Name: Saranya R Student ID: 100981198 Date: 28 Feb 2018 Assignment: 1

<u>State Space</u>

Game sends a Hashmap of index as string and pegs at index as values to state. State create a 2d array and populates comparing it to the index keys of Hashmap. Boundaries is set as Hashmap with row as key and arrays as list as columns, and is used extensively in move class to check look ahead valid moves do not go beyond boundaries. Print is done on the basis of counting spaces and dashes with respect to the size of each row and centralization of board view. Board prints "1" when a peg exists, "0" when there is no peg and nothing when it is out of valid board boundaries.

1) The A*Search with the heuristic neighbor nodes with associated cost estimate worked best.

<u>BFS:</u>

Moves to goal: 14 Nodes expanded: 7520

Note: However, this started at the 52 second board configuration.

<u>DFS:</u>

Moves to goal: 36 Nodes expanded: 1032385

<u>A*Star heuristic neighbor nodes:</u>

Moves to goal: 36 Nodes expanded: 1009

<u>A*Star heuristic possible valid moves:</u>

-bad heuristic

<u>A*Star heuristic Manhattan distance (extra):</u>

-bad heuristic

<u>A*Star heuristic average (neighbor nodes + possible valid moves):</u>

Moves to goal: 36 Nodes expanded: 17501

2) Heuristics used: Neighbor Nodes, Possible valid moves, Manhattan distance(extra)

<u>Neighbor nodes</u>: Corner grid pegs have higher cost than pegs closer to the goal

<u>Possible valid moves:</u> Total valid moves of all existing pegs at a board state

<u>Manhattan distance(extra):</u> To the neighboring nodes of the final peg in goal state

3) Combination of the two heuristics works better than one the neighbor node heuristic and worse than the possible valid moves heuristic.