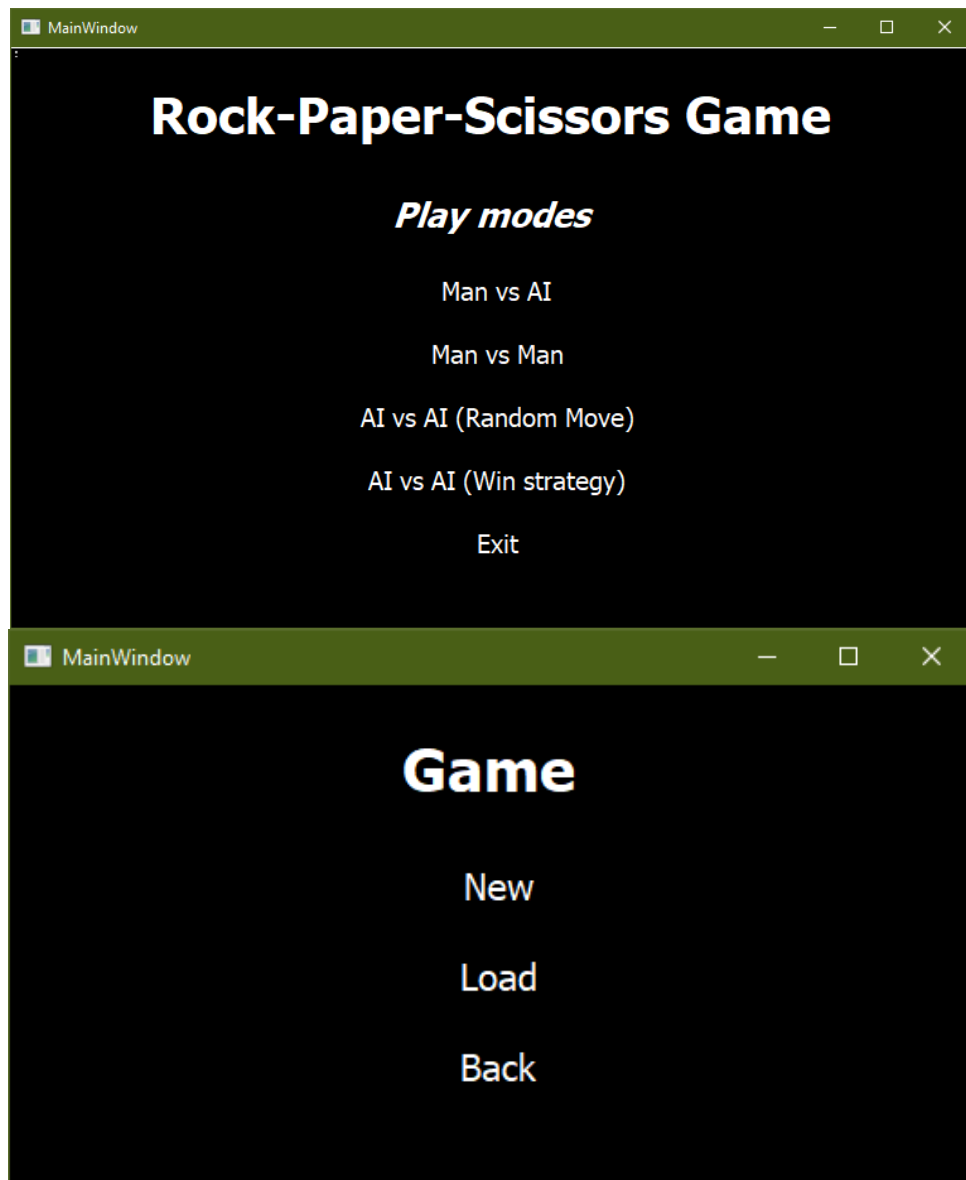
