Name: 1v1 Basketball

TP3 Update:

I removed the blocking function and instead incorporated the blocking rating with the probability of making layups because blocks usually occur near the basket. I purposely created a non-parabolic arch for the layups to differentiate those shots from the normal jump shots, which have a parabolic arch as their shot trajectory. For the players' animations, instead of using a sprite strip, I simply flipped the image as the players dribbled to show the dribbling motion; with my edited images, as the players move, it will seem like the players are dribbling. For my AI, I am forcing my AI to shoot the ball whenever it is near the rim so that it will not just stall due to the user being too close to it. I added a player selection page, and instructions page. In the game, there is a pause button, escape button to leave the game and a message center that will indicate major events that occur (e.g. a goal will show "swish") I also added a power bar indicator for the user to estimate the amount of power for their jump shots only.

TP2 Update:

This project is incomplete. I have created rectangular collisions, basketball movement and physics. For the AI comp player, it can move, chase the ball, and pick up the ball. But I am still trying to program the AI so it can attack and shoot the ball; the user can control the power of their shot by holding the "s" button down. For the user, it can do most moves except for stealing, blocking, and laying up the ball; I still need to create functions for these 3 moves. I also created a scoreboard and backboard. I also still need to include sprites for my players and incorporate probabilities to each move for both the AI computer player and the user. Other features I need to include are a main menu page, player selection, instructions, and game over function. The difficulty levels will depend on the player selection. The AI will not be run based on backtracking, but instead based on distance to the ball, distance to the player, and distance to the basket. There will not be a dunking feature and a special abilities feature.

TP1 Update:

The project will still be a 2D 1v1 basketball game. But instead of including 3 players per team, I will select a few players as the process would be simpler. There will be no difficulty levels and each player will just have attributes that depend on their real-life playstyle. I hope to still have enough time to add the special abilities feature for each player. The ball physics for dribbling and free bouncing is already completed. But I am struggling with jump shot ball physics as I am trying to derive a formula that creates a ball trajectory depending on the distance of the ball from the hoop. So I may have to hard code the trajectory to simplify the project. For the AI computer-player, I will be using backtracking. Instead of purely using distance, I will, on offense, loop over each move (e.g. 3 pointer, drive then layup, dunk) that will produce the best result and, on defense, loop over the left and right direction to see which direction will allow the AI get closer to the user. The backtracking code will stop recursing when the ball is not in possession of the AI or the game is paused, on offense, and stop recursion when the ball is in possession of the AI or the game is paused, on defense. I am also considering making this game based on my favorite player Russell Westbrook to simplify the game and only allowing the user pick the opposing AI computer-player.

Description:

This project will be a 2D 1 versus 1 basketball game where the user plays against the computer-player. The user can choose an NBA player to play against the computer in 3 different levels (easy, medium, hard) with the 3 best players on each NBA team representing each of the different difficulty levels. The user will start off by choosing a team out of the 30 NBA teams and will be assigned the 3rd best player on the team the user chose. In order to unlock more players, the user must play against a specific player. To unlock the easy mode player, the must just needs to beat the easy level player; to unlock the hard mode player, the user must beat all 3 levels for that particular team.

The gameplay would be similar to real basketball with a 24 seconds shot clock. Each game will be played with first to 11 points. Each NBA player will have an ability according to their real playstyle and the ability will appear if the user makes a certain number of shots/prevents the opponent from scoring a certain number of times (e.g. if user makes a shot/prevents opponent from scoring 4 times, then they will gain the ability which may include 100% success rate in scoring or 100% success rate in blocking the shot).

Similar Projects:

There are similar basketball term projects. The 2k112 basketball game by David Luries on the TP gallery is similar to what I envision my game to be, but I would have real players with data on players' realistic attributes and their playstyle. It would also be a 1v1 game instead of a 5v5 game and I would have abilities for each player. This player ability part of my project is also distinctive from the video game NBA 2k which has a 1v1 mode. (There is also a "One-on-one basketball game" on the TP gallery but the video is not available so I can't compare my plan to their project.)

Structural Plan:

I will have a class that stores all the players' attributes and data. For example, for the "Player" class, I could add player "LeBronJames" with the name of LeBron James, jersey number of 6, Lakers as the team, mode as hard mode (meaning he's the best player on the Lakers), ability of running through the opponent and dunking with 100% success rate, speed of 90, strength of 80, 3 point rating of 75, etc. I will also have a function for the main menu, the player selection page, and the gameplay page. For my gameplay, I will have 1 class for the users' control, and another class for the computer-player. For my user's control class, it will feature the key press functions heavily. For the computer-player's class, it will make use of the database of the players to determine how well the computer-player will defend or score.

