**Final – Water Pong - Brag Sheet:**  
Michael Panighetti

Interactions:

\* Interactions:

\* - The goal of the game is to throw the ball into all of your

\* cups before your opponent does.

\* - Press 'R' to Start The Game.

\* - Hold Left-Click over ball to drag into desired position.

\* - Higher position = longer Shot.

\* - Lower position = shorter Shot.

\* - Press 'Space Bar' to throw the ball.

\* - Scoreboard is at the top of the screen.

\* - There will be a mark next to the player whose turn it is at top of screen.

\* - Also, cup color will signify whose turn it is.

\* - Red = Player 1

\* - Blue = Player 2

Accomplishments and Challenges:

* I’ll start by saying that I had bigger plans for this assignment than what I could handle with the time given, so I had to adjust. Did we have less time than the other assignments? or was I just busier in other classes during this time of the semester? Maybe a little of both.
* I started with wanting to do a swept surface object with the solo cup. That took a while to figure out, especially due to the rows and columns being dependent on the curve design and parameterization.
* Secondly, the collision. I really couldn’t get the sphere intersection with a plane working – I don’t know what I was overlooking, but after hours of trying to get it to work I decided to do an array of small spheres. I made the space between them smaller than the width of the ball so that nothing got through, except everything needed adjustments and tweaking.
* I think the best part is the lighting and textures. I created the textures using clippings of things I had found on google images and such. I quite like how the scene looks!
* The game logic was a tad extensive. I needed it to flow properly from start to finish.
* And lastly, the tracking of the ball’s coordinates was a challenge because I wanted to use glRotated for the movement, but then I thought, how do I have coordinates for the collision aspect then? …and because of this, I had to translate all the movements made to the ball over to coordinates as well, ugh.
* Overall, it all seems simple until you run into problems... and then have to troubleshoot for hours debugging.
* I wanted to use 6 cups on each side instead of 3, but it cost too much performance/lag. I tried lowering the number of columns/rows with my cup rendering, but I didn’t want to lose the smoothness of the curve.

List of features:

* The game works. Yay.
* Lights – overhead with lamp, glow of cups on score and miss (green and red respectively). The lighting has an incredible ambience in my opinion. Very polished.
* Many self-created custom textures for tabletop, walls, carpet, and ball.
* Everything has material settings.
* Rendered text on the screen to guide the user and show the score of the game. The text positioning adjusts according to the window size.
* Thorough collision detection for scoring and misses.
* Color-picking with separate buffer for clicking and dragging the ball before throwing.
* The balls ‘highlights’ while doing this by providing a bit of emission.
* Table has a shadow that can be seen during the side-view – this shadow is transparent with blending. This creates a nice realistic effect on the carpet.
* The ball is thrown with a timer function and adjustments made to a rotation with an offset pivot (origin/center of rotation).
* Ball has shadow on the table when thrown.
* Ball starting position uses random coordinates within a specified range.
* The game turns are guided with a message plus countdown.
* **See planning notes included below:**