CPS630 – Web Applications

Lab 1

In this lab, you will develop a classic