IIT Jodhpur

ARTIFICIAL INTELLIGENCE 2

ASSIGNMENT 1

Instructor: Debarati Bhunia Chakraborty

Submitted by: Shristy Gupta

Roll Number: M20CS015

PROBLEM STATEMENT:

PROBLEM 1

AIM: To make students understand the utility theory and bring it to practice by working on Utility functions, policies, single agent game, agent interaction with envrionment etc.

The task is to model an agent to perform **assisted goal** shootout in the game of soccer. This is a simple implementation of policies, no learning is expected by the agent.

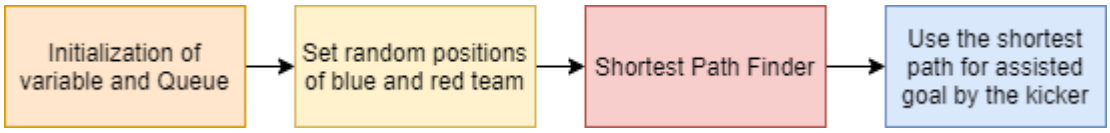
- There are 2 teams, team **RED** and team **BLUE**
- Team **RED** has 3 players and team **BLUE** has 4 players (1 Kicker and 3 in team **RED** play area).
- The players can not move from their respective places once the game starts.
- Team **BLUE** is performing an assisted goal shootout but from the Center Circle position. **(Condition 1)**

- So, one player from the **BLUE** team must remain at the center circle to take an assisted goal shoot.
- One player from each team will be staying in the Team **RED** goal box and will not leave it. **(Condition 2)**
- Apart from the center kicker from team **BLUE** players, the rest of the players will remain in the upper part as shown (**RED** Team area). **(Condition 3)**
- Shootouts must be done from the center by the team **BLUE** player as shown in the below image.
- **How to Play:**
  - The assisted goal shoot will be taken by a TEAM **BLUE** player from the center.
  - The kicker is our agent.
  - The agent needs to decide the shortest goal path, this will be your heuristic cost. (has to be assisted goal) **(Condition 4)**
  - With every run, the position of players will be changed, which has to be randomized and should satisfy the previous condition. **(Condition 5)**
- **Evaluation Components:**
  - System Design and architecture explanation **[20 Marks]**
    - You can include block diagrams, algorithms, flowchart, whichever **you** feel is necessary to understand the approach.
    - A detailed explanation of each component that is used.
  - Implementation of Environment and Agents. **[15 Marks]**
  - Conditions followed. **[6*5 = 30 Marks]**
  - Printing the cost of the steps (Top 2) for every iteration. (The ways the agent can perform assisted goal. **[10 Marks]**
  - Video of the implementation **[25 marks]**

SOLUTION 1

System Architecture:

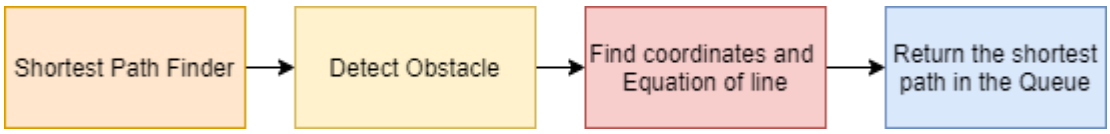
Pygame has been used to create the game



The above diagram is a simple flow diagram. Variables are initialized, and random positions of blue and red team is set as per the condition.

1. Team BLUE is performing an assisted goal shootout but from the Centre Circle position.
2. One player from each team will be staying in the Team RED goal box and will not leave it.
3. Apart from the centre kicker from team BLUE players, the rest of the players will remain in the upper part as shown (RED Team area).