Algorithmic Plan:

My most algorithmically complex sections include the ball physics. I would have to code how the ball bounces when a player dribbles the ball, and how the ball travels in the air during a shot. Especially for programming the ball physics in the air, I would need to derive a formula that depends on the players' location (x,y coordinates) of shooting the ball. And the type type of shot (e.g. 3 pointer, dunk, layup) would also alter the ball physics, which would involve conditional statements (e.g. if x,y coordinates are beyond 3 point line, then the 3 pointer ball physics for that location would be executed).

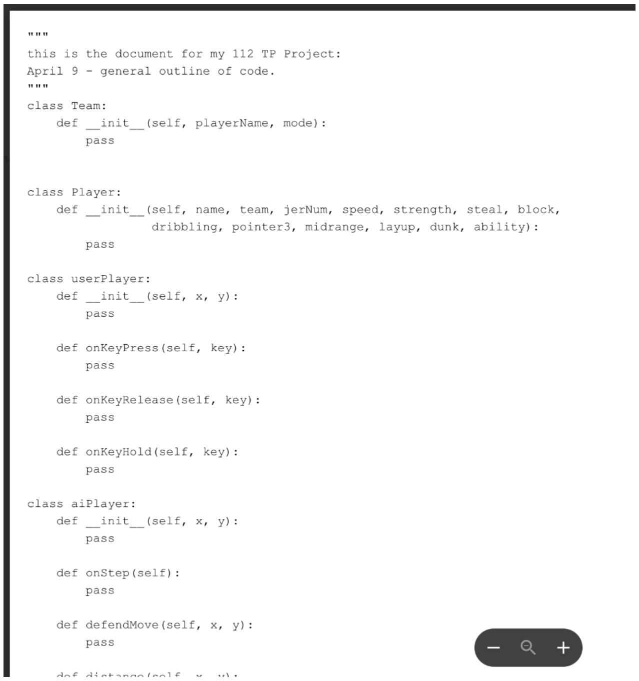
The AI component of the computer-player is also algorithmically complex. For example, if the computer-player has a slow player, then the ability that the computer-player stays on the user's player is low and so it would take more time for the computer-player and the user-player to reduce the distance which means it would be easier for the user to score if they ran quickly. So I must also derive some arbitrary formulas for the computer-player that will correctly represent the way the NBA player plays in real life.

Timeline Plan:

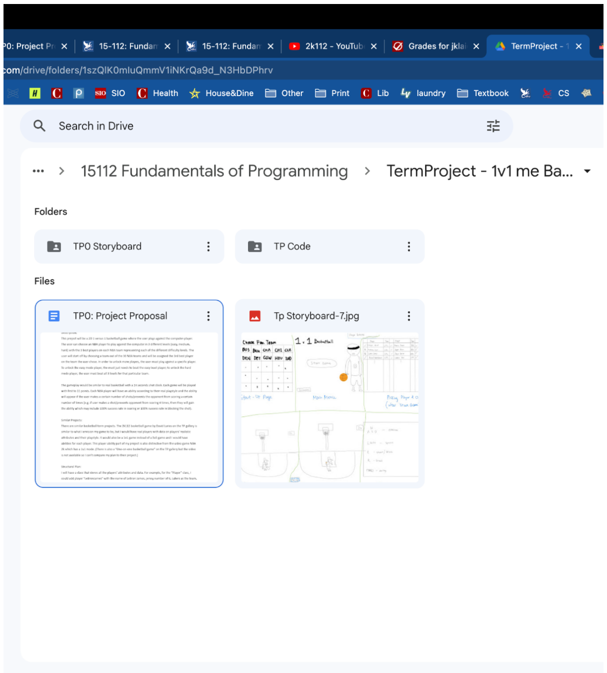
I would want to complete the database of my player with a Player Class and the main menu page and player selection page by Thursday April 13. I would like to complete the ball physics code by TP1's April 17 deadline . By TP2's Apr 22 deadline, I want to complete both the user's code and the AI computer-player's code. And before TP3's April 26 deadline, I would like to complete my project by including additional features (e.g. pause button, cool animations for using abilities,

Version Control:

I will use Google Drive to store my code. I will create a folder for the TP project and store revisions with a note in the beginning explaining the additions I made every time, showing my actual timeline as I code.



```
"""
this is the document for my 112 TP Project:
April 9 - general outline of code.
"""
class Team:
    def __init__(self, playerName, mode):
        pass


class Player:
    def __init__(self, name, team, jerNum, speed, strength, steal, block,
                 dribbling, pointer3, midrange, layup, dunk, ability):
        pass

class userPlayer:
    def __init__(self, x, y):
        pass

    def onKeyPress(self, key):
        pass

    def onKeyRelease(self, key):
        pass

    def onKeyHold(self, key):
        pass

class aiPlayer:
    def __init__(self, x, y):
        pass

    def onStep(self):
        pass

    def defendMove(self, x, y):
        pass

    def distance(self, x, y):
```



Module: None for now