Assignment – 2

By,
Prerana Kamath
Vivek Patani
Supreeth Suryaprakash

Part 1:

Explanation:

The problem was formulated in the following way:
State Space: All possible moves that involve placing a marble in an empty space on the board

State Space: All possible moves that involve placing a marble in an empty space on the board with one marble occupying exactly one place.

Initial State: Any legit move that satisfies that each marble occupy one space and should not be a terminal state.

Successor function: All possible successors from a given initial state where each successor would have an empty tile filled with a marble of a particular color (depending on the player). Minimax prunes successors based on their evaluation score so some successors would not be generated/explored.

Heuristics: We evaluate each state by applying the following rules.
- If it is Max:
    We find contiguous runs of w or b (depending upon turn) and raise 10 to the power of n. So if you find 'www' in any row/column/diagonal.
    we would score it 1000.
- If it is Min:
    We do the same but negate the cost value.

The search algorithm works by finding whether the max or the min player is playing and recursing from the bottom of the tree to find the best possible next move by either maximizing or minimizing the score depending on the Player.

There were a few problems faced such as recursion depth and time trade off. You either had to chose from the two, we have come halfway.
The other idea was to implement the tree iteratively to a depth until the time given as an input was left.

Part 2:

Explanation:

We have built the Tetris AI bot by making use of the genetic algorithms and minimax. The things we took in to consideration were,

- Height
- Bumpiness
- Number of Holes
- Altitude
- Completed Lines

We first look at all the possible places where a piece can be placed with all kinds of orientation, then we consider the next piece and place it in all the possible places in all orientation. We calculate the score based on this formula,

Score = a * Height + b * Bumpiness + c * Number of holes + d * Completed lines

We calculate the score based on the above formula for each position of the next piece with respect to the position of the current piece. We take the final score at the End.

We ran a fitness function that gave me the best possible values for the constants a, b, c, d. The factor Height is undesirable and we give it a maximum penalty, whereas the Completed Lines is a very desirable thing to have and hence we reward it by a positive number.

The feature Number of Holes and Bumpiness are undesirable too, therefore we penalize it. After trying different values, the bot started playing well for these values,

a = -1.5, b = -1.5, c = -3 and d = 4.5

While Testing the bot performed exceedingly well by crossing 50K mark more than thrice. It succumbs to random distribution once in a while.

How to run the code?

- As mentioned by the Professor in the assignment.

References:

1. http://www.cs.uml.edu/ecg/uploads/AIfall10/eshahar_rwest_GATetris.pdf
2. Discussed the approach on a higher level with Supreeth and team, Tanmai and team and Raghuveer and team.
3. https://codemyroad.wordpress.com