Topic and Group Selection Template
Group members: Ricardo Larios, Leslie Xie, Eric Tatar, Daniel Nachum

Group name: BCC (Backgammon C++ Committee) (please change if you come up with a better name)

CCIS GitHub location: https://github.ccs.neu.edu/dpn703/CS3520-2017FA-PROJ.git

Provide a high-level prose description of your project.

Our project is an implementation of the classic game Backgammon. In the game, there are two players and in our current implementation, they will both be two human players (i.e. no AI or computer). Players start their pieces on either side of the board and must move their respective pieces to the capture slot on their side of the board. The size of the move is determined by two dice rolls that can each be used as moves or combined for one larger move, with a double roll giving the player twice as many moves (i.e. rolling a 4 and a 4 gives the player four 4 rolls to use). Pieces alone on their board spots can be captured by enemy pieces that land on them and are sent to the middle of the board. The player whose piece has been captured then has to roll to move it back onto their end of the board and cannot take any other actions until they get a useable roll. A useable roll is one that will not land the piece on a spot with more than 1 enemy piece and applies to normal rolls as well. The final score of the game is determined by the number of pieces left on the board by the loser and how far they are from that player's capture slot (i.e. 5 pieces at 1 space away from the capture slot will count as 5 points and 1 piece at 5 spaces away will also count as 5) and these points will go to the winner. These points persist among games and the player will the most points after 3 games will be the winner.

Describe the major features of your project. Provide this in a "checklist" format. Be as specific as possible, and provide *at least* five distinct features.

- RNG through the dice rolls.
- 2D Graphics, which will generate the image of the board/player pieces.
- Working implementation of the capture pieces mechanic and determining which rolls are useable.
- A points system to keep track of the scores between games and determine a winner.
- A doubling cube, which will act as a score multiplier that the players can decide to use.
- If possible, deterministic AI for the user to play against.

Describe the advanced feature(s) of your project, and the library/SDK/API you plan to use.

We're planning using 2D graphics to create a visual version of Backgammon. These graphics will definitely include the pieces, the board, the die, the doubling cube, and the score. We plan on using OpenGL for the 2D graphics, although we are not married to that idea. Our plan is to first "program" the game, and then "design" it.

Describe plans for what kind of user input your program will take and how it will affect the state of the program.

Choosing which rolls to use and for what pieces through some form of choosing input (whether through a selection on the keyboard or with a mouse) will change the state of the game through players moving their various pieces across the board and potentially capturing enemy pieces. This will change the states of the pieces.

Briefly describe plans for dynamic memory management and class inheritance structure.

All the pieces are being allocated at the beginning of the game generation, and nothing will necessarily be deallocated until the end of the game/the game is quit. We could potentially deallocate the pieces once they arrive in the captured slot after changing them into values.

As for class inheritance structures, an abstract class for a player piece will be created and then the separate two player classes will be derived from that. They have separate colors and can move in opposite directions, which may be the only differences.