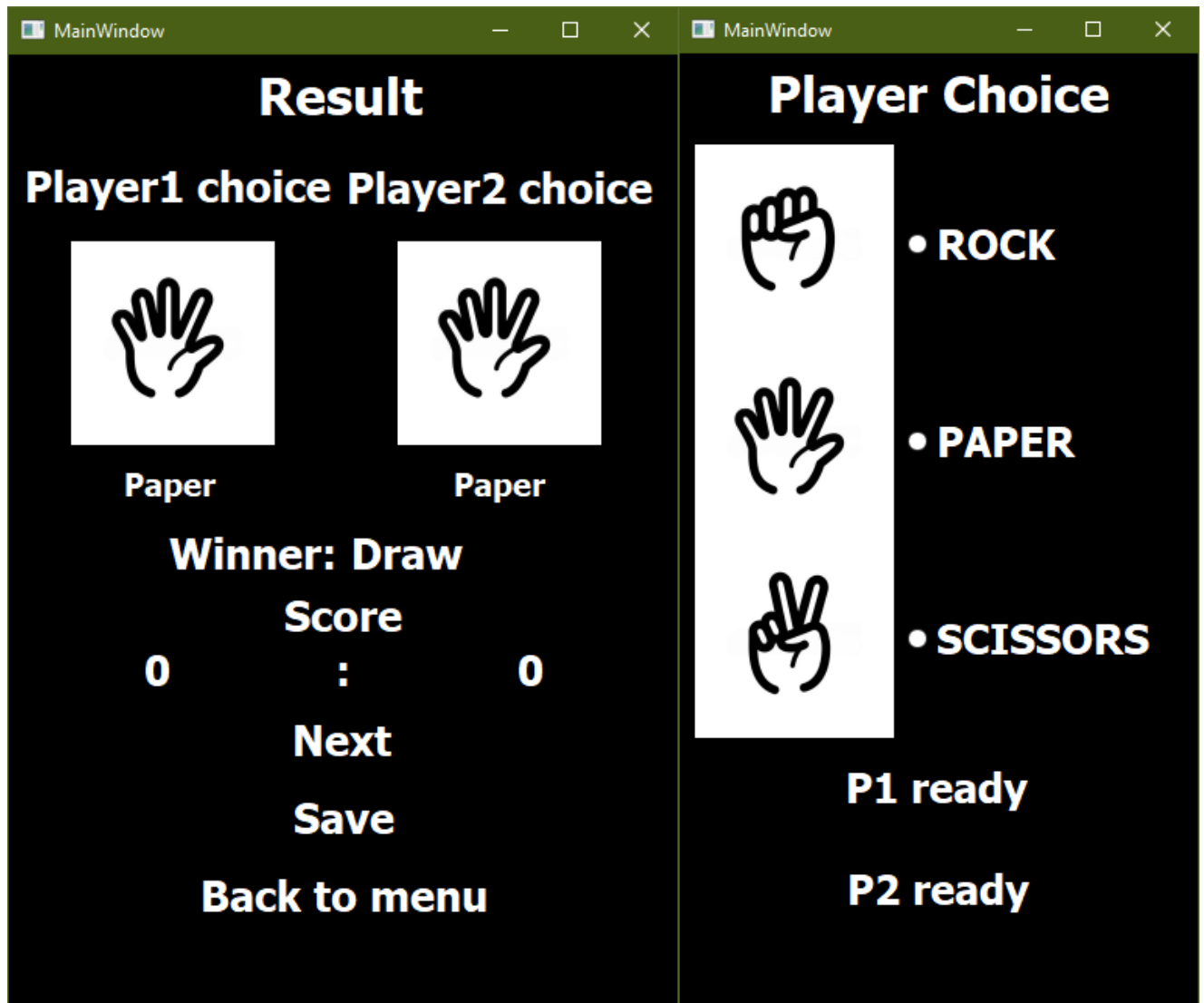


