Alex Petitjean

Outline of search.py & searchAgents.py

search.py

class SearchProblem:

Defines the structure of a SearchProblem. A search problem defines the state space, start state, goal test, successor function and cost function.

def tinyMazeSearch:

Solves 1 specific maze

def depthFirstSearch:

def breadthFirstSearch:

def uniformCostSearch:

def aStarSearch:

These are all different search methods.

def nullHeuristic:

Trivial heuristic returning 0 always.

searchAgents.py

class SearchAgent:

Defines the structure of all other SearchAgents and finds a path using a supplied search algorithm. By default, uses DFS.

class PositionSearchProblem <- SearchProblem:

Finds paths to a particular point in Pacman

class StayEastSearchAgent <- SearchAgent:

SearchAgent favoring positions more East

class StayWestSearchAgent <- SearchAgent:

Same as above, but favors West

def manhattanHeuristic:

Manhattan Distance

def euclideanHeuristic:

Euclidean Distance

class CornersProblem <- SearchProblem:

Search Problem that find a path through all four corners

def cornersHeuristic:

class AStarCornersAgent <- SearchAgent:

SearchAgent for FoodSearchProblem using A\* and FoodHeuristic

class FoodSearchProblem <- SearchProblem:

SearchProblem for finding a path that collects all the food (dots) in Pacman

def FoodHeuristic:

class ClosestDotSearchAgent <- SearchAgent:

class AnyFoodSearchProblem <- PositionSearchProblem <-SearchProblem:

SearchProblem for finding a path to any food. Same as PositionSearchProblem, but with a different goal.

class ApproximateSearchAgent <- SearchAgent:

Contest entry?

def mazeDistance:

Distance between two points in a maze.