Menu

Previous

Next

Unit 2 / Lesson 2 / Assignment 1

Street Fighter: Project Overview

Estimated Time: **1-2 hours**

You will now work on a project that brings together DOM manipulation, events and effects, as you build a Street Fighter themed interactive web page. You'll practice using the jQuery API and get a better sense of what your workflow might look like when you're using jQuery for interactive effects on a web page.

Your end product will look similar to this demo page (Be forewarned: This page plays some groovy music. You might want to turn your volume down before visiting it). We will walk you through everything you need to do to achieve the basic layout and implement the click behavior. You will then be on your own to implement a key effect. There is also no walk-through for creating the initial animations of the Streetfighter and jQuery logo. When we get to the final part of this project where you implement the "x" key feature, you will also show off your knowledge of jQuery by adding some text to the bottom of the page that explains how the page works.

This project should be fun, but even if 1990s videogames are not your cup of tea, this project will still teach you valuable skills that you can use to build other interactive web pages.

This project will be completed in several steps. In this assignment, you will fork a GitHub repository with starter files, review what's in the repository and go over a high-level approach for implementing your page. In the next assignment, you'll do some initial work on layout and style. After that assignment, you will learn about click behavior. Then, in the final assignment, you'll implement the "x" key down behavior independently.

Forking the Repository

To start, you'll need to fork a repository that contains starter files for the project. In git, when you want to use an existing repository as a starting point for your own project, you can fork it. This creates a new copy of the project at its current commit. If you make changes to a forked repo, it does not affect the original, and if the original changes, the forked repo is unaffected. Go to this repository on GitHub. In the top right hand corner of the page, you'll see a "Fork" button. Click on this button (note that you'll need to be logged in to GitHub to do this). You'll see a message that GitHub is forking the repository, and after a few moments you'll be rerouted to https://github.com/YOUR-ACCOUNT/streetfighter-starter. You'll see several files there.

Before cloning this repo to your local computer, let's change the repository name — "streetfighter-starter" is not a great description for your repo, since you'll actually be building out the full project. From https://github.com/YOUR-ACCOUNT/streetfighter-starter, click on the "Settings" button on the right hand side of the screen. On the following page, change the repository name to something more descriptive — for instance, "jquery-streetfighter" — then click the "Rename" button.

Now you're ready to clone this repository. Open your GitHub client. Click on your account name under the "GitHub" header on the left of the app. Then, find the repo you just renamed in the list of repositories and click the "Clone" button. If you're using Windows, you'll have to hover over the repo for the "Clone" button to appear.

