



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки

Розробка програмного забезпечення на платформі Node.JS

Лабораторна робота №7

“ Розгортання програмного забезпечення”

Виконали:

студенти групи ІА–24:

Бондар А. Д.

Кармазіна А. В.

Перевірив:

Нечай Д. О.

Київ 2025

Тема: Калькулятор характеристик та пошкоджень

Огляд проєкту

Цей проєкт має на меті створити калькулятор для обчислення характеристик та пошкоджень персонажів. Додаток дозволить користувачам обрати персонажа з наперед визначеного списку (збереженого в базі даних) та розрахувати підсумкову шкоду, враховуючи різні фактори, такі як атака, резисти, таланти та модифікатори. Додатково калькулятор забезпечить обчислення шкоди за секунду (DPS) на основі швидкості атаки вибраного персонажа.

Цей інструмент допоможе гравцям аналізувати та оптимізувати шкоду, експериментуючи з різними налаштуваннями та умовами. Він надасть зручний інтерфейс для візуалізації та налаштування цих значень у реальному часі.

Завдання:

- Налаштувати CI. При створенні нового ПР або коміта в ньому — необхідно перевірити код на відповідність обраному стилю(рекомендую prettier), відсутність помилок (linter\statyc analyzer eslint\tslint), те що він ""збирається-билдиться"", тести проходять(поки можна стаб який проходить завжди), а коміти відповідають обраному стилю описання (Рекомендую conventional commits). Github налаштувати так, щоб змерджити ПР можна було лише, якщо пройшов CI.
- Налаштувати CD. При додаванні нових комітів в обрану гілку (наприклад мастер), необхідно автоматично розгортати нову версію ПО на виділеному Стейдж-сервері, або оновлювати її в маркеті.
- Сервіс доступний з інтернету

Хід роботи

```
"scripts": {  
  "start": "node index.js",  
  "test": "mocha",  
  "build": "echo \"No build step specified\" && exit 0",  
  "dev": "nodemon index.js",  
  "format": "prettier --write .",  
  "lint": "eslint .",  
  "prepare": "husky install",  
  "commitlint": "commitlint --edit"  
},
```

ESLint, Prettier встановлені наче ще з другої лабораторної роботи

```
PS C:\Styding\Node\ocd> npm run format
```

```
> ocd@1.0.0 format
```

```
> prettier --write .
```

```
.prettierrc 63ms
commitlint.config.js 8ms
eslint.config.mjs 15ms
index.js 14ms (unchanged)
package-lock.json 108ms
package.json 2ms
public/index.html 66ms (unchanged)
public/script.js 101ms (unchanged)
public/styles.css 48ms (unchanged)
README.md 41ms (unchanged)
server/server.js 28ms (unchanged)
test/test.js 3ms
testDbConnection.js 6ms (unchanged)
workflows/ci.yml 23ms
PS C:\Styding\Node\ocd> npm run lint
```

```
> ocd@1.0.0 lint
```

```
> eslint .
```

```
PS C:\Styding\Node\ocd> █
```

Все працює

```
PS C:\Styding\Node\ocd> git commit -m "L7 test1"
```

```
> ocd@1.0.0 format
```

```
> prettier --write .
```

```
npm error npx canceled due to missing packages and no YES option: ["commitlint@19.6.1"]
```

```
npm error A complete log of this run can be found in: C:\Users\Void\AppData\Local\npm-cache\_logs\2025-01-19T11_11_49_909Z-debug-0.log
```

```
husky - commit-msg hook exited with code 1 (error)
```

```
PS C:\Styding\Node\ocd> █
```

Можливо занадто

```
[detached HEAD 02a76d0] test: тестування комітів для лаби 7, номер 1
Author: BAntonD <121023898+BAntonD@users.noreply.github.com>
Date: Sun Jan 19 12:48:35 2025 +0200
```

