РОЗРОБКА ГРИ НА UNITY/C# ІМПЛЕМЕНТАЦІЄЮ ШТУЧНОГО ІНТЕЛЕКТУ

Виконав студент 4 курсу спеціальности «Прикладна математика»
Кириняченко Ростислав Вадимович

Науковий керівник Борозенний Сергій Олександрович

МЕТА РОБОТИ

- Розробка комп'ютерної гри на платформі Unity
- Імплементація штучного інтелекту на основі математичних алгоритмів

ПРЕДМЕТ ДОСЛІДЖЕННЯ

Комп'ютерна версія гри шьоґі розроблена з використанням алгоритму Minimax для імплементації штучного інтелекту

ГРА ШЬОҐІ

Шьоґі - це японська версія шахів.

- Грається на дошці 9x9
- У кожного гравця по 20 фігур
- Майже всі фігури можуть отримувати просування, коли досягабть зони перетворення
- Захоплені фігури можна повернути на дошку
- Гра завершується, коли король опонента отримує шах і мат

АЛГОРИТМ МІНІМАКС

Minimax – алгоритм штучного інтелекту для ігор з нульовою сумою для двох гравців

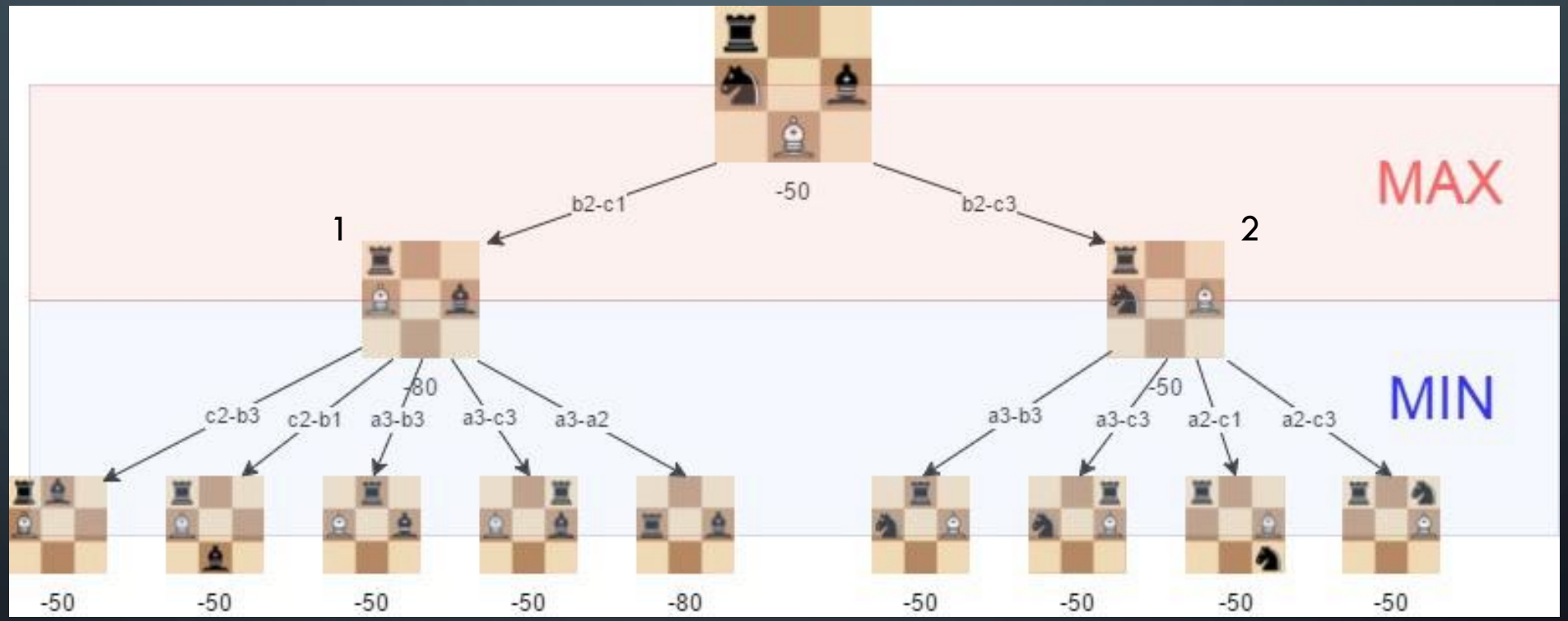
Переваги:

- Здатен гарантувати оптимальний хід
- Простий у реалізації
- Дає можливість використовувати евристичну інформацію

Недоліки:

- Вимагає побудови повного дерева гри
- Низька ефективність у випадку гри з великою кількістю можливих ходів
- Потребує значних обчислювальних ресурсів

