ICsci 164 AI Project

Simplified Soccer

Jairo Reyes

­­

5/2/2016

**Domain**

* Problem: Design and implement a simplified version of soccer and determine what effect changing some of the parameters have on the outcome of the game. More specifically, after implementing the game,

determine what effect changing the structure of a team -- for example, changing from 4 midfielders and 3 defenders to 3 midfielders and 4 defenders -- has on the outcome of the game.

**Problem Representation**

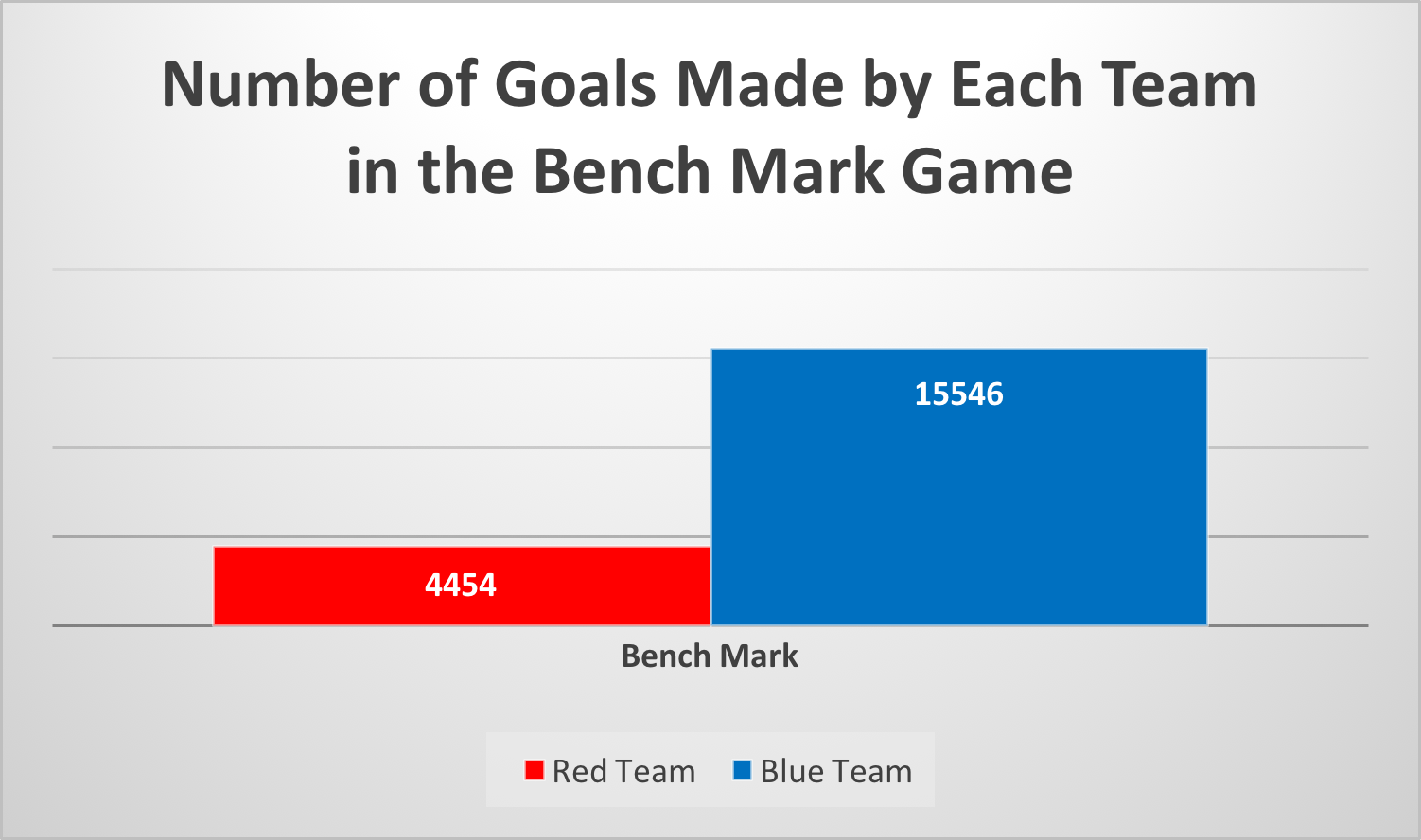
* Performance measure: Using my knowledge of soccer, the game should look like and act something like soccer, the game should be stable without crashes. An accurate measure of the average percentage of goals each team makes in the initial model should be determined.
* Environment: The soccer field with players and ball
* Actuators: Wait, Pass, Dribble, Chase, Shoot
* Sensors: Current state of the game
* I implemented the model using python along with Tkinter for the graphics. At each step of the game each player is assigned a state of Wait, Pass, Dribble, Shoot, Chase, or Home and then is moved accordingly.

