Assignment Report

<u>Let's Play Soccer</u>

System Design And Architectural Details:

- **Component Description:**

    - **Blue Team Player:** A blue team player is represented by a circle of radius 10px as shown in below figure. There are 4 blue team players. One blue player (called blue center) stands in the center of the field. One player stands randomly inside the red team's box and other three blue players are randomly located in the upper side of the field i.e red team's area.

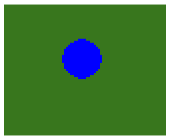


Fig. Blue Agent

    - **Red Team Player:** A red team player is of the same size and shape as blue players i.e circle of radius 10px. One red player is randomly assigned a position inside the box and other red players are randomly located in the same upper half of the field.

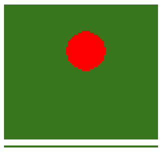


Fig. Red Agent

    - **Football:** The football is represented as a circle of size 3px. The football in the possession of the blue player is shown in the below figure.

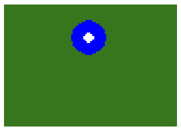


Fig. Football in possession of the blue player

- One Example of the field with initial random position of agents according to the given conditions is shown below:

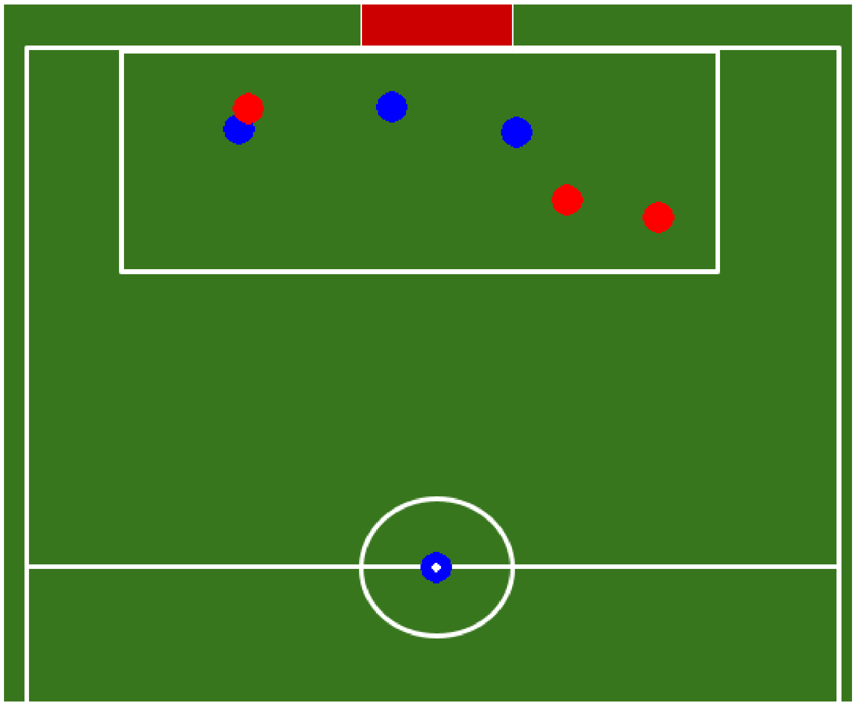


Fig. Field

Since, the location of the agents are randomly selected, by chance all the agents get placed inside the box.

- **Goal Post:** The goal post object stores the coordinates of the goal's left post and right post.

- **Note that:** here the field, the ball , the goal and the players are implemented as an environment and agent system, extensively using OOPs.

- All the objects contain getter and setter methods to update and excess their internal parameters for e.g. location.

- **Flow Chart:**

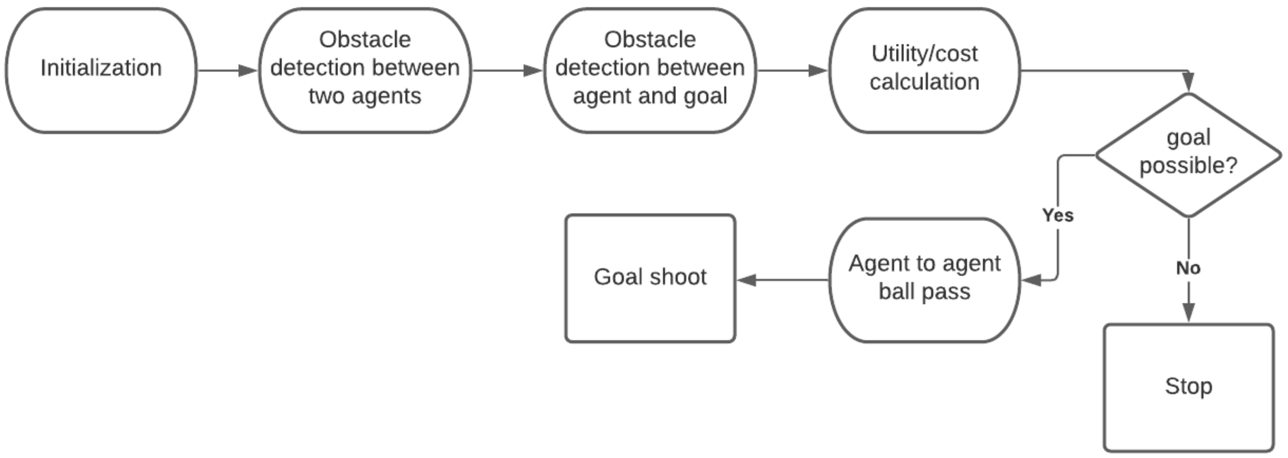


Fig. Flow chart

Assumption: The utility is the negative of the cost therefore, higher utility and lower cost is desirable.

