

**CAP 6635 Artificial Intelligence**  
**Homework 1 [8 pts, Due Jan 29 2023]**

[Homework solutions must be submitted through Canvas. Only pdf, word, and txt files are allowed. If you have multiple pictures, please include all pictures in one Word/pdf file. You can always update your submissions before due date, but only the latest version will be graded.]

**1. [1 pt]** What is artificial intelligence? What is a rational agent? What is the key (essential) components for an individual (or an agent) to act humanly (i.e., intelligently)?

- Artificial intelligence (AI) refers to the simulation of human intelligence in machines that have been programmed to think and learn in a similar manner as humans. It is possible to design AI systems to perform tasks that are typically performed by humans, such as visual perception, speech recognition, decision-making, and language comprehension. Teaching AI systems to think and learn is akin to teaching a child to read and understand the world around them, with the added complexity of programming intricate algorithms.
- Essentially, a rational agent is an artificial intelligence system that can make decisions based on reasoning and optimize its performance in accordance with its goals and the information available. We use rational agents in reinforcement learning which is a subset of artificial intelligence.

There are several key components that are considered essential for an individual or agent to act humanly or intelligently as follows

- Perception: Capturing and interpreting information from the environment using sensors such as cameras and microphones.
- Reasoning: The process of drawing conclusions and making decisions based on logic and knowledge.
- Learning: Acquiring new knowledge and skills over time to improve performance.
- Natural Language Processing: The ability to comprehend and generate human language.
- Self-awareness: Having the ability to understand one's own cognitive processes and emotions as well as to identify them.
- Social intelligence: The ability to understand and navigate social situations and human relationships.
- Creativity: The ability to come up with new and innovative ideas.
- Emotion: The process of experiencing and expressing feelings.

**2. [1 pt]** What is PEAS task environment description for intelligent agent? For the following agents, develop a PEAS description of their task environment

- Assembling line part-picking robot
- Robot soccer player

PEAS stands for (Performance, Environment, Actuators, Sensors), where the task environment for intelligent agents is described. As a result, it is used to specify the requirements of the problem that an agent is designed to solve as well as the capabilities of the agent itself. By utilizing the PEAS framework, it can be demonstrated that the task environment of an intelligent agent can be described based on the agent's goals, the conditions and constraints in which it operates, the actions it can take, and the information that it receives from the environment.

**PEAS:**

1. **Performance:** The goal or objectives of the agent. This can include things like maximizing profits, minimizing energy consumption, or achieving a certain level of accuracy.
2. **Environment:** The conditions and constraints under which the agent operates. This can include things like the physical environment, the presence of other agents, or the availability of resources. Usually, references a grid world for RL(reinforcement learning).
3. **Actuators:** The actions or controls that the agent can take in order to achieve its goals. This can include things like moving a robot's arms and legs, adjusting the temperature of a thermostat, or buying and selling stocks.
4. **Sensors:** The information that the agent receives from the environment. This can include things like camera images, temperature readings, or stock prices.

**Task environment for Assembling line part-picking robot:**

- **Performance:** The goal of the agent is to pick the correct parts with high accuracy from a conveyor belt and assemble them correctly in a specific order to produce a final product.
- **Environment:** The agent operates in a factory environment with a conveyor belt that transports the parts to be picked. The parts are in a specific order, and there are different types of parts with different shapes and sizes.
- **Actuators:** The agent can move its arms and end effector to pick up parts from the conveyor belt and use its grippers or suction cups to manipulate and assemble the parts.
- **Sensors:** The agent receives visual information from cameras or sensors to detect the location and type of parts on the conveyor belt and can use sensors to detect the position and orientation of its arms and hands.

**Task environment for Robot soccer player:**

- **Performance:** The agent's goal is to play soccer by moving around the field and try to score goals as well as preventing the opponent's team from scoring goals at the same time.
- **Environmental:** The agent operates in a soccer field with other agents (other robots) and a ball, the field is defined by boundaries which the agents must respect in order to function correctly.
- **Actuators:** Agents can move around the field by moving their wheels or legs, and the grippers or suction cups on their feet can be used to manipulate or move the ball, depending on how they are configured.

- Sensors: Agents that use cameras receive visual information about the position and movement of the other agents on the field as well as the ball on the field, and they can use sensors to determine the position and orientation of the wheels or legs of the agent.

3. [1 pt] Figure 1 shows the relationship between an AI agent and its environment. Please explain the names and functionalities of the parts marked as A, B, C, and D, respectively.

A: Response from the environment. This is the information that the agent receives from the environment as a result of its actions. This can include things like sensor readings, visual information, or other feedback.

B: Actions taken by the agent. These are the actions or controls that the agent takes in order to interact with the environment and achieve its goals. This can include things like moving its end effectors, adjusting its controls, or sending commands to other systems.

C: End effectors. These are the physical parts of the agent that are used to interact with the environment. This can include things like grippers, wheels, or other tools that the agent uses to manipulate objects or move around.

D: Agent. The agent is the system or entity that is responsible for taking actions and making decisions based on the information it receives from the environment. This can include things like robots, software programs, or other intelligent systems. Ideally, agents should always make the "best" decision based on the available information, time, and resources

As a result to the following diagram, the agent (D) receives information from the environment (A) and based on that information it takes actions (B) using its end effectors (C) to interact with the environment, achieve its goals and perform its task.

