Assignment #11: Web Games

Description:

Now that we have a basic handle of JavaScript, we're going to take a look at our Ruby Games again. This time however we're going to rebuild one of them into a JavaScript web application. In the process we'll be solidifying the JavaScript fundamentals we learned about yesterday, while learning about prototypes and jQuery.

JavaScript's Prototype object model is a little different from the Object Oriented Model in Ruby. From Wikipedia:

> Prototype-based programming is a style of object-oriented programming in which classes are not present, and behavior reuse (known as inheritance in class-based languages) is performed via a process of cloning existing objects that serve as prototypes. This model can also be known as classless, prototype-oriented or instance-based programming.

Moreover, instead of using the terminal to interact with the user, we're going to use the browser instead. This assignment will introduce a JavaScript API present in the browser that allows us to interact with the HTML elements present on our page, called the Document Object Model (DOM). The DOM gives our client side code the ability to access the HTML and modify almost any property on the elements. We can even use JavaScript to create new elements on a page!

This is where jQuery comes in, it will help us deal with the DOM. From jquery.com:

> jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Pick one of the Ruby games you wrote in the first week (between Pig Latin, Hangman, and Tic Tac Toe) and work on porting it over to the browser.

Learning Goals:

jQuery

- Selectors
- Manipulation
- Events

Process:

Build the UI

The first thing you should look at doing is creating the HTML and CSS for your game. Build out the web page first, add the elements you think you will need. If it helps, try drawing this out on paper before coding. Getting it down onto paper sometimes helps you figure out what you need to show the user and how to display that information.

Get input from the user

There are a few ways you can get input from the user. You can take a look at the `confirm()` function, which you should be familiar with from the Arms Network assignment. You could provide a keyboard of buttons (for Hangman, e.g.). You could also just provide a text box and allow the user to type in their input. There are pros and cons for each of these methods, and part of a developer's job is to evaluate and pick one that fits your application needs.

Displaying information to the User

One way of displaying data back to the user would be to look through the jQuery docs, and how you can create elements in jQuery. Once these elements are created in JavaScript, you can use the jQuery `append()` function to add that element to the page.

Tips and Tricks:

- In JavaScript, we use special prefixes on variable names to define what they contain. Generally, variable names that start with a '`$`' are jQuery objects, variables that start with an '`_`' are private variables. So if you see somewhere `$_variable`, you can infer that it is a private jQuery object. When you create or store jQuery objects in variables, make sure that they start with a $.
- Try to modularize your code as much as possible. The more modular it is, the easier it will be to fix bugs and find issues. Make sure to use the most out of prototypes for this.