Week 6: Assignment 6: Working with Sprites + User Input

Due Oct 5 at 12:59am **Points** 30 **Questions** 3
Available Sep 26 at 1am - Dec 15 at 12:59am 3 months **Time Limit** None

Instructions

Description

For this assignment, you may create an original game of your choosing, based on some of the examples from the Screencast. You might also find inspiration in the Phaser Examples: Games. There are nine games there with more complex gameplay than we have attempted yet. Each one demonstrates more sophisticated coding patterns:

1. Breakout: **https://phaser.io/examples/v2/games/breakout (https://phaser.io/examples/v2/games/breakout)**
2. Defender: **https://phaser.io/examples/v2/games/defender (https://phaser.io/examples/v2/games/defender)**
3. Gem Match: **https://phaser.io/examples/v2/games/gemmatch (https://phaser.io/examples/v2/games/gemmatch)**
4. Invaders: **https://phaser.io/examples/v2/games/invaders (https://phaser.io/examples/v2/games/invaders)**
5. Matching Pairs: **https://phaser.io/examples/v2/games/matching-pairs (https://phaser.io/examples/v2/games/matching-pairs)**
6. Simon: **https://phaser.io/examples/v2/games/simon (https://phaser.io/examples/v2/games/simon)**
7. Sliding Puzzle: **https://phaser.io/examples/v2/games/sliding-puzzle (https://phaser.io/examples/v2/games/sliding-puzzle)**
8. Starstruck: **https://phaser.io/examples/v2/games/starstruck (https://phaser.io/examples/v2/games/starstruck)**
9. Tanks: **https://phaser.io/examples/v2/games/tanks (https://phaser.io/examples/v2/games/tanks)**

Before the term is over, you will learn all of the Phaser methods and properties used in these more advanced examples. Many of the concepts are covered in the weekly live screencast and assigned reading from specific sections of Chapter 6 in *An Introduction to HTML5 Game Development with Phaser.js.*

Requirements

The assignment challenges you to integrate your own custom JavaScript code and external graphic asset files. You will use the Phaser JavaScript library to explore some of the possibilities of working with sprites and user input.

Build a game that takes user input via keyboard or pointer (mouse). Your game must incorporate multiple sprites and make use of a Phaser sprite group. Your game does not need to display a score, but should utilize the Phaser render state to display some debugging information about the gameplay (in my example, I show how many bullets are flying around at any moment).

1. Use your own bitmap images for sprite art, or assets that the author licenses as open source. A great online repository is OpenGameArt found at **https://opengameart.org/** **(https://opengameart.org/)** .
2. Set the dimensions of your game world to at least 800 pixels wide by 600 pixels tall. Use the *Phaser.CANVAS* mode for rendering the game.
3. Incorporate use of the *Phaser.Input* class (object value) to detect and respond to user-triggered events from the keyboard and/or mouse.
4. Display and manage multiple sprites using the *Phaser.Group* class (object value). Not all of your game's sprites need to be in a group, but some must be.
5. Include a block of code to run during the *render* state of your Phaser game that displays some debugging information about the current gameplay.
6. Validate the JavaScript in your work and provide a screenshot: **http://esprima.org/demo/validate.html** **(http://esprima.org/demo/validate.html)**

Purpose

Begin working with sprite management concepts to deliver user experiences with optimal performance. Learn how to use Phaser library's *Input* class to respond to a variety of user input.

Tools

- Cloud9 IDE code editor and file manager
- Chrome browser with Chrome developer tools
- Phaser JavaScript library found at **http://phaser.io/** **(http://phaser.io/)**
- JavaScript code validator found at **http://esprima.org/demo/validate.html** **(http://esprima.org/demo/validate.html)**
- OpenGameArt asset repository found at **https://opengameart.org/** **(https://opengameart.org/)**

Submission Directions

1. If you have not already done so, share your Cloud9 workspace with the instructor's account, ***srjcewilde.*** For instructions on sharing a workspace, see **https://docs.c9.io/docs/share-a-workspace** **(https://docs.c9.io/docs/share-a-workspace)** .
2. Create a folder inside of your Cloud9 workspace.
3. Name your folder "module06".
4. Upload a copy of the latest version of the Phaser.js library file (phaser.min.js) to your "module06" folder, along with any image assets used for sprite display.
5. Complete all JavaScript coding needed to meet assignment requirements.
6. Make a screenshot of the validation confirmation screen. Upload the screenshot to question 6.1.

7. Preview your HTML file containing your JavaScript in Cloud9 using the running application, and copy the URL where your file can be viewed on the Internet. Enter the preview URL for your page for question 6.2.
8. Copy the contents of your custom JavaScript code and paste into your response to question 6.3.

Take the Quiz