ПРИКЛАД РОБОТИ АЛГОРИТМУ МІНІМАКС



ПРОБЛЕМА ВИСОКОЇ ЧАСОВОЇ СКЛАДНОСТІ МІНІМАКСУ І ШЛЯХИ ЇЇ ВИРІШЕННЯ

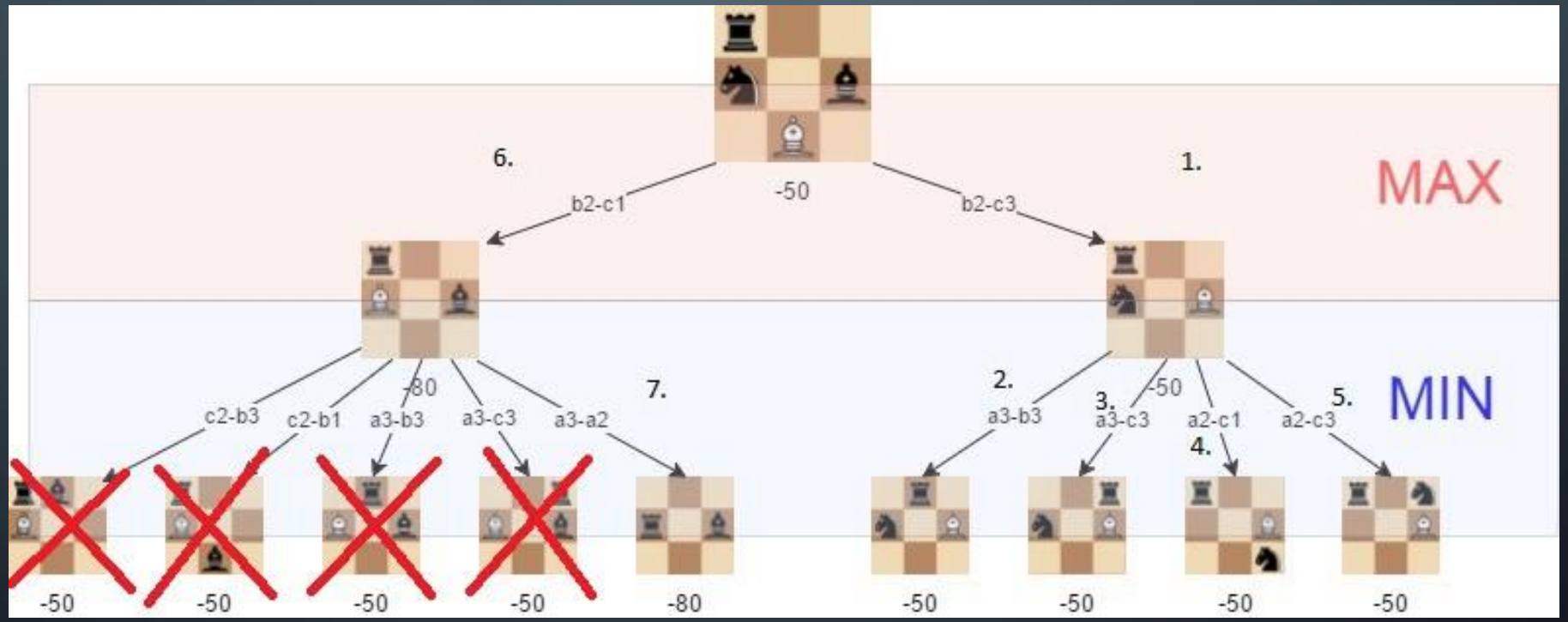
Часова складність мінімаксу становить O(b^m).

- b - коефіцієнт розгалуження
- m - глибина дерева

Альфа-Бета відсічення - метод оптимізації мінімаксу

- Відсікає непотрібні гілки дерева гри
- Знижує часову складність до O(b^(m/2)) в найкращому випадку

ПРИКЛАД РОБОТИ АЛЬФА-БЕТА ВІДСІЧЕННЯ



РЕАЛІЗАЦІЯ АЛГОРИТМУ

Алгоритм роботи ШІ

1. Створюється масив усіх можливих ходів противника
2. Створюється клон ігрової дошки на якому робиться хід
3. Викликається функція Minimax() для кожного ходу, щоб отримати його оцінку.
4. Обирається хід з найбільшою кількістю балів і виконується на реальній ігровій дошці

Функція Minimax(TestBoard board, int depth, bool isMaximizingPlayer, int alpha, int beta)

1. Симулює всі можливі ходи, клонує дошку, рекурсивно викликає Minimax() для протилежного гравця та зменшує глибину
2. Коли доходить до нульової глибини викликає метод EvaluateBoard для оцінки стану дошки.

ОЦІНКА ДОШКИ

Цінність фігур*:

- Пішак – 10
- Спис - 50
- Кінь - 60
- Срібний генерал - 80
- Золотий генерал - 90
- Перетворений срібн. генерал – 90
- Перетворений спис-100;
- Перетворений кінь-100;
- Перетворений пішак-120;
- Слон – 130;
- Тура-150;
- Перетворений слон-150;
- Перетворена тура-170.

