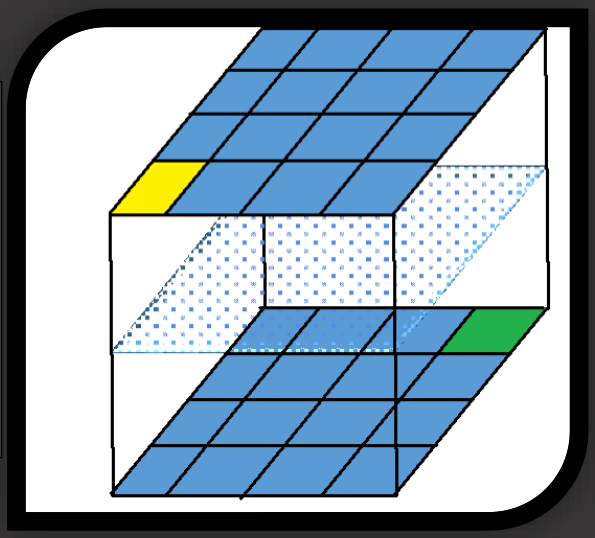
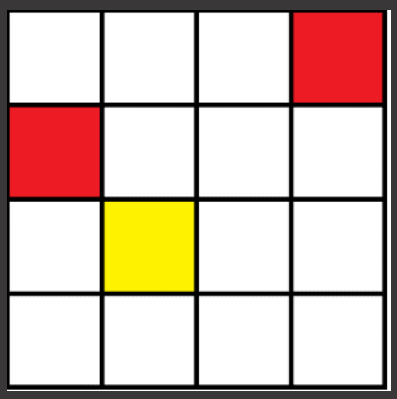
A Genetic Algorithms To Solve the Markov's Prison Puzzle



Purpose

To create a system with an Artificial Intelligence that can create a path for a prisoner to follow from the start of the puzzle to the end.

Each guard moved randomly so the main objective was to ensure that the path selected had the highest chance of success. If the prisoner reaches the end, it wins. If it collides with a guard, it loses.



Moves

Guards, in red, can move in any direction, including up and down floors (not able to be shown on a single floor diagram). If the direction they seek to move in cannot be travelled through, they stay put at their current position.

The prisoner, in yellow, can move in any direction like the guards. However the prisoner is incapable of staying put in a single location and instead is constantly attempting to reach the goal. Thus, if it cannot travel in a particular direction then it does not consider that direction as a valid direction to take for its next move.

Functionality

Player Controls:

- The user can take control of the prisoner at any stage so long as the current run hasn't reached an end state yet.
- Can alter the floor shown and monitor the number of moves made, the current floor and if the run is still active.

AI Controls:

- Can create a basic random path.
- Can test the current loaded path with 100 runs and find out how many times it succeeds.
- Can call the Genetic Algorithm, taking in the parameters of repeats and population below it, and finds the best path it can find.
- Can reset all parameters.

Simulation Controls:

- Can start a simulation of the loaded path. Moves through the path step by step with the guards acting randomly to each move.
- Any attempts by the user to take control will render any attempts to continue the simulation invalid due to deviating from the set path.