Daniel Kantor and Erin Li

Homework 2 – Question 2.5

**Playing Soccer**

Performance: Winning the game and the number of goals scored

Environment: The soccer field, the soccer ball, players on team

Actuators: body parts of soccer players used for movement/kicking

Sensors: eyes, ears, touch

**Exploring the subsurface oceans of Titan.**

Performance: amount of ocean explored, quality of picture/video recorded

Environment: subsurface oceans of Titan

Actuators: parts of rover (wheels, arms), remote control

Sensors: camera (picture and video), location tracker, depth, sensor

**Shopping for used AI books on the Internet.**

Performance: getting the correct book, getting for cheapest price

Environment: internet shops

Actuators submit data, links, display data

Sensors: user input, content on websites

**Playing a tennis match.**

Performance: winning the tennis match, score of the tennis match, tennis statistics (unforced errors, winners, aces, etc)

Environment: tennis court, net, tennis ball, players

Actuators: tennis player body, tennis racket

Sensors: ears, eyes, touch

**Practicing tennis against a wall.**

p: keeping long rallies, number of missed balls

e: wall, balls, player

a: legs, arms, moving direction, tennis racket

s: tennis skills, eyes, footwork, ears

**Performing a high jump**

p: height of jump, distance traveled

e: jumper, bar to jump over

a: legs, arms, jumping pole

s: eyes

**Knitting a sweater.**

p: number of mistakes made, size of sweater

e: yarn, instructions

a: hands, needles

s: eyes, touch

**Bidding on an item at an auction.**

p: correct item will be paid fir, price, winner

e: items, customers, auctioneer, auction building

a: money in customers’ account, voice

s: hands, eyes, ears

**Report**

**Agent's rules:**

Our agent is vey simple. The determine how the agent will move it will choose a random number between 0 and 3 and depending on the number that is generated the agent will either move right, left, up, or down by one spot in the grid. If the agent attempts to move in a direction that will lead it out of the boundaries of the environment then the agent will not move and stay in its location that it was at. If the agent is at a location that is dirty then it will clean that location

**Any modifications you made to the environment or to the agent**

The performance score of the agent is calculated by recording the number of spots for each iteration run. These values go into a list that when the experiment stops running will have 100 values to represent each of the 100 iterations run. The values are then added up and divided by 100 to produce an average number of spots cleaned. This is done for all four of the experiments run.

**Average performance scores for each experiment**

Text

Description automatically generated

The average number of spots cleaned for experiments 1 and 3 was usually between 14 and 16 each time we ran the program. While the average number of spots cleaned for experiments 2 and 4 was typically between 26 and 27

**Conclusion on why one agent is better than the other**

The agent that jumped around randomly performed much better than the one that moved up, down, left, and right by one stop in the grid each movement. The random movement agent performed better than the other one because the other agent when at a boundary and attempting to move out of boundary would not be able to move and would stay in the same location, missing out on opportunities to clean other locations, an issue that the randomly “teleporting” agent didn’t have. Another reason that the random movement agent performed better is because the environment was randomly generated. Because the environment was randomly generated it is improbable that many dirty locations are next to each other. Therefore, an agent that moves around the environment randomly would have more success encountering the randomly spread-out dirty locations as opposed to an agent that would be focused on a subset of the environment space by its movement rules, perhaps not exploring all corners of the environment.