**Updated Project Proposal** (as of Wed, Dec 6)

Name: Bo Xie AndrewID: boxie

TP3 Update:  
There are no changes to the design since TP1. I believe MVP has been hit and there are extra features post-MVP, like the driving range feature, which I have implemented into the code.

CITATION: All the images in the project are hand drawn by myself, no other images were used.

Project demo:

https://youtu.be/JYKwdSm8bjk

**Project Description:**

The project is called Golf Across CMU. It is a golf game that takes place across the CMU campus, divided into 4 holes. The players will have to control the clubs, the directions and powers on the golf swings, in order to get the ball into the holes in the least amount of strokes as possible. A second player (A.I. or second human player), competes against the first player (post MVP).

**Similar Projects:**

My game will model two very well-established golf games on the market, *EA Sports PGA Tour 2023*, and *WGT Golf Mobile*. These are the two best golf games that are available. In the games, players can select golf courses, golf clubs, and try to control the golf swing of the character such that they get the ball into the hole in the least number of strokes. There is a green and a hole for every hole of the course, and score is counted after the ball enters the hole. There are obstacles like trees, water hazards, roughs, out of bounds areas, and the wind which will add to the challenge of the players.

Similar to these games, Golf Across CMU will allow players to select golf clubs and try and control how the ball is hit, like the direction, distance, and height. The project will be done in 2D, top-down view for the golf shots, so that height and direction can still be visualized by changing the size of the ball. Different to these projects, the golf course would be CMU, and the holes will be designed to be playing across the different buildings and the CUT at CMU. Another difference is that two players are always playing together, much like real golf.

**Structural Plan:**

The project would need the Player class, Clubs class, and Hole class, (which draws on a “terrain class called object class”). The Player class is able to track properties of each player, such as the ball position, score, moves, and their golf clubs. The Clubs class would help make golf clubs including putter, wedges, 9 iron, 8 iron, …3 iron, hybrid, 3 wood, and driver. Each club will have properties like where they can be used, possible distances and heights, how difficult it is for players to control (longer clubs are harder to control, so the user must be more exact with the timing of their clicks). The Terrain class has properties like penalty (if in hazards), different roll/bounces (if ball lands in areas like rough or sand), and difficulty hitting out of (increase difficulty to hit out of on next shot).

For the player control functions, the player must decide what clubs to hit, and try to select the perfect power and distances by clicking a shifting arrow at the right time. Functions like mouse press and distance functions will be heavily used. As the selections are all done, the swing and the resulting ball path will be shown through draw functions and onstep functions.

For the ball flight/physics functions, some math/physics calculations must be done to calculate and simulate how the golf ball would react when it encounters different types of terrain. This could be simplified or really complex, so functions are not confirmed yet. Some work has been done here, but still have not gone too deep.

For the computer/AI second player functions, some more complex functions should make this player play differently every time, yet average a similar total score for the level of difficulty that the human player chooses. These functions only run if in single player mode.

Other potential additional features include hints, mode of competition (stroke play or match play), and adding a practice range. These would require more functions.

**Algorithmic Plan:**

The ball flight/physics related functions may turn out very difficult. I expect myself to be able to manage this with some practice and some advice from mentors or TAs who have done some similar projects in the past. Also, if proven too difficult, I believe I can lessen the complexity of this part of the game yet still present a very complete game overall.

The other difficult part would be the computer/AI second player functions. Without any experience in this, it may be difficult to code a whole player who plays without any user input. It would be hard to control randomness and overall stability. However, I am able to receive help about Game AI from mentors. Also, since my computer second player does not have to react to what the human player is doing (although I want it to), it can be implemented with less difficulty than most other Game AI.

**Timeline Plan:**

Mon Nov 20: TP0

* Finish TP0, have a meeting, and finalize some of the parts of this proposal.
* By Friday, have the graphics set up, and have some preliminary code that proves this project would run.
* Implement some single player controls

Mon Nov 27: TP1

* Submit preliminary code from before
* Work on the physics part of the game and ensure the smoothness of the gameplay, for two players.
* Improve the graphics and interface, maybe add the practice range section.
* Begin on the computer/AI programming

Fri Dec 1: TP2

* Submit working demo of the player controls and how the ball moves after being hit.
* Shows how the scores are counted, the movement from hole to hole.
* Work this whole week on the computer/AI second player, and get as far as possible.

Wed Dec 6: TP3 (due):

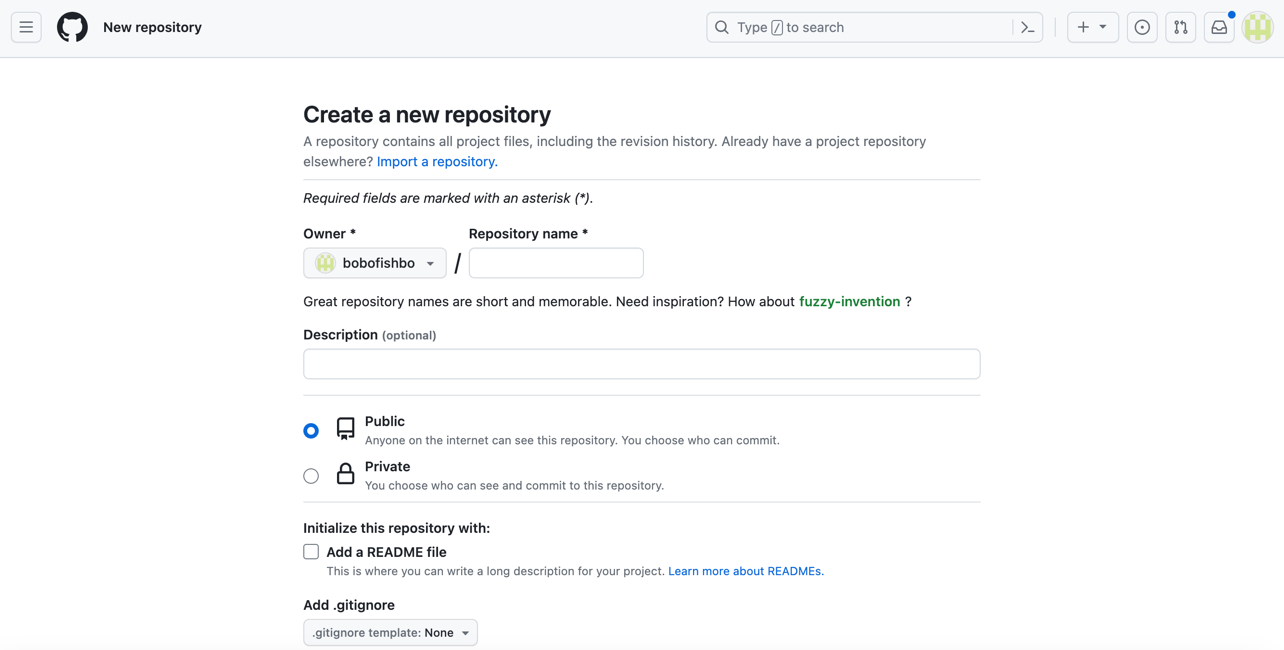
* Submit everything

Thu Dec 7: Showcase in lecture

* Presentation

**Version Control Plan:**

Use a GitHub repository. Make a new repository for this project and link this with VS code so the versions do not get messed up.



**Module List:**

I plan to only use CMU graphics, and no tech demos needed. I could use procreate or other platforms to draw my own images, but those should not require new modules.

In terms of story board, nothing was really changed except the 3d design is more 2d now and in top-down view, which you will see if you run my code, but not too different from what I had drawn.