* узято з книги Коджі Таніґави «Як мислити, щоб вигравати в шьоґі»

Цінність позиції фігури на прикладі слона

```
PieceType.Bishop,
new int[,]
{
    { 2,  2,  2,  2,  2,  2,  2,  2,  2 },
    { 2,  3,  3,  3,  3,  3,  3,  3,  2 },
    { 2,  3,  4,  4,  4,  4,  4,  3,  2 },
    { 2,  3,  4,  5,  5,  5,  4,  3,  2 },
    { 2,  3,  4,  5,  6,  5,  4,  3,  2 },
    { 2,  3,  4,  5,  5,  5,  4,  3,  2 },
    { 2,  3,  4,  4,  4,  4,  4,  3,  2 },
    { 2,  3,  3,  3,  3,  3,  3,  3,  2 },
    { 2,  2,  2,  2,  2,  2,  2,  2,  2 }
}
```

РЕЗУЛЬТАТИ ТЕСТУВАННЯ АЛГОРИТМУ МІНІМАКС БЕЗ АЛЬФА-БЕТА ВІДСІЧЕННЯ

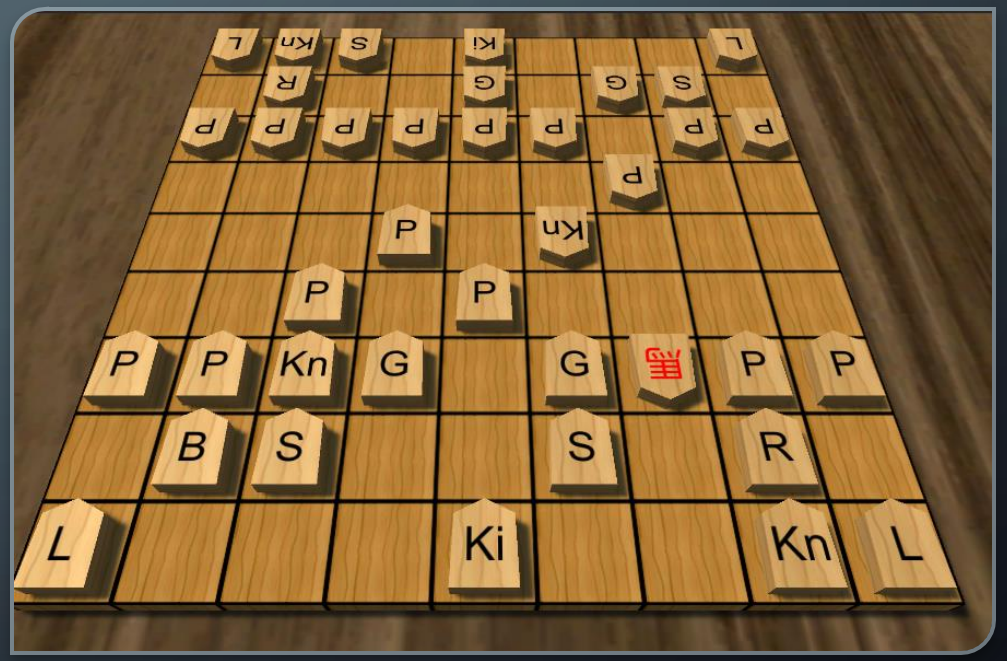
Виміряно середній час, що необхідний штучному інтелекту, щоб зробити хід при відповідній глибині пошуку:

- Глибина 1: ≈ 0,0005 секунди на хід
- Глибина 2: ≈ 0,03 секунди на хід
- Глибина 3: ≈ 0,65 секунди на хід
- Глибина 4: ≈ 36 секунд на хід
- Глибина 5: ≈ 747 секунд на хід

РЕЗУЛЬТАТИ ТЕСТУВАННЯ АЛГОРИТМУ МІНІМАКС З АЛЬФА-БЕТА ВІДСІЧЕННЯМ

Виміряно середній час, що необхідний штучному інтелекту, щоб зробити хід при відповідній глибині пошуку:

- Глибина 1: ≈ 0,0005 секунди на хід
- Глибина 2: ≈ 0,02 секунди на хід
- Глибина 3: ≈ 0,25 секунди на хід (було 0,65 сек)
- Глибина 4: ≈ 5 секунд на хід (20+ секунд в окр. вип.) (було 36 сек.)
- Глибина 5: ≈ 67 секунд на хід (було 747 сек.)



РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

- Розроблено гру шьоґі на платформі Unity
- Імплементовано алгоритм мінімакс для роботи штучного інтелекту у грі
- Алгоритм мінмакс успішно оптимізовано за допомогою альфа-бета відсічення

ДЯКУЮ ЗА УВАГУ!