**Prog 109 Project 2: Number Guessing Game**

* Due Jun 3, 2016 by 11:55pm
* Points 30
* Submitting a website url or a file upload
* File Types zip
* Available May 24, 2016 at 7:40pm - Jun 17, 2016 at 11:59pm 24 days

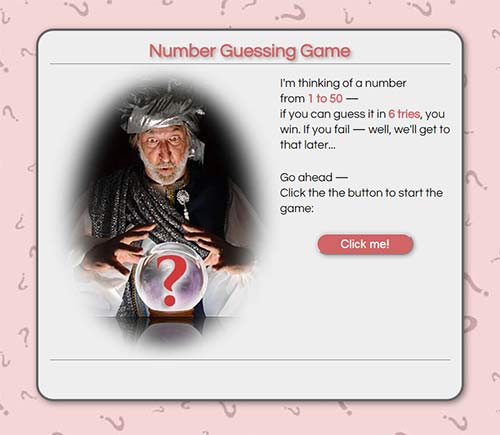
This assignment was locked Jun 17, 2016 at 11:59pm.

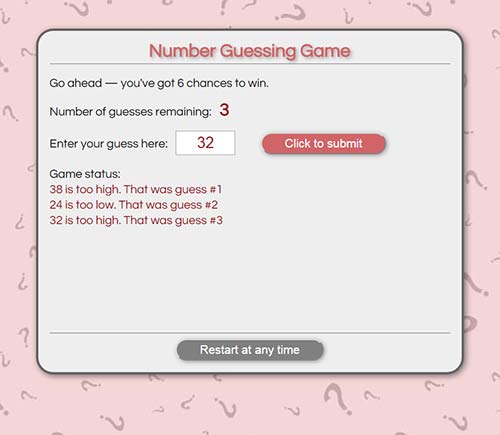
**Description:**

Develop a web-based number guessing game. The computer will select a number from 1 to 50. The player will then have six tries to guess the correct number using basic logic skills.

**Specifications:**

* Your project should consist of an HTML document and 3 directories for each of the following: **1)** an external JavaScript file which contains all your JavaScript code, **2)** an external CSS file, and **3)** at least 1 image file. There should be **NO** JavaScript code (like event attributes) in your HTML file *otherwise points will be deducted*.
* Your program will consist of 2 game screens. The first will be a **start screen** which should contain an image, a description of the game, and a button. Below is a sample screenshot of what this might look like:



* When the button is clicked, the start screen will be replaced by the **game screen**, which is where all the user interaction will take place. Here is a sample screenshot of what the game screen (already in progress) might look like:  
    
    
  ***NOTE:*** *We haven't discussed how to show/hide HTML content yet, but I will reveal all in class. If you plan on starting the project right away, try working on the game screen in the meantime. That is the most important part.*
* The game screen will contain at a minimum:   
  **1)** a counter of some sort that shows how many guesses have been made (or how many are left, your choice)  
  **2)** a textbox for entering the player's guess   
  **3)** a button that will evaluate the player's guess when it is clicked   
  **4)** some kind of message that tells the user their status after each guess -- the more information the better  
  **5)** a restart button that will refresh the browser when clicked, effectively restarting the game. This feature is **optional**, but the code to do this is quite simple:

location.reload(true); // reload browser window to restart the game

* The game will evaluate the player's guess and determine if it is less than, equal to, or greater than the computer's number.  If it is equal to the computer's number than the player wins and the game ends. After each guess, the computer should generate an appropriate message, like "Your guess was too high" or "Your guess was too low".
* The player will be allowed six guesses. If they exceed six guesses then the computer wins and the game ends. The computer is permitted to gloat.
* The game screen should let the player know whether they have won or lost and what the random number was they were trying to guess (to prove the computer wasn't cheating, of course). ***NOTE:*** *Reasonable (understandable and helpful) messages to the user will be part of your project grade.*
* Comment the JavaScript code to describe how it is working. ***NOTE:*** *Adding informative or helpful comments will be part of your project's grade. This lets me know that you understand the code you are creating.* Don't just throw together 2 or 3 comments to meet the requirements; as a rule, I would expect 1 informative comment every 5-8 lines of code.
* You should use CSS so that the game is well formatted. You must also indent your HTML, CSS and JavaScript to make everything readable. Child elements should be indented inside parent elements, CSS declarations should be aligned vertically, the code inside JavaScript blocks should be indented, and so on. Indentation and script organization should also be consistent.***NOTE:*** *Reasonable and consistent indentation will be part of your project's grade.*
* The program should check to make sure that the user entered a number, rather than text or a blank field. If the user did not enter a number then the computer should reply with something like: "Please Enter a Number". You can decide whether this will cost the player a guess or not.  You should also test if the user entered a value within the proper range, and display an appropriate message if they did not.
* Utilize input text fields for all user input and utilize the **innerHTML** property to write all output to the browser window. Your program should NOT contain any popup window methods (*alert*, *prompt*, *confirm*) or document.write methods. You may leave in *console.log* statements for debugging purposes.
* The HTML and the CSS should pass W3C validation. If you are unable to have it pass validation then include the W3C errors with your submission. You do not have to pass JSLint or JSHint, but these sites are ***very***useful for spotting errors in your code.

**Submission:**

Zip up the files you created for the project and upload the zip file to the assignment submission area by the due date. Alternatively, you can upload the assignment files to your website and provide the website URL in the assignment submission area. Let me know if you have any questions or issues regarding this project; by now, you know how much effort is involved in completing projects so give yourself plenty of time. *To help out,* *I will also be providing tips and hints in the Discussion Forum and in class, so be on the lookout...*

**Rubric**

| PROG109: Number Guessing Game | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning Outcome Display of Start Screen  When the game opens in the browser, only the Start screen is displayed, which contains at least 1 image, game description, and button. Clicking the button hides the Start screen and displays the Game screen. | |  |  | | --- | --- | | 4.0 pts  Full Marks | 0.0 pts  No Marks | | 4.0 pts |
| This criterion is linked to a Learning Outcome Display of Game Screen  The Game Screen is only displayed once button in the Start Screen is clicked, and contains counter of guesses played (or guesses remaining), a textbox for user entry, a game play button, status messages showing the results of each guess, and game replay button (optional). | |  |  | | --- | --- | | 4.0 pts  Full Marks | 0.0 pts  No Marks | | 4.0 pts |
| This criterion is linked to a Learning Outcome Input and Output  Document methods are properly used to input user text into game and display output information. | |  |  | | --- | --- | | 4.0 pts  Full Marks | 0.0 pts  No Marks | | 4.0 pts |
| This criterion is linked to a Learning Outcome Generates random number  A random number in the range 1 to 50 is correctly generated. | |  |  | | --- | --- | | 1.0 pts  Full Marks | 0.0 pts  No Marks | | 1.0 pts |
| This criterion is linked to a Learning Outcome Evaluates user guess  Use conditionals to evaluate user input: 1) Was it a number? 2) Was it lower or higher than the random number? 3) Was it in the proper range? Does the program convey this information to the user? | |  |  | | --- | --- | | 2.0 pts  Full Marks | 0.0 pts  No Marks | | 2.0 pts |
| This criterion is linked to a Learning Outcome Game play and termination  Does the game end properly? This should happen when the user either guesses correctly or 6 attempts have passed. Does the program display an appropriate message for each situation? (see Project description). | |  |  | | --- | --- | | 3.0 pts  Full Marks | 0.0 pts  No Marks | | 3.0 pts |
| This criterion is linked to a Learning Outcome Interface design  Is CSS used effectively to style the Start/Game screens and their contents to make the game look appealing to the user? Use alignment, color highlighting, spacing, etc. to enhance your game's organization. | |  |  | | --- | --- | | 2.0 pts  Full Marks | 0.0 pts  No Marks | | 2.0 pts |
| This criterion is linked to a Learning Outcome HTML/CSS validation and Project Organization  Markup and CSS should be error free. Use W3C validators to identify and remove errors in the HTML and CSS. JavaScript should be properly formatted (proper use of variables, code blocks, loops, and so on). HTML, CSS, and JavaScript must be separated out into their own files. There should be no inline style attributes and no JavaScript code (like event attributes) in the HTML. | |  |  | | --- | --- | | 2.0 pts  Full Marks | 0.0 pts  No Marks | | 2.0 pts |
| This criterion is linked to a Learning Outcome Program execution.  The JavaScript program should run as expected. To avoid errors, check the code in the console window, verify with JSLint or JSHint, and test as many conditions as you can (are all input situations covered? does the game counter increment correctly? etc.). | |  |  | | --- | --- | | 4.0 pts  Full Marks | 0.0 pts  No Marks | | 4.0 pts |
| This criterion is linked to a Learning Outcome Project readability  The JS code should contain comments to explain how it is working. Reasonable indentation and spacing should be used on your HTML, CSS, and JavaScript to make your document and scripts readable. (see Project description for more details). | |  |  | | --- | --- | | 4.0 pts  Full Marks | 0.0 pts  No Marks | | 4.0 pts |
| Total Points: 30.0 | | | |