Menu

Previous

Next

Unit 2 / Lesson 2 / Assignment 4

Street Fighter: Animation Walkthrough

Estimated Time: **1-2 hours**

In this assignment, you'll write the code to handle when a user clicks on Ryu. When this happens, you want Ryu to throw a Hadouken (and yell "Hadouken!"). You will code the `.hide()` and `.show()` behavior first, then animate the Hadouken, and finally add the "Hadouken!" sound.

Ryu Throwing Position

You want Ryu to fire Hadoukens when users click on him. More specifically, when the mousedown event is detected, you want Ryu to lean forward. When the user releases the mouse, Ryu should return to his still position.

Follow the same procedure you did before by first proving that your events are firing. This time, though, instead of using alert(), use console.log(). Because you're testing mouseup and mousedown behavior, and having to click to dismiss an alert would interfere with this. Also, put in placeholder comments about the behavior you want in your event callback functions. Change app.js to look like this:

```
$(document).ready(function() {
  $('.ryu').mouseenter(function() {
    $('.ryu-still').hide();
    $('.ryu-ready').show();
  })
  .mouseleave(function() {
    $('.ryu-ready').hide();
    $('.ryu-still').show();
  })
  .mousedown(function() {
    console.log('mousedown');
    // play hadouken sound
    // show hadouken and animate it to the right of the screen
  })
  .mouseup(function() {
    console.log('mouseup');
    // ryu goes back to his ready position
  });
});
```

The console will now log both the mousedown and mouseup events. Save these changes, then return to Chrome. This time, before refreshing the page, open Developer Tools and click on the Console tab. Then, reload the page and try clicking on Ryu. You should see a message logged in the console for the mousedown and mouse up events.

Focus next on swapping out Ryu images on mouse down — animating the Hadouken and playing the Hadouken sound will come later. Hiding and displaying looks like it should work here. By definition, when you click on Ryu, the mouse already has to be over him, so that means you'll need to hide div.ryu-ready, not div.ryu-still. Then you'll want to show the Ryu throwing and Hadouken images. Alter your .mousedown event handler so it looks like this:

```
  .mousedown(function() {
    // play hadouken sound
    $('.ryu-ready').hide();
    $('.ryu-throwing').show();
    $('.hadouken').show();
    // animate hadouken to the right of the screen
  })
```

Note that the comments in our code to remind us to implement the Hadouken sound and animation are retained. Save these changes, then go back to the browser, refresh the page and verify the behavior. If you release the mouse while still hovering over Ryu, he'll remain in his throwing position because we haven't implemented the mouseup event handler yet. Let's take care of that now. Modify `.mouseup()` so it looks like this:

```
  .mouseup(function() {
    $('.ryu-throwing').hide();
    $('.ryu-ready').show();
  });
```

Save these changes, then return to the browser to verify everything works. Note that when the mouse is released, although Ryu returns to his ready position, the Hadouken doesn't disappear. This is a good thing because you will animate across the screen, and then want it to disappear.

You've made great progress, so take a moment to add and commit your changes. Here's what app.js should look like at this point:

```
  $(document).ready(function() {
    $('.ryu').mouseenter(function() {
      $('.ryu-still').hide();
      $('.ryu-ready').show();
    })
    .mouseleave(function() {
      $('.ryu-ready').hide();
      $('.ryu-still').show();
    })
    .mousedown(function() {
      // play hadouken sound
      $('.ryu-ready').hide();
      $('.ryu-throwing').show();
      $('.hadouken').show();
      // animate hadouken to the right of the screen
    })
    .mouseup(function() {
      $('.ryu-throwing').hide();
      $('.ryu-ready').show();
    });
  });
```

Animating the Hadouken

To animate the Hadouken, use jQuery's animate() method. The animate method allows you to specify style properties and values, and jQuery will handle transitioning from the current settings to the new ones in a specified interval of time.

To animate the Hadouken's motion from Ryu's hands to the right of the screen, you will animate the left property on div.hadouken. When div.hadouken reaches its right most point, you'll use a callback function (that is, a function that gets executed after a prior function completes) to .hide() the div.

