

## **Monster Game Project**

**Instructions:** You will work individually to create a “Monster Game” website.

**Submission Guidelines:** You must upload a zipped archive containing the following items:

- monster\_game\_project\_YOURLASTNAME.html
- All image files (and any other files) needed for your website to behave appropriately
- A 1-minute (or less) screencast (video file) of you playing your game. Play at least two games in the video to show that it is possible to win/lose the game.

### **Required Features:**

1. Standard HTML and CSS elements (a title, custom CSS styling, a header)
2. Descriptive code comments (including a comment before every function, code citations when appropriate, and citations for all images used)
3. At least 5 monster names and images
4. At least 5 hero names and images
5. Store important data in global variables:
  - a. The Hero's current health
  - b. The Monster's current health
  - c. The number of monsters defeated
  - d. The index of the currently-selected monster
  - e. The index of the currently-selected hero
  - f. Two parallel arrays of hero names and image filenames
  - g. Two parallel arrays of monster names and associated image filenames
6. A select box that allows the player to change avatars at any time during the game
7. Appropriate <div> elements that allow you to display updated information on the screen:
  - a. Counters that keep track of the Hero Health, Monster Health, Number of Monsters Defeated
  - b. Descriptive messages that give the player information about the game (how much damage the hero dealt, how much damage the monster dealt, whether the player has defeated the monster, and whether the player has won/lost the entire game)
  - c. A button that has the correct value and onClick event for the current state of the game
    - i. Example: “Fight!”, “Next Monster”, and “Play Again”

### **Webpage Behavior:**

1. When the page loads (and each time you defeat a monster) a random monster is selected and its associated image is displayed.
2. The Hero and each new Monster starts with 100 health.
3. Each time you click "Fight," the hero **randomly** deals between 10 and 17 damage (inclusive) to the Monster and the Monster **randomly** deals between 1 and 4 damage (inclusive) to the Hero.
  - A descriptive message should appear on the screen summarizing the battle. For example, the message might say something like "Batman has dealt 11 damage, and the Joker has dealt 3 damage." If you click "Fight!" a second time, the message might say "Batman has dealt 15 damage, and the Joker has dealt 5 damage."
  - Each time you click "Fight," all counters on the screen for Hero and Monster health should be updated for each round of battle.
4. When the Hero defeats a monster, show a message on screen showing that the monster was defeated and update appropriate counters.
  - Give the main button a value of "Next Monster" and give it an appropriate event handler.
5. When the Hero defeats 5 monsters, tell the player that he/she won the game.
  - Give the main button a value of "Play Again" and give it an appropriate event handler.
6. When the player runs out of health before defeating 5 monsters, tell the player that he/she lost.
  - Give the main button a value of "Play Again" and give it an appropriate event handler.
7. The "Play Again" and "Next Monster" buttons should generate a new random Monster, update global variable values, and update displayed information as appropriate.

### **Video Demo:**

Here is a video of the basic behavior your website should exhibit: <https://youtu.be/hiDUPd8AgZI>

*Note: Page layout, CSS, images, etc should match your personal tastes. Don't just reproduce what you see in the video!*

### **Optional Enhancements:**

1. Easy/Hard Mode: Allow the user to switch between Easy Mode and Hard Mode. These modes should appropriately modify the range of values for the damage the Hero and Monster can deal. Note that this enhancement might make it easier for you to test your program and may help you keep your screencast video under a minute.
2. Score: Display a counter tracking the player's "Score." You can determine how to give a player points for his/her score, but it should probably be based on things such as the total amount of damage the Hero has dealt, the number of monsters the Hero has defeated, or how many hits it took the Hero to defeat a particular Monster.
3. High Score: Display the highest score a player has earned since the page was last refreshed.
4. You may add additional enhancements to your game if desired.

### **Code of Conduct Reminder:**

While collaboration is always encouraged, plagiarism is never allowed. This is an individual project. Please refer to the Code of Conduct for more information regarding course expectations and consequences for violating these expectations.