

# Розробка гри "Морський бій" на C++

Дмитро Чучман

## Анотація

У роботі розглянуто розробку комп'ютерної гри "Морський бій" на мові програмування C++ з використанням основних принципів об'єктно-орієнтованого програмування (ООП).

## 1 Вступ

Морський бій — це популярна стратегічна гра, де гравці змагаються в пошуку і знищенні кораблів супротивника на обмеженому полі. Класичні правила цієї гри передбачають два варіанти змагання: між двома гравцями або між гравцем та комп'ютером. У сучасних комп'ютерних версіях гри часто реалізується штучний інтелект для ведення бойових дій на боці комп'ютера, а також зручні інтерфейси для гравців, що дозволяють грати через мережу або локально. Метою цього реферату є розробка гри "Морський бій" на мові програмування C++ з використанням основних принципів об'єктно-орієнтованого програмування (ООП) та створення двох рівнів складності для комп'ютерного супротивника.

## 2 Огляд гри та її принципи

Гра "Морський бій" передбачає наявність двох гравців або одного гравця та комп'ютера. Поле для гри складається з двох ідентичних сіток 10x10, на яких розміщуються кораблі. Кожен корабель займає кілька клітинок на полі, і його розташування має бути унікальним, тобто кораблі не можуть перетинатися або стикатися між собою.

Мета гри — знищити всі кораблі супротивника, завдаючи попадань по його полю, при цьому потрібно відшукати правильне розташування кораблів на полі, стріляючи по клітинках.

## 3 Архітектура та компоненти гри

Для реалізації цієї гри на C++ використовуються наступні компоненти:

- **Field.cpp** — файл, в якому написана вся логіка гри, реалізація всіх методів для коректного процесу гри.
- **ComputerPlayer.cpp** — клас, що генерує випадкові координати для пострілів комп'ютера.
- **Main.cpp** — головний файл, який об'єднує всі елементи гри, керує логікою гри, чергуванням ходів гравця та комп'ютера, а також виводить інтерфейс гри.

Крім того, використовуються функції для відображення поля гри, введення координат пострілів, перевірки стану гри та визначення перемоги.

## 4 Реалізація гри

### 4.1 Клас Field

Клас `Field` відповідає за побудову та управління ігровим полем. Кожен гравець має своє поле, яке містить кораблі та порожні клітинки. При створенні об'єкта поля ініціалізується сітка 10x10 клітинок. Важливими функціями є:

- `setShip()` — встановлює корабель в певну клітинку.
- `isCellEmpty()` — перевіряє, чи клітинка є порожньою.
- `canPlaceShip()` — перевіряє, чи можна розмістити корабель у певному місці поля з врахуванням умов (кораблі не повинні стикатися один з одним).
- `shootAt()` — функція для здійснення пострілу по певній клітинці.
- `placePlayerShips()` — розміщує кораблі гравця.
- `placeComputerShips()` — розміщує кораблі комп'ютера, враховуючи позиції кораблів гравця.
- `isCellFreeWithBuffer()` — перевіряє, чи є вільне місце навколо корабля.
- `areAllShipsSunk()` — перевіряє, чи знищено всі кораблі.
- `areShipsAdjacent()` — перевіряє, чи є кораблі поруч.

### 4.2 Клас ComputerPlayer

Клас `ComputerPlayer` відповідає за логіку поведінки комп'ютера. У найпростішому випадку комп'ютер здійснює постріли випадковим чином. Для цього використовується функція `makeShot()`, яка генерує випадкові координати на полі.

### 4.3 Режими гри

У програмі заплановано три режими гри:

1. **1 VS 1** — два гравці грають один проти одного на одному комп'ютері.
2. **You VS Random Computer** — гравець грає проти комп'ютера, який стріляє випадковим чином.
3. **You VS Smart Computer** — в планах розробити складнішого комп'ютерного супротивника, який буде більш стратегічно обмірковувати свої ходи.

У даному варіанті реалізований лише другий режим гри, в якому комп'ютер робить випадкові постріли.

## 4.4 Головний цикл гри

Головний цикл гри організований так, щоб після кожного ходу гравця комп'ютер здійснював свій постріл. Після цього перевіряється, чи знищено всі кораблі одного з гравців, і відповідно виводиться повідомлення про перемогу чи поразку.

## 5 Оцінка складності гри та штучного інтелекту

Програму можна удосконалити за рахунок покращення алгоритму для комп'ютерного гравця. Наразі комп'ютер стріляє випадковим чином, що робить гру передбачуваною. У майбутньому можна впровадити більш складну стратегію, таку як:

- Стратегії для пошуку кораблів (наприклад, за допомогою алгоритмів пошуку або методу "сліпого вогню").
- Оцінка ймовірності попадання в залежності від результатів попередніх пострілів.

## 6 Висновок

Розробка гри "Морський бій" на мові програмування C++ є цікавим і корисним проектом, який дозволяє ознайомитися з основами об'єктно-орієнтованого програмування, а також з техніками реалізації штучного інтелекту для комп'ютерного супротивника. У процесі розробки гри важливим є не лише створення логіки гри та графічного інтерфейсу, а й реалізація складних механізмів взаємодії між об'єктами гри, такими як кораблі, поле та гравець. Створення двох рівнів складності для комп'ютерного супротивника дозволяє зробити гру більш цікавою та динамічною.

Крім того, цей проект демонструє важливість тестування та оптимізації програмного коду, щоб забезпечити коректну роботу гри в різних сценаріях. В результаті розробки, можна побачити реальні можливості використання C++ для створення ігор, що мають інтерактивні елементи та використовують принципи штучного інтелекту для взаємодії з гравцем.

Загалом, розробка гри "Морський бій" є корисним досвідом для будь-якого програміста, що бажає покращити свої навички в об'єктно-орієнтованому програмуванні та алгоритмах штучного інтелекту.

## Література

- [1] <https://www.geeksforgeeks.org/how-to-use-cin-fail-method-in-cpp/>
- [2] <https://acode.com.ua/urok-74-generatsiya-randomnyh-chysel/#toc-1>
- [3] <https://en.cppreference.com/w/cpp/utility/pair>
- [4] <https://chatgpt.com/>