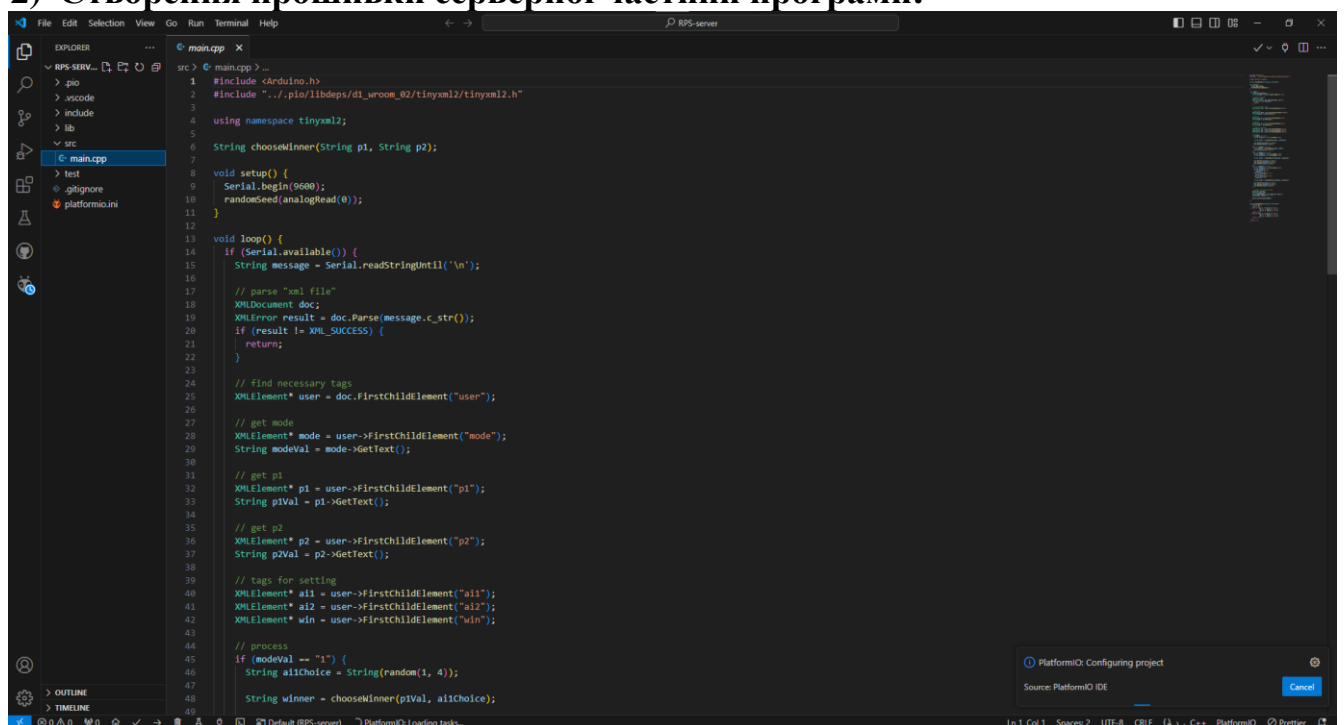
Рис. 1. Середовище розробки Qt creator з відкритим проектом



