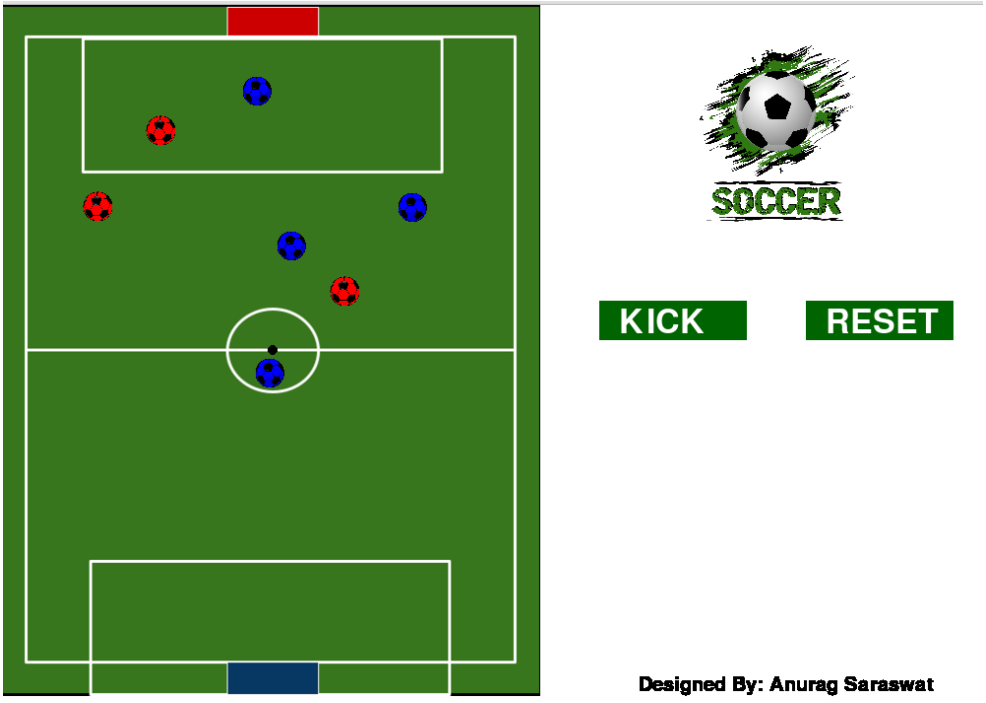
AI-2 Assignment 1 Report

- Submitted By : Anurag Saraswat (M20CS066)

Question 1: Programming: Let's Play Soccer

The GUI and functionality of the game is designed using PyGame library. Game involve arrangement of players following constraints/conditions mentioned in the question. Player in the center circle performs assisted goal shootout.



- **GUI Design**
  The game has a simple GUI divided into two parts. Left part consists of the game environment and in the right part two buttons are present: KICK and RESET. The KICK button is used to perform assisted goal from the center position. RESET is used to RESET the environment and on every click it randomly places players at positions following the condition mentioned in the question.

- **Environment and Agent**
  Image provided in the question is used to represent the game environment. Players act as agents. Initially, the player in the center circle is an active agent. It then passes the ball to the nearest player not surrounded by red players. Now this agent becomes an active agent. For implementation purpose, coordinate of goal is added to players' coordinate array after first pass(to avoid direct goal). If all players are surrounded by red players then the program performs direct goal and printed goal not possible on terminal.

- **Cost Function**
  Euclidean Distance between two players is used as cost function. Agent chooses a player which is at least distance from the current player. Searching for the least distant player is performed in space decreasing in vertical direction. Since the environment is designed in such a way that all agents try to pass the ball forward. Agents also take care of red players. If a red player lies on the line of sight between two blue players then passing to that player is rejected. To find a path is clear or not collinearity between three points is used. If point A, B, and C lies on same line then,

  $$distance(AB) + distance(BC) - distance(AC) = 0$$

  To make it more flexible error is introduced in the above equation. Now ,

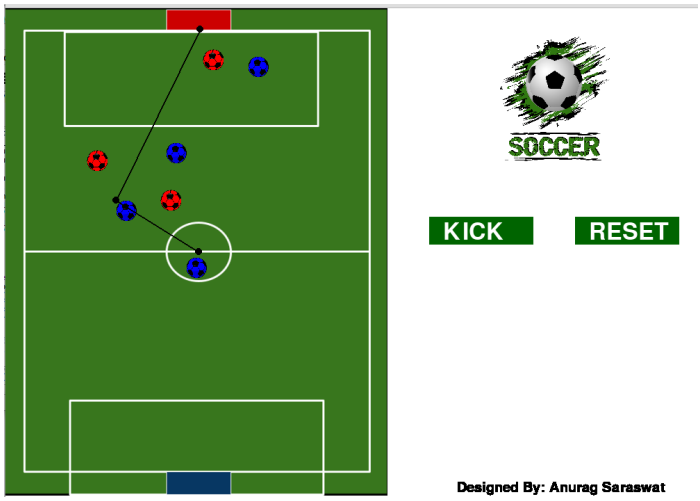
  $$distance(AB) + distance(BC) - distance(AC) = delta$$

  Here , delta is some small value calculated by hit and trial.