- **Procedure:**

The steps involved in the above flow chart are as follows:

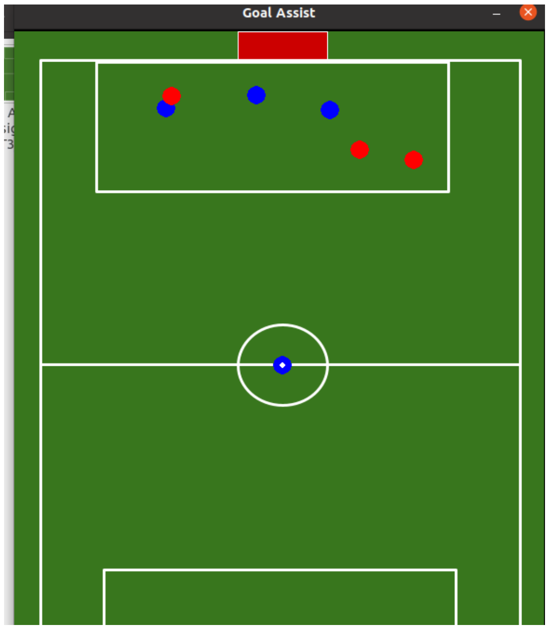
1. Initialization: The positions of the red and blue player are assigned randomly according to the conditions given. The goalpost object is initialized using its fixed coordinates obtained from the background image and the ball's initial position is assigned to the center of the field in the possession of the center blue player.
2. Obstacle detection between two agents: To get an assisted goal, the center blue player passes the ball to another blue player according to the utility or cost. If a red agent is present in between the center blue player and another blue player then the ball can't be passed on to that particular player therefore, the presence of red player between the agents is detected and if found the cost of that agent is made too high so as to not pass the ball to that agent.

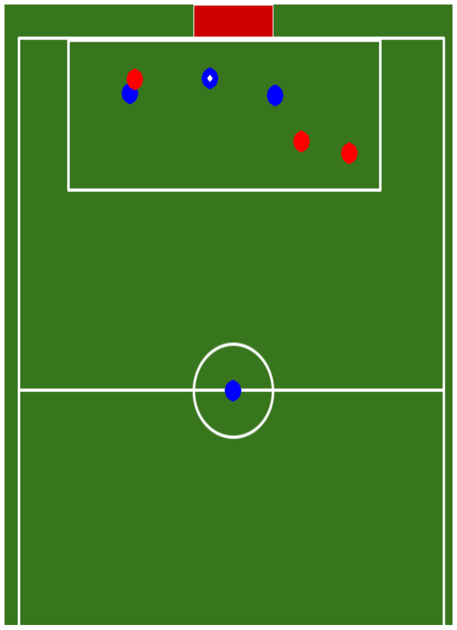
3. Obstacle detection between agent and goal: To make the final goal, the similar process as above is repeated to find the obstacles in between the agent and the goal.
4. Utility/cost calculation: The total cost is calculated as the sum of the cost between the blue player and first blue player to receive the ball and the cost from that player to the goal. Also the cost depends on the obstacle present in the path. To calculate the cost from agent to goal, an iterative approach is used to obtain the minimum distance between the goal and agent without obstacle since, the goal is line not a point (which is the case between two agents)
5. Depending on the cost/utility calculated in the previous step if the goal is possible then the ball is passed to the agent having minimum cost and if the goal is not possible then a message box pops out stating to try again..
6. Finally, the intermediate blue player shoots the ball on the goal through the minimum cost path.

Note that: The ball passing is done on mouse click event (can be done anywhere on the screen).

● Observation:

The results obtained for a single run are attached below and the results for the multiple runs are presented in the video implementation.





