**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 3 з дисципліни

«[Програмування інтелектуальних інформаційних систем](https://classroom.google.com/c/NTQ3MTk0ODU2OTY3" \t "https://classroom.google.com/c/_self)»

„**Алгоритм Negamax , алгоритм Negamax з альфа-бета відсіканням, алгоритм NegaScout**”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*Заранік Богдан ІП-01*

**Перевірив**

(прізвище, ім'я, по батькові)

*Вавіленкова А. Д.*

Київ 2021

Зміст

[1 Мета лабораторної роботи 3](#_Toc52291748)

[2 Завдання 4](#_Toc52291749)

[3 Виконання 6](#_Toc52291750)

[3.1 Програмна реалізація алгоритму 6](#_Toc52291752)

[3.1.1 Вихідний код 6](#_Toc52291753)

[3.1.2 Приклади роботи 6](#_Toc52291754)

[3.3 Тестування алгоритму 6](#_Toc52291755)

[Висновок 7](#_Toc52291756)

[Критерії оцінювання 8](#_Toc52291757)

# Мета лабораторної роботи

Мета роботи - вивчити основні алгоритми пошуку стратегії наступного ходу за допомогою алгоритмів Negamax і NegaScout.

# Завдання

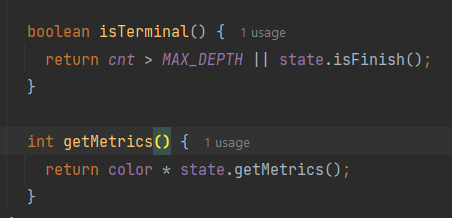
Написати код (мову програмування обираєте самі), для реалізації двох алгоритмів: алгоритму мінімакс та алгоритму альфа-бета відсікань, розібратися в роботі алгоритмів, вміти пояснити та аргументувати написаний код.