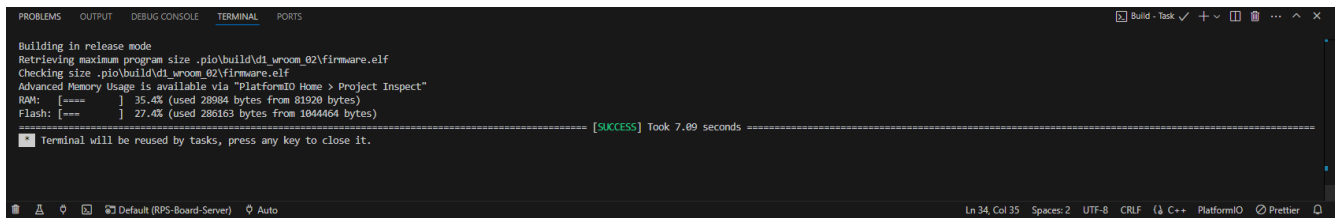


*Рис. 2.* Результат: графічна програма-клієнт (вікна з вибором режиму, завантаження/нова гра, власне вікно гри та вікно вибору)

## 2) Створення прошивки серверної частини програми:

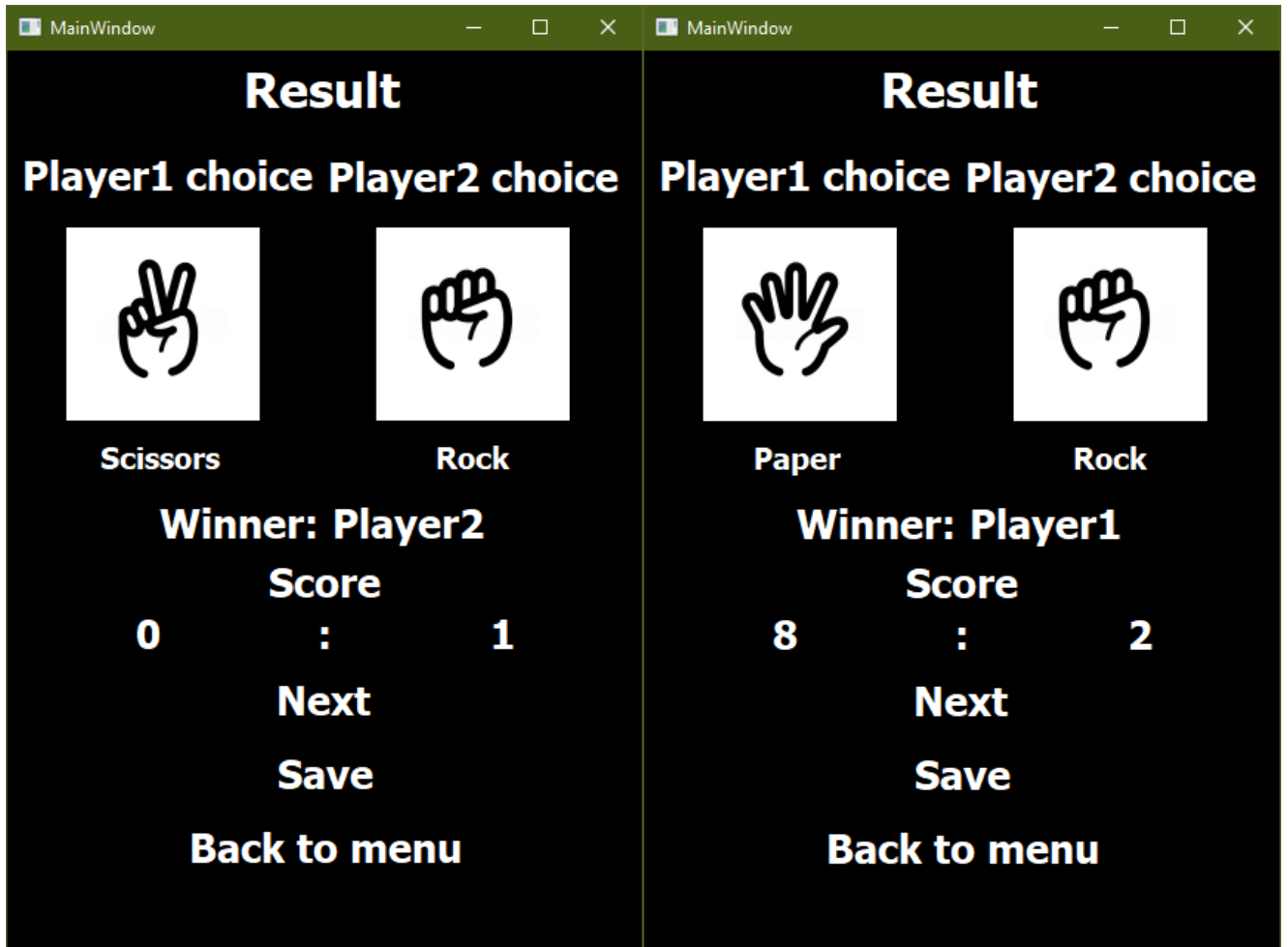


*Рис. 3.* Середовище розробки VS Code з плагіном PlatformIO та відкритим проектом



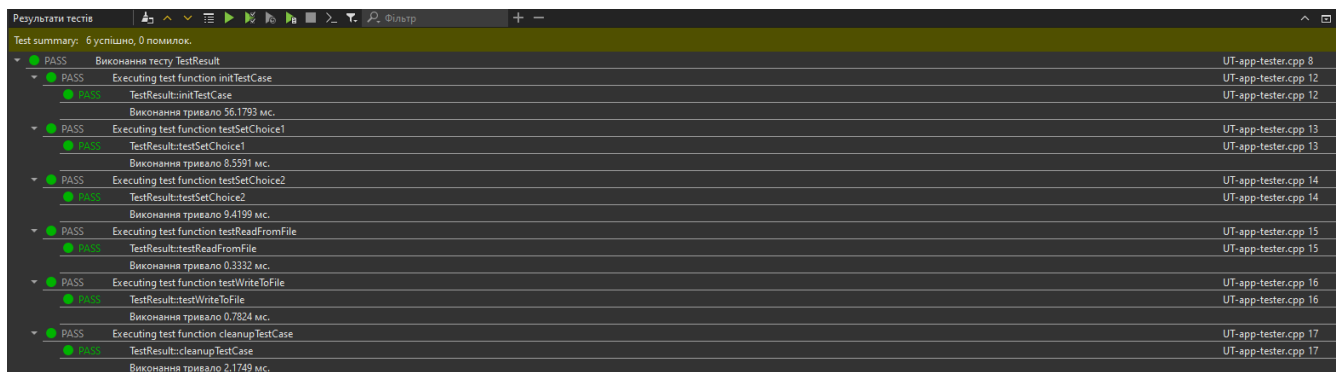
*Рис. 4.* Білд та завантаження програми на плату

