**Group Members: Sparsh Rawlani and Carlos Betancourt**

**CS 371-Final Project Documentation**

**Date: January 24, 2019**

**Instructor: Matthew Bell**

**Design:**

Our design for the Pong game is made using a WPF main window and the source code is written in C#. Our WPF window has a Canvas, two rectangles(which are the paddles), an ellipse (which is the ball) and a line(which is the center division where the ball respawns), three buttons(one for start and pause, one for resetting the game, and one for exiting the game). We also have a couple of text fields that take in the name of the users and message boxes that display the name of winner after the game has ended. The ball spawns in the center and goes in a single direction for a player to win. After a player has gone up to a score of 7 points, the speed and color of the ball changes to increase the difficulty level of the game, another twist is that when either of the players get to the score of 15, the difficulty level increases to the GOD MOD and the game becomes unplayable for more than some seconds. And last but not the least, to make the game look more real, we added some background music and beeps whenever the ball hits the paddles. The players use Up and Down keys and the A and S keys to move the two paddles.

**Code Design:**

We have two classes in our code, namely Ball and Paddles. The class ball just has private member variables with their getter and setter properties defined so that they can be called in the main window file. Class Paddle on the other hand is derived from an interface called INotifyPropertyChanged. We are using this interface to assign different properties to our player models/objects in the main window. Every model has a different binding property that is derived from the interface. In order to reset the position of the ball, we have a method reset ball that sets the x and y position of the ball using the setter methods from the ball class. In order to create collisions, we did the math and based on the height and width of the canvas, and based on the position of the paddles, we are stopping the ball at that point and redirecting it, making the collision to work accordingly. We use a vector struck to store the x and y values of radian which is basically the deviation angle of the ball. We have another method called checkBounds that checks the upper and lower bounds based on the height of the canvas. We have buttons to start/pause the game, and a couple of other buttons to restart and exit the game.

**Problems Faced:**

We had two big problems that we really struggled with over the weekend. One was the collision with the ball and the other was the dealing with the deviation of the ball using the vector and the position of the X and Y axis of the ball. The second big problem was to implement collision using some math, we use the height and width of the canvas to create collision and bounce the ball back in the opposite direction. Other small problems that we faced were inconsistent paddle speeds, inconsistent changes in the speed of the ball, and score count. At last, I would say that we worked hard and we were satisfied by our results.