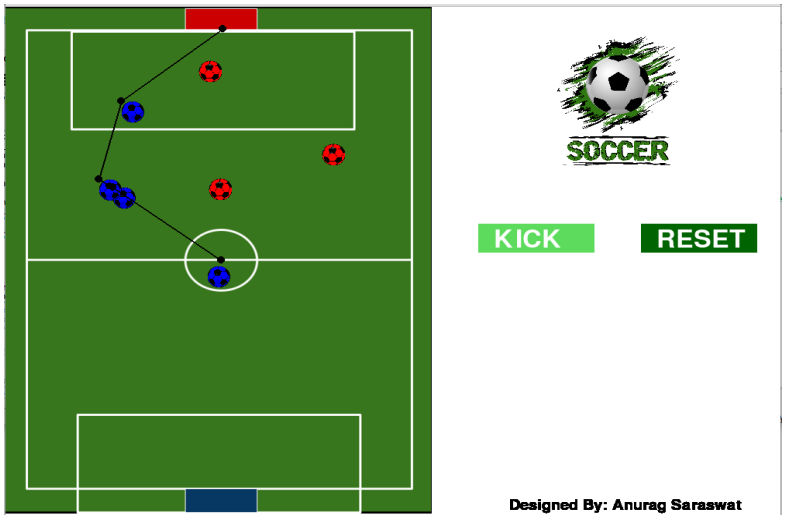
- **Algorithm**
  Algorithm for finding next best agent is summarized as follow:
  1. Initialise list containing coordinates of blue player and red player. And the boolean flag goal_reached to False.
  2. For the first turn the central player passses ball to the nearest non colliding player. After the first turn, the coordinate of the goal post is added to the blue player list. So intuitively after the first pass we are finding if the goal is near or another player is near. Cost of reaching all the players is added to the cost list.
  3. After finding optimal cost to all players, cost for the top two players is stored. If only strategy is possible then top second cost is printed as not possible.
  4. Repeat step 2 and 3 till the goal is not reached.

- **Example of different scenarios covered**

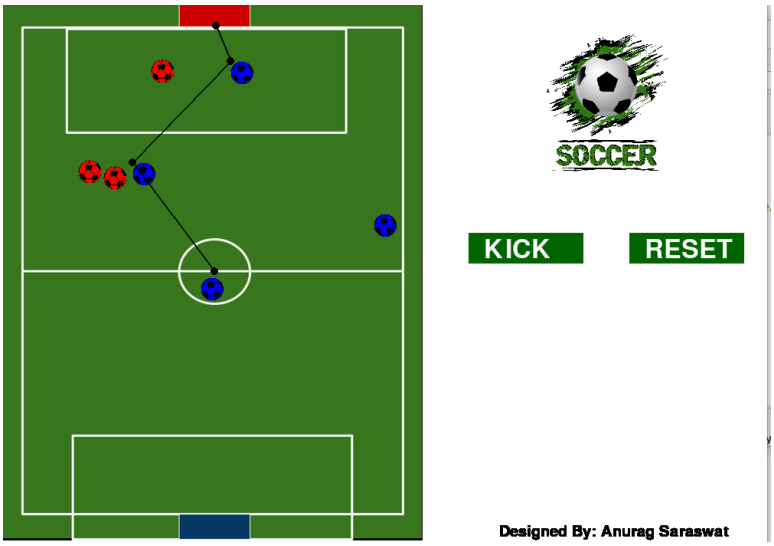
- Single Pass Goal



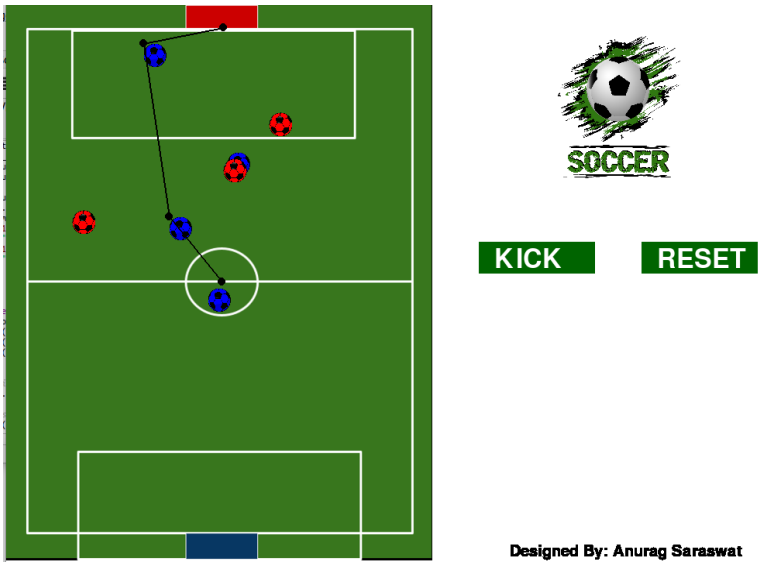
- Goal with triple pass



- Goal with double pass



- Another example of double pass goal



Q-3 Making Simple Decision Networks

Given:

Cost of single ticket = Rs 2000

Cost of combined ticket = Rs 3000

Value of going to movie = Rs 2000

Value of going to concert = Rs 2000

Probability of finding the time for both event = 0.4

$P(t_1)$ = P (time of movie known) = 0.4

$P(t_2)$ = P (time of concert is known) = 0.4

Option	$P(t_1).P(t_2)$ (P = 0.16)	$P(t_1).\sim P(t_2)$ (P = 0.24)	$\sim P(t_1).P(t_2)$ (P = 0.24)	$\sim P(t_1).\sim P(t_2)$ (P = 0.36)
Combined	cost = 3000 value = 4000 Total = 1000	cost = 3000 value = 2000 Total = -1000	cost = 3000 value = 2000 Total = -1000	cost = 3000 value = 0 Total = -3000
Single	cost = 4000 value = 4000 Total = 0	cost = 2000 value = 2000 Total = 0	cost = 2000 value = 2000 Total = 0	cost = 0 value = 0 Total = 0

Expected value of buying a combined ticket for probability of 0.4

$$= Rs\{0.4 \times 0.4 (4000-3000) + 0.4 \times 0.6 (2000-3000) + 0.6 \times 0.4 (2000-3000) + 0.6 \times 0.6 (0-3000)\}$$

$$= Rs\{0.16 \times 100 + 0.24 \times (-10) + 0.24 \times (-10) + 0.36 \times (-30)\} \times 100$$

$$= Rs(160 - 240 - 240 - 1080)$$

$$= Rs(-1400)$$

Expected value of buying a single ticket for probability of 0.4.

$$= Rs((0.16 \times 0) + (0.24 \times 0) + (0.24 \times 0) + (0.36 \times 0))$$

$$= 0$$

E.V (single ticket) > E.V (combined ticket)

∴ It is more beneficial to buy single Ticket.

Considering probability of finding time be 0.5

then,

E.V (single ticket) = 0

E.V (combined ticket) = P(0.25 x (1000) - 0.25x(1000) - 0.25 x (1000) - 0.25(3000))

= Rs (250 - 250 - 250 - 750)

= Rs -1000

Here, E.V (single ticket) > E.V (combined ticket)

∴ at probability 0.5, it is more beneficial to buy single ticket

Considering probability of finding time be 0.6

then, E.V (single ticket) = 0

E.V (combined ticket) = Rs (0.36 x1000) -(0.24 x1000) -(0.24 x 1000) - 0.16 x3000

= Rs (360 - 240 - 240 - 480)

= Rs (- 600)

Here, E.V (single ticket) > E.V (combined ticket)

∴ at probability 0.6 it is more beneficial to buy single ticket

Considering probability of finding time be 0.9

then,- E.V (single ticket) = 0

E.V (combined ticket) = Rs ((0.81 x 1000) -(0.09 x 1000) - (0.09x1000) -(0.01 x 3000))

= Rs (810 - 90 - 90 - 30)

= Rs (600)

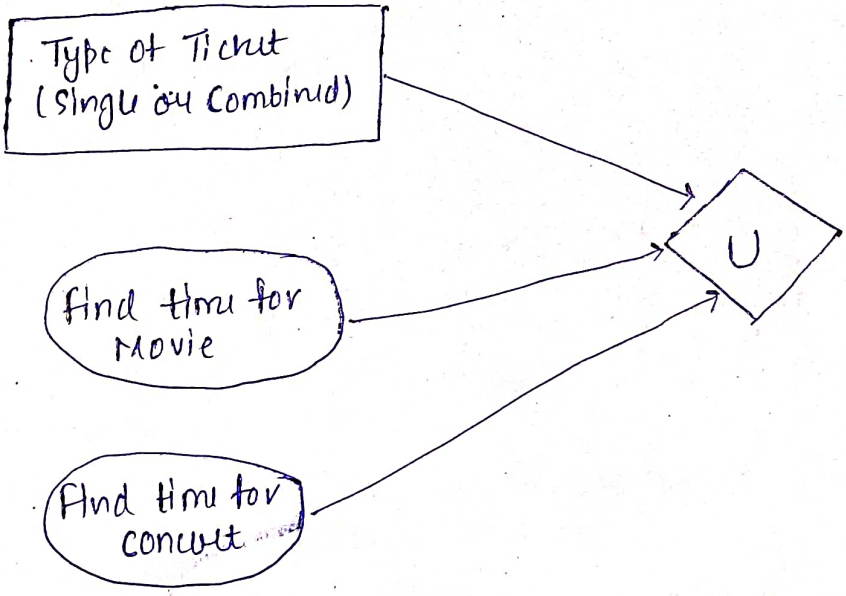
Here, E.V (single ticket) < E.V (combined ticket)

∴ at probability 0.9 it is more beneficial to buy combined ticket.

Observation:

As probability of finding the time of both event increases. Expected Value of buying Combined Ticket Increases.

Decision Network for Buying Tickets.



⇒ Decision Node

⇒ Chance Node

⇒ Utility Node

Decision Tree for buying tickets