

Simple Hangman

Summary

Using HTML and JavaScript, create the functionality and UI for a simple Hangman game.

Make sure you have an understanding of JavaScript arrays, strings, the onClick event on an element, and how to use the innerHTML property of an element.

Specification

A Hangman game goes as follows:

1. A word is randomly selected from a set of words
2. The player is given a number of single-letter guesses to figure out the word
 1. A correct guess adds to the word and does not result in another body part being added to the person
 2. An incorrect guess increments the number of misses and adds another body part to the person
3. The game is over either when the player has successfully guessed the word or has missed too many times (typically seven times in an average game)
4. The body is constructed at the following points:
 1. One miss: One rope piece
 2. Two misses: Two rope pieces
 3. Three misses: Two rope pieces, head
 4. Four misses: Two rope pieces, head, neck
 5. Five misses: Two rope pieces, head, neck, torso (chest and arms)
 6. Six misses: Two rope pieces, head, neck, torso, one leg
 7. Seven misses: Two rope pieces, head, neck, torso, two legs

User Interface

The user interface should contain the following:

1. A set of buttons (one letter each, A-Z) that, when clicked, send a letter guess to the game logic
2. A Start New Game button that resets the game after a win or loss and begins a new game
3. A display of the current state of the body (can be ASCII art, actual images, or a plain-text description of the body) that changes based on the number of misses; maxing out the misses should create a full body
4. A display of the misses left to make
5. The current word with only the correctly-guessed letters shown; all other characters should be underscores or dashes

I'm not looking for anything stellar for the UI but just make sure all of the components are there.

Functionality

1. The game should begin when the page is loaded
2. Each letter guess should add to and check the set of guessed letters against the selected word
 1. If the letters all match, the game has been won
 2. If the guess is not part of the word, add a miss
3. Each guess (right or wrong) should update the user interface appropriately (all correct letters shown in the word, an image representing the state of the body, number of misses left, etc)
4. Missing out the number of available misses loses the game
5. When the game has been won OR lost, display an appropriate message
 1. If the game has been lost, however, also display the word that was selected so the player knows what the win condition would have been

Grading

This project is worth 20 points, weighted at 10% of the grade. Implementing all of the functionality is considerably more important than how the UI looks.

I'm going to be using Chrome to run everything since Firefox can be funky sometimes.

Extra Credit

Clicking on the letter buttons should hide them (set the "visibility" CSS property). On game start, all letter buttons should be shown as well.

The extra credit is worth 2 points.