CS531 Programming assignment #1

Matt Meyn, Durga Harish, and Jon Dodge

January 20, 2017

1 Introduction

The goal of this project is to assess the results of various agent control strategies in a simple path planning domain. The three strategies we explore are

1. Memoryless deterministic reflex agent
2. Randomized reflex agent
3. Deterministic Model-based reflex agent

We found... FIXME provide summary of results once that is known

2 Description of agents

Here we will describe our agents as a set of rules, and give a diagram illustrating the path it takes in each environment.

2.1 Memoryless deterministic reflex agent

Essentially, this agent will move forward until it hits a wall, then turn right.

FIXME typeset this pseudocode better

```
if isSpaceDirty: return self.ActSuckDirt
if not isFacingWall: return self.ActMove
if isHome: return self.ActTurnOff else: return self.ActTurnRight
```

FIXME generate diagrams

2.2 Randomized reflex agent

Essentially, this agent will randomly choose actions with a uniform distribution between the available actions. In general, available actions are to move, turn left, and turn right, though the existence of a wall may remove the availability of the move option. The case that the agent is on the home square is treated similarly, it has equal probability to turn off or try to continue vacuuming.

FIXME typeset this pseudocode better

```
if isSpaceDirty: return self.ActSuckDirt
if isHome: if random.randint(0, 5) == 0: return self.ActTurnOff
if isFacingWall: return self.TurnRandom()
else: if random.randint(0, 3) == 0: return self.TurnRandom() else: return self.ActMove
```

FIXME generate diagrams

2.3 Deterministic Model-based reflex agent

FIXME implement this our first plan is to do a spiraling space filling curve, and our second plan is to do a snaking space filling curve

FIXME typeset this pseudocode better

FIXME generate diagrams

3 Experimental Setup

4 Results

5 Discussion

1. What is the best possible performance achievable by the simple reflex agent in the two environments? What prevents it from achieving the goal of cleaning the rooom perfectly in each case?

2. How well does the random agent perform? Tune any parameters of the random agent to improve its performance. Give a table showing the number of actions it took to clean 90% of the room for each trial. What is the average of these numbers for the best 45 trials? What are the costs and benefits of randomness in agent design?

3. How does the memory-based deterministic agent perform in the two environments? Is it able to clean the room perfectly? How long does it take? Can the agent be improved with more memory?

4. What are the tradeoffs between the random and determistic agents? How would you design better agents for more complex environments, say with polygonal obstacles?

5. What did you learn from this experiment? Were you surprised by anything?