Seega For Noobs

MIFY SARL

November 2, 2018

1. What is Seega ?
    - Board Game
    - The Game

2. The code
    - What you have to do.

3. End

What is Seega ?

- Seega is a game dating back over a century, it is still practiced in present day in Egypt, Ethiopia and especially in Somalia.
- The goal of the game is to capture all the pieces of your opponent (or as much as possible when the game arrives at a blocking configuration).

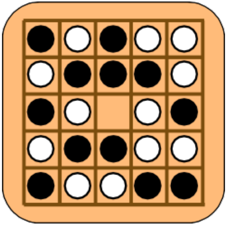
Board Game

- The Seega is played on a board of 5 squares out of 5.
- Each player has 12 pieces.
- It happens to play on board larger in size, for example, 7 squares out of 7; the players then having 24 pieces each; even 9 boxes out of 9.

Board Game

The Game

- **First step:** placing the pieces.
  At first, the players deposit two pieces in turn on empty squares of the game board, **with the exception of the central box**, which must remain free. For example, here is what to give an early game after the installation of 24 pieces.

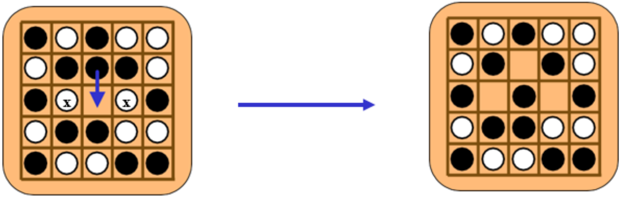


The Game

- **Second step:** moving and taking pieces.
  - The **last player** who landed his last 2 pieces can then start moving a piece.
  - Each piece can move to an **adjacent empty space** horizontally or vertically.
  - A piece **can move** on the central square.
  - A piece **can capture** an opponent's piece when, during his move, his arrival square allows him **to frame** the opponent's piece with another of his pieces.
  - The framed piece is removed from the game.
  - **It is possible to take several pieces simultaneously.**

The Game

In the example below, the black piece, moving, will simultaneously frame 2 white pieces (marked with a cross), which will be captured and removed from the game.



The Game

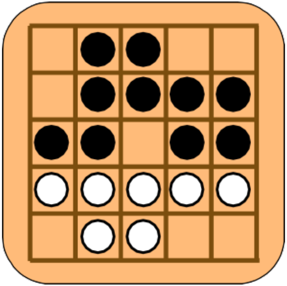
- **A piece on the central square can not be captured.**
- **The catch is always made during a trip**. This is why a piece moving between 2 other opponent pieces is not captured.
- If a player can not play, he passes his turn and his opponent must make a move allowing the first player to play.

At Seega, it can happen that a position is reached where each player has managed to build a "**barrier**" (continuous line of his pieces), behind which he can move his pieces without any danger of catch. The position remaining in the status quo, **the player having captured more pieces at this time of the game wins**.

The Game

For example, on the diagram, 2 white and black barriers are formed, the 2 players can then move their pieces behind each of their barrier without danger.

At this stage of the game, the black player will be declared the winner, since he will have captured more pieces than his opponent.



The Code

The game Seega was developed in **python** as part of this contest.
It consists of classes that interact to make the game work.
The most important classes are:

- The **Board** class that represents the game board. It contains the functions to update the board for each action.
- The **BoardSquare** class which represents a box of the game board.
- The **RulesGames** class where the rules of the game are implemented.
- The **GameWindows** class representing the game window. It allows the global management of the application.

The Code

The participants of this competition aim to build an artificial intelligence that is effective enough to beat their opponent's. To do this, apart from the effort they will provide themselves, it is made available to them several functions that may be useful:

- **play()**: This function takes as parameter a Board object and a step variable indicating which step the game is in (0 for the step of "placing pieces" and 1 for the step of "moving and taking pieces"). It returns two values wehen step equals 0 and four when step equals to 1.
- **canPlayHere()**: This function takes in parameter a Board object, the step variable and the coordinates of the point of which one wants to check if it is playable. It returns True if the coordinate box (x, y) is playable and False if no.

The Code

- **getRealsMoves()**: This function takes as parameter a (x, y) coordinate of a piece and returns an array of all the actual possible piece destinations.
- **getMovingPiece()**: This function takes as parameter an object board and a color and returns all the list of pions having the color received in parameter and which can be moved.
- **pieceCanMove()**: Takes into parameter an object Board and the coordinates of a piece (origin), the color of the piece and returns True if there is possible movement for this piece.

What you have to do.

One of the classes in this game is the **Player** class. This is an abstract class containing the prototype of the **play()** function from which the classes of artificial intelligences must inherit. These subclasses must be named IA and the name parameter must be updated with the name of the team. These teams must therefore imperatively implement the play () function of the class **IA** except their classes and functions to them and return the expected result within a **time limit**.

End

Thank you for your attention.