### 3) Результат роботи програми:



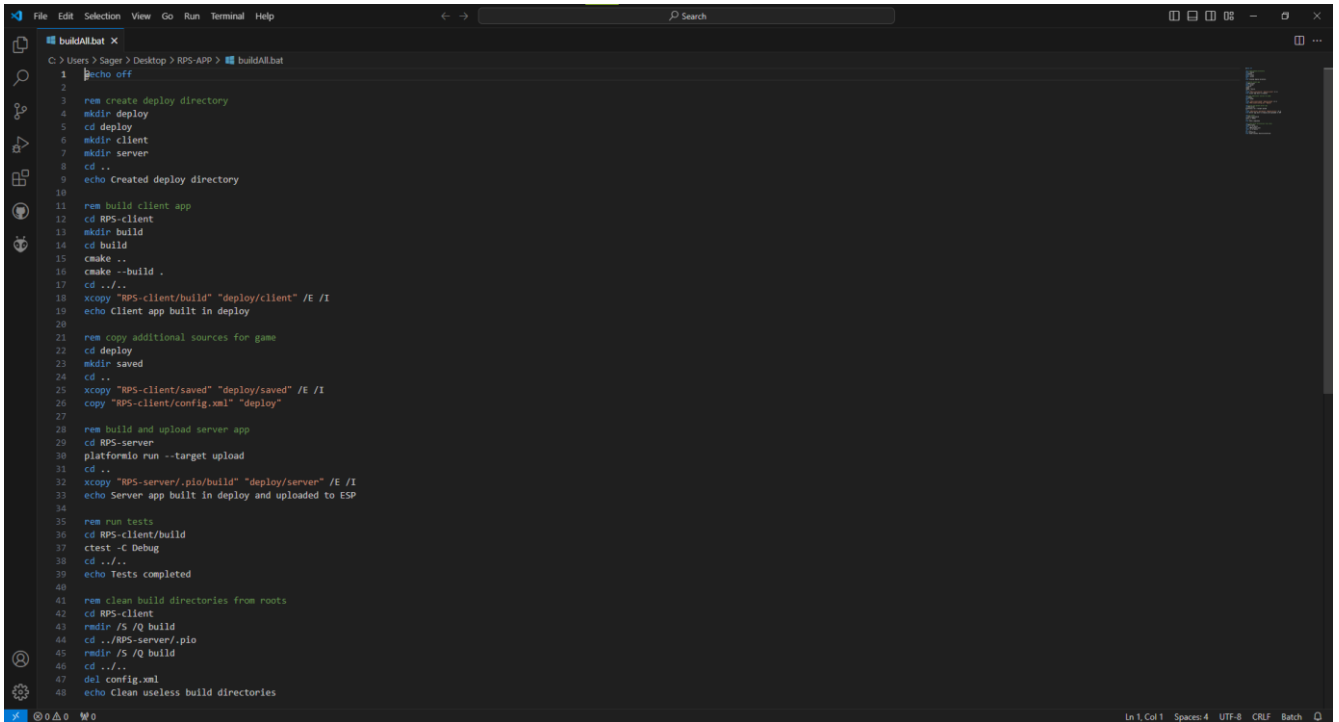
*Рис. 5.* Процес гри в одному з режимів

### 4) Результат роботи тестів:



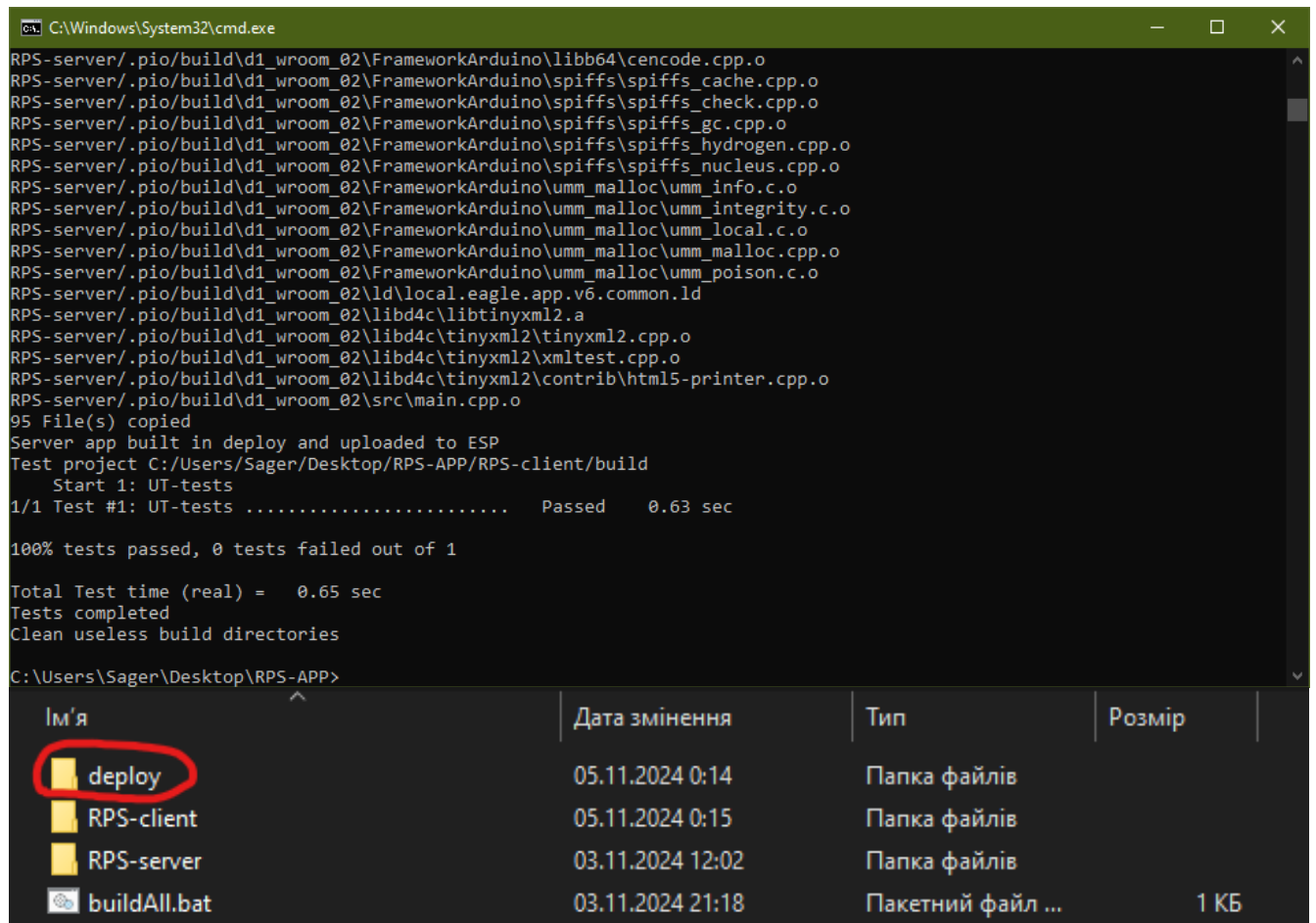
*Рис. 6.* Вікно, з успішним результатом тестів

