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 video game (Breakout) 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 Links to an external site.**](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 should use Google Chrome or any Chromium-based browser (such as Brave) to run and debug your JavaScript code. To run your program on your browser, go to file:/// and navigate through your files until you find your project1/breakout.html. On your browser, to get the developer tools, you go to the main menu, select More tools, and then Developer tools.  
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 TutorialLinks to an external site.](http://www.w3schools.com/js/default.asp)
* [JavaScriptLinks to an external site.](http://www.cs.rochester.edu/courses/210/spring2011/lectures/009/)
* [JavaScript DOMLinks to an external site.](http://www.cs.rochester.edu/courses/210/spring2011/lectures/010/)

Project Description

You are asked to write a JavaScript program that implements the Breakout game. Please watch the following demo video of how your game should look like:

Download [https://lambda.uta.edu/cse5335/project1.zip Links to an external site.](https://lambda.uta.edu/cse5335/project1.zip)and unzip it. You should put your code in the JavaScript file breakout.js in project1, which is used by the file breakout.html, that implements the following actions:

* initialize: initialize the game
* startGame: starts the game (when you click the mouse inside the court or you push Start)
* resetGame: resets the game (restores bricks, clears Tries, but doesn't clear Score)
* movePaddle: moves the paddle left and right by following the mouse

The court is 805x500px and the ball is 20x20px. They are all specified in breakout.html. When you click on the Start button or left-click on the court, the ball must start from the middle of the paddle at a random angle between 3π/4 and π/4. The paddle can move left and right at the bottom by just moving the mouse (without clicking the mouse). The ball bounces on the left, top, and right borders of the court and on the bricks. If the ball hits a brick, the brick becomes invisible (it is removed). If you remove all the bricks (there are 4\*20 bricks), your score is incremented. If the ball crosses the bottom border but doesn't hit the paddle, the Tries counter is incremented. You would need to click on the Start button or click on the court to continue.

I have already provided the HTML and the code to lay out the bricks, the paddle, and the ball. Also I have provided the method movePaddle that moves the paddle and ball by following the mouse move. I have also provided code that checks whether the ball hits the brick (i,j).

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.

* 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).
* 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 bounce 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.
* It will be easier to develop your code by first ignoring the paddle and the bricks, and by making all the court borders solid, so that the ball will bounce on every border (but not on the bricks nor on the paddle). After you make this work, you can change your code so that the ball that crosses the bottom border bounces if it hits the paddle but goes through and stops otherwise. Then you can finish the program by bouncing and removing bricks when hit. When hitting a brick, you simply set the visibility="hidden" for the brick and dy=-dy.

**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

Zip your project1 directory that contains your breakout.html and breakout.js files. Submit your project1.zip