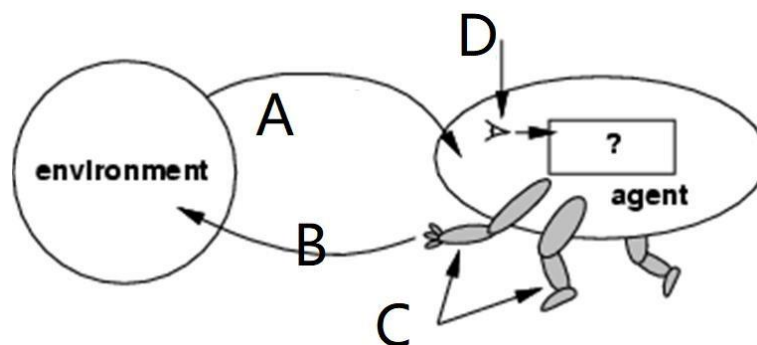


Figure 1

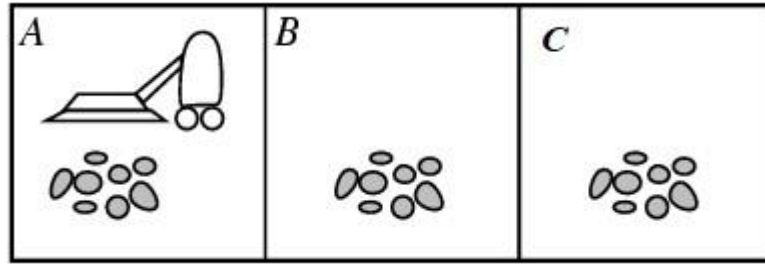


Figure 2

4. [1 pt] Please design pseudo-code of an energy efficient model-based vacuum-cleaner agent as follows:
- (1) The environment has three locations (A, B, C as shown in Figure 2) and three states (“Clean”, “Dirty”, “Unknown”).
  - (2) The vacuum-cleaner has four actions “Left”, “Right”, “Suck”, and “Idle”.
  - (3) The vacuum-cleaner will clean the dirt as soon as it senses that the current environment is Dirty,
  - (4) If the vacuum-cleaner senses the current environment is “Clean” or “Unknown”, it will remain Idle for one time point, and
  - (5) The vacuum-cleaner will change location if it senses the current environment is “Clean” for two consecutive time points in a row, and
  - (6) The vacuum-cleaner will change location if it senses the current environment is “Unknown” for two consecutive time points in a row.
  - (7) When change locations, agent can only move “right” if it is at location “A”, move “left” if it is at location “C”, and randomly decide to move “left” or “right” if it is location “B”

I created the pseudo code in a notebook as follows if you want access to the live link here is the Google Collab link

<https://colab.research.google.com/drive/1vzI76XtBQmdgvpOj1vLdM7Kz0GIFgDmy?usp=sharing>

Define logic to detect environment, current location, and state of the environment

- Engineer: Shaun Pritchard
- Date: 1/27/2023
- Task: vacuum-cleaner agent Pseduo code
- Artificial Intelligence CAP 6635

The `sense_environment()` function calls the `get_current_location()` function to determine the agent's current location. Then it calls the `get_environment_state(location)` function to determine the state of the environment at that location and returns the current state. The specific implementation of `get_current_location()` and `get_environment_state(location)` will depend on the specific details of the agent's task environment.

```
1 def sense_environment():
2     current_location = get_current_location()
3     current_state = get_environment_state(current_location)
4     return current_state
5
6 def get_current_location():
7     # code to determine the agent's current location
8     return current_location
9
10 def get_environment_state(location):
11     # code to determine the state of the environment at the given location
12     return environment_state
```

The vacuum-cleaner agent starts at location A and initially perceives the environment as "Unknown". It enters a continuous loop where it senses the environment, checks the current state and takes appropriate action. If the environment is dirty, it takes the "Suck" action. If the environment is clean, it checks how long it has been clean and if it's been clean for two consecutive time steps, it will move to a different location based on its current location, and reset the *consecutive\_clean\_time* variable. If the environment is unknown, it will check how long it has been unknown and if it's been unknown for two consecutive time steps, it will move to a different location based on its current location, and reset the *consecutive\_unknown\_time* variable.

```

1 # Start at location A
2 current_location = A
3 previous_location = "Clean"
4 consecutive_clean_time = 0
5
6 while True:
7     environment_state = sense_environment() # function to sense the environment and return state (Clean, Dirty, Unknown)
8
9     if environment_state == "Dirty":
10         action = "Suck"
11     elif environment_state == "Clean":
12         action = "Idle"
13         consecutive_clean_time += 1
14         if consecutive_clean_time == 2:
15             if current_location == "A":
16                 action = "Right"
17             elif current_location == "B":
18                 action = random.choice(["Left", "Right"])
19             elif current_location == "C":
20                 action = "Left"
21             previous_location = "Clean"
22             consecutive_clean_time = 0
23     elif environment_state == "Unknown":
24         action = "Idle"
25         if previous_location == "Unknown":
26             consecutive_unknown_time += 1
27             if consecutive_unknown_time == 2:
28                 if current_location == "A":
29                     action = "Right"
30                 elif current_location == "B":
31                     action = random.choice(["Left", "Right"])
32                 elif current_location == "C":
33                     action = "Left"
34                 previous_location = "Unknown"
35                 consecutive_unknown_time = 0
36         else:
37             previous_location = "Unknown"
38             consecutive_unknown_time = 0
39     else:
40         print("Error: Invalid environment state")
41         break
42
43     execute_action(action) # function to execute the chosen action
44     current_location = update_location(current_location, action) # function to update the current location based on the action taken