Inside of app.js, replace `$('.hadouken').show();` and the placeholder comment in the mousedown method call about animating the Hadouken, with the following code:

```
$('.hadouken').show().animate(
  {'left': '1020px'},
  500,
  function() {
    $(this).hide();
    $(this).css('left', '520px');
  }
);
```

Before previewing the animation in the browser, go over each of the parameters supplied. If you glance back at main.css, you'll see that the .hadouken class' left property is set to 520px. The first parameter says to animate our jQuery object so its left value is 1020px (500px to the right). The 500 (milliseconds) is how long you want this animation to take. Finally, the anonymous function at the end will be called when the animation completes. It says to hide the element with the .hadouken class then set its left property back to its original value of 520px.

Make sure to save your changes and then preview the page in Chrome. When you click on Ryu, you should see the Hadouken fire. If you click on Ryu repeatedly, however, you'll notice that the Hadoukens no longer originate from his hands. This is because the previous animation hasn't completed when you're retriggering the lever.

There are different ways of dealing with this bug, but you'll use jQuery's .finish() method. .finish() completes all currently running animations on an element. Add the .finish() method immediately after the section of code where you select the .hadouken in the mousedown method. Modify your code so it looks like this:

```
$('.hadouken').finish().show()
  .animate(
    {'left': '1020px'},
    500,
    function() {
      $(this).hide();
      $(this).css('left', '520px');
    }
  );
```

Playing the Hadouken Sound

Your nearly finished coding the responses wanted when users click on Ryu. Now you need to trigger the sound. For this, you're going to write a named function that you'll call on the line with your comment about audio.

A dive into HTML5 audio is beyond the scope of this project, so some basic code will be provided. If you're curious about HTML5 audio, check out the Mozilla Developers Network's articles about the audio element and using HTML5 audio and video.

First, you need to embed an audio element to index.html. Inside of that file, right after the closing tag for div.main, add `<audio id="hadouken-sound" src="sound/hadouken.mp3">`. Save your changes to this file.

Now inside app.js, the bottom of the file, after the document ready function, add the following code:

```
function playHadouken () {
  $('#hadouken-sound')[0].volume = 0.5;
  $('#hadouken-sound')[0].load();
  $('#hadouken-sound')[0].play();
}
```

When this function gets run, it will load and play the sound file indicated in audio#hadouken-sound. Volume values can range from 0 to 1. Set the volume to 0.5, so it's audible but not blaring. The final step for your audio effect is to call playHadouken within the body of the .mousedown() code block. Modify the mousedown code so it looks like this:

```
.mousedown(function() {
  playHadouken();
  $('.ryu-ready').hide();
  $('.ryu-throwing').show();
  $('.hadouken').finish().show()
  .animate(
    {'left': '1020px'},
    500,
    function() {
      $(this).hide();
      $(this).css('left', '520px');
    }
  );
})
```

When mousedown fires, it will now call the playHadouken() function. Save this change, then refresh the page in your browser. Fire a Hadouken, and this time you should hear sound.

Take this opportunity to add and commit the changes you've made. If your code isn't working, note that this is what app.js should look like at this point:

```
$(document).ready(function(){
    $('.ryu').mouseenter(function(){
        $('.ryu-still').hide();
        $('.ryu-ready').show();
    })
    .mouseleave(function() {
        $('.ryu-still').show();
        $('.ryu-ready').hide();
    })
    .mousedown(function(){
        playHadouken();
        $('.ryu-ready').hide();
        $('.ryu-throwing').show();
        $('.hadouken').finish().show().animate(
            {'left': '1020px'},
            500,
            function(){
                $(this).hide();
                $(this).css('left', '520px');
            });
    })
    .mouseup(function(){
        $('.ryu-throwing').hide();
        $('.ryu-ready').show();
    });

});

function playHadouken() {
    $('#hadouken-sound')[0].volume = 0.5;
    $('#hadouken-sound')[0].load();
    $('#hadouken-sound')[0].play();
}
```

You're almost done with this project. In the next assignment, you'll be responsible for implementing an event listener that makes Ryu "look cool" when the user holds down the "x" key.

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

✓ Mark as completed

Previous

Next