

Simple Pong Game Using JavaScript

The Plan

What I plan on doing is implemented a game of the classic 'pong' game by manipulating HTML and CSS properties and functions using JavaScript. The original inspiration for this idea was this page on the W3Schools website:

https://www.w3schools.com/js/js_htmldom_animate.asp

Specifically what inspired me is the final example on the bottom. Clicking "Try it Yourself" opens up a new page in which you can test the simple code and functionality. My first thought was how I could implement this where the button/mouse was not the input(s) required to start the animation but a keyboard press of a specific key.

After messing around with the idea and changing around some of the code (mostly unsuccessfully) I stumbled across the 'onkeypress' event written into HTML:

https://www.w3schools.com/jsref/event_onkeypress.asp

With this event, not only can we invoke the function to move an element on the page, but we can also send that function exactly which key was pressed, meaning that a single invoked function can have different functionality (up, down) depending what was entered in the text box.

Current State/Problems

The current state of the proof of concept is simply a single maneuverable paddle (using the 'w' and 's' keys), the gameboard, a div that will eventually become the ball, and the input textbox. What I eventually would like to implement is to add some sort of event listener to the whole page that listens for keyboard inputs. That way, I can get rid of the need for the textbox as well as the need for the cursor to be inside the textbox to collect the text. I will implement a second paddle that will take the inputs from some keys on the right side of a keyboard ('o' and 'k' maybe?) to accept a second player's input. The other immediate things that needs to be addressed is the stationary ball.

Once the ball is animated then the next major hurdle for the game will be collision testing on a web page. Luckily, this page seems to point me in a good direction:

https://developer.mozilla.org/en-US/docs/Games/Techniques/2D_collision_detection

After collision detection has been implemented, as well as the necessary functionality that comes with what happens after a collision is detected, the very last thing to implement is the rest of what makes a game. Things like a scoring and reset system, some minor decorations, a reset button, etc.