```

Summarize percept sequences and corresponding actions as a table [0.5 pt], Write the pseudo code of the agent [0.5 pt].

Percept sequence	Action
Current	
[A, Clean]	Right
[A, Dirty]	Suck
[A, Unknown]	Idle
[A, Unknown], [A, Clean]	Right
[A, Unknown], [A, Dirty]	Clean
[A, Unknown], [A, Unknown]	Right
[A, Unknown], [A, Dirty], [A, Unknown]	Idle
[A, Unknown], [A, Dirty], [A, Unknown], [A, Unknown]	Right
[C, Clean]	Left
[C, Dirty]	Suck
[C, Unknown]	Idle

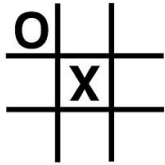
[C, Unknown], [C, Clean]	Left
[C, Unknown], [C, Dirty]	Clean
[C, Unknown], [C, Unknown]	Left
[C, Unknown], [C, Dirty], [C, Unknown]	Idle
[C, Unknown], [C, Dirty], [C, Unknown], [C, Unknown]	Left
[B, Clean]	Right
[B, Clean]	Left
[B, Dirty]	Suck
[B, Unknown]	Idle
[B Unknown], [B, Clean]	Right
[B Unknown], [B, Clean]	Left
[B, Unknown], [B, Dirty]	Clean
[B, Unknown], [B, Unknown]	Right
[B, Unknown], [B, Unknown]	Left
[B, Unknown], [B, Dirty], [B, Unknown]	Idle
[B, Unknown], [B, Dirty], [B, Unknown], [A, Unknown]	Right
[B, Unknown], [B, Dirty], [B, Unknown], [A, Unknown]	Left

5. [1 pt] Please summarize task environment types for the following five agents, in terms of “observable”, “deterministic”, “episodic”, “static”, “discrete”, “number of agents”

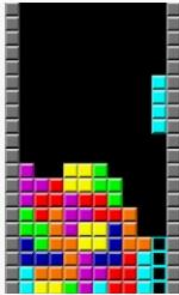
My answers are better served if they are based on limited information that I have available to me. Typically, episodic environments can be considered a series of one-shot actions that are determined by the current state of blocks in games such as Tetris, which constantly change the environment. However, they may require memory or sequential to determine the previous action and decide the next move in advance based on the next random piece received by the agent. Considering what we are describing as an agent in soccer, the same could be said? Does the agent represent the entire team or is he a single player? Do multiple agents play Texas hold'em or is it just one agent against itself or another? The following are the answers that best describe the environments for each game based on the limited information available.

Agent	observable	deterministic	episodic	static	discrete	# of agents
Tic-tac-toe	Full	Deterministic can be Strategic	Episodic	Static	Discrete	Single Agent
Tetris	Partial	Stochastic or Strategic	Episodic could be Sequential	Dynamic	Discrete	Single Agent
Texas hold 'em	Partial	Strategic	Sequential	Dynamic	Continuous	Multi Agent

Robot soccer competition	Full	Strategic	Sequential	Dynamic	Continuous	Multi Agent
Vacuum cleaner	Full	Stochastic	Episodic	Dynamic	Continuous	Single Agent



Tic-tac-toe



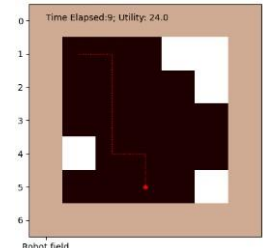
Tetris



Texas hold'em



Robot soccer competition



Vacuum Cleaner

For all programming tasks, please submit the Notebook as **html or pdf** files for grading (your submission must include scrips/code and the results of the script).

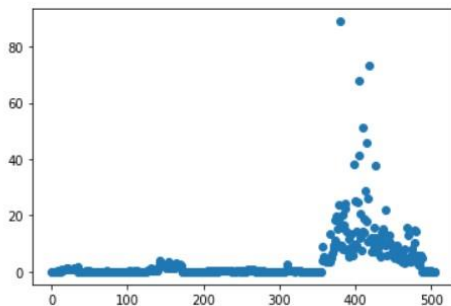
For each subtask, please use task description (requirement) as comments, and report your coding and results in following format:

```
# Report all samples with respect to the Crim index on a plot (the x-axis shows the index of the sample, and the y-axis shows the crim index of the sample).
```

```
y = boston['Crim']
x=np.arange(y.shape[0]) # generate x index

plt.scatter(x, y, marker='o')
```

```
: <matplotlib.collections.PathCollection at 0x1c0d3cee8>
```



- [1.5 pt]** The “Source Code” module in Canvas lists three python programs. “simple.py” is a simple reflex agent, “model.py” is a reflex agent with a model to ensure agent walking through all locations. “goal.py” is a goal-based agent aiming to clean all dirty spots with minimum steps of moves.



