Maharishi University of Management, CS472

Lab 9: Mouse Maze

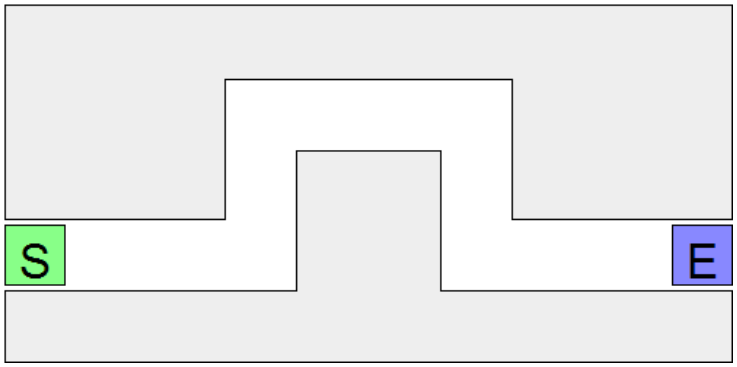
original lab idea and code by Victoria Kirst and Jeff Prouty; revised by Brian Le, Katlyn Edwards, Roy McElmurry IV, and Marty Stepp

Basic lab instructions

- You may want to **bring your textbook** to labs to look up syntax and examples.
- Have a question? **Ask a TA** for help, or look at the book or lecture slides.
- We encourage you to **talk to your classmates**; it's okay to share code and ideas during lab.
- You are not expected to finish all of the exercises. Just do as much as you can in the allotted time. You don't need to finish the rest after you leave the lab.
- Before you leave, **check in** with a TA to get credit for your work.

Today's lab

This lab practices unobtrusive JavaScript events and the Document Object Model (DOM). We'll write a page with a **"maze"** to navigate with the mouse. You will write `maze.js` to implement the maze behavior.



Info about the maze

Download the file below (right-click, Save Target As...) to get started:

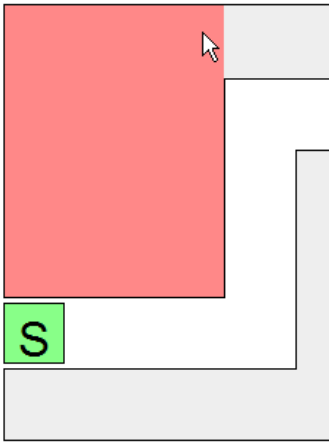
- maze.html

The difficulty is in having the dexterity to move the mouse through **without touching any walls**. When the mouse cursor touches a wall, all walls turn red and a "You lose" message shows. Touching the Start button with the mouse removes the red coloring from the walls.

The maze walls are 5 `div` elements. Our provided CSS puts the `div`s into their proper places.

```
<div id="maze">
    <div id="start">S</div>
    <div class="boundary" id="boundary1"></div>
    <div class="boundary"></div>
    <div class="boundary"></div>
    <div class="boundary"></div>
    <div class="boundary"></div>
    <div id="end">E</div>
</div>
```

Exercise : Single boundary turns red (~15 min)



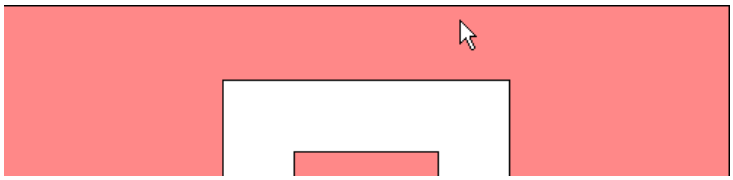
Write code so that when the user moves the mouse onto a single one of the maze's walls (`mouseover`), that wall will **turn red**. Use the top-left wall; it is easier because it has an `id` of `boundary1`.

- Write your JS code **unobtrusively**, without modifying `maze.html`.
- Write a `$(document).ready();` handler that sets up any event handlers.
- Handle the event on the wall by making it turn red.
- Turn the wall red by setting it to have the provided CSS class `youlose`, using jQuery's `addClass` method.
- You might want to use the jQuery event handler assignment methods mouseover and mouseleave. E.g.,
  $("#someId").mouseover(function() { alert('You just moved your mouse over the #someId element!');

Exercise : All boundaries glow red on hover (~10 min)

Make it so that ***all* maze walls turn red** when the mouse enters any one of them.