**Solution**

* The game is a simulation of a simplified version of soccer. At each step of the game, an evaluation function is used to assign a state to each player. Each player is then moved according to the state that has been assigned. The input to the evaluation function is the player who has ownership of the ball. The evaluation function determines if there is a player that can receive a pass before any player of the opposite team reaches that player. (Of course, the ball might be intercepted along the way.) If such a player exists, the ball is PASSED to such player closest to the goal. If no such player exists, the ball is DRIBBLED or SHOT depending on the distance from the goal. All players on the opposite team CHASE the ball, if it is within their assigned territory. Several aspects of the game are randomized (the position of the goalies, who owns the ball when a cycle occurs, the position of the defenders).

**Results**

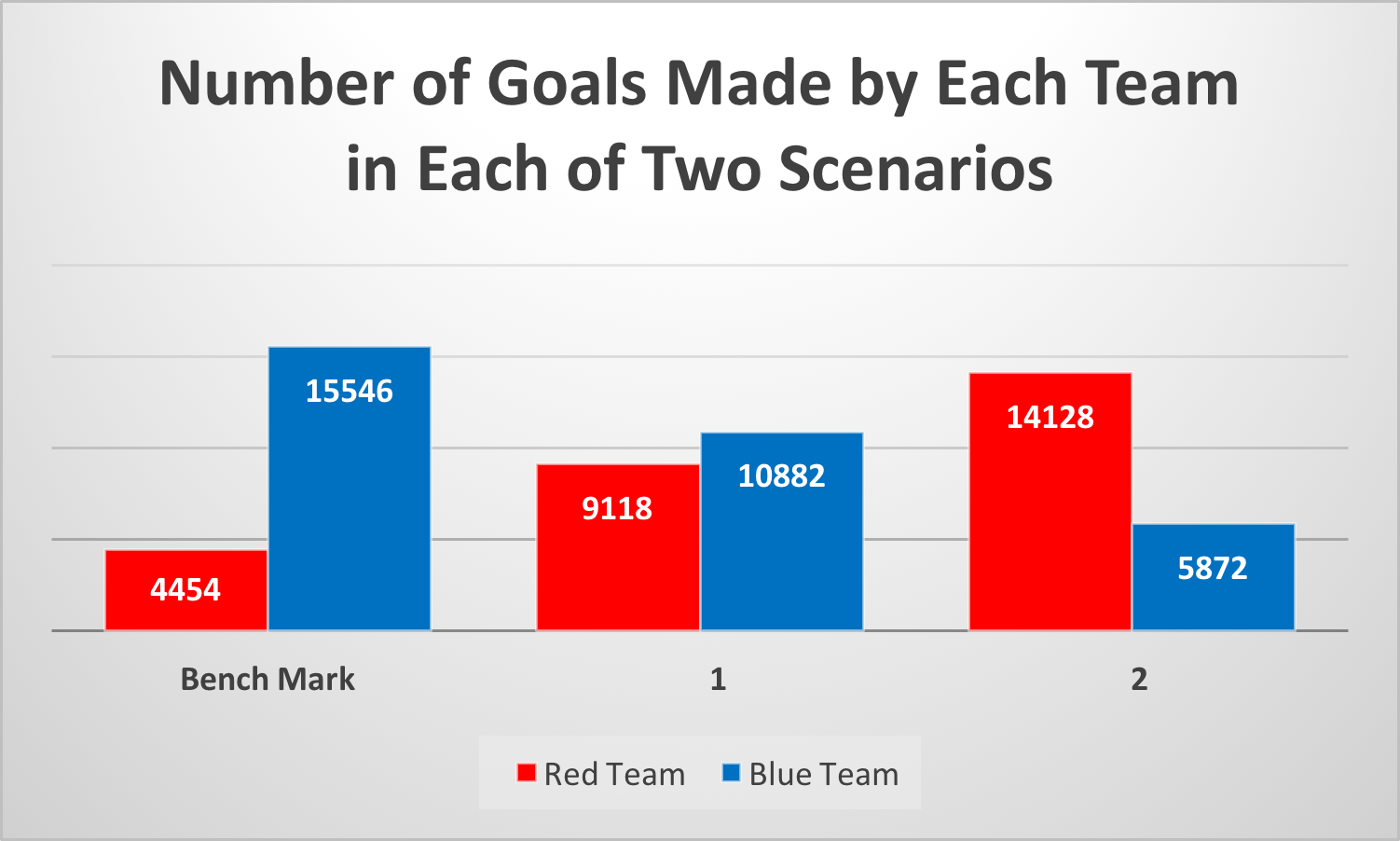
* The Bench Mark Game
  + File: blue.py
  + Red Team: 3 Defenders, 4 Midfielders, 3 Forwards, 1 Goalie
  + Blue Team: 3 Defenders, 4 Midfielders, 3 Forwards, 1 Goalie
  + Speeds of all players on both teams equal.
* The Bench Mark Game was run (without graphics) until 20,000 goals were made with the following results.



