

CSCI 3202 - Introduction to Artificial Intelligence
Instructor: Hoenigman
Assignment 3
Due Friday, September 18 by 4pm

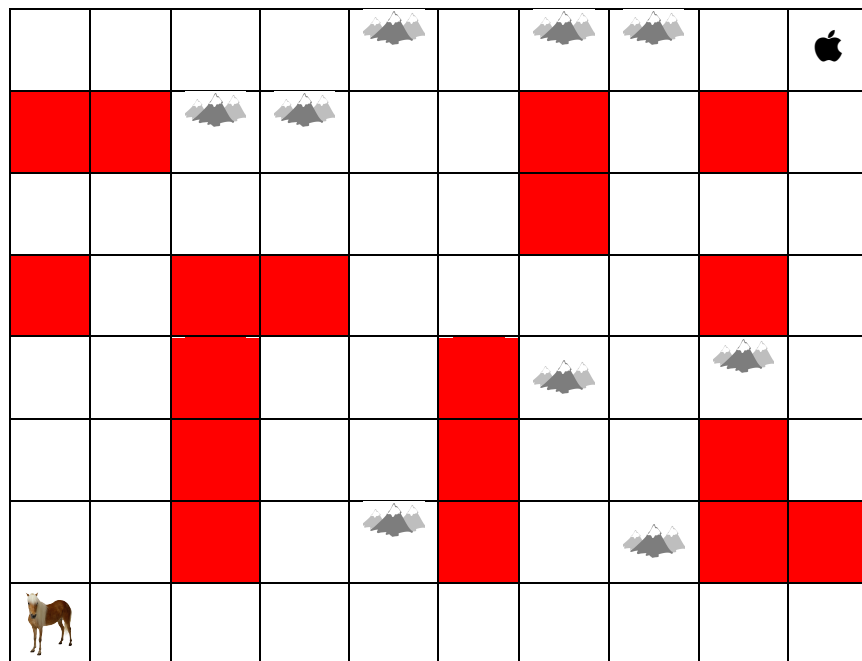
Problems:

Github

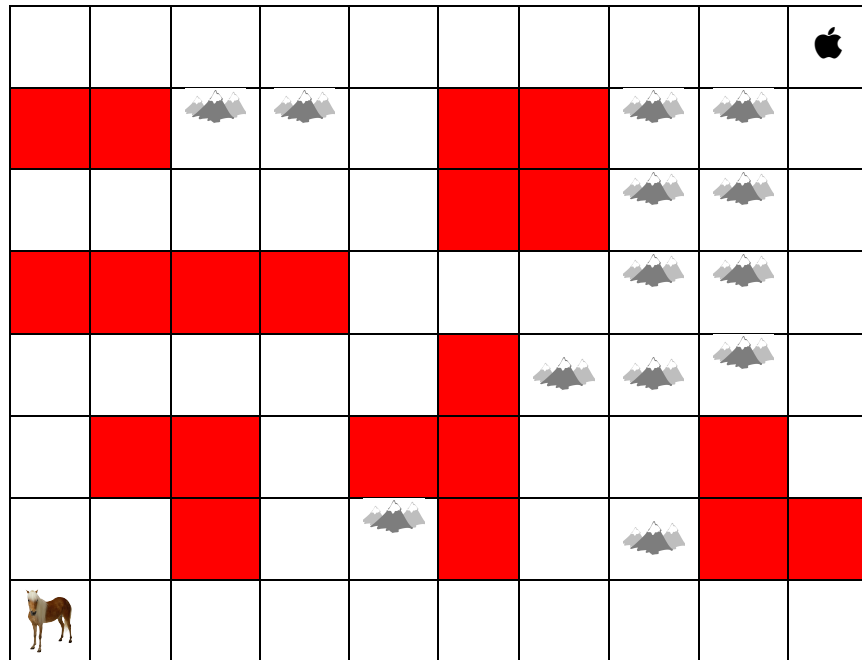
Create a directory called *ai3202* using the github username you submitted for Assignment 1. Within the directory, create a repository called *Assignment2*, and place all code for this assignment in that repository. We will be grading your assignment by pulling your code from that repository.

A* Search

The following pictures show fictional worlds where the horse needs to navigate the maze to get to the apple. Along the way, there are walls that he needs to go around (shown in red), and mountain that require additional cost to go over. Use A* search to find the lowest cost through the maze, and output the path that the horse takes through the maze to get to the apple.



World 1. Navigate the maze to get the horse to the apple.



World 2. Navigate the maze to get the horse to the apple. Don't get stuck in the mountains.

Rules for the search:

- Use a cost of 10 for each horizontal or vertical move and a distance of 14 for each diagonal move for non-mountain squares.
- Add an additional cost of 10 for each move into a mountain square. For example, a horizontal move into a mountain square would have a cost of 20 and a diagonal move into a mountain square would have a cost of 24.
- The horse can't move through a wall.
- Use the Manhattan distance as the heuristic function for the first search.
- Implement another heuristic function of your choosing and run the search again.

Implement a second heuristic

For the first part of the problem, you will use the Manhattan distance as your heuristic. Run the search again using a different heuristic of your choice. You are free to look up other heuristics online, or to invent your own heuristic that you think will work for this problem.

In github Read Me file for your assignment, explain the second heuristic that you tried and how the results compared to the results from the Manhattan distance.

Your explanation should include:

- The equations for the heuristic
- Motivation for selecting the heuristic, i.e. why you thought it would produce a good solution
- The performance of your heuristic compared to Manhattan distance, including the number of squares explored and the final path discovered

Implementation details

- The world is represented as a matrix, where 0 is an open square, 1 is a mountain square, and 2 is a wall.
- The worlds are on Moodle, called World1.txt and World2.txt. Your program needs to read in the file, using the filename as a command line argument for your program.
- Your program should also have a command line argument for which heuristic to use. Include an explanation of the command line arguments in your Read Me.
- The output of your program needs to include:
 - The cost of the path
 - The locations along the path
 - The number of locations evaluated
- Once you have the world data read into your program, there are two suggested options for implementation. You can either create a graph, where each square in the matrix is a node in the graph. You are welcome to use any built-in graph functionality to do this. The other option is to build nodes as you explore the matrix and add the nodes to the open and closed list during the search. In either case, there are certain properties that you may want your nodes to have:
 - Node:
 - location: (int, int)
 - distanceToStart: int
 - heuristic: int
 - f: int
 - parent: Node