2016

IUT Lyon 1

Bryan Boni G2S4

[AI: REPORT PRACTICAL 1]

This report talk about the project created to response of the practical 1.

The Maze

Programming Part

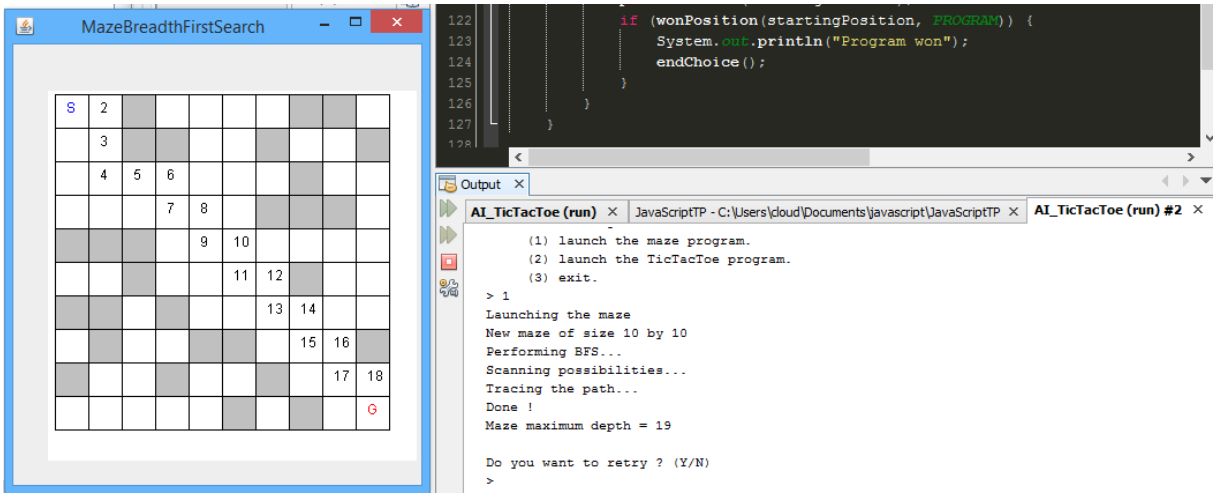
In this part I implement the class **BFS** and **PositionQueue** as expect from the subject and also I implement the function **endChoice()** that allow the user to retry the program.

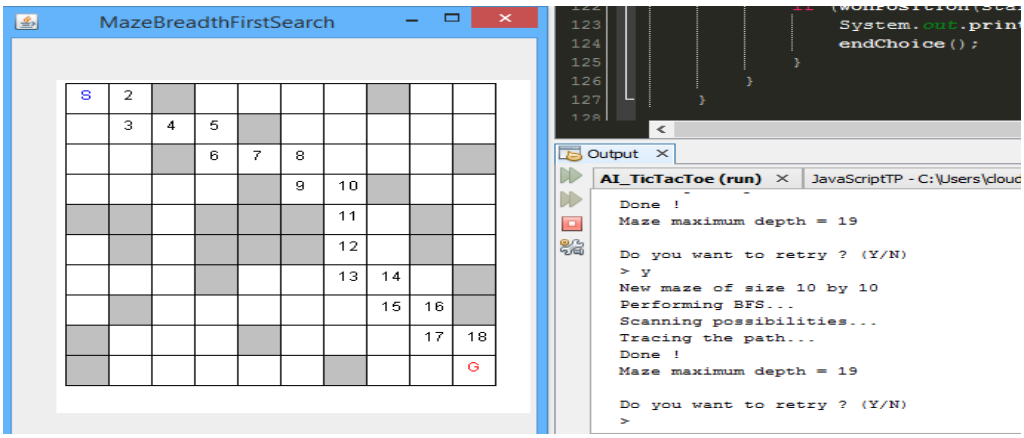
How they work is describe in commentary in each function implemented.

Testing part

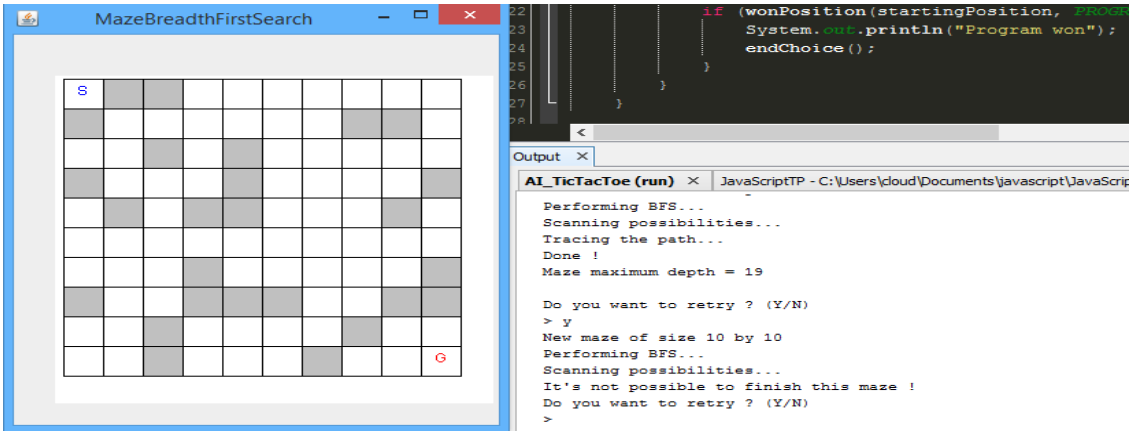
For this part I tested a few times the algorithm and I found it face 2 general cases:

- When the maze is feasible :





- When the isn't feasible :



Possible Improvement

- In the maze class, change the generation code with a procedural generation in order to always have a feasible maze for the AI.

The TicTacToe

Programming Part

I've create:

- The class TicTacToeMove with the variable moveIndex (int).
- The class TicTacToePosition who's implementing the game variables BLANK,HUMAN, PROGRAM and board, and the function toString(): printing the board.

I've implement:

- All the methods ask in the exercice in the class TicTacToe.
- The method endChoice() in the class GameSearch.

Possible Improvement

- In the gameSearch/TicTacToe class, modify or add code in order to check if the case were the player want to move is already taken or not, and if it's the case, don't make the move and re-ask to the player is next move.
- Implementing the entire minimax algorithm (which means the players will never win anymore!).