The heuristic cost is calculated as per the flow diagram



The shortest Path Finder follows the below step. For more clarity refer the video

1. Initialize all the possible path. In our scenario there is ${}^5C_2 - 3$ paths. Exceptions are direct pass from source (kicker) to goal and nodes that don't have kicker
2. Find if there is any red team in the path of the blue team, if yes then eliminate that path. This is done by **detect Obstacle** function
3. **Detect Obstacle** function finds the **perpendicular distance** of the red player with the line segment that connects the blue player. If the distance is at certain range it detects it as a obstacle. The range is decided as per the pixels of red team.
4. After detecting the obstacles and eliminating the useless paths. The minimum distance nodes are chosen. Note that the **Euclidian distance is taken as the cost.**
5. These nodes are entered into a queue. E.g., If the node kicker k, one blue agent b and destination d, is having shortest path. Then the nodes entered in the queue will be k, b, d
6. Now the driver work is simple. It simply dequeues the node from the queue and finds the corresponding line segment. The ball follows the coordinates of the line segment.
7. The board shows the two optimal scores and the team and ball movement.

Functionality of functions:

```
setredrand
```

Sets the random position of red team

```
setbluerand
```

Sets the random position of blue team

```
playerBlue
```

Displays the Blue player on the screen/board

```
playerRed
```

Displays the Red player on the screen/board

```
ball
```

Defines the position of the ball at each instance. Takes the coordinates x, y i.e location at which the ball should be positioned

```
def coordinates(agent):
```

This returns the coordinates of each “agent” in the blue team

```
def sourcedest(path):
```

This function takes path as input and returns the coordinates of that segment

```
def lineequation(line):
```

This function takes the input segment and returns the equation of the line. It internally uses sourcedest function

```
def detectobstacle(line):
```

This function detects the obstacle. It checks all the red agents and calculates if they are on the shortest path line segment. If yes then return true else false

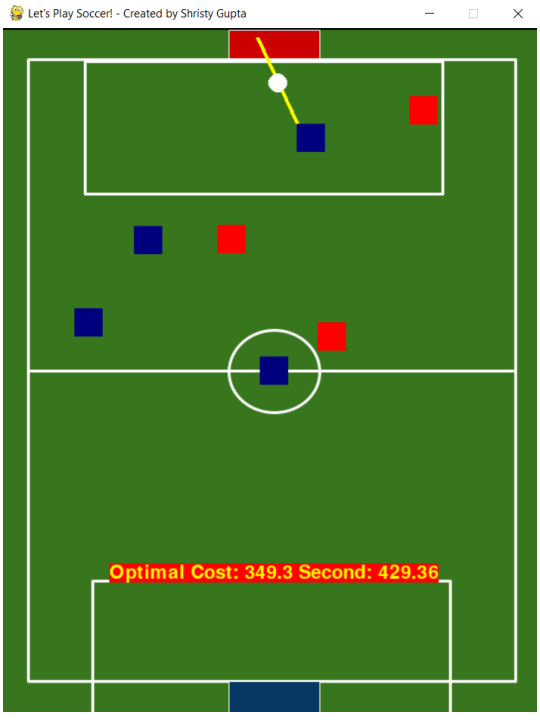
```
def pathcost(path):
```

This function just returns the Euclidian distance as a cost for the line segment

```
def shortestpath():
```

This is the heart of the logic. This calculates the shortest path and returns all the nodes corresponding to the shortest path in the queue.

The final game board looks like below



PROBLEM 2

Topic: POMDP's / Game Theory/ Utility Function

1. Paper review format: Link
2. Plagiarism Accepted: 20%
3. Papers (Select any 1)
    a. Planning and acting in partially observable stochastic domains (Can be done in group of 2)
    b. Learning Policies from Partially Observable Environment (Can be done in group of 3)
    c. Acting Optimally in Partial Observable Stochastic Domains (Must be done individually)
4. Format:
    a. **Title Page.** On the title page include the title, your name, and the date.
    b. **Abstract**. An abstract is a brief summary of **your review.**
    c. Introduction
    d. Discussion
    e. Conclusions
    f. References

SOLUTION 2

The paper chosen for review is ACTING OPTIMALLY IN PARTIAL OBSERVABLE STOCHASTIC DOMAINS

PROBLEM 3

Topic: Making Simple Decision Networks

The new movie of the Marvel Universe "Falcon and the winter soldier" is releasing soon in the nearest IMAX theaters and you being a big fan of the marvel universe, want to watch the movie at IMAX theaters only. On the same day of the movie release, you come to know that there is a concert of your favorite pop star, happening in the same city of 1 hr duration (consider timing is not known and entry will be allowed max 10 min before the start of the concert). Now, you have 2 options. Either you can buy tickets for both now, at some discounted rates or you can buy them separately after your Marvel Universe Movie so that you can avoid time clashing and wasting your money on buying the ticket. It is given that, the probability of finding the time for both the events is 0.4. A single ticket costs you around Rs. 2000/- each and combined ticket costs you around Rs. 3000/-. The Value of going to the movie is Rs. 2000/-. Which ticket you should buy? Calculate the expected value of buying a combined ticket for the probability of 0.4.

Expectations from the question:

- **Decision Network** for purchasing the tickets (Block Diagram) P.S. Please check how the blocks are modeled for building a decision network [10 Marks]
- **Decision Tree** for buying tickets [5 Marks]
- Calculate the Expected value of buying tickets for 3 different probabilities and explain the outcome. (Consider different Scenarios) [15 Marks]

SOLUTION 3

Given:

Probability of finding the time: 0.4
Single Ticket Cost: 2000 Rs
Combined Ticket Cost: 3000 Rs
Value of going to a movie: 2000 Rs

Let A be the case of buying a movie ticket, A' be the case of not buying a movie ticket

Let B be the case of buying a concert ticket, B' be the case of not buying a concert ticket

I. Case A and B combined
Cost = 3000 Rs
Value = 2000 + 2000 Rs = 4000 Rs
Difference = 1000 Rs
Probability = 0.4 * 0.4 = 0.16

II. Case A and B independently
Cost = 2000 + 2000 Rs = 4000 Rs
Value = 2000 + 2000 Rs = 4000 Rs
Difference = 0 Rs
Probability = 0.4 * 0.4 = 0.16

III. Case A and B' combined
Cost = 3000 Rs
Value = 2000 Rs
Difference = -1000 Rs
Probability = 0.4 * 0.6 = 0.24
(This case basically applied when we want to buy both and go to only one event)

IV. Case A and B' independently
Cost = 2000 Rs
Value = 2000 Rs
Difference = 0 Rs
Probability = 0.4 * 0.6 = 0.24

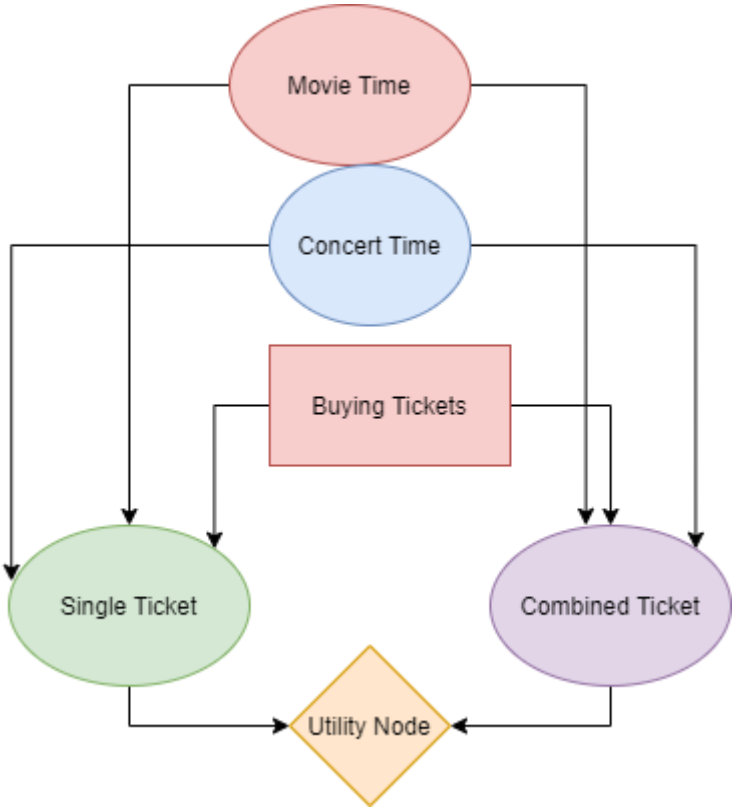
V. Case A' and B combined
Cost = 3000 Rs
Value = 2000 Rs
Difference = -1000 Rs
Probability = 0.4 * 0.6 = 0.24

VI. Case A' and B independently
Cost = 2000 Rs
Value = 2000 Rs
Difference = -1000 Rs
Probability = 0.4 * 0.6 = 0.24

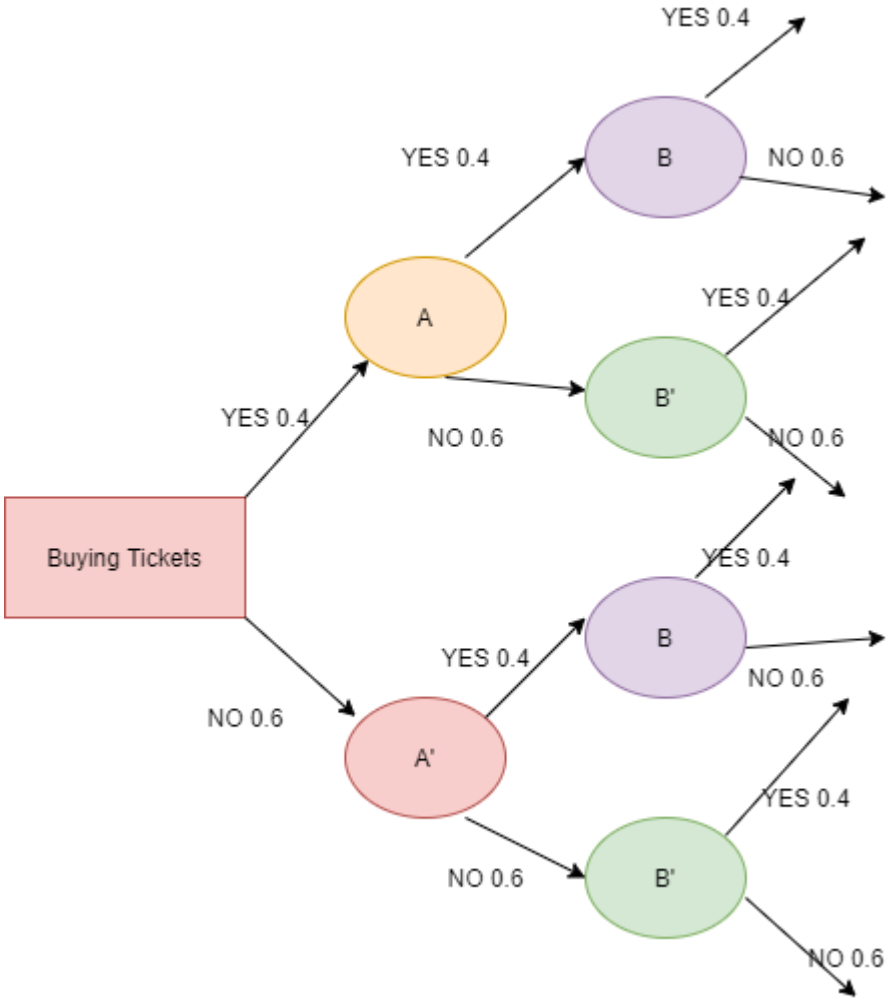
EVM of buying the tickets in the combination of the cases above

(0.16 * 1000) + (0.24 * -1000) + (0.24 * -1000) + (0.36 * -3000) = -1400

Decision Network



Decision Tree To buy the Tickets


YES 0.4
B
NO 0.6
YES 0.4
A
YES 0.4
B'
NO 0.6
NO 0.6
Buying Tickets
YES 0.4
B
YES 0.4
NO 0.6
NO 0.6
A'
YES 0.4
NO 0.6
B'
NO 0.6