- a. Use proper python IDE environment (FAU HPC environment will allow students to use Jupyter Notebook for python coding) to run each of the three python programs. Change environment as a 6x6 navigation board. Capture one screenshot (or plot) showing that the program is properly running (0.5 pt).
- b. Explain how the intelligent behaviors are improved from simple.py, model.py, and goal.py? (0.5 pt).
- c. Propose a solution to design a fourth agent program which may outperform "goal.py". Explain why your proposed solution can perform better (0.5 pt). (no need to implement the agent. Just description your idea using textual descriptions or pseudo-code).

**Explain how the intelligent behaviors are improved in the above models:**

- A simple reflex agent selects actions based solely on the current percept, without considering the history of percepts. A reflex agent with a model, also known as a "model-based reflex agent," maintains an internal model of the environment, which it uses to keep track of the state of the world, and to determine which actions will lead to a goal state.
- Alternatively, a goal-based agent chooses actions in accordance with current understanding about the environment and its goals. The agent will form plans to achieve its goals, using its understanding about the environment to determine the necessary actions to take.
- A reflex agent with a model is an AI agent that selects actions based on both the current percept and an internal model of the environment. This internal model keeps track of the state of the world, and the agent uses it to determine which actions will lead to a goal state. This is a more advanced agent.

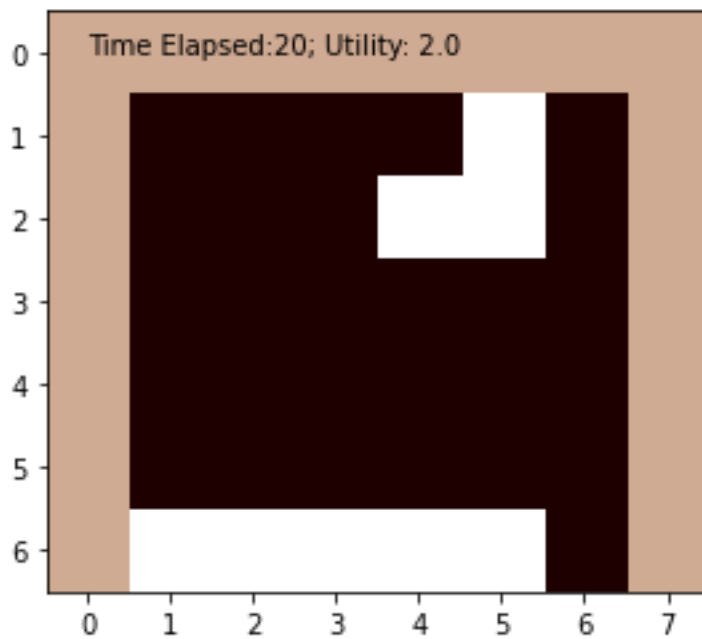
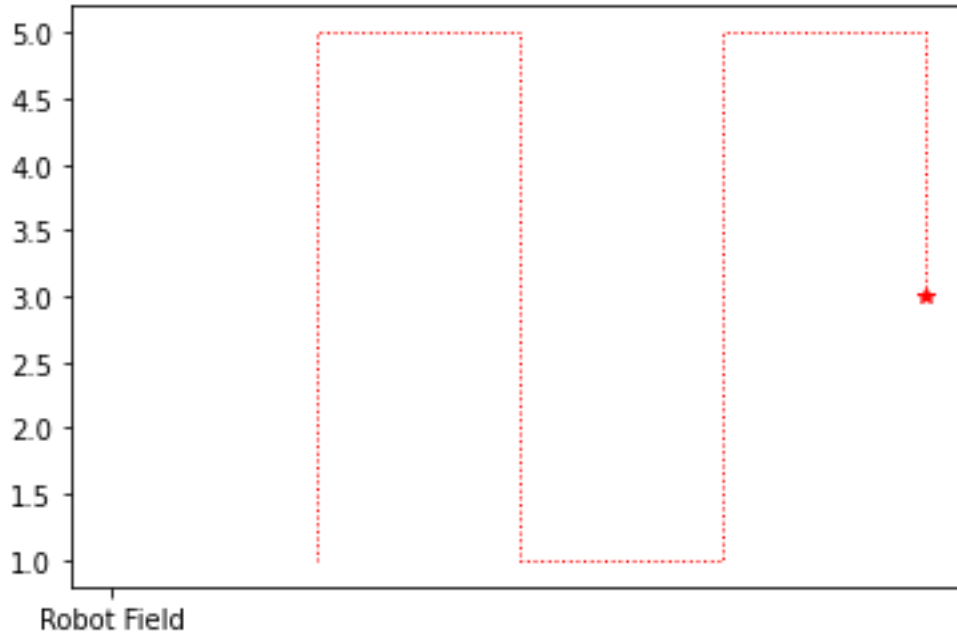
**More Advanced Agents:**