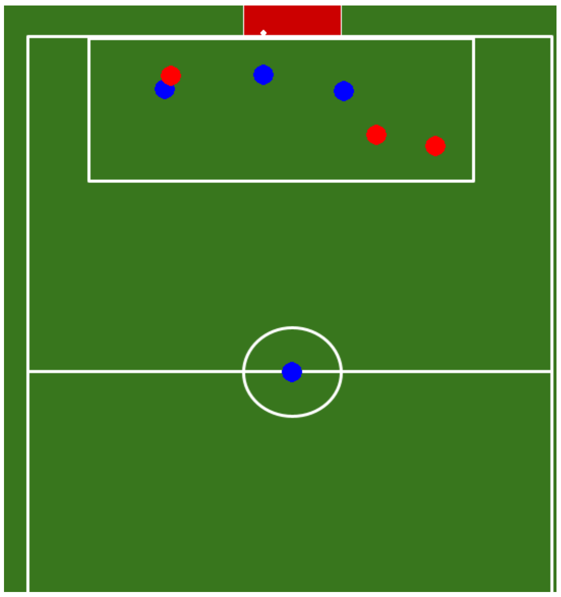
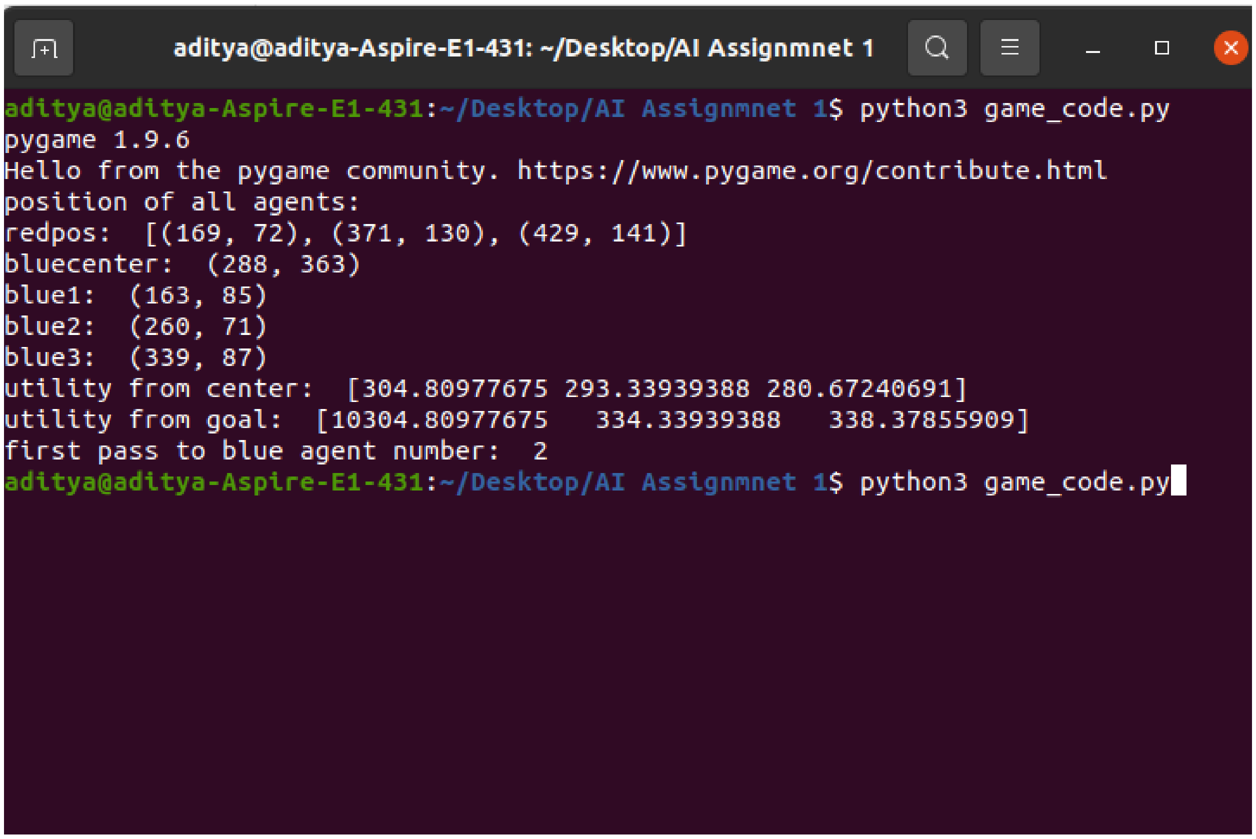


Fig.1 (Top Left Corner) Initial position, ball with center blue player **Fig.2** (Top Right Corner) First Pass to the minimum cost taking agent **Fig.3** (Bottom left) final shoot in the goal via minimum distance path



```
aditya@aditya-Aspire-E1-431:~/Desktop/AI Assignmnet 1$ python3 game_code.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
position of all agents:
redpos:  [(169, 72), (371, 130), (429, 141)]
bluecenter:  (288, 363)
blue1:  (163, 85)
blue2:  (260, 71)
blue3:  (339, 87)
utility from center:  [304.80977675 293.33939388 280.67240691]
utility from goal:  [10304.80977675   334.33939388   338.37855909]
first pass to blue agent number:  2
aditya@aditya-Aspire-E1-431:~/Desktop/AI Assignmnet 1$ python3 game_code.py
```

Fig. Result

- **Conclusion:**

As shown in the figure 1,2 and 3, the best agent is selected according to the situation and given the ball in the first pass and then the receiving agent shoots the ball into the goal.
The results printed in the Fig. Result contains the negative of utility i.e cost, the total cost of the 2nd blue player is minimum therefore got selected. Also the value in “utility from goal” list is very high which is corresponding to the blue player completely blocked by a red player in between the goal (See Fig.1).