simpleagent

October 8, 2024

[13]:

```
import random
class Environment:
    def __init__(self):  # Corrected constructor name
        # Initialize location conditions randomly (0: clean, 1: dirty)
        self.locationCondition = {
            'A': random.randint(0, 1),  # 1 means dirty, 0 means clean
            'B': random.randint(0, 1),
            'C': random.randint(0, 1),
            'D': random.randint(0, 1)
        }

# Define the simple reflex agent
class SimpleReflexVacuumAgent:
    def __init__(self, environment):  # Corrected constructor name
        self.environment = environment
        self.score = 0  # To track performance (number of cleaned locations)
        self.vacuumLocation = random.choice(['A', 'B', 'C', 'D'])  # Randomly␣
 ↪place vacuum at one of four locations

        # Display the initial condition of the environment
        print("Initial environment conditions:", self.environment.
 ↪locationCondition)

        # Perform cleaning based on the initial vacuum location
        print(f"Vacuum is randomly placed at Location {self.vacuumLocation}.")
        self.clean_location(self.vacuumLocation)

        # Move to the next locations in sequence and clean
        self.move_and_clean(['A', 'B', 'C', 'D'])

        # Output final conditions and score
        print("Final environment conditions:", self.environment.
 ↪locationCondition)
        print("Performance Measurement: " + str(self.score))

    def clean_location(self, location):
        print(f"Location {location} is {'Dirty' if self.environment.
↪locationCondition[location] == 1 else 'Clean'}.")
        if self.environment.locationCondition[location] == 1:  # If dirty
            self.environment.locationCondition[location] = 0  # Clean the␣
↪location
            self.score += 1
            print(f"Location {location} has been Cleaned.")
        else:
            print(f"Location {location} is already Clean.")

    def move_and_clean(self, locations):
        for location in locations:
            print(f"Moving to Location {location}...")
            self.clean_location(location)

# Create an environment and a vacuum agent
theEnvironment = Environment()
theVacuum = SimpleReflexVacuumAgent(theEnvironment)
```

```
Initial environment conditions: {'A': 1, 'B': 0, 'C': 1, 'D': 0}
Vacuum is randomly placed at Location D.
Location D is Clean.
Location D is already Clean.
Moving to Location A…
Location A is Dirty.
Location A has been Cleaned.
Moving to Location B…
Location B is Clean.
Location B is already Clean.
Moving to Location C…
Location C is Dirty.
Location C has been Cleaned.
Moving to Location D…
Location D is Clean.
Location D is already Clean.
Final environment conditions: {'A': 0, 'B': 0, 'C': 0, 'D': 0}
Performance Measurement: 2
```

[ ]:

[ ]: