COMP2132 :: PROJECT

OBJECTIVE Create an interactive game using HTML, CSS, images and Javascript

REQUIREMENTS You may team up with one other classmate for this project, or you may complete it yourself.

Select **one** of the following two options to implement:

a) Dice game (easier)
b) Hangman (harder)

The dice game will be easier to code, the hangman game will be more challenging. Both of them will be marked the same., so make your choice according to your strength with Javascript.

Details for the rules and requirements of each game will be provided on the following pages.

No matter which game you implement:

- The game page should be nicely styled with CSS and have a well laid out, usable user interface in a viewport >= 1024px wide. Submitting a project with poor or little styling will result in a greatly reduced final mark. It does not need to be responsive.
- 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 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: functions authored by you, and one or more objects 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 SASS variables and at least one SASS mixin. Both .css and .scss files should be included with the project submission
- Project code base must be published to a public repository on Github.com

Game Parameters

Create a dice game where a user plays against the computer. The user and the computer each roll a pair of dice 3 times. After the third roll of the dice the player with highest score wins

The scoring for the game works as follows:

- If any of the players two dice comes up as a 1 then the score for that round for the player is 0. eg: if the player rolls a 6 and 1, they get a score of 0
- If the player rolls a pair of the same numbers then the players score is the total of the two dice times 2. eg: if he player rolls 5 and 5, they get a score of (5+5)*2=20
- If the player rolls any other combination of dice other than the ones mentioned above then the players score is the total value of the two dice, eg: player rolls a 3 and 2, player gets a score of 3+2=5
-

The game should provide a text or graphical output showing the following:

- The current rolled dice by the player and the computer
- The score for this round for the player and the computer
- The total score for the game

The game should provide a button that will roll the dice for the player and the computer

After three rolls of the dice the game should total up the scores and display a message displaying who the winner was

The game should provide a button that will reset the game and start a new game

NOTE: a video demonstration of the required functionality for the Dice Game can be found on the learning hub > Session10 > Content

HANGMAN **Game Parameters:**

Create your own version of the hangman game

- The hangman game should randomly select a word and hint from a collection of words and 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
- 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

Additional notes:

After the user chooses a letter, disable the button so they cannot choose the same letter more than once in a game

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. A new game should reset everything (eg enable all buttons)

NOTE: a video demonstration of the required functionality for the Hangman can be found on the learning hub > Session10 > Content

SUBMISSION **Project Assigned**: Session 10

Progress Report Due: Session11

Completed Project Due: 11:59pm the night before Session 12

Progress Report

You (or your team) must contact your instructor by email at

jeffrey_parker@bcit.ca

before Session11 to provide a progress report with the following information:

- The names of those who is on your team
- Which game you have chosen to make
- A public Github URL where the code repository is published (the code does not need to be completed at this stage, but the current state of the project code should be kept up to date on Github.com)
- If working in a team, how you plan to allocate the tasks
- Any questions or concerns you may have

Completed Project

Submitting the completed project includes two requirements:

Submission A

Compress all files (HTML, CSS, JS, etc) and folders into a single **.ZIP** file and name it

Lastname_firstname_project.zip

OR

Lastname_firstname.zip

And upload it to the Drop Box:

https://learn.bcit.ca > COMP2132 > Session12 > Project Drop Box

If completed with a partner, be sure each of you upload your own copy of the code base to the Drop box.

Submission B

Publish your completed code base to a public repository on Github.com. Email your instructor the Github.com URL before the due date.

If completed with a partner, only one public repository is required. The code on the public repo should be identical to that uploaded to the Drop box