ABALONE AI PROJECT

Agenda

Introduction

Branching Factor/Complexity

Algorithms Used

Implementation

Results/Improvements

Abalone

Abalone is a deterministic two-player perfect information board game that was created in 1988 and was based around Japanese sumo wrestling (N.P.P.M Lemmons, 2005).

A player can move 1-3 pieces diagonally, vertically, or horizontally as long as the pieces are in a line.

The goal of the game is to “push” 6 of the opponent’s pieces off the board.

Branching Factor/Complexity

According to Lee and Noh, “The branching factor is estimated to be around 60 to 80, which is at least twice as high as chess, 8 times as high as Checkers, and 16x as high as Othello [1, 2]”.

This makes the game complexity of abalone around 5 x 10^154 , similar to Xiangqi and Shogi (Lemmens).

```
if (killboost == 100)
{
    return FMS.centercount  + killboost;
}
else
{
        return (FMS.centercount) + killboost;
}
```

```
if (pshd == 2)
{
    p = 70;
}
else if (pshd == 1) { p = 60;
else
{
    p = 0;
}
```

Algorithms and Functions Used

Minimax was used with Alpha-Beta Pruning.

The Static Evaluation functions used were:

- Distance From the Center
- Opportunity to Push
- Opportunity to Take a Player Piece

Depth of Search

Minimax

Depth	Time	Competitive Performance
1	Very Fast	Bad
2	Fast	Bad
3	30 sec	Okay
4	>30 min	Good

Minimax with Alpha- Beta Pruning

Depth	Time	Competitive Performance
1	Very Fast	Bad
2	Fast	Bad
3	<5 sec	Okay
4	<10 sec	Good

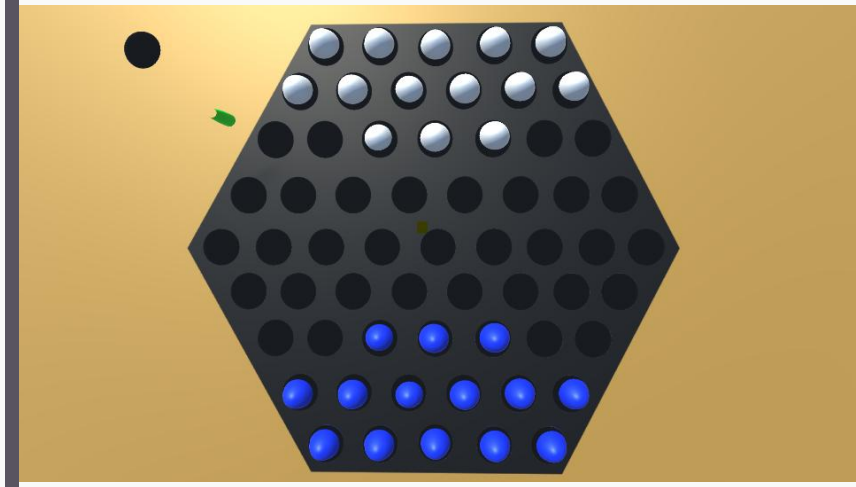
Results

Since the average game of Abalone consists of on average 87 moves not much actual data was collected.

I was able to beat it constantly after some practice.

It tends to perform well against beginner and inexperienced players.

https://youtu.be/6bPRArzxO-E



References

Lague, S. (n.d.). Alpha-beta pseudocode. Retrieved from https://pastebin.com/rZg1Mz9G

Lemmens, N. P. P. M. (n.d.). Retrieved from https://project.dke.maastrichtuniversity.nl/games/files/bsc/Lemmens_BSc-paper.pdf

Lee, B., & Noh, H. J. (n.d.). Retrieved from https://www.cs.cornell.edu/~hn57/pdf/AbaloneFinalReport.pdf

Tim, John, Cho, Forerun, Konro, Tomberli, G., ... Offline. (2024). How to convert the mouse position to World Space in Unity (2D + 3D). Retrieved from https://gamedevbeginner.com/how-to-convert-the-mouse-position-to-world-space-in-unity-2d-3d/#:~:text=Converting%20the%20position%20of%20the%20mouse%20on%20the,the%20Scene.%20Like%20this%3A%20Vector3%20worldPosition%20%3D%20Camera.main.ScreenToWorldPoint%28Input.mousePosition%29%3B