Programming Assignment 1

Due Sep 29, 2020 by 11:59pm **Points** 100 **Submitting** a file upload **Available** until Oct 2, 2020 at 11:59pm

This assignment was locked Oct 2, 2020 at 11:59pm.

A JavaScript Game

This project must be done individually. No copying is permitted. The goal of this project is to learn client-side web programming using JavaScript. More specifically, you will create a game that runs on a web browser using JavaScript.

This project must be done individually. No copying is permitted. **Note: We will use a system for detecting software plagiarism, called Moss (http://theory.stanford.edu/~aiken/moss/) , which is an automatic system for determining the similarity of programs.** That is, your program will be compared with the programs of the other students in class as well as with the programs submitted in previous years. This program will find similarities even if you rename variables, move code, change code structure, etc.

Note that, if you use a Search Engine to find similar programs on the web, we will find these programs too. So don't do it because you will get caught and you will get an F in the course (this is cheating). Don't look for code to use for your project on the web or from other students (current or past). Just do your project alone using the help given in this project description and from your instructor and GTA only. Finally, you should not post your code nor deploy your project on a public web site.

Platform

You will do this project on your own PC/laptop. You have to use the Mozilla Firefox web browser to run and debug your JavaScript code. You need to activate the Debugger from the browser Developer tools. You may click on "Show in Separate Window" to get the debugger on a separate window.
Note: You should not use any JavaScript library, such as JQuery or d3.js. You should not use the JavaScript canvas object nor svg graphics.

Documentation

The following web pages contain various tutorials. Use them as a reference only. The class slides contain enough information on JavaScript and DOM.

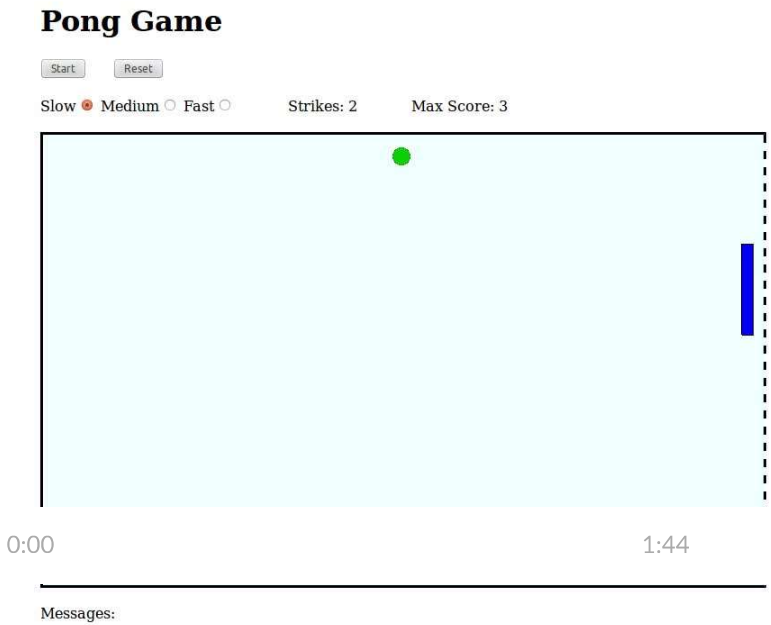
- **JavaScript Tutorial** **(http://www.w3schools.com/js/default.asp)**
- **JavaScript** **(http://www.cs.rochester.edu/courses/210/spring2011/lectures/009/)**
- **JavaScript DOM** **(http://www.cs.rochester.edu/courses/210/spring2011/lectures/010/)**

Project Description

You need to write a JavaScript file `pong.js`, used in the file **pong.html** (after you click on this link, you can get the HTML source if you right-click and use View Page Source). Your code must implement the following actions:

- initialize: initialize the game
- startGame: starts the game (when you click the mouse)
- setSpeed: sets the speed to 0 (slow), 1 (medium), 2 (fast)
- resetGame: resets the game
- movePaddle: moves the paddle up and down, by following the mouse

Please watch the following demo video of how your game should look like:



The pong court is 800x500px, the pong ball is 20x20px, and the paddle is 102x14px. They are all specified in pong.html. When you click on the Start button or left-click on the court, the ball must start from a random place at the left border of the court at a random angle between -π/4 and π/4. The paddle can move up and down on the right border by just moving the mouse (without clicking the mouse). The ball bounces on the left, top, and bottom borders of the court. Every time you hit the ball with the paddle, you add one strike. If the ball crosses the right border (the dotted line), the game is suspended and the strikes so far becomes your score. You would need to click on the Start button or click on the court to restart with a zero number of strikes. So the goal of this game is to move the paddle to protect the right border by hitting the ball.

Hints:

- The position of any element is dictated by the three style properties: position, left, and top. If an element is nested inside another and its position is "relative", the top and left properties are relative to the enclosing element.

  ```
  <p id="x" style="position: relative; left: 50px; top: 100px;"> ... </p>
  ```

  To move this element, just change the left/top attributes using code:

  ```
  document.getElementById("x").style.top = "10px";
  ```

Note that the values that you set the left/top attributes must have units (e.g., "10px"). It will not work if you set them to numbers.

- You can get the X and Y coordinates of the mouse using the pageX and pageY attributes of an event (e.g., from the event that is passed on the `onmousemove` handler).
- You can get the top coordinate of a regular element `x` (such as, the court) using `x.getBoundingClientRect().top` (also: left, width, and height).
- To animate an element, it must be moved by small amounts, many times, in rapid succession. For example, you can use `setTimeout("fun()", n)` that calls `fun()` after a delay of n milliseconds (you have to put it in a loop or use recursion).
- It will be easier to develop your code by first ignoring the paddle and making all the court borders solid, so the ball will bounce on every border. After you make this work, you can change your code so that the ball that tries to cross the right border bounces if it hits the paddle. You need to define a time period (the "tick") `dt` to calculate the new `x/y` coordinates from the current. The speed coordinates `vx/vy` are determined when the ball is kick-started (from the kick angle). The new `x` is `x+vx*dt`, but if the new value is beyond the right border, then the ball must be bounced by setting `vx = -vx` and `x = 2*width-x`, assuming that the court x-coordinates are from 0 to `width`. You do something similar for the left, top, and, bottom borders.

Note: You should use plain JavaScript. You should not use any JavaScript library, such as JQuery and d3.js. You should not use the JavaScript canvas object or svg graphics.

What to Submit

Submit your `pong.html` and `pong.js` files. After you submit the files, please double-check that your submitted files are correct.