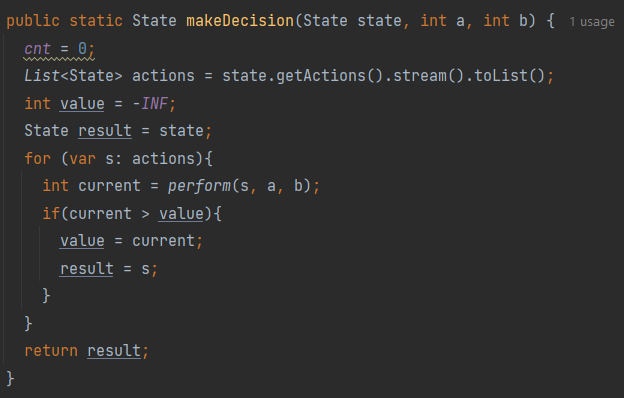
# Виконання

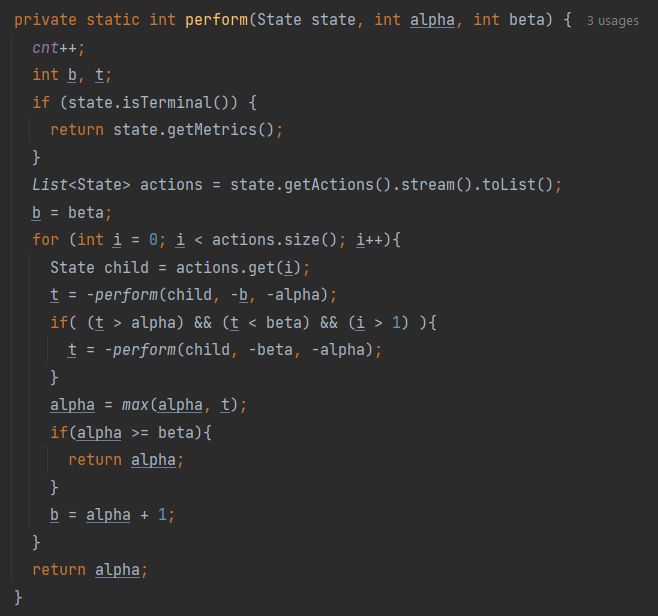
## Програмна реалізація алгоритму

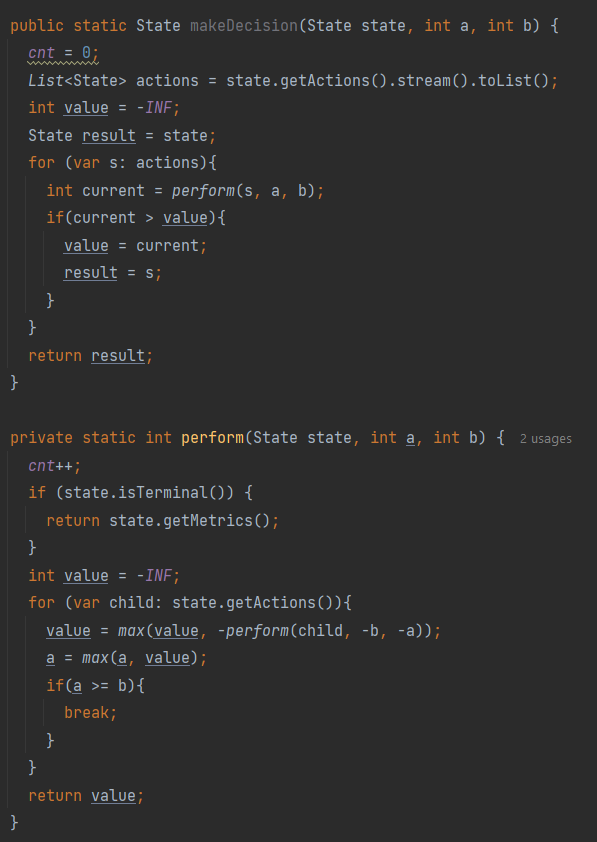
### Вихідний код











### Приклади роботи

На рисунках 3.1 і 3.2 показані приклади роботи програми.

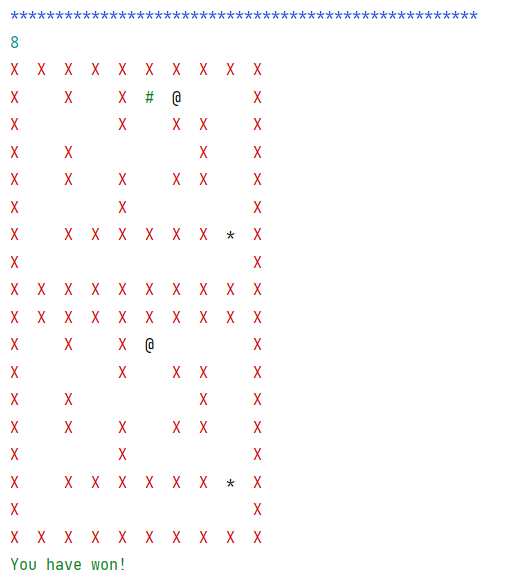


Рисунок 3.1 – Фрагмент виводу результатів роботи алгоритму minimax

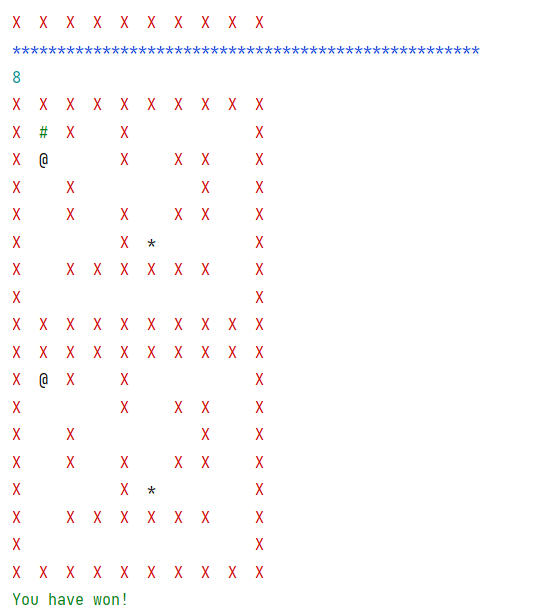


Рисунок 3.2 – Фрагмент виводу результатів роботи алгоритму NegaMax з альфа-бета відсіканнями

