Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра СКС



**Звіт**

З лабораторної роботи №5

З дисципліни: «Автоматизоване проектування комп’ютерних систем»

На тему: «Тестування проекту»

Виконав:

ст.гр. КІ-403

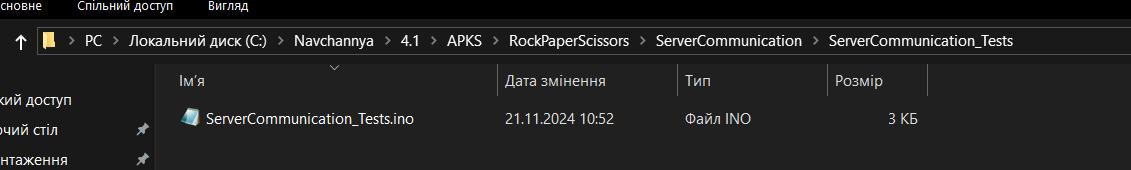
Соболь К.Д.

Перевірив:

Кіцера А.О.

**Львів 2024**

**Хід роботи**

**  
Рис. 1 Створив тест для серверної частини**

#include <Arduino.h>

#include <ArduinoJson.h>

#include <AUnit.h>

enum Move { Rock, Paper, Scissors };

// Функції, які тестуються

Move getStupidMove(Move playerMove) {

  switch (playerMove) {

    case Rock: return Scissors;

    case Paper: return Rock;

    case Scissors: return Paper;

    default: return getRandomMove();

  }

}

Move getRandomMove() {

  int randMove = random(0, 3);

  return static\_cast<Move>(randMove);

}

Move getCounterMove(Move playerMove) {

  switch (playerMove) {

    case Rock: return Paper;

    case Paper: return Scissors;

    case Scissors: return Rock;

    default: return getRandomMove();

  }

}

String getMoveString(Move move) {

  switch (move) {

    case Rock: return "Rock";

    case Paper: return "Paper";

    case Scissors: return "Scissors";

    default: return "Unknown";

  }

}

Move getMoveFromString(String move) {

  move.trim();

  if (move == "Rock") return Rock;

  if (move == "Paper") return Paper;

  if (move == "Scissors") return Scissors;

  return Rock;

}

void setup() {

  Serial.begin(9600);

  randomSeed(analogRead(0));

  delay(2000);

  Serial.println("Setup complete. Running tests...");

}

void loop() {

  // Запускаємо тести. AUnit сама обробляє результати.

  aunit::TestRunner::run();

}

// --- Тести ---

test(getStupidMove\_Rock) {

  assertEqual(getStupidMove(Rock), Scissors);

}

test(getStupidMove\_Paper) {

  assertEqual(getStupidMove(Paper), Rock);

}

test(getStupidMove\_Scissors) {

  assertEqual(getStupidMove(Scissors), Paper);

}

test(getRandomMove) {

  for (int i = 0; i < 10; ++i) {

    Move move = getRandomMove();

    assertTrue(move == Rock || move == Paper || move == Scissors);

  }

}

test(getCounterMove\_Rock) {

  assertEqual(getCounterMove(Rock), Paper);

}

test(getCounterMove\_Paper) {

  assertEqual(getCounterMove(Paper), Scissors);

}

test(getCounterMove\_Scissors) {

  assertEqual(getCounterMove(Scissors), Rock);

}

test(getMoveString\_Rock) {

  assertEqual(getMoveString(Rock), "Rock");

}

test(getMoveString\_Paper) {

  assertEqual(getMoveString(Paper), "Paper");

}

test(getMoveString\_Scissors) {

  assertEqual(getMoveString(Scissors), "Scissors");

}

test(getMoveFromString\_Rock) {

  assertEqual(getMoveFromString("Rock"), Rock);

}

test(getMoveFromString\_Paper) {

  assertEqual(getMoveFromString("Paper"), Paper);

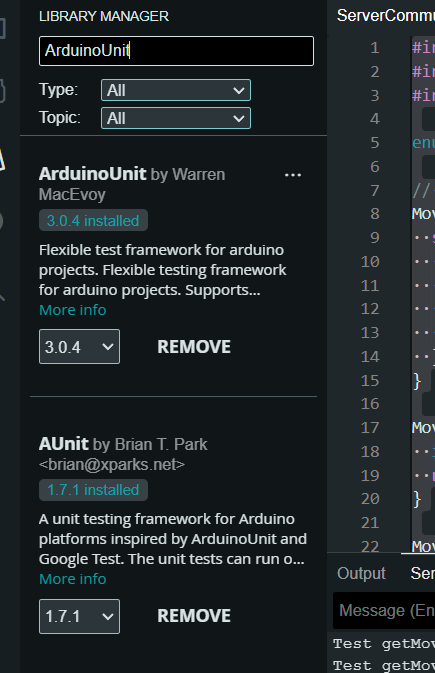
}

test(getMoveFromString\_Scissors) {

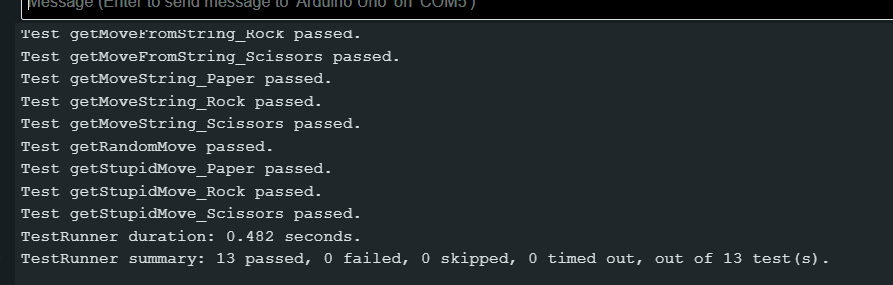
  assertEqual(getMoveFromString("Scissors"), Scissors);

}

Написав скетч з юніт тестами.



**Рис. 2 Завантажив необхідні бібліотеки**

****

**Рис. 3 Тести пройдені успішно**

Висновок: Я протестував проект за допомогою unit тестів.