Student: Brendan Turner

P number: P15232006

Module: IMAT1213(HTML JavaScript)

Tutor: Justice

Assignment: create a pong game in JavaScript

**What we had to do**

Due to the implications with our previous module tutor’s absence, this assignment was a lot less involved than it should have been. so, the basic template for the pong game had already been set out, A function for creating the ball and one paddle along with some to-do messages at various points.

**Other paddle and boundaries**

By referencing the exercises that we have done in previous weeks, drawing and positioning the boundaries and second paddle took no time at all.

**Bouncing the ball of the boundaries**

To bounce the ball off the boundaries was a simple task of multiplying the y speed by negative 1 each time it reaches the x position of where the boundaries would be.

**Moving the paddles**

To move the paddles, I set up a function to check for keyboard presses and saved the key code to a variable. I used that variable in a switch statement to call the movement for either paddle.

The movement itself is a small function that checks whether the paddle has room to move and if so it will then move a certain number of pixels (10), if there is no room it will simply do nothing.

**Resetting the ball and incrementing the score**

To reset the ball and increment the score I put a check in the ball update function to see whether the x position of the ball falls outside of the canvas boundaries on either side. If it does, then it will call the ball reset function.

The ball reset function simply resets the starting position back to the centre, sets the x speed to either 1 or minus1(left or right) and increments either players score depending on whether the ball left the right or left side of the screen (this is done with a Boolean upon creation of the ball).

**Bouncing the ball of the paddles**

Bouncing the ball off the paddles is where I had most trouble, if I had the ball moving too fast (too many pixels per update) then the check to see whether it had hit the paddles position would be bypassed and the ball would appear to pass through the paddle.

To solve this I put the entire update function in a for loop using a sort of global speed variable for the ball as the counter. It would then do all the checks and move the ball at a speed of 1 multiplied by the time elapsed a specific number or times per update.

The other problem was getting the ball to bounce off at a specific angle. Initially, the ball would travel to the right and then no matter where it would collide with the paddle, it would simply start traveling towards the left with no up and down movement.

I solved this by having the ball target a point further back then the paddle as it collides with it and calculate an angle off that. Depending on how far away from the centre of the paddle, the sharper the angle. This is done only on the first off centre bounce and the angle is then set for the rest of that round.

**Small problems I had during the assignment**

Once I had got it all working a couple of the bugs that I noticed were:

* The origin point of the paddles of which I had initially thought were the centre, were on the top left corner instead, and so the ball was bouncing off an invisible paddle on one side and passing through on the other. This was a simple case of adding the equivalent width and height onto the value.
* If the ball was left to bounce from both paddles without either being moved, due to how the initial bounce check was done the balls y speed would be its initial value of 0 multiplied by negative 1 and then that would be multiplied again by negative 1 on the other side. This would cause an overflow. To solve this I put a further check in that line of code to make sure the paddle wasn’t in the exact centre and if it was, then the y speed would stay at 0.

**Conclusion**

Although the assignment was very simplified due to the time constraints we have had, I’m reasonably happy with how it has turned out. I’m pleased with my understanding of the class structure and the adding attributes to classes with the prototyping. And it was nice to put some of the mathematics we have learned into practice.