**Week 4 Assignment**

**Overview**

In this assignment, you'll create another fun and interactive game for web browsers. This time, your app must dynamically update your HTML pages with the jQuery library.

**Remember**

You will be fully capable of doing this homework by the end of Saturday's class.

**Before You Begin**

1. Create a new GitHub repo called week-4-game, then clone it to your computer.
2. Inside the week-4-game folder, create the following: index.html, composer.json and index.php.
   * Place the appropriate content in the composer.json and index.php files, so you can deploy your game to Heroku.
3. Still inside the week-4-game directory, make a folder called assets.
   * Inside the assets directory, make three additional folders: javascript, css and images.
     + In the javascript folder, make a file called game.js.
     + In the css folder, make a file called style.css.
     + In the css folder, make a file called reset.css. Paste into it the code found from the Meyerweb Reset.
     + In the images folder, save any of the images you plan on using.
4. Push the above changes to GitHub.
5. Choose whichever game you want to make from the choices below. The CrystalsCollector game is easier than making the Star Wars RPG, but you'll get a more comprehensive review with the Star Wars exercise.

**Option One: CrystalsCollector Game (Easy)**

1. [Watch the demo](https://github.com/RutgersCodingBootcamp/08-16-JC-Class-Content/blob/master/Class-Content/04-jquery-rpg/2-Homework/Instructions/homework_demos/crystalsCollector_demo.mp4).
2. The player will have to guess the answer, just like in Hangman. This time, though, the player will guess with numbers instead of letters.
3. Here's how the app works:
   * There will be four crystals displayed as buttons on the page.
   * The player will be shown a random number at the start of the game.
   * When the player clicks on a crystal, it will add a specific amount of points to the player's total score.
     + Your game will hide this amount until the player clicks a crystal.
     + When they do click one, update the player's score counter.
   * The player wins if their total score matches the random number from the beginning of the game.
   * The player loses if their score goes above the random number.
   * The game restarts whenever the player wins or loses.
     + When the game begins again, the player should see a new random number. Also, all the crystals will have four new hidden values. Of course, the user's score (and score counter) will reset to zero.
   * The app should show the number of games the player wins and loses. To that end, do not refresh the page as a means to restart the game.

**Game design notes:**

* The random number shown at the start of the game should be between 19 - 120.
* Each crystal should have a random hidden value between 1 - 12.