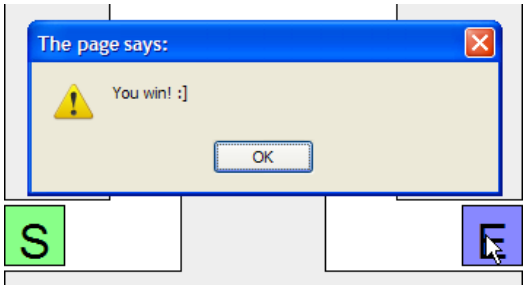
- You'll need to attach an event handler to each `div` that represents a wall of the maze.
- It is harder to select all of these `div`s, since they do not have `id` attributes.
- But they do all have a `class` of `boundary`. jQuery's `$()` function will find all elements that match the CSS selector.



Exercise : Alert on completion of maze (~10 min)

Make it so that if the user reaches the end of the maze, a "You win!" `alert` message appears.

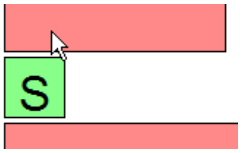
- The end of the maze is a `div` with an `id` of `end`.
- Don't pop up "You win!" unless the user makes it to the end **without touching any walls**.
- Keep track of whether any walls were hit, so you'll know what to do when the end square is hit.

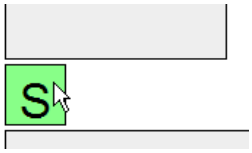


Exercise : Restartable maze (~10 min)

Make it so that when the user clicks the mouse on the **Start** square (a `div` with an `id` of `start`), the maze state will **reset**. That is, if the maze boundary walls are red, they will all return to their normal color, so that the user can try to get through the maze again.

- You'll need to use the `$()` function again to select all of the squares to set their color.





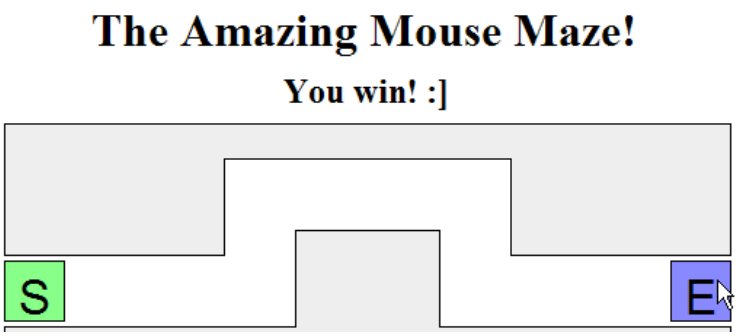
Exercise : JSLint / Upload Page (~5 min)

- Verify your JavaScript code by making sure it passes **JSLint** with no errors.
- Then follow the directions at our Uploading Files page to **upload your page** to Webster and make sure it still works there.

Exercise : On-page status updates (~10 min)

Instead of an `alert`, make the "You win" and "You lose" messages appear **in the page** itself.

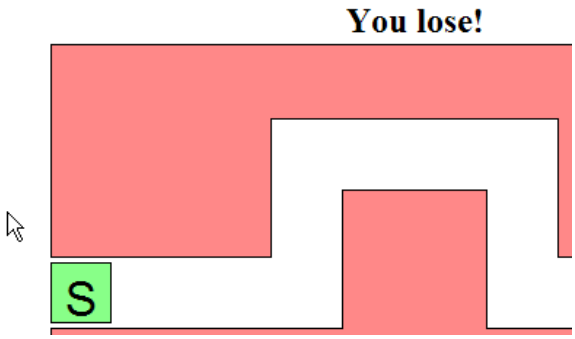
- The page has an (initially empty) `h2` element on the page with an `id` of `status`. Put the win/lose text into that `div` when the user finishes the maze.
- Hint - use jQuery's `.text()` function to edit the text inside of the `h2`.



Exercise : (h4x0rz only): Disallow cheating

It's too easy to **cheat**: Just move your mouse around the outside of the maze!

- Fix this by making it so that if the user moves the mouse anywhere outside the maze after clicking the Start area, the walls will light up red and the player will lose the game.
- To do this, you'll need to listen to other kinds of mouse events on other elements.



If you finish them all...

If you finish all the exercises, you can add any other content or code (or bling!) you like to your page.

If the lab is over or almost over, check with a TA and you may be able to be dismissed.

Great work!