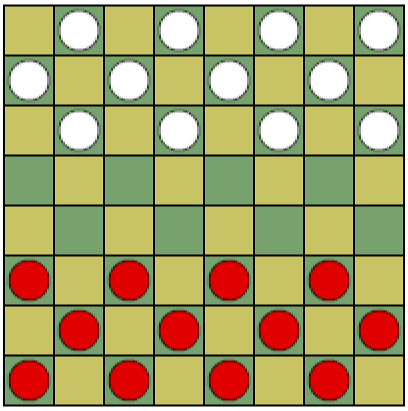
Assignment #1

The purpose of this assignment is to create an AI program that can masterfully play Checkers or solve Traveling salesman problem. You need to choose one of them.

Checkers

Checkers is a very popular two player game all over the world. One player has dark pieces, and the other has light pieces. They take turns moving their pieces.
Players can only move their pieces diagonally from one square to another square. When a player jumps over their opponent's (the other player's) piece, you take that piece from the board. The game of checkers is considered a complicated game with 10^{20} possible legal positions in the (8*8 board) alone (much more on higher dimensions). There for we need a good AI algorithm to go through this game.



Methodology

Your program should allow:

- Two player game.
- One player game (Human vs. Computer (Alpha-Beta AI)).

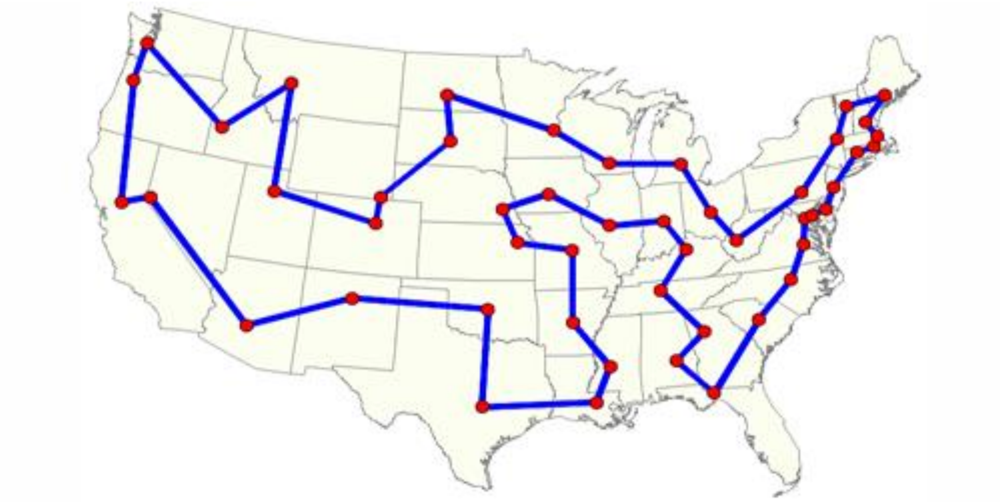
Your program should use:

- Game strategy:
    - Use utility and evaluation functions of your choice. The better your game strategy, the better your grade in the project.
- You can design the graphical user interface (GUI) as you like, but make sure that your program is user friendly.

More information on how to play the game can be found here:
https://gametable.org/games/checkers/

Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is the challenge of finding the shortest yet most efficient route for a person to take given a list of specific destinations. It is a well-known algorithmic problem in the fields of computer science and operations research.
There are obviously a lot of different routes to choose from but finding the best one (the one that will require the least distance or cost) is what mathematicians and computer scientists have spent decades trying to solve for.



Methodology

You need to develop a tool to solve the TSP using a good AI optimization algorithm. Your program should allow the user to:

- Randomly connect between the cities (reset)
- Change number of cities (default 10)
- Other parameters depending on the choosing algorithm
  - Example if you chose the simulated annealing algorithm then you should allow the user to set some parameters like initial temperature, cooling rate, maximum iteration (or final temperature)

You can design the graphical user interface (GUI) as you like, but make sure that your program is user friendly.

Languages

You may use the programming language that you prefer.

Marking scheme

Look and feel of interface	1-5	Algorithm implementation	1-5
Quality of design and coding	1-5	Demo	1-5