PROJECT PROPOSAL

Project Description

Project Name: Virtual Soccer (really creative, I know)

This project will be a game where the user represents one team on a soccer field, and can pass, throw, and shoot with a certain player using arm and leg motions. The user will be able to switch team members and make them perform these actions. Ideally, the game will incorporate soccer rules and have a team that can play against the user's team. By the MVP stage, the user should be able to play a short game against either another player, a stationary opposing team, or an AI opposing team.

Competitive Analysis

I have found several projects online that create virtual soccer games using the Kinect. One by Lemon&Orange allows the user to be the goalkeeper and save penalty shots, with the goalie hands shown onscreen mimicking the user's movement and a computer soccer player that continuously takes shots on goal. Another game by Artefactoestudio allows the user to take the penalty shots, with a computer goalie attempting to save the shots. A third game by The Hand Creative is an interactive two-player juggling game, where a virtual soccer ball is kept up in the air by movements of the two players, whose real video is shown onscreen.

Similarly to these games, my project will translate user movement into movement of the ball onscreen, and in a direction similar to that indicated by the user's motion.

However, none of these games offers full soccer game playability, and focuses on very specific aspects of the game. Additionally, they do not incorporate multiple inputs (i.e. arms as well as legs) at the same time. My game will attempt to do these things. Also, currently my project does not aim to play from the user's perspective, as these games do.

Structural Plan

I plan to have different files or classes for the Kinect-based code and the different visual elements of the game. For example, I intend to have a file for visuals, containing a separate class for 'Player', 'SoccerBall', and 'Goal'. I will create another file with subclasses of 'Player' based on soccer positions. Under these classes, I will implement the possible actions each object has. For example, under each subclass of 'Player', I would have the function 'move' and have methods to determine possible actions if the player has the ball. Under 'Ball', I would include a 'move' function as well. Under all these classes, I will include a 'draw' function so that the elements can be drawn into the PyGame window.

I will have functions in a class that obtains user input from the Kinect and stores body data in the form of coordinates, so that this data can be accessed in other functions that detect certain actions required for the game (such as pass, shoot, and throw in).

I'll have a function (or several) that implement the game rules, and return the next required action for the user to complete (e.g. throw in, or goalkick.)

In essence, I will use separate files when possible (for example, the game run file will be separate from the files that define classes and other functions) and utilize classes to define objects on the screen. I will use functions to define characteristics of these objects and to obtain user input from the Kinect.

Algorithmic Plan

One of the trickiest parts of the project would be to implement an AI team to play against, or to include another player that controls another team. Depending on what I decide on, the AI part could be implemented using starter code from online that helps me create AI game bots that I can tune to each player on the opposing team. I will need to either use a specific module (maybe Unity-AI-Module?) to do this, or create a more simple if-then-else type of decision-making sequence that I would need to write more specifically for each player. For example, for a midfielder I could write a sequence corresponding to something like "if ball with defender, move up the field" or "if ball with opposing defender, move toward defender".

If another player is included, I would first figure out how the Kinect would distinguish between the two players, then process both their inputs to the Kinect. Then, I will translate each player's actions specifically to their assigned teams by tracking their individual body positions. They would both use the same gestures but the gestures would only apply to their team members onscreen.

Another difficult part of the project is assigning movement to each player onscreen based on the ball's position. To do this, I can look into defining standard desirable formations of teams based on how the ball moves (from my prior knowledge about soccer) and turning this into a function for movement of each player. This would be similar to the method described above for creating another team with an "if-then" decision making sequence.

Timeline Plan

November 21st: I plan to have full Kinect functionality integrated with pygame and the visual aspect of the project.

November 22: Have direction of kicks, throw ins, and passes shows onscreen.

November 23: be able to pass to teammates and switch players automatically based on direction of passes/kicks. Have ball bounce off or stick to players.

November 24: Implement soccer game rules, so that if ball goes out of bounds, player moves to take a throw in.

November 25: Implement onscreen player movement.

