Jaylon Kiper

CS-320-01: Artificial Intelligence

Dr. Johnson

9/8/21

Summary Document

With the Vacuum class of this project, this program provides the functions for the vacuum to move in any direction and count the number of sucks in a dirty unoccupied square. The methods in this class allow the vacuum to move on the board and which direction to move on the board. This involves a while loop to move the vacuum and a switch statement to determine which direction to move in. There is a suction method which sucks the dirty square on the board and counts how many times it sucked dirt. Other methods set the vacuum location, as well as returning the number of moves and sucks the vacuum went through. Finally, the last methods allow the vacuum to move up, down, left and right.

Onto the Square class, this program will set the squares as if they are dirty or clean, while also setting if the space is occupied or unoccupied. Two separate Square methods set the space as either dirty unoccupied and dirty occupied and the direction for each square to move in. Two boolean methods return if the space is dirty unoccupied and dirty occupied. The next voided method will tell if the square is dirty by setting it as true or false. Eight separate methods are setters and getters to set the vacuum to move in that direction and return that direction. The last toString method will return strings of the squares as how they are presented.

Finally, the Board class will implement the floor that will be vacuumed and show which squares are dirty and clean. Also this will count the number of moves and sucks by the vacuum. The first two Board methods initialize the position of the vacuum on the board and the size of the board itself. Both the runVacuum and createFloorAndRandomState method run other methods inside themselves. The setFloor method sets the floor’s size by its length and width. The generateSquare method produces the squares that are clean or dirty with two for loops. The voided linkSquaresAndRandomSetDirty method links each square to dictate what direction the vacuum can go in. There are two randomlyChangeStates methods that set the location of the randomly generated squares and provide said random squares on the board. The provideVacuumLocation sets the vacuum in a random position based on the size of the board with a do while loop. The cleanFloor method checks to see if the entire floor is clean or if the vacuum still needs to move. The cleanAndMoveVacuum method uses two other methods from the Vacuum class. The last couple of methods print the layout of the floor with its squares using food loops and checks for clean squares with a try catch exception.

The vacuum stops cleaning when each dirty space is clean. The vacuum will clean each square in all instances and be precise about it. Depending on the randomness of each square, the number of moves for the vacuum to take always will be different no matter what. If the grid was larger, more moves would be made to move across the board and the number of sucks would increase for each dirty square. Even if the space is clean, the vacuum will continue to move on to find dirty spaces. The best performance measure for the vacuum agent is being static, since it requires some time to compute if a square is dirty or clean.