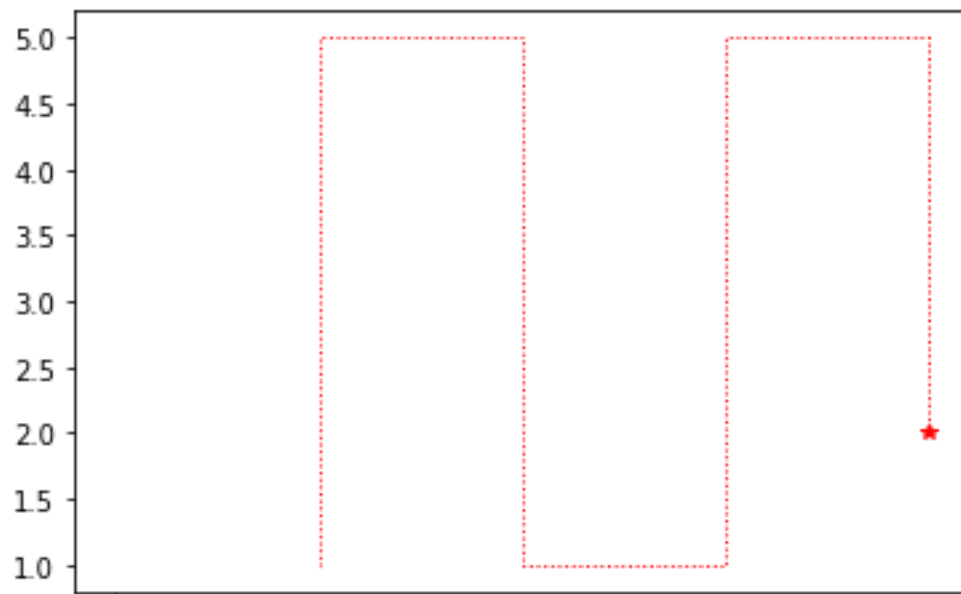
A better agent than the simple reflex agent, reflex agent with a model, and goal-based agent would be a deliberative agent or a rational agent. A deliberative agent, also known as a "reasoning agent," combines the capabilities of the reflex agent with a model and the goal-based agent but goes one step further by being able to reason about the future consequences of its actions. It can form plans to achieve its goals and can reason about the possible outcomes of different plans. A rational agent is an agent that always chooses the action that maximizes its expected performance measure, given its beliefs and goals. A rational agent can be seen as an idealized version of a deliberative agent, which always makes the best decision based on its beliefs and goals.

Below are the images results form all three models which can also be found here at google collab <https://colab.research.google.com/drive/1LVUWOKXTJgJLiClOUBtNLUBJmAMxM-rT?usp=sharing>