* Using these results as bench mark, changes were made and results were recorded.
* For each of the 2 scenarios below, the Bench Mark Game was altered and played until 20,000 goals were made.

1. The speed of player 11 (a Forward on Red Team) was increased by a factor of 1.1.

2. The speed of player 11 (on Red Team) was doubled.

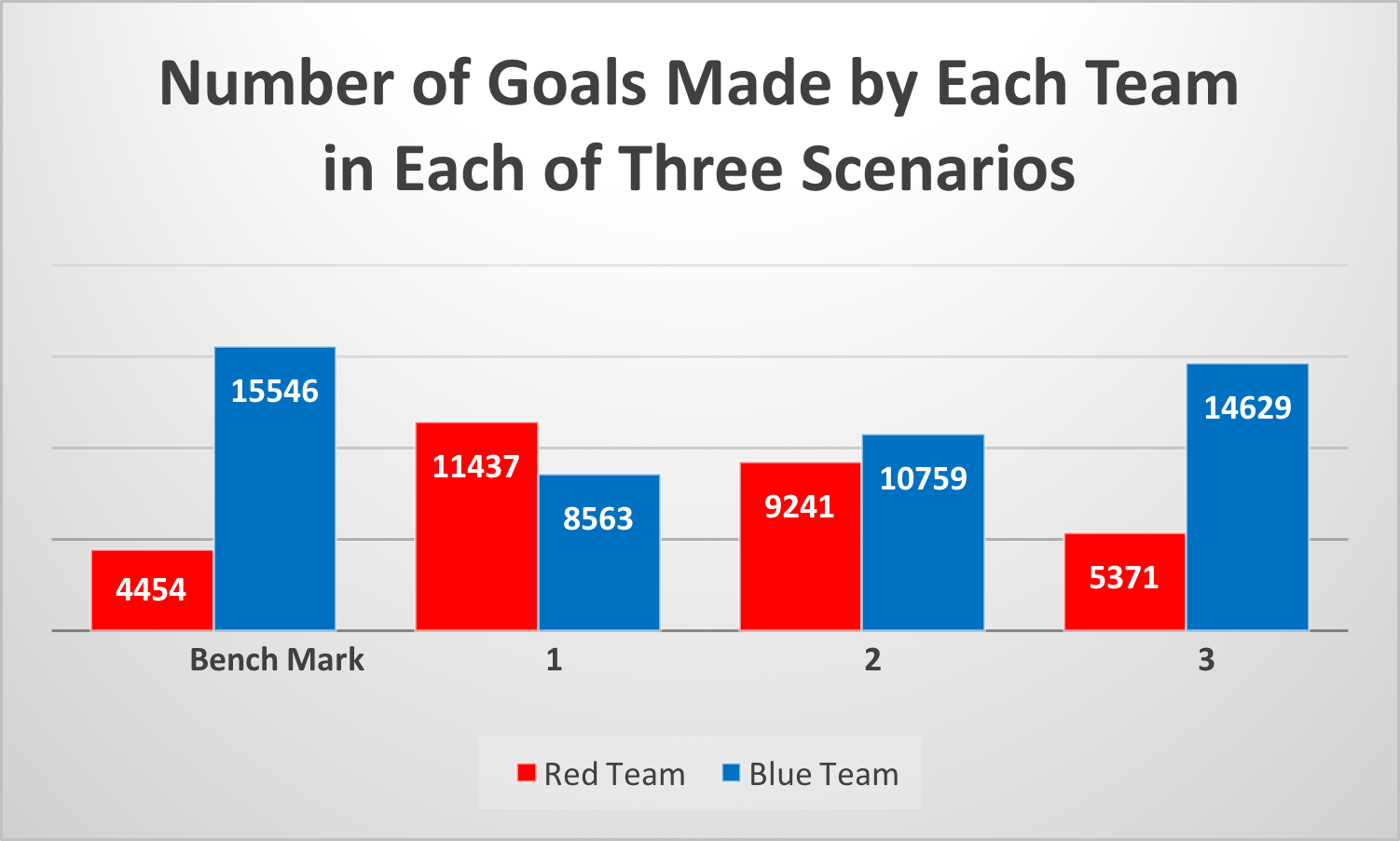


* Conclusion: In this model, changing the speed of a single player can significantly effect the outcome of the game.
* Changing the Team Structure
  + File: gamet.py
  + Red Team: 4 Defenders, 3 Midfielders, 3 Forwards, 1 Goalie.
  + Blue Team: 3 Defenders, 4 Midfielders, 3 Forwards, 1 Goalie.
  + This game referred to as CS (Changed Structure).
* For each of the 3 scenarios below, the game was played until 20,000 goals were made.

1. CS, players have equal speeds.

2. CS, the speed of player 11 (a Forward on Blue Team) was increased by a factor of 1.1.

3. CS, the speed of player 11 (on Blue Team) was doubled



* Conclusion: In this model, changing the structure of the team or changing the speed of a single player can significantly affect the outcome of the game.

**Game Details**

* Graphics
  + Used Tkinter(same package used by PacMan)
  + Designed field to be scaled version of a legal soccer field according to FIFA (100-110m by 64-73m). I used 1000px by 640px
  + Player and ball icons designed.

G:\Presentation\Original_Blue_Start.tiff

* Classes
  + Ball
    - Attributes
      * x and y coordinates of the current position
      * x and y coordinates of the starting position
  + Players
    - Attributes
      * x and y coordinates of the current position
      * x and y coordinates of the starting position
      * x and y coordinates of the home position
      * Team (A-blue or B-red)
      * Role (Goalie, Defender, Midfielder, Forward)
      * Role (Goalie, Defender, Midfielder, Forward)
      * Speed (initialized to 1.0)
      * State (WAIT, PASS, DRIBBLE, CHASE, SHOOT, HOME)
  + Score
    - Attributes
      * Red
      * Blue
* Players
  + Eleven players for each team.
    - 1 Goalie
    - 3 Defenders
    - 4 Midfielders
    - 3 Forwards
  + Original locations symmetric except for the starting player.
* Ball Ownership
  + The closest player to the ball owns the ball provided the player is within a minimum distance of the ball.
  + Ties are broken randomly. Once a player owns the ball, the player remains in ownership until another player gains ownership. For example, in passing the ball, the ownership of the ball remains with the passing player until it is either intercepted or it reaches the intended player.
