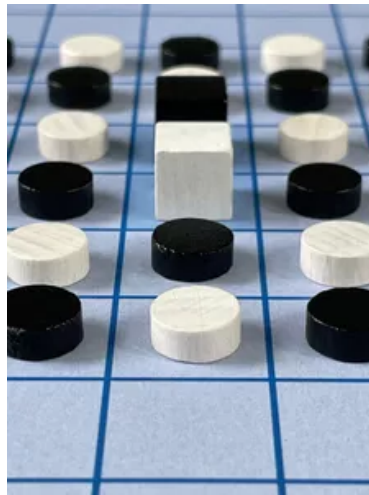


Artificial Intelligence - Splinter Board Game



IA - Group 44_2D

Group elements:

Diogo Tomás Câmara - up201905166
Mário Pedro Soares Pinto Mota Ferreira - up201907727
Pedro Albernaz Moreira - up201904642

April 2022

Specification of the work to be performed

We choose Topic 2, which consists of Adversarial Search Methods for Two-Player Board Games, more specifically Splinter board game, which is an abstract strategy game for two players. The objective of the game is to get more connected board pieces to your king than your adversary is able to, this is to splinter your opponent's king into a group that is smaller than your king's group. The game is characterized by the type of board and pieces, the rules of movement of the pieces (operators) and the finishing conditions of the game with the respective score.

Game related work

The development of our game was done following the guidelines found in this [link](#) provided on this course Moodle Page, consisting of its components, lay-out and rules of the game.

Problem as a search problem

- **State representation, initial state**

Play begins with the pieces arranged in a single, inter-connected group, with the kings occupying the board's two central squares. This initial state is to be represented by a 2d list in the Game Class.

- **Objective test**

The game ends if a splinter causes the kings to occupy two different groups and the player whose king occupies the larger group wins the game. If both groups are of the same size, the game ends in a tie. If a move causes one of the kings to get pushed off the board, the other king's player wins.

- **Operators**

The game as a unique operator with some restrictions such as:

- ☐ you may move one of your own pieces one square in any direction (including diagonally) so that it lands on an adjacent square.
- ☐ you may push a single piece or a row of connected pieces of either or both colors.
- ☐ you may move or push any piece of either color off the board.
- ☐ Kings may be moved (and may push and be pushed) in exactly the same way as pawns.

- **Heuristics/evaluation function**

If a splinter causes the kings to occupy two different groups, the game ends and the player whose king occupies the larger group wins the game.

The heuristic chosen is based on counting the score of the pieces that are connected to a king, counting its value and subtracting this value with the other king connected pieces values.

Implementation work

Our game is being developed on Python 3.10 . We used the Colorama library to manage the coloring of our game. So far we implemented the game representation, the movement functions, score counting, main game loop and an initial version of Minimax Algorithm. For this implementation we used Classes and mainly lists.