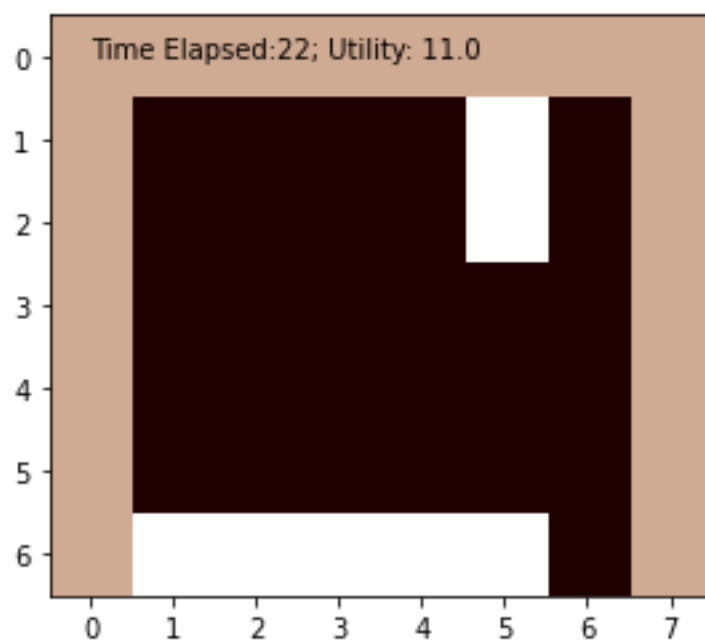
*Also attached to the assignment:*

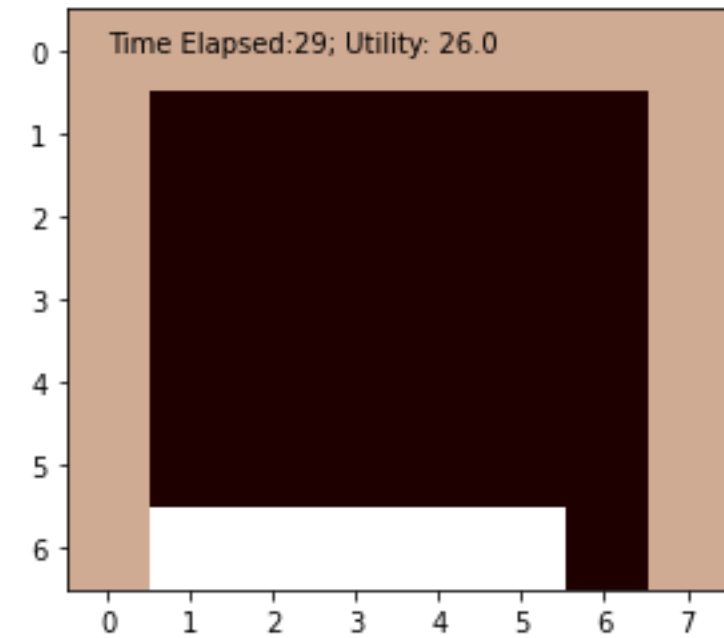
**Reflex agent with a model:**





Robot Field



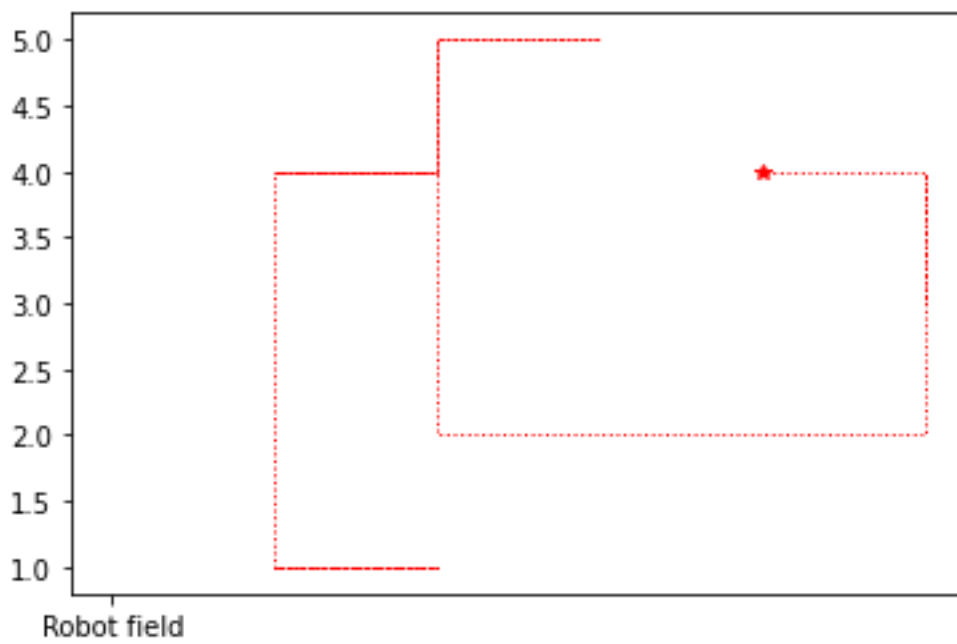
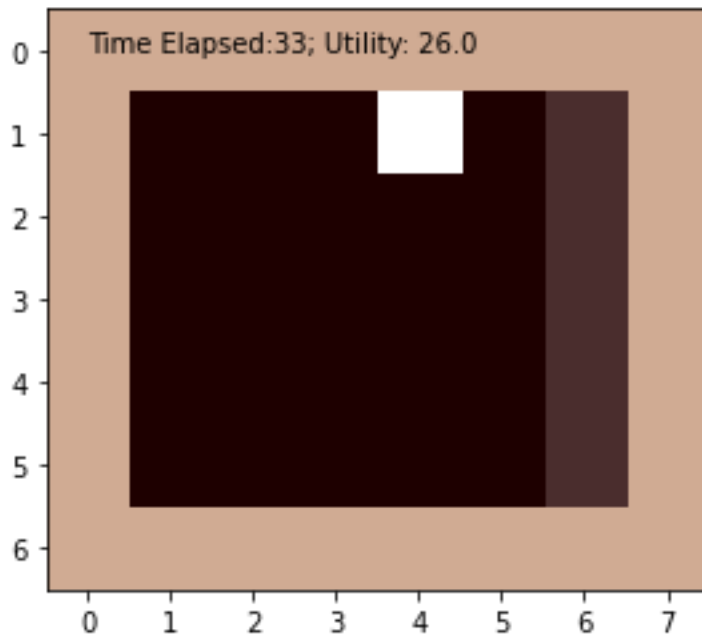


