Neuromancer @ Tablut Student Challenge 2021

Kevin Michael Frick, Corinna Marchili, Davide Ragazzini, Antony Zappacosta

May 25, 2021

Alma Mater Studiorum Università di Bologna
Corso di Fondamenti di Intelligenza Artificiale 2020/2021

Specifications

The project is developed in Java. It includes the following modules:

- the **Game module**, consisting of the pre-existing game implementation plus AIMA interfaces;
- the **Genetic module**, employed to define the most suitable set of weights for the heuristic evaluation functions.

Strategy

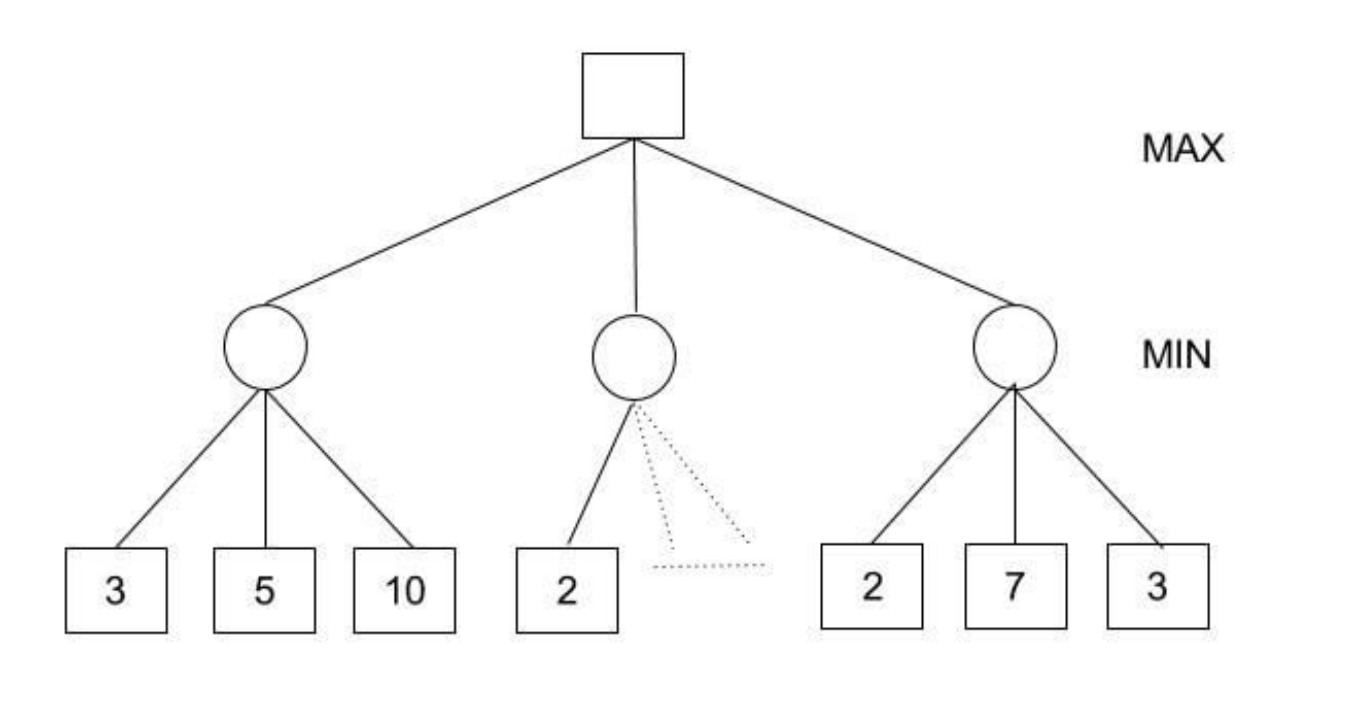

Figure 1: AIMA implementation of the iterative deepening MiniMax algorithm, coupled with alpha-beta pruning.

White Heuristics

The set of weights comprehends:

- **King Position**: distance between the king and the throne;
- **Surrounding Black Pawns**: number of black pawns surrounding the king;
- **White Near King**: number of white pawns close to the king;
- **Black Near King**: number of black pawns close to the king;
- **Number of Whites**: number of white pawns on the board;
- **Number of Blacks**: number of black pawns on the board;
- **Threat**: risk of getting eaten with a single move of the opponent;
- **Victory**: chance of winning with one final move.