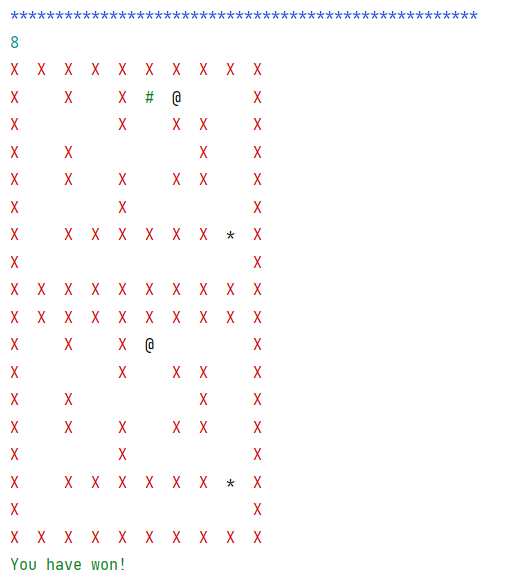


Рисунок 3.3 – Фрагмент виводу результатів роботи алгоритму NegaScout

Висновок

В рамках даної лабораторної роботи я реалізував алгоритми мінімакс та альфа-бета відсікань, розібрався в роботі цих алгоритмів, можу пояснити та аргументувати написаний код.

NegaMax— це правило прийняття рішень, що використовується в [теорії ігор](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D1%96%D0%B3%D0%BE%D1%80" \o "Теорія ігор), [теорії прийняття рішень](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D1%80%D1%96%D1%88%D0%B5%D0%BD%D1%8C" \o "Теорія рішень), [дослідженні операцій](https://uk.wikipedia.org/wiki/%D0%94%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9" \o "Дослідження операцій), [статистиці](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0" \o "Математична статистика) і [філософії](https://uk.wikipedia.org/wiki/%D0%A4%D1%96%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D1%96%D1%8F" \o "Філософія) для мінімізації можливих втрат з тих, які особа, яка приймає рішення не може уникнути при розвитку подій за найгіршим для неї сценарієм. Засноване на алгоритмі Minimax.

Альфа-бета відсікання - це по факту покращення NegaMax. Головною проблемою першого є те, що кількість розглянутих станів може бути достатньо великою і розгортання деяких гілок є взагалі безкорисним: вони вже точно не покращать результат. Саме на таке відсікання і направлений алгоритм альфа-бета відсікань.

NegaScout – це один із варіантів використання алгоритму альфа-бета

відсікання, направлений алгоритм пошуку для обчислення мінімаксного

значення вузла. Алгоритм дає прискорення , не вносячи ніякої додаткової

похибки обчислень. Алгоритм не буде розглядати вузли, які можна відсікти за

алгоритмом альфа-бета, але деякі гілки може переглядати декілька разів.

Алгоритми протестовано. Коректність їх роботи показано. Результати роботи наведено вище. Також наведено фрагменти коду програми, що спрощую розуміння коди при швидкому перегляді звіту. Для більш детального ознайомлення із кодовою базою <https://github.com/GeniusDP/PIISLab3>.

Критерії оцінювання

При здачі лабораторної роботи до 10.12.2020 включно максимальний бал дорівнює – 5. Після 10.12.2020 максимальний бал дорівнює – 1.

Критерії оцінювання у відсотках від максимального балу:

* програмна реалізація алгоритму – 95%;
* висновок – 5%.

+1 додатковий бал можна отримати за реалізацію рівнів складності.