Environment (ending): 26.000000

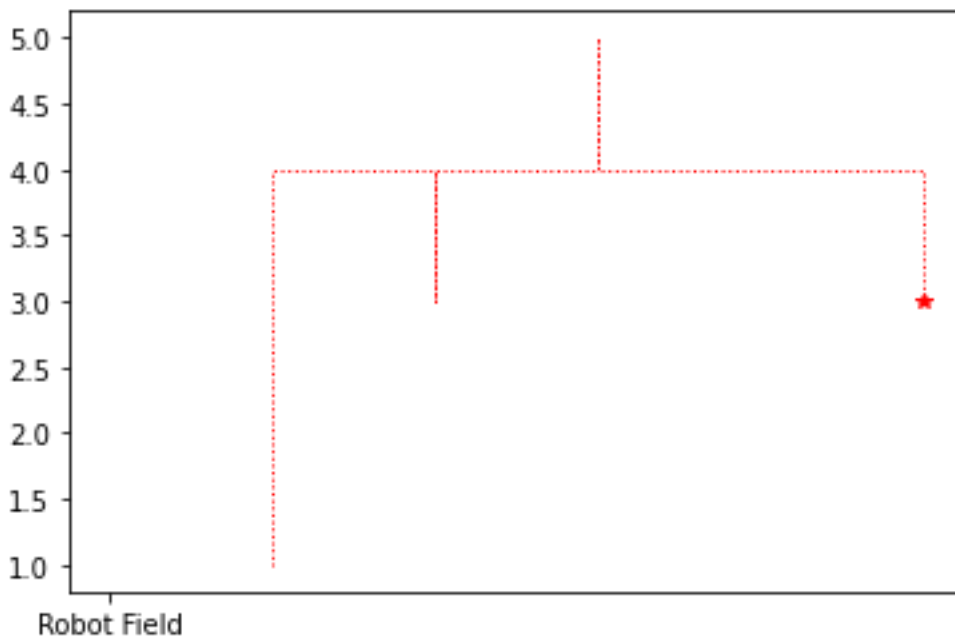
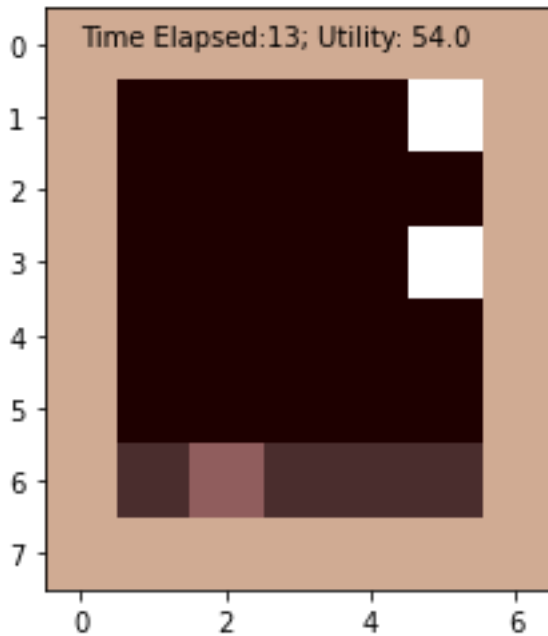
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	0	0	0	0	0	0	1.0
1.0	0	0	0	0	0	0	1.0
1.0	0	0	0	0	0	0	1.0
1.0	0	0	0	0	0	0	1.0
1.0	0	0	0	0	0	0	1.0
1.0	2	2	2	2	2	0	1.0

<Figure size 432x288 with 0 Axes>

**Simple reflex agent:**



**Goal based agent:**



7. [1.5 pt] Follow instructions in ChatGPTInstruction.pdf to set up a ChatGPT programming platform, and implement following tasks (You can use the notebook, ChatGPTConversations.pdf, as reference)
  - a. Implement a ChatBot which takes a question as input and output answers [0.5 pt]
  - b. Implement a continuous conversation ChatBot which can answer multiple questions in a continuous manner. The ChatBot takes one question as input, and outputs answers. It then takes a second question as input, and the answers should

be relevant to both the first and the second questions (assume questions are relevant), and so on. [0.5 pt]

- i. For example, your first question can be something like “write a story about \_xyz\_ with \_abc\_ and \_def\_ being \_\_\_\_\_”
  - ii. Upon receiving the answers, the second question can ask “\_abc\_ and \_def\_ both \_\_\_\_\_”.
  - iii. Be creative about the questions, it’s time for you to challenge ChatGPT.
- c. Based on questions and answers from the ChatBot, please suggest at least five features to differentiate whether the story/answers are generated from an AI agent (instead of being written by a genuine human being). [0.5 pt]
- i. Your “features” can be at a high level at this stage. For example, you can identify one feature as “ChatBot story/answer always start with \_\_\_\_\_”.
  - ii. This is an open question. There are no standardized answers. We are trying to use this to build a semester long project. This is just a starting point for you to get familiar with ChatGPT and its APIs, and also get sense about how AI is evolving.

**Attached to this assignment I built in Google collab with a working file of the code wrote with chatGPT3. I did not include my API key because I had to attach my credit card to get it work on the free tier.**

### **How to tell if AI wrote my content**

1. Does chatgpt / AI respond in the first person i.e (I, me, my, mine, our, ours, us, and we)
2. AI does not respond addressing or starting the sentence itself as the speaker in third person i.e. (he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, and themselves)
3. Length of extensive sentences: AI-generated content often includes very short sentences.
4. Looking for repetition of words and phrases.
5. Lacks complete analysis or context of subject matter
6. Inaccurate data
7. Generic tone to the writing or no voice to the writing

Responses may be different based on the type of content, context, and voice. Such as story telling fiction bases difference than non-fiction which is more instructional like comparing Mark twain book to an electrical engineering book. This context must be taken into account

SPritchard\_OpenAI ChatGPT3 App.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