## 5) Створення .bat файла для локальної збірки проекту і збірка:



```
1 echo off
2
3 rem create deploy directory
4 mkdir deploy
5 cd deploy
6 mkdir client
7 mkdir server
8 cd ..
9 echo Created deploy directory
10
11 rem build client app
12 cd RPS-client
13 mkdir build
14 cd build
15 cmake ..
16 cmake --build .
17 cd ../..
18 xcopy "RPS-client/build" "deploy/client" /E /I
19 echo Client app built in deploy
20
21 rem copy additional sources for game
22 cd deploy
23 mkdir saved
24 cd ..
25 xcopy "RPS-client/saved" "deploy/saved" /E /I
26 copy "RPS-client/config.xml" "deploy"
27
28 rem build and upload server app
29 cd RPS-server
30 platformio run --target upload
31 cd ..
32 xcopy "RPS-server/.pio/build" "deploy/server" /E /I
33 echo Server app built in deploy and uploaded to ESP
34
35 rem run tests
36 cd RPS-client/build
37 ctest -C Debug
38 cd ../..
39 echo Tests completed
40
41 rem clean build directories from roots
42 cd RPS-client
43 rmdir /S /Q build
44 cd ../RPS-server/.pio
45 rmdir /S /Q build
46 cd ../..
47 del config.xml
48 echo Clean useless build directories
```