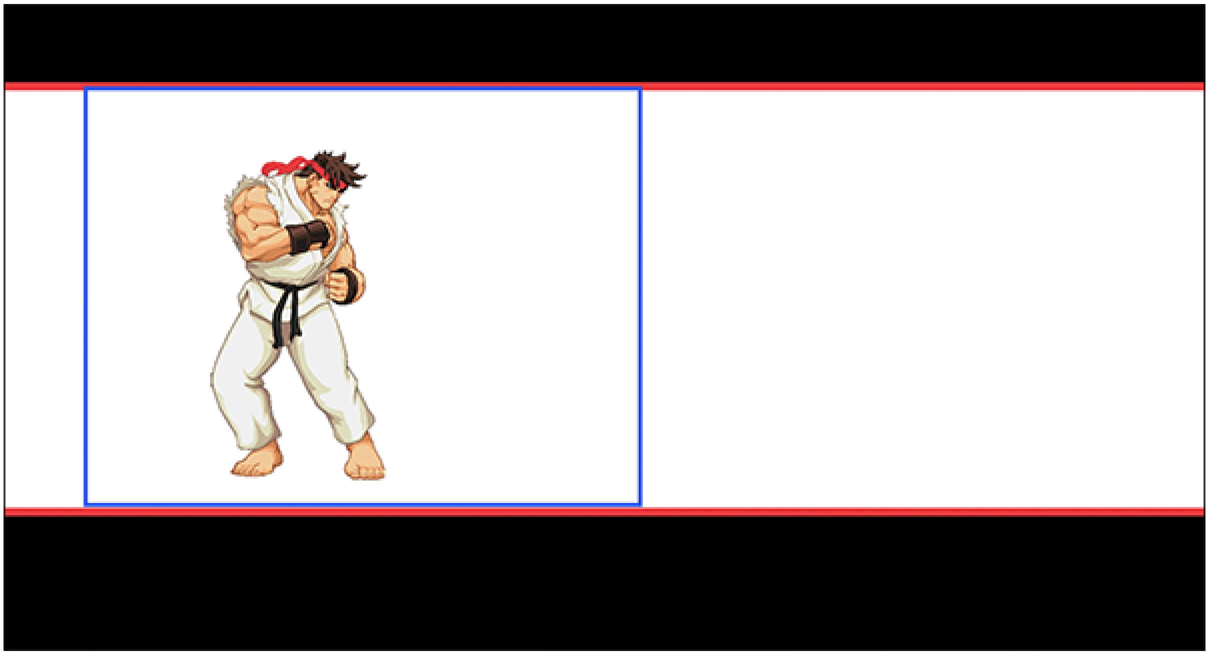
Project File Overview

After you've cloned the repo, you should open it in Sublime Text. You'll see that we have the following contents: a single html page (index.html); a folder for CSS with two files in it — a blank main.css file and a normalize.css file; a folder called images with several image files in it; and finally, a folder for our JavaScript file with a blank app.js in it. If you open up index.html, you'll see that the JavaScript file has already been linked to jQuery and the two CSS files. You also have boilerplate HTML code. Aside from this, though, all the code needed to get this site working still needs to be written.

Before moving on, look at the images in the images folder. If you're on a Mac, you can open the images folder in Finder. Then, click on each image and look at the preview and the information about its dimensions. If you're on Windows, open the images folder in Windows Explorer, click on each image and note its dimensions. Mac users will notice from the image previews the following files are animated gifs: ryu-cool.gif, ryu-ready-position.gif, and ryu-throwing-hadouken.gif. Windows users, you'll need to open each file (for instance, in Chrome) to see the animations. Note that all of the Ryu files have the same dimensions, and Ryu is positioned at roughly the same position in each one. This will be important later in this project when creating effects.

Layout and Strategy

The picture below demonstrates the basic layout wanted for the page. There is a black background with a white container in the middle of the page. Inside of this is the Ryu standing still image, which is pushing away from the left side of the page. You will write the CSS to achieve this basic layout in the next assignment.



Start planning by outlining what you want for the user experience. The page loads, Ryu is standing still. You hover the mouse over Ryu, and he shifts to his ready pose. You stop hovering over Ryu and he goes back to still. You click on Ryu, and he leans forward and throws a Hadouken. He remains in that pose until you release your click, which causes Ryu to return to his ready pose. When you hit the "x" key, Ryu assumes his "looking cool" pose, and when you release "x" he returns to his ready pose.

Most of the work will lie in implementing the visual aspects of this page. When you reviewed the image files earlier, you noted that each of the image files has the same dimensions. You also noted that Ryu's starting position for each image is comparable. This means that if you were to simply swap out the different Ryu images based on the user inputs, it would create the illusion of fluid animation. It looks like you should be able to simply `.hide()` whichever image is currently displayed, and then `.show()` the right one. For the Hadouken sequence, you'll have to both swap in the right Ryu image and load the (appropriately positioned) Hadouken image. The Hadouken image file is an animated gif, too, but the animation only captures the flickering of the Hadouken. It will be up to you to make the Hadouken move from left to right. You'll also have to figure out how to play the Hadouken sound effect.

The items described above can be broken down into a set of events you want jQuery to respond to. The initial state is Ryu still. You should listen for mouseenter and mouseleave events on Ryu and move him into and out of his ready position accordingly. You should listen for click events and initiate the Hadouken sequence in response. There's a mousedown and mouseup component to this sequence. You should listen for "x" for keydown (and keyup) events on Ryu and respond with the looking cool sequence.

In the next assignment, you'll begin implementing this vision, focusing on base HTML and CSS.

☆ ☆ ☆ ☆ ☆ · Report a typo or other issue

✓ Mark as completed

Previous

Next