Christopher Morcom
CS 487

Homework 4

Information Modeling: You are designing World Cup penalty kick simulator

- **Briefly describe the penalty kick scenario: focusing on 5 objects – kicker, goalie, ball, net, and scoreboard**

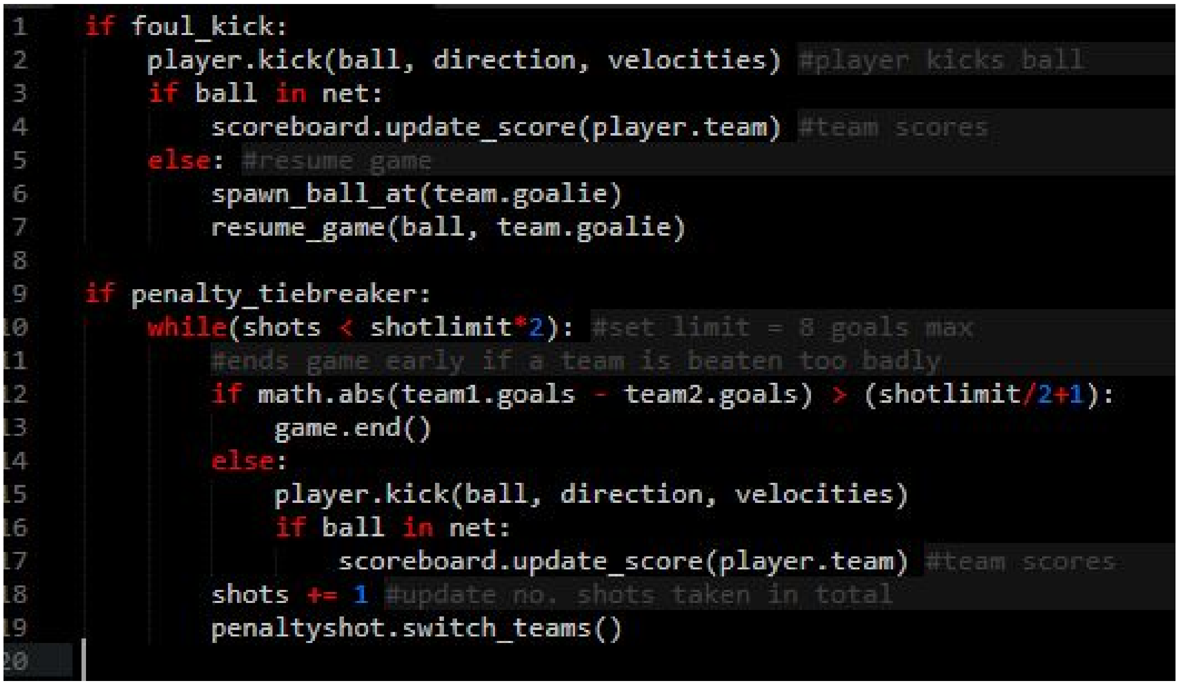
  Penalty kick occurs in a penalty tiebreaker game scenario or foul inside goal box case from current side's team. Kicker interacts with ball with accuracy and kick velocity parameters. Second player controls goalie and can move goalie in xz-plane at speeds corresponding to player attributes. Ball can interact with all objects. If the ball touches the net or crosses into the net ares, a goal is scored and a signal is sent to the scoreboard. In a penalty tiebreaker game scenario, the team with a winning number of goals in a set limit will win. In a foul scenario, the game may continue or end based on the scoreboard time/score attributes.

- **Describe each object by listing its attributes (including data type) and methods (including parameters and return values)**

- Kicker
  - Overrides a player abstract class (contains team and basic player attributes)
  - Interacts with other players, the ball, and referee
  - Has a series of stats that correspond to virtual player "skill"
  - Signals referee if game rules are broken
  - Has x,y, and z movement parameters as well as kick/control parameters
- Goalie
  - Overrides a player abstract class (contains team and basic player attributes)
  - Interacts with other players, the ball, and referee
  - Has a series of stats that correspond to virtual player "skill"
  - Able to use hands in bounding area
  - X & Z axis controls set during penalty kick scenario
  - Signals referee if foul occurs
- Ball
  - Interacts with any objects in the bounding field area
  - Has z,y, and z axis velocities and rotational velocity that interacts with a static air friction parameter
  - 3D hitbox
- Net
  - Bounding area that interacts with all objects
  - Goal posts are immovable

    - Signals scoreboard when ball enters bounding area
- Scoreboard
    - Updates the score when signal received from net object
    - Keeps track of game time
    - Signals referee during set time/score intervals

- **Use pseudo-code to depict the scenario – be sure your code supports the interactions between your objects**



```
1   if foul_kick:
2       player.kick(ball, direction, velocities) #player kicks ball
3       if ball in net:
4           scoreboard.update_score(player.team) #team scores
5       else: #resume game
6           spawn_ball_at(team.goalie)
7           resume_game(ball, team.goalie)
8
9   if penalty_tiebreaker:
10      while(shots < shotlimit*2): #set limit = 8 goals max
11          #ends game early if a team is beaten too badly
12          if math.abs(team1.goals - team2.goals) > (shotlimit/2+1):
13              game.end()
14          else:
15              player.kick(ball, direction, velocities)
16              if ball in net:
17                  scoreboard.update_score(player.team) #team scores
18          shots += 1 #update no. shots taken in total
19          penaltyshot.switch_teams()
20
```

Note that the ball would have its own physics to detect if it collides with anything in its path. For instance, a goalpost or a goalie could stop the ball so it would be meaningless to check for specific collision with certain objects. It would be easier to check if the ball hitbox intersects with anything during play.

For instance: the player.kick() method sets parameters for the player's interaction with the hitbox and passes said parameters to the ball on contact.

Shotlimit is a parameter that allows for any number of rounds for each team. If a team scores N/2+1 shots greater than the opposing team, it is impossible for the opposing team to win.

- **Enhance the code to simulate a championship tie-breaker scenario (2 teams, 5+ rounds)**

(see line 9 and below for scenario)