Pong Narrative

Cameron Cuzmanko

December 2015

1 What does the RTICA do

My RTICA simulates one of America's classic video games, Pong, in a 3D environment using VPython. The RTICA allows the user to control a paddle and play against a computer controlled paddle in an attempt to get the ball past the computer's paddle.

2 How did you do it

First, I needed to make the environment. To make the environment, I used VPython's built-in box class to create the boundary walls and the paddles. Next, I created the ball, using VPython's built-in sphere class, and made a simulation of the ball bouncing off of the walls work (see my vPythonBallInCage project). Next, I limited the ball's z-rebounds to only occur if the ball hits the paddle of the user or computer.

Once all of that was done, I moved on to implementing the user controls, as the game would be woefully incomplete without them. After searching VPython's library, I was able to find a method that properly received keypresses. I then used these keypresses as parameters to another method which would move the user's paddle according to which key was pressed. At this point, the user is now able to move left, right, up, down, into the screen, and out of the screen.

With user control out of the way, it was time to work on the computer's paddle movements. This proved to be rather tricky as the AI (artificial intelligence) needed to be imperfect. By that, I mean that the AI shouldn't lose very easily, but it should also not win every time. To accomplish this, I set the AI's speed to a constant value. At first, this value should allow the AI to reach the ball with no problem. To account for this, I added increasing speed to the ball depending on how it hit the paddles. Eventually, with a nice bounce or quick speed, the ball's trajectory will prove to be too difficult for the AI's paddle to catch.

3 How to use it

Once the program is running, the game should pop up with text saying "Click to start." Once the user has done as it says, the ball will begin to travel. The user now has control over the paddle. The controls are as follows:

- W = Move the paddle up
- S = Move the paddle down
- A = Move the paddle left
- D = Move the paddle right
- Q = Move the paddle into the screen
- E = Move the paddle out of the screen
- Escape (esc) = End game

With these controls, the user can move the paddle to meet the ball as it comes toward the user's end. If the user is able to meet the ball as it comes to the user's end, the ball will reflect back toward the AI's side. If the user is unable to meet the ball, the program will end and the user will need to restart it if they wish to play again. I must give a slight disclaimer regarding running the program. Sometimes, if the program has been edited or not been run before, the first couple of runs may be extremely prone to lag and make it difficult for the user to move the paddle. If this is experienced, just restart the program and it should work properly within the next few runs.