Project Proposal – Junho Park

**Project Description**

* The name of my project is the NBA Simulator. The user will be able to select an NBA team, with real NBA players and statistics, and play a simulated game against a random NBA team, controlled by the AI. The user will be playing the role as a coach, substituting players and changing the tendencies of how the team plays (Ex.) shoot more three pointers, pass the ball more, etc.). The objective of the game is to win against the AI. Good luck!

**Competitive Analysis**

* A similar project that I’ve found is the [15112- Soccer Manager](https://www.youtube.com/watch?v=NQSU42N-9k4&ab_channel=BoXia) by Bo Xia, which was a past 15-112 term project. The project is similar to mine in that it is a game about simulating a sports game. Like the Soccer Manager, my project has several screens, such as the roster screen, and the actual game screen. Both projects use AI to move the players in different directions to simulate a real sports game. The differences between this project and mine is that my project implements real player data from APIs. Each player has their own rating and skill set based on their real-life statistics in the NBA. The 15-112 Soccer Manager, on the other hand, uses randomly generated players with randomly generated ratings. Another difference is that my project is a complete simulation, with the user only able to play the role as a coach. The 15-112 Soccer Manager, however, is a simulated game, where the user controls a player in the game, playing the role as the actual soccer player.
* Another similar project that I found online is a NBA game simulation, which can be found [here](https://www.youtube.com/watch?v=PDuKpPJ4nmU&t=127s&ab_channel=DunderData). This project is a simulation of real NBA games, animating each play that has happened. Although both projects simulate NBA games on a court, the main difference is that this project uses real game data to animate the games. Using camera data from an API, the project maps out each player’s movement from a real NBA game to visually represent what happened. My project, however, is a simulation of a NBA game based on real players, not a simulation of a NBA game that actually took place. My game implements AI, while the NBA animation project does not.

**Structural Plan**

* My project will be structured into a main python file, with many functions. The file will be divided into three sections, one for the model, one for the controller, and one for the viewer.
  + The model will also be divided into multiple sections, with each section containing ui parameters for different parts of my game. For example, one section will contain the parameters needed for the game screen, while another section will contain parameters needed for the roster screen.
  + The controller will also be divided into the usual three controllers (keyPressed, mousePressed, timerFired). The mousePressed function will be divided into multiple sections, with each section designated for a specific screen. For example, there will be different code for the mousePressed function for the home screen, and there will be different code for the mousePressed function for the game screen. Also, for the timerFired function, I will have 2 helper functions (doStep functions) that will be run based on who has possession of the ball. For example, the doStep1 function will run if the user’s team has possession of the ball, while the doStep2 function will run if the AI(opponent)’s team has possession of the ball.
  + The redrawAll function for the viewer also be contain multiple redrawAll functions. Each redrawAll function will be for a specific screen, and each screen redrawAll function will have multiple helper functions for organizational purposes. For example, the redrawAll function will contain the gameScreen\_redrawAll function, which will contain helper functions such as drawThreePointLine or drawScoreBoard.

**Algorithmic Plan**

* The trickiest, most complex part of my project is simulating a game, with AI. Because I want the players to move by themselves and the players to score the games based on the user selection of tendencies and players, I need to create a way to move the players and make them score based on their ratings.
* To approach this part of my project, I decided to run separate functions within the timerFired; 1 doStep function for when the user has possession, and 1 doStep function for when the opponent has possession. Each function will move the players to designated spots. Then, I will set a state in the model that keeps track of which player has possession of the ball. Based on the player’s distance from the basket and the tendency set by the user, the player with the ball will perform an action (pass, shoot, shoot three). Once the action is decided, the percentage of completion of that action will be calculated based on the player’s rating. If complete, the game will be updated respectively; if a pass was completed, then the player receiving the pass will go through the same decision-completion cycle. If a shot is completed, the score will be updated, the possession will change, and the other doStep function will be called, calling the same process, but in a different direction for a different team.
  + Ex.) the user has possession of the ball -> the players move to respective positions -> player with ball makes decision (pass, shoot, shoot three) based on tendency and distance to basket -> player executes action -> completion of action is calculated
    - -> if action completed -> update game state/score
    - -> if action not completed -> change possession -> call other doStep function

**Timeline Plan**

* TP0 (4/20)
  + By TP0, I plan to
    - finish all graphics for all screens
      * Home screen, roster screen, team select screen, game screen, substitution screen, tendency screen, game over screen
    - Implement API for real life data and images
      * Roster screen -> player images, player ratings, team names, etc.
* TP1 (4/26)
  + By TP1, I plan to
    - complete a working simulation of a game
      * player movements, player substitutions, changing player tendencies, scoring games
* TP2 (5/1)
  + By TP2, I plan to
    - Implement a smarter AI for the simulation games
    - Add the ‘stamina’ feature, which makes players move slower with time
    - Use matplotlib to graph player ratings
    - Add more tendencies for more dynamic gameplay
* TP3 (5/5)
  + By TP3, I plan to
    - Add additional features
      * One on one game mode
    - Perfecting gameplay and reducing bugs
      * Make gameplay more accurate (better players in real life should be performing better in the game)
    - Adding statistics for each player
      * gameOver screen will show stats of each player

**Version Control PlanGraphical user interface, application, Word

Description automatically generated**

* As shown in the image above, I am backing up my code using google drive. After each day of working on my term project, I upload my python file to the TP backup folder to ensure that my file is safe and backed up. The number between the parentheses indicate which version of the TP it is. The file with the highest number between the parentheses thus indicates the latest version.

**Module List**

* Matplotlib
* ImageTk
* API (requests, requests-cache)