Black Heuristics

The set of weights comprehends:

- **Rhombus**: rhombus-shaped configuration employed to block escaping tiles;
- **Row-Column Coverage**: rate of occupied rows and columns;
- **Surrounding Black Pawns**: number of black pawns surrounding the king;
- **Black Near King**: number of black pawns close to the king;
- **Number of Whites**: number of white pawns on the board;
- **Number of Blacks**: number of black pawns on the board;
- **Threat**: risk of being eaten with a single move of the opponent;
- **Victory**: chance of winning with one final move.

Genetic Module: fitness

A script synchronizes the game and the genetic module, collecting the result of each match against the best players from the last two competitions. Each individual codifies a set of weights. If the player scores a better fitness value in the game its set is used as an input for subsequent computations.
The fitness value is defined according to the game result, given w = number of moves leading to the best fitness score so far and n = number of moves to end the current game:

$$f = \begin{cases} 0 & \text{if the result is L and n<w} \\ 1 & \text{if the result is L and n>w} \\ 2 & \text{if the result is D} \\ 3 & \text{if the result is W and n>w} \\ 4 & \text{if the result is W and n<w} \end{cases}$$

Genetic Module: trained values

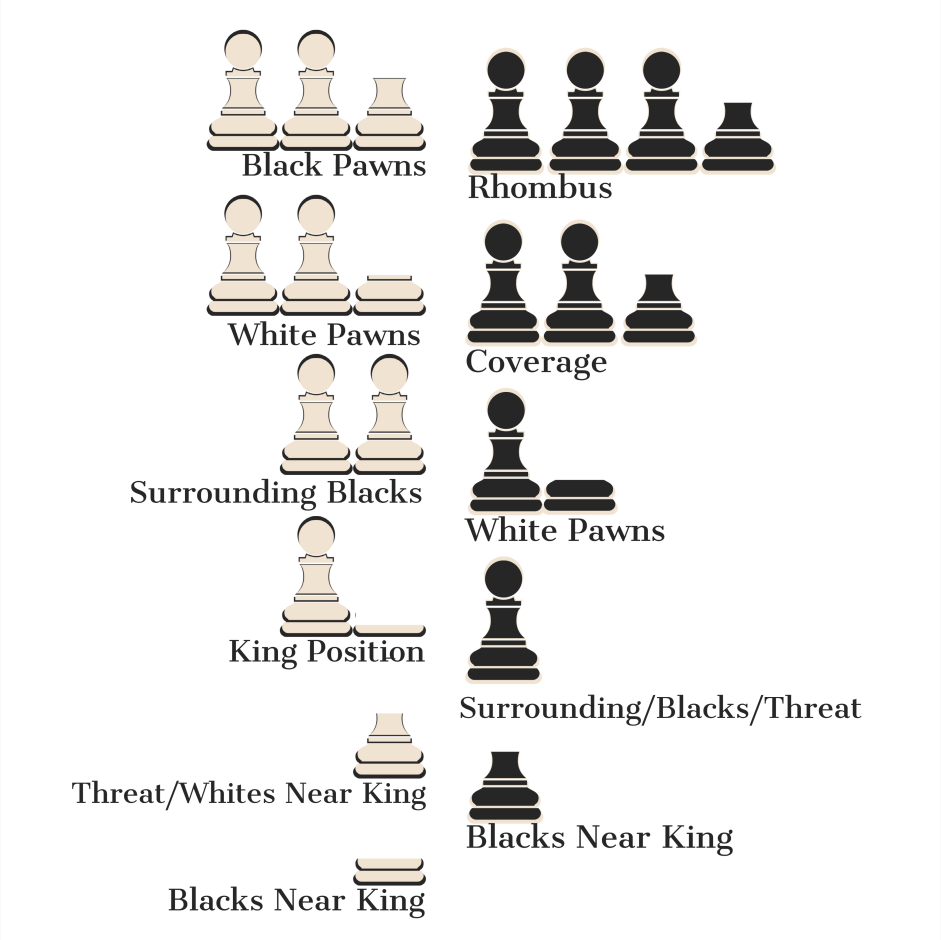

Figure 2: Final set of weights (1 Pawn = 10%)

The end

Thanks for your attention!