November 26: Address unanticipated setbacks/additional ideas to implement

November 27: implement goalies

November 28: Have full one team functionality, with goal scoring. At least stationary opposing team; Have working demo: MVP stage.

November 29:

November 30: Implement two-player game

December 1:

December 2: Implement AI team

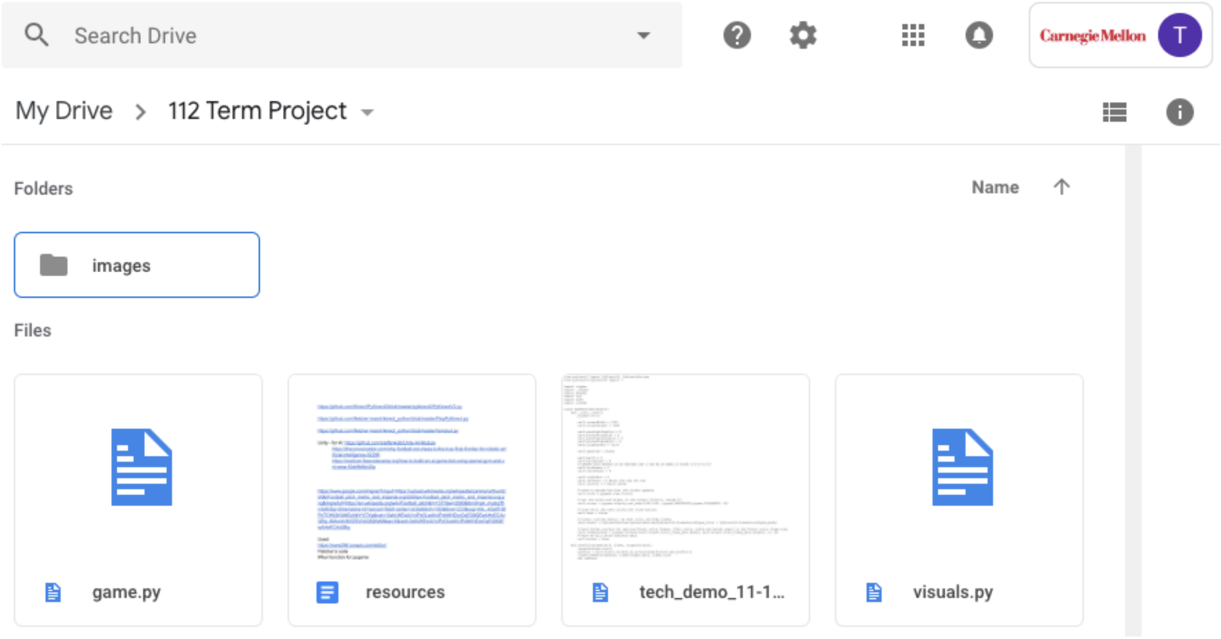
December 3:

December 4: complete UI

December 5: Add any extra features desired (sound, et cetera)
December 6: Complete final term project

Version Control Plan

I plan to use Google Docs to upload my code and other necessary files to a designated folder at different points in time. This way I can revert to older versions if necessary while retaining the current code. Image: of current folder (with code from tech demo and other TP1 files)



Module List

Hardware Used: Microsoft Kinect V2
Additional Modules: Pygame

TP2 Update

A user gesture is used to switch selected players instead of being done automatically. Also, player movement is controlled by keypresses for MVP. Otherwise, design has not changed.

TP3 Update

Player movement is now continuous, and additional gestures are used to stop movement. Interceptions are carried out by kicks on part of the Kinect player, and by colliding directly with the ball on the keyboard player's side. There is an information screen that can be accessed by a gesture during the game, and the start screen can be controlled by the user in the Kinect screen. The throw ins and passes go directly to the nearest player on the team, and players are unable to move completely offscreen, so they can't be lost. The timer is paused when any screens pop up. And there's background sound from the World Cup stadium to make the experience all the more immersive! I also added referees and automatic player movement for both teams (i.e. when one player on a team moves, some of its teammates also move with it so the users do not have to move every single player separately.)