Рис. 7. Створений .bat файл



```
C:\Windows\System32\cmd.exe
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\libb64\cencode.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\spiffs\spiffs_cache.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\spiffs\spiffs_check.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\spiffs\spiffs_gc.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\spiffs\spiffs_hydrogen.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\spiffs\spiffs_nucleus.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\umm_malloc\umm_info.c.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\umm_malloc\umm_integrity.c.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\umm_malloc\umm_local.c.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\umm_malloc\umm_malloc.cpp.o
RPS-server/.pio/build/d1_wroom_02\FrameworkArduino\umm_malloc\umm_poison.c.o
RPS-server/.pio/build/d1_wroom_02\ld\local.eagle.app.v6.common.ld
RPS-server/.pio/build/d1_wroom_02\libd4c\libtinyxml2.a
RPS-server/.pio/build/d1_wroom_02\libd4c\tinyxml2\tinyxml2.cpp.o
RPS-server/.pio/build/d1_wroom_02\libd4c\tinyxml2\xmltest.cpp.o
RPS-server/.pio/build/d1_wroom_02\libd4c\tinyxml2\contrib\html5-printer.cpp.o
RPS-server/.pio/build/d1_wroom_02\src\main.cpp.o
95 File(s) copied
Server app built in deploy and uploaded to ESP
Test project C:/Users/Sager/Desktop/RPS-APP/RPS-client/build
Start 1: UT-tests
1/1 Test #1: UT-tests ..... Passed    0.63 sec

100% tests passed, 0 tests failed out of 1

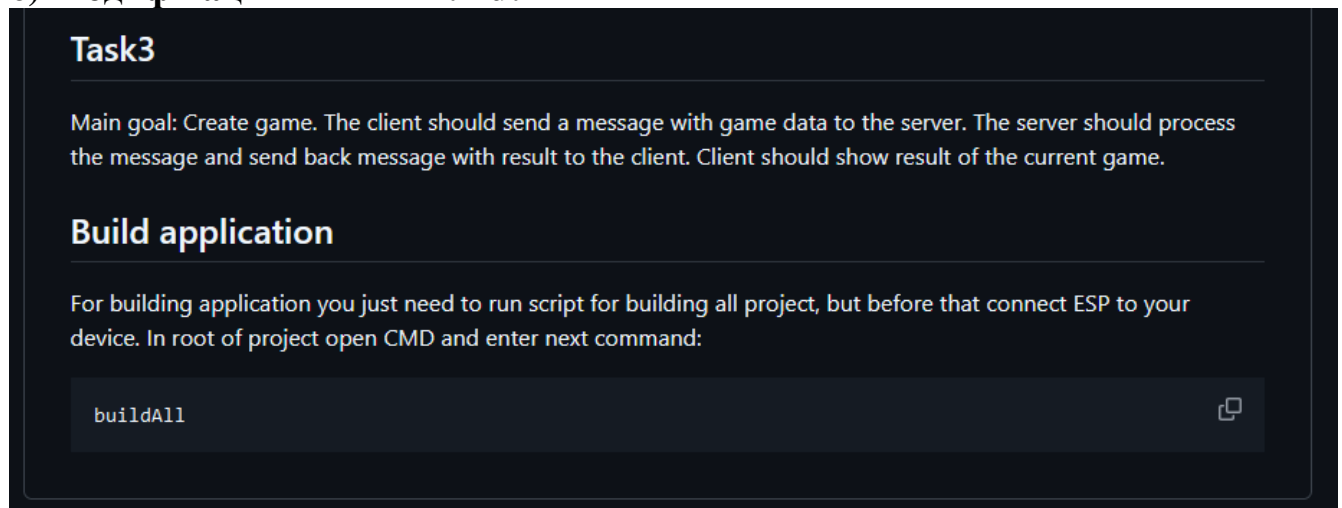
Total Test time (real) = 0.65 sec
Tests completed
Clean useless build directories

C:\Users\Sager\Desktop\RPS-APP>
```

Ім'я	Дата змінення	Тип	Розмір
deploy	05.11.2024 0:14	Папка файлів	
RPS-client	05.11.2024 0:15	Папка файлів	
RPS-server	03.11.2024 12:02	Папка файлів	
buildAll.bat	03.11.2024 21:18	Пакетний файл ...	1 КБ

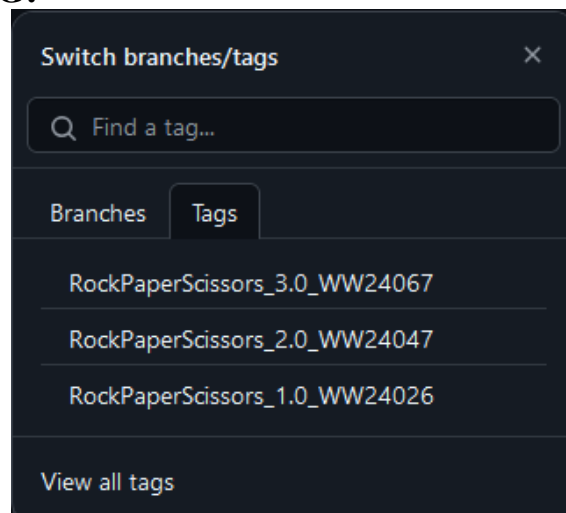
Рис. 8. Результат роботи .bat файла для автоматизованої збірки