Тепер наче все добре, можна переходити до додавання цього на гітхаб

```
name: CI Workflow

on:
  pull_request:
    branches:
      - main # або інша ваша основна гілка

jobs:
  lint_and_test:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Set up Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '22' # Ваша версія Node.js

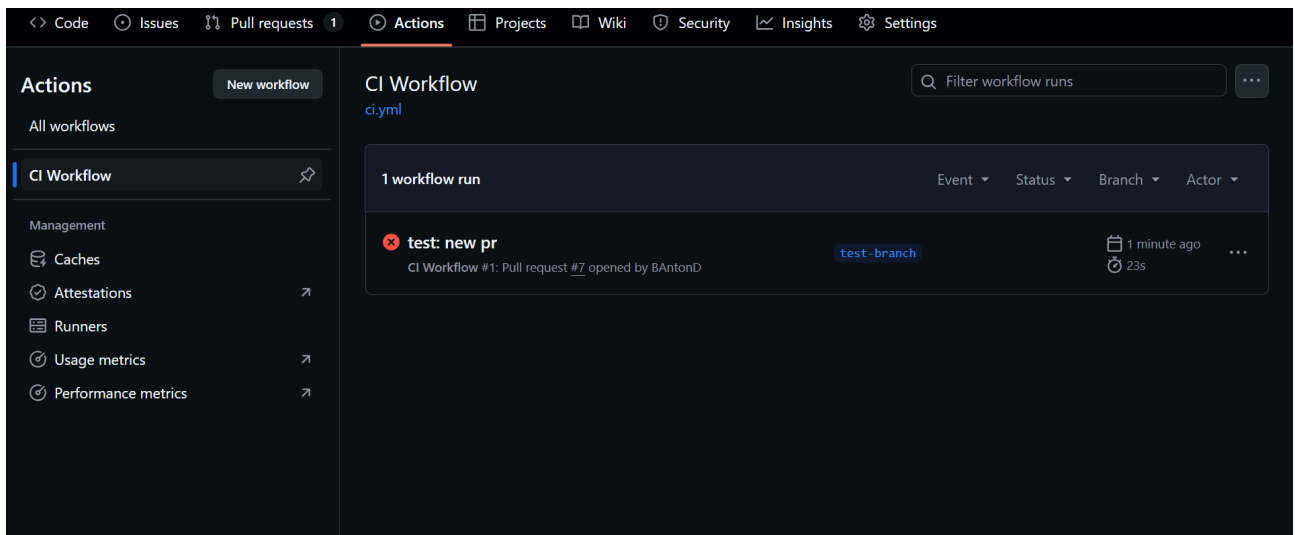
      - name: Install dependencies
        run: npm install

      - name: Run ESLint
        run: npm run lint

      - name: Run Prettier
        run: npm run prettier --check .

      - name: Run tests
        run: npm test # Додайте тут ваш тест
```

Створюємо ci.yml



CI працює, хоча не успішно але це можна виправити

- **name:** Install Husky globally
run: npm install -g husky
- **name:** Install dependencies
run: npm install
- **name:** Install Husky
run: npx husky install
- **name:** Ensure Husky files are executable
run: chmod -R +x .husky/
- **name:** Ensure ESLint is executable
run: chmod +x node_modules/.bin/eslint
- **name:** Run ESLint
run: npx eslint .
- **name:** Ensure Prettier is executable
run: chmod +x node_modules/.bin/prettier
- **name:** Run Prettier
run: npm run format -- --check .
- **name:** Ensure Mocha is executable
run: chmod +x node_modules/.bin/mocha

Додаю Ensure

15 workflow runs		Event ▾	Status ▾	Branch ▾	Actor ▾
✓	test: new pr	CI Workflow #15: Pull request #7 synchronize by BAntonD		test-branch	now 18s
✗	test: new pr	CI Workflow #14: Pull request #7 synchronize by BAntonD		test-branch	2 minutes ago 22s
✗	test: new pr	CI Workflow #13: Pull request #7 synchronize by BAntonD		test-branch	18 minutes ago 21s

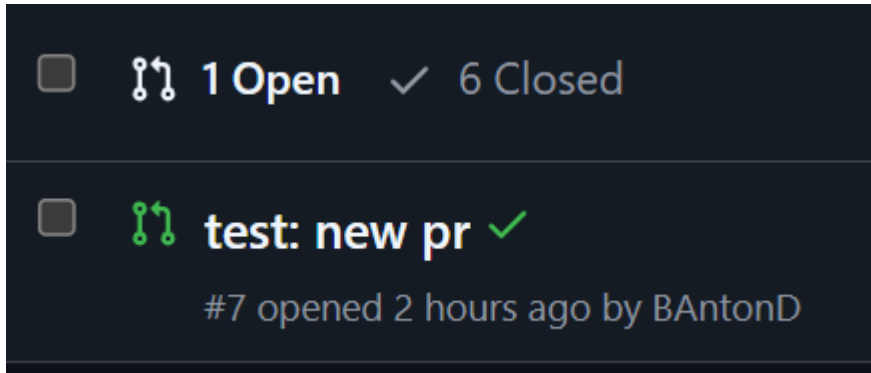
ci.yml
on: pull_request

✓ **lint_and_test** 10s

✓	Set up job	0s
✓	Checkout repository	2s
✓	Set up Node.js	0s
✓	Ensure Husky files are executable	0s
✓	Install Husky globally	0s
✓	Install dependencies	2s
✓	Install Husky	0s
✓	Ensure Husky files are executable	0s
✓	Ensure ESLint is executable	0s
✓	Run ESLint	0s
✓	Ensure Prettier is executable	0s
✓	Run Prettier	1s

✓	Ensure Mocha is executable	0s
✓	Run tests	0s
✓	Post Set up Node.js	1s
✓	Post Checkout repository	0s
✓	Complete job	0s

І через велику кількість спроб все починає працювати



Мерджу цей пулл реквест

```
on:
  push:
    branches:
      - main # або інша основна гілка
  pull_request:
    branches:
      - main # або інша основна гілка
```

Додаю для гілки мейн перевірку

Branch protection rule



Protect your most important branches

[Branch protection rules](#) define whether collaborators can delete or force push to the branch and set requirements for any pushes to the branch, such as passing status checks or a linear commit history.

