Marcos Ondruska
Student id: 2685885

Assignment 1

1) i) Rationality: My take on reading the course material is that “rationality” in terms of AI is the agent’s ability to be in a situation and make a decision that results in a functional outcome. There seems to be a curious/interesting parallel in psychology between human value-oriented belief systems and some of the material in the book where agents are “guided” in uncertain outcomes through either goal oriented, performance-oriented guides or others, and the results that are obtained.
   ii) Autonomy: When an agent is using its own percepts and learning abilities to “rationally” make choices it is said to be autonomous. The more it relies on the previous knowledge of whomever built it, the less it is autonomous.
   iii) Reflex agent:
   This is the simplest form of agent. It selects actions based on the current percept only. Most of the code-based rules are pre-engineered in the form of if/else if statements or Booleans. So, in my mind, there is not a lot of decision making that is being made and not much intelligence in this model. The rules are fixed, and actions are static based on the percepts.
   iv) Model-based-agent:
   This type of agents has a more dynamic view of its world and will include multiple models that will influence its decision making. There are two main models that are used – a transition model and a Sensor model. Using the AI that plays tennis below, and example of transition models for it might be 1) playing against a wall, 2) playing against another agent, 3) playing when the conditions are wet. A sensor model for the playing when wet scenario would include something like, droplet-shaped objects appear in the camera images making it harder to see the ball.
   v) Goal-based-agent:
   This type of agent improves on the former models by giving the agent a method for dealing with ambiguity. For example – if a car is stopped at a stop sign, and has the choice to go straight, left, or right, then the goal (destination) will guide this choice allowing for an improved set of outcomes overall.
   vi) Utility-based-agent:
   This is the highest level of thinking in the agents surveyed so far. While in the above example of the car and its goal of arriving at a particular destination solves some problems, there are also many other considerations that might be pertinent to the outcome. Things such as safety of drive, time taken, profit factor of the drive in the case of a taxi, passenger comfort, and many others. To allow for this utility based agents have what is called a utility function, which is just a performance score of how well the considerations were catered to during the ride. Part of the decision making on the route would take into account multiple routes and the best expected utility of these routes.

2) Automated Taxi:

Performance measure: Safety, speed, comfort, trips, maximize profits.
Environment: Roads -> urban, suburban, highway | other traffic, other road users -> pedestrians, cyclists, motorbikes | customers | weather
Actuators: Steering, accelerator, brake, signals, horn, display, speech.
Sensors: Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

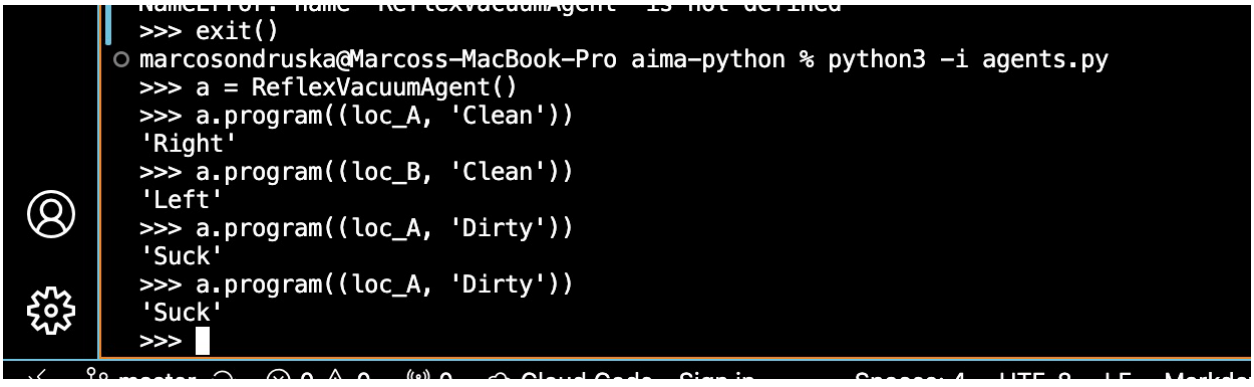
Practicing Tennis Against a Wall:

Performance measure: Safety, accuracy, hitting power, spin on ball, reaction time (from wall and bounces), speed of movement, consistency.
Environment: physical space that includes a wall (tennis court or a practice area), wall, net demarcation, weather conditions (rainy, windy, sunny), other agents (humans, other robots, pets, coaches, spectators).
Actuators: Arms, hands, legs, feet, neck, tennis racket.
Sensors: Camera, Motion sensors, Touch sensors, position sensors, environment sensors, proximity sensors, audio sensors.

Shopping for used AI Books on the Internet:

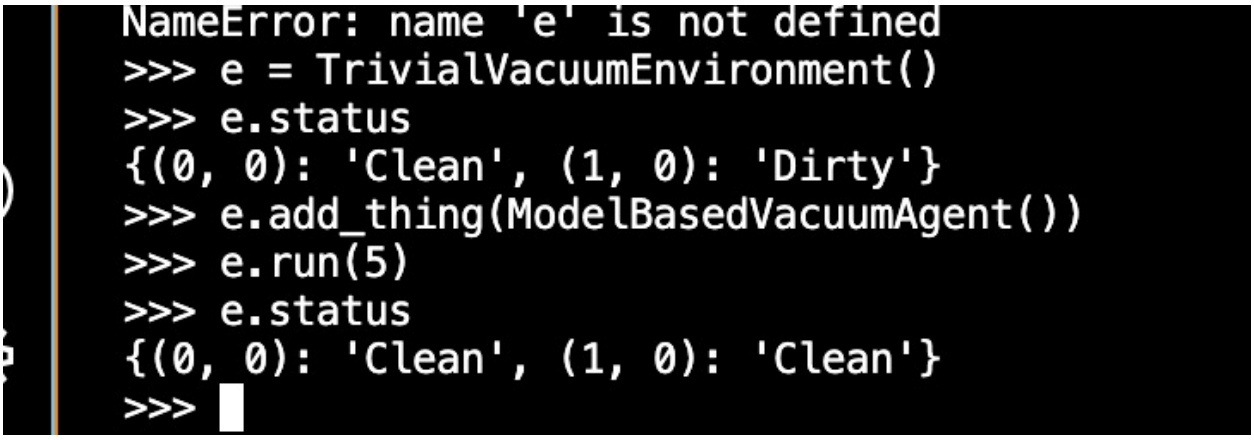
Performance measure: Successful find of books that user wants, reduced cost, speed of find.
Environment: websites, search engines, other agents (other humans listing books)
Actuators: feedback, software to find the relevant books on the internet.
Sensors: Keyboard entry.

3)



This output initializes an instance of the ReflexAgentVacuum() and then “tests” to see what is programmed as the world for the agents. It seems to be 2 locations –> loc_A and loc_B, and it has 2 conditions -> clean and dirty. The code entered propagates a programmed action.

4)



This code prints the status of e. Initially e is initialized as a TrivialVacuumEnvironment(), and the status of that is called and printed. A ModelbasedVacuumAgent is then added to e (or maybe even defined for e), and instructed to run, after which we check the status. The status tells us that there is a change in that area “1, 0” is now clean.