COMP2132 PROJECT

OBJECTIVES

Implement a JavaScript web application game for the web browser.

DUE DATE

Assigned during session10, due 11:59pm the night before session12.

REQUIREMENTS

Create your own version of the standard hangman game.

General Requirements

- The hangman game should randomly select a word and hint from a collection of words and helpful hints.
- The user must guess the correct word by entering letters into an input box.
- If the user guesses a letter that is contained in the selected word than the game should display the correctly guessed letters in the position of the word where they are located.
- If the user makes an incorrect guess, then the program should display part of the hangman graphic.
- After the user guesses a letter, disable the option so they cannot choose the same letter more than once per game.
- If the user makes too many incorrect guesses and the entire hangman graphic is displayed, then the user loses the game. Most hangman games allow 6 incorrect guesses
- If the user correctly guesses all the letters in the selected word than the user wins the game.
- When the game is over either from making too many incorrect guesses or correctly guessing the word, then the game should display the results (Tell them if they Won or Lost the game) and give the user the option to **'Play Again'**.
- When a game is over, ensure the user cannot keep guessing letters, but must choose a **'Play Again'** option before they can play a new game.
- If the user chooses to **'Play Again',** reset everything (eg enable all buttons), and start a new game.

More on next page

Technical Requirements

- The words and hints should be defined in a JSON formatted file, and read by JavaScript using fetch request(s) over HTTP.
- The game page should be carefully designed and styled with CSS to present a high-quality user experience. Submitting a project with poor or little styling will result in a greatly reduced final mark.
- There must be at least 6 images used.
- All paths used in HTML, CSS and JavaScript files must be relative paths. Do NOT use server root paths that begin with **/** or client specific paths like **C:/**
- HTML, CSS and JavaScript files must be free of serious errors (warnings are ok).
- Code must be well tabbed and use descriptive variable names.
- JavaScript code must include the use of one or more **function**s authored by you, and one or more **Object**s authored by you.
- Must include at least one JavaScript animation (for example, a fade in effect).
- jQuery may be used if desired.
- CSS should be compiled from SASS. SASS file(s) should demonstrate the use of at least one SASS mixin. Both .css and .scss files should be included with the project submission.
- Project must be published to a public repository on Github.com

SUBMISSION

Before the due date, publish all project resuorces to a public repository on **Github.com**. Email your instructor the Github.com URL before the due date:

To: jeffrey_parker@bcit.ca or **jparker6@my.bcit.ca** (send to one, not both)

From: Use your **@my.bcit.ca** email accessible from **learn.bcit.ca**

Subject: COMP2132 Project

Body: The URL to your project repository on Github.com

If completed with a partner, only one public repository is required with one email notification.

As a backup, also compress all project resources (HTML, CSS, JS, images, folders, etc) into a single **.ZIP**. Upload **.ZIP** to

learn.bcit.ca > Content > Session10 > Project Dropbox