[Your GitHub Free plan](#) can only enforce rules on its public repositories, like this one.

Branch name pattern *

main

Protect matching branches

☐ Require a pull request before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

☒ Require status checks to pass before merging

Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

☒ Require branches to be up to date before merging

This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

lint_and_test

Status checks that are required

lint_and_test



GitHub Actions



☒ Require deployments to succeed before merging

Choose which environments must be successfully deployed to before branches can be merged into a branch that matches this rule.

Налаштовую CI для всіх.

Наче з CI закінчено


```

name: CD Pipeline

on:
  push:
    branches:
      - main # або ваша основна гілка

jobs:
  deploy:
    runs-on: ubuntu-latest







    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Set up SSH key
        uses: webfactory/ssh-agent@v0.5.3
        with:
          ssh-private-key: ${ secrets.SSH_KEY } # Приватний ключ з секретів GitHub

      - name: Deploy to Server
        run: |
          ssh -o StrictHostKeyChecking=no ${ secrets.SSH_USER }@${ secrets.SSH_HOST } << 'EOF'
          cd /path/to/your/project
          git pull origin main
          # додайте сюди команду для перезапуску вашого додатку, наприклад:
          npm install
          pm2 restart your-app
          EOF

```

Прописую deploy.yml

	test: test	main	 1 minute ago	...
	CD Pipeline #1: Commit 5e55e65 pushed by BAntonD		 12s	
	test: test	main	 1 minute ago	...
	CI Workflow #19: Commit 5e55e65 pushed by BAntonD		 18s	

З'являється що вже добре, потрібно тільки проблеми вирішити як це було з CI

```
PS C:\WINDOWS\system32> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH.Server*'

Name : OpenSSH.Server~~~~0.0.1.0
State : NotPresent

PS C:\WINDOWS\system32> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

Path      :
Online    : True
RestartNeeded : True

PS C:\WINDOWS\system32> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH.Server*'

Name : OpenSSH.Server~~~~0.0.1.0
State : InstallPending
```

```
PS C:\WINDOWS\system32> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH.Server*'

Name : OpenSSH.Server~~~~0.0.1.0
State : Installed
```

Створюю SSH сервер

```
PS C:\WINDOWS\system32> Start-Service sshd
PS C:\WINDOWS\system32> Set-Service -Name sshd -StartupType Automatic
PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "OpenSSH Server" -Direction Inbound -Protocol TCP -LocalPort 22 -Action Allow

Name                : {a8cb9375-8fb3-43d5-8e36-bfd949f7efbc}
DisplayName          : OpenSSH Server
Description          :
DisplayGroup         :
Group                :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Inbound
Action               : Allow
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : Правило успішно проаналізовано зі сховища. (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource    : PersistentStore
PolicyStoreSourceType : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId          :
```

Запускаю та відкриваю 22 порт

steps:

```
- name: Checkout code
  uses: actions/checkout@v2

- name: Install sshpass
  run: sudo apt-get install sshpass

- name: Create deployment folder on server
  run: |
    sshpass -p "${{ secrets.SSH_PASSWORD }}" ssh -o StrictHostKeyChecking=no ${{ secrets.USERNAME }}@${{ secrets.SERVER }}

- name: Copy files to server
  run: |
    sshpass -p "${{ secrets.SSH_PASSWORD }}" scp -r ./ ${{ secrets.USERNAME }}@${{ secrets.SERVER }}:${{ secrets.DEPLOY_PATH }}

- name: Restart server or services
  run: |
    sshpass -p "${{ secrets.SSH_PASSWORD }}" ssh -o StrictHostKeyChecking=no ${{ secrets.USERNAME }}@${{ secrets.SERVER }}
```

81 workflow runs		Event ▾	Status ▾	Branch ▾	Actor ▾
✓	test: test	CI Workflow #51: Commit 23c633f pushed by BAntonD	main	7 minutes ago	...
✗	test: test	Deploy to Server #30: Commit 23c633f pushed by BAntonD	main	7 minutes ago	...
✓	test: test	CI Workflow #50: Commit e22486c pushed by BAntonD	main	8 minutes ago	...
✗	test: test	Deploy to Server #29: Commit 46b4698 pushed by BAntonD	main	12 minutes ago	...

Як виявилось не змогли ми це зробити, поспробувавши пару варіантів завжди на деплої щось ставалося не так