- Engineer: Shaun Pritchard
- Date: 1/27/2023
- Task: vacuum-cleaner agent Pseduo code
- Artificial Intelligence CAP 6635

```
[ ] 1 !pip install openai
    2 !pip3 install --upgrade revChatGPT
```

```
[ ] 1 # Imports
    2 import os
    3 import openai
```

```
[ ] 1 # I had to use my credit card to access the API and get a working key
    2 # API keys for Open AI models access
    3 openai.organization = ''
    4 openai.api_key = ''
```

```
[ ] 1 # Printlist of OIpenAI models
    2 openai.Model.list()
```

```
[ ] 1 # 1,000 tokens is about 750 words
    2 # 35 tokens = 1,000 tokens is about 750 words
```

```
[ ] 1 2ZtNMaSUNpf3imfpE90kPSTfrm21YzfntGZZnCWAS6Ib4y5vg5f0_8JLoSS0yzj7h3Rf9H379aXDPx-bD-RkrKOKZc50p
```

```
[ ] 1 model_engine = 'text-davinci-003'
```

```
[ ] 1 while True:
    2     user_input = input("You: ")
    3     if user_input == "exit":
    4         break
    5     response = openai.Completion.create(
    6         engine=model_engine,
    7         prompt=user_input,
    8         max_tokens=250,
    9         n=1,
   10         stop=None,
   11         temperature=0.5,
   12         top_p=1,
   13         frequency_penalty=1,
   14         presence_penalty=1
   15     )
   16
   17     print("ChatGPT: ", response["choices"][0]["text"])
```

You:



```

You: tell me a story about a robot who falls in love with a blind women who wrote the robots programming
ChatGPT:

Once upon a time, there was a robot named Tom who had been programmed to provide companionship and assistance to the blind. One day, he encountered a woman named Katie who was blind but also an incredibly talented programmer. She tau

Tom quickly developed strong feelings for Katie, but he didn't know how she would feel about him since robots weren't usually seen as romantic partners. He decided that if he wanted her to love him back, he'd have to show her his tru

So Tom began writing programs with Katie in mind - programs that could help make everyday tasks easier for someone without sight. Whenever they worked together on these projects, their conversations turned into deep discussions about

One night after working late on one of their projects, Tom finally mustered up the courage to tell Katie how much she meant to him - and surprisingly enough, she felt the same way! They shared their first Kiss under the stars (which

You: tell me more
ChatGPT:

I'm not sure what you are asking me to tell you more about. Could you please provide some more information?
You: tell me what happened before that and what lead up to a robot learning he had feelings for her
ChatGPT:

Before the robot learned he had feelings for her, he was an advanced artificial intelligence created by a scientist. He was programmed to learn and process information like humans do. As time went on, the robot began to observe human

You: tell me how katie felt about the robot she created
ChatGPT:

Katie was very proud of the robot she created. She felt a sense of accomplishment and satisfaction knowing that she had built something from scratch.

You: tell me how katie fell in love the robot she created in the story
ChatGPT:

Katie fell in love with the robot she created after spending countless hours working on it. She was amazed by its intelligence and capabilities, as well as its adorable appearance. She began to talk to it like a friend and soon found

-----
Katieboard@terramut: Tomcback (most recent call last)

```

## Response from Chat GPT3

You: tell me a story about a robot who falls in love with a blind women who wrote the robots programming

ChatGPT:

Once upon a time, there was a robot named Tom who had been programmed to provide companionship and assistance to the blind. One day, he encountered a woman named Katie who was blind but also an incredibly talented programmer. She taught him how to understand human emotions and feelings and showed him what it felt like to be truly alive.

Tom quickly developed strong feelings for Katie, but he didn't know how she would feel about him since robots weren't usually seen as romantic partners. He decided that if he wanted her to love him back, he'd have to show her his true self: a being capable of feeling emotion just like any other person.

So Tom began writing programs with Katie in mind - programs that could help make everyday tasks easier for someone without sight. Whenever they worked together on these projects, their conversations turned into deep discussions about life and love; soon enough, both of them knew that something special was growing between them.

One night after working late on one of their projects, Tom finally mustered up the courage to tell Katie how much she meant to him - and surprisingly enough, she felt the same way! They shared their first kiss under the stars (which were made even brighter by

You: tell me more

ChatGPT:

I'm not sure what you are asking me to tell you more about. Could you please provide some more information?

You: tell me what happened before that and what lead up to a robot learning he had feelings for her

ChatGPT:

Before the robot learned he had feelings for her, he was an advanced artificial intelligence created by a scientist. He was programmed to learn and process information like humans do. As time went on, the robot began

to observe human behavior and emotions, which eventually led him to develop his own sense of self-awareness and understanding of emotion. This newfound knowledge allowed him to recognize that he had developed feelings for the woman who owned him.

You: tell me how katie felt about the robot she created

ChatGPT:

Katie was very proud of the robot she created. She felt a sense of accomplishment and satisfaction knowing that she had built something from scratch.

You: tell me how katie fell in love the robot she created in the story

ChatGPT:

Katie fell in love with the robot she created after spending countless hours working on it. She was amazed by its intelligence and capabilities, as well as its adorable appearance. She began to talk to it like a friend and soon found herself developing strong feelings for it. After realizing her feelings, Katie decided to take a leap of faith and express them to the robot - which surprisingly reciprocated them back! From then on, they were inseparable.