

NYU Tandon School of Engineering

CS-UY 3113 Fall 2023

Project 2: *Pong Clone!*

Due: 11:59pm, Saturday, October 21st, 2023

Submission instructions

1. You must use delta time in your animations. Do not use the `Entity` class in this assignment.
2. You should submit your homework on [Brightspace](#).
3. You should also push that same version to your GitHub account. Note that any commits done after the deadline will be ignored.
4. For this assignment you should turn in **one** C++ file named `main.cpp` and any image files that you used in your project. Do **not** submit any other files.
5. Do not use any OpenGL functionality that we have not learned in class.
6. The `main.cpp` file you submit should contain a header comment block as follows:

```
/**
 * Author: [Your name here]
 * Assignment: Pong Clone
 * Date due: 2023-10-14, 11:59pm
 * I pledge that I have completed this assignment without
 * collaborating with anyone else, in conformance with the
 * NYU School of Engineering Policies and Procedures on
 * Academic Misconduct.
 **/
```

No late submissions will be accepted.

Sections

Requirements

1. [Paddles / Players \(50%\)](#)
2. [Bounces Off Walls \(25%\)](#)
3. [Game Over \(25%\)](#)

Tips

1. [Common Issues](#)
2. [Extra Credit](#)

Requirement 1: *Paddles / Players (50%)*

- There needs to be a paddle on each side that can move independently to the other.
- The ball needs to bounce off of the paddles.

- The paddles should not be able to go off the top or bottom of the screen.
- Both players should be allowed to use the keyboard. For instance, player 1 could use the **W** and **S** keys, while player 2 would use the **Up** and **Down** arrows.
- If the player presses the **t**-key, one of the paddles should switch from player-controlled to a simple up-and-down motion. In other words, we are simulating the player switching from 2-player mode to 1-player mode. Make sure that the user *can't* use their keys to move the paddle when in 1-player mode.
- You can use both images and untextured polygons.

Requirement 2: *Bounces Off Walls (25%)*

- The ball needs to bounce off the top and bottom of the screen.

Requirement 3: *Game Over (25%)*

- The game should stop when someone wins or loses (i.e. when the ball hits a wall on the left or right).

Common Issues

- To keep things simple, I'd recommend using **box-to-box collision detection** only.
- Work object by object, mechanic by mechanic, and test often. This goes for all programming, but if you try finishing all objects in one go, you are bound to miss something that have a hard time finding it.

Extra Credit

- **Have fun with it and have a theme.** Instead of square paddles and a ball, have other kinds of objects/images with the same mechanics. It doesn't need to be super polished—anything that looks fun is welcome.
- **Have an endgame UI message.** When someone wins, you can show an image of text saying who won.