**Final Project Document**

Work with your team members to complete this final project document.

FINAL PROJECT GROUP 8

Lane Albrecht, Gabriel Galarza, Connor Moore , Timothy Zarlenga

# What is the name of your project?

# ULTRA VIRTUAL MINI TENNIS 3000 – X

(THE REMAKE) v 1.01

# Which game/concept are you building?

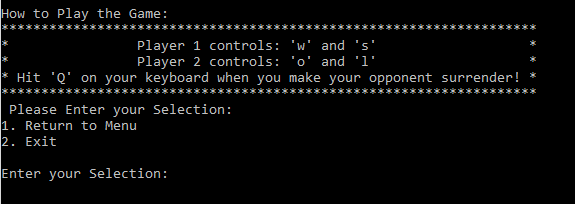
Old school Pong

# Give a brief description of what data structures (arrays, classes, struct, files,…etc.) you will be using and how.

3 classes for Ball, Paddle, and a manager for movement, controls, score keeping, drawing etc.

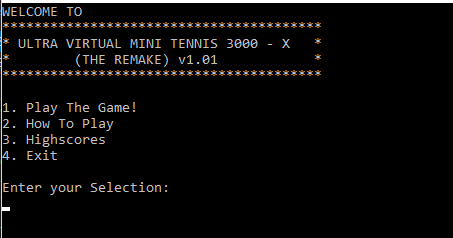
Also a struct and vector for sorting the high score board file.

How to play the game:



In the program there is a menu option, “How to Play,” to see how to play the game. However, this is a two player game where player 1 uses ‘w’ and ‘s’ to move their paddle up and down and player 2 uses ‘o’ and ‘l’. If you wish to quit the game press ‘q’.

# What will your interface look like?



# How did you benefit from our class to create your project?

We learned how to write classes, the in class time for planning the project out helped, the code in the discussion helped us implement sound. We also learned how to override operators.

Also the in class discussion helped us get the if statement for reading keyboard hits during gameplay.

# How do you plan on implementing Polymorphism, inheritance, and encapsulation?

Manager class will inherit ball and have pointers pointing to the paddle classes in order to reset them after scores and get positions. Also we are overloading the () operator using a vector to sort the high score board, and have private variables for each class that are only accessed by get functions.

# Which part of your game is the most difficult to program? How are you going to overcome the difficulty?

# 

Movement of the ball, it needs to be random and collide with objects/reset game when it passes each players paddle.

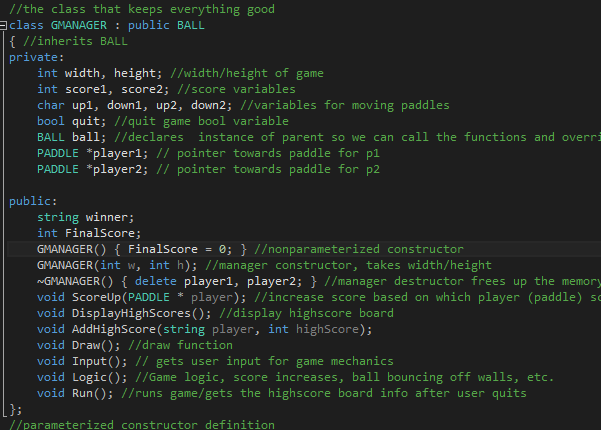
To fix it, we use enum class that declares the directions so we can use them. Then take the current position of the balls x and y coords, and with some basic math, make it so when the ball hits a certain location (height or width of the game border) it will “bounce” back in a direction or increase the score then reset if it hits the left/right walls, we could possibly use the rand() function built into C++ to randomly make it bounce in a direction. For hitting paddles, we use similar method to bouncing off walls but also read the positioning of the paddle.

Creating and sorting the high score board was complicated. First we had to make a file and have the program read and write to it. We needed to sort it by score and alphabetical order as well as make sure it output in the correct format to make sure it actually was legible.

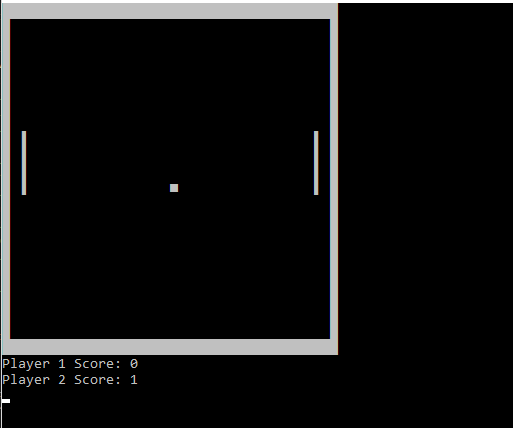
Persisting issues:

Sometimes during play the pong ball will clip through the right (player 2) paddle, which will end up giving a point to player 1. This is only an issue seen on the current computers in class, as our personal computers ran the program much more smoothly and the clear screen command was not as obvious with the amount of flashing. Since the issue never occurred for the player 1 (left) paddle, we believe its just a case of the clear screen command and the interaction between this computer when it clears the screen as the ball moves.

What is the main class in your game (write the complete code with prototypes)? [[1]](#footnote-1)

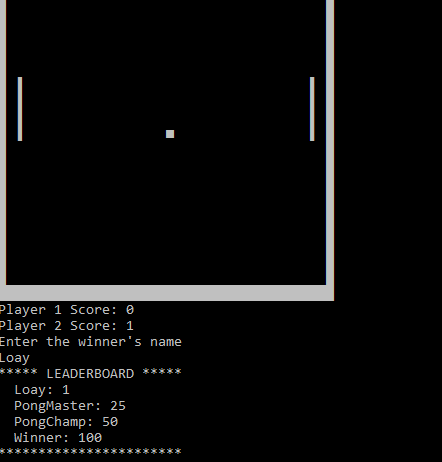


What the game looks like when played (during then after):



During play ^

After game ends ( user hits ‘q’ ) :



1. For CS10A you’re not using classes, list at least 4 functions that your final project will be using. Include parameters and what each function does. [↑](#footnote-ref-1)