## 6) Модифікація README.md:



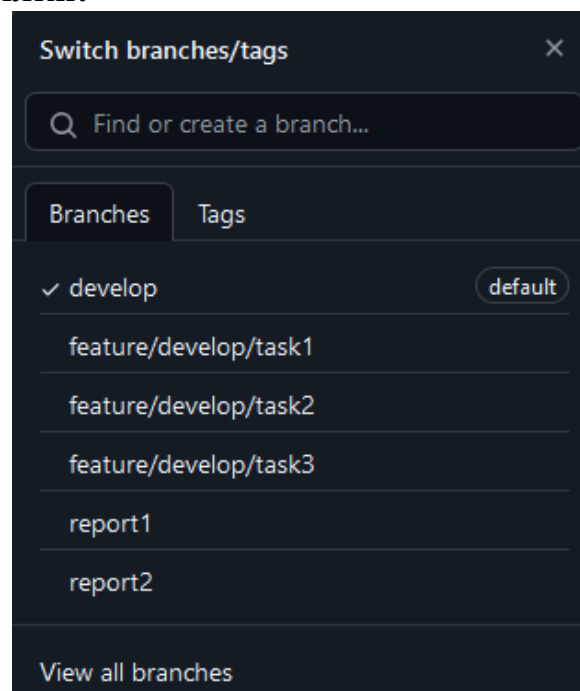
*Рис. 9.* Змінений README.md

## 7) Створення GIT TAG:



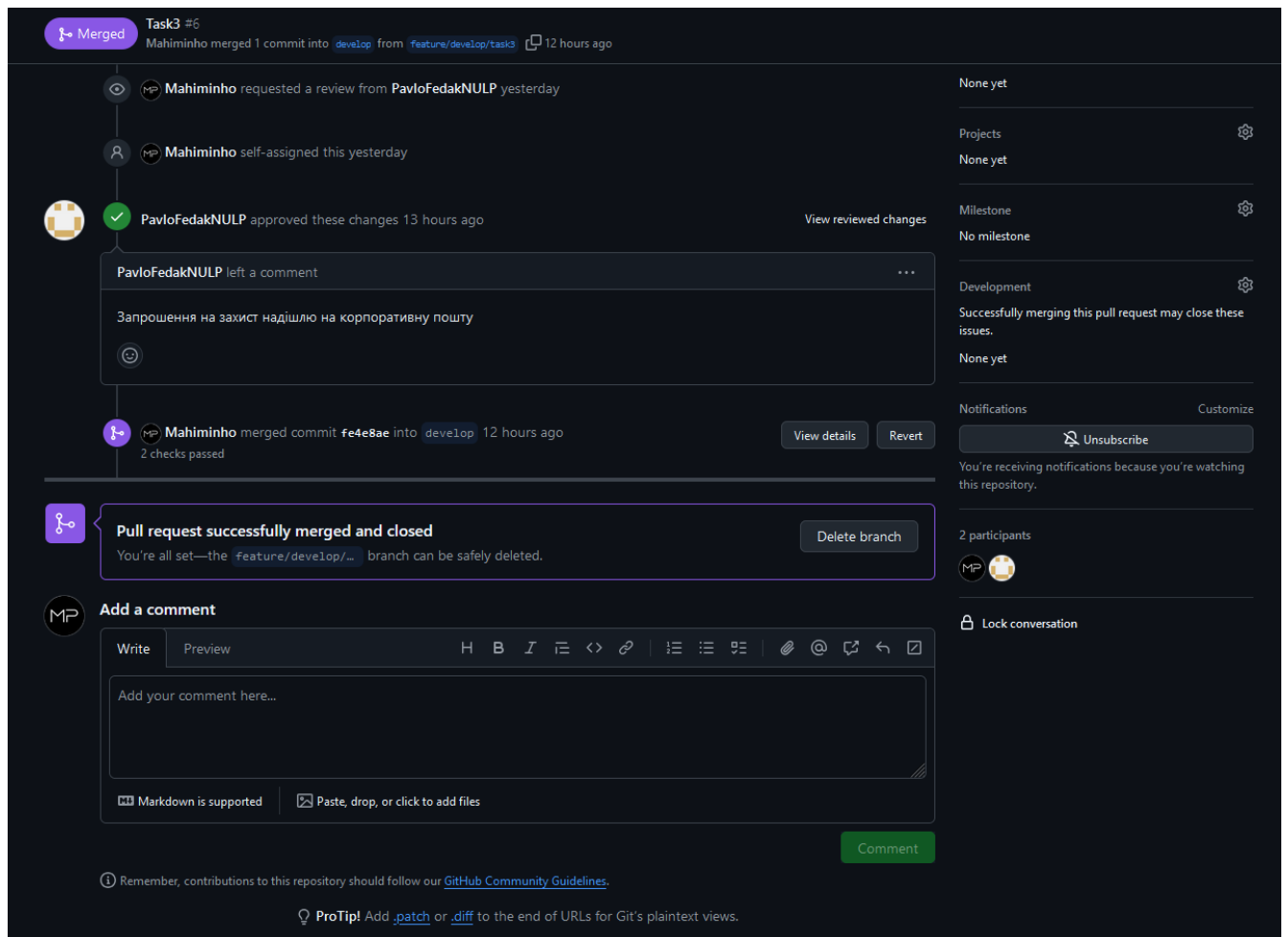
*Рис. 10.* Створений GIT TAG

## 8) Створення нової гілки:



*Рис. 11.* Створена нова гілка

## 9) Створення пул реквесту, отримання апруву та мердж в головну гілку:



*Рис. 12.* Отриманий апрув в запит на пул реквест, та мердж в develop

### Висновок:

Під час виконання даної лабораторної роботи, я познайомився з скриптуванням для локальної збірки проекту та пригадав CI та Unit tests, навчився та попрактикував використання усіх попередньозгаданих технологій, а також написав гру камінь-ножиці-папір, яка має 4 режими: людина проти AI, людина проти людини та два режими AI проти AI, але з різною тактикою, та підтримує збереження гри.