* Evaluation Functions
  + There are two evaluation functions, one for the A team and one for the B team.
  + The evaluation function for the player having ownership of the ball is called.
  + The evaluation function determines if there are any players that the player with the ball can pass to before any player of the opposite team can reach that player. The function returns a list (possibly empty) ordered by players being closest to the goal.
  + The evaluation function for the player owning the ball determines the states of all of the players of both teams.
    - If the player owning the ball is less than 280px from the player’s goal, the player’s state is set to SHOOT. If he is between 280px and 350px from the goal or there is no player (according to the evaluation function) to pass to, the player’s state is set to DRIBBLE. Otherwise his state is set to PASS.
    - The state of each of the other players on the same team as the player owning the ball is set to HOME.
    - The state of each player on the opposite team is set to CHASE if that player and the ball are in the player’s assigned territory, otherwise is set to HOME.
      * The field is divided into thirds (length wise). Each player is restricted to stay in the one third of the field where the player’s home position is, unless the player has possession of the ball.
  + If a player DRIBBLES, his direction is toward the average of the y-coordinates of the two closest players of the opposite team that are in front of him. If there is only one player in front of him he misses that player, and if there are no players in front of him, he heads for the goal.
  + The game continues until a goal is scored at which time the players and the ball are reset to their Initial positions except that the defenders on both teams are randomly moved slightly.
* Randomization
  + Each goalie moves back and forth randomly between the goal posts.
  + As mentioned above, upon reset, the defender’s position are moved slightly at random.
  + Ties for ball ownership are broken randomly
  + Infinite cycles are broken randomly.
  + When a Player shoots the ball towards the goal, a random spot between the goal posts is chosen.

**Conclusion**

* By simulating the soccer game, I was able to see what affect changing parameters, such as the speed of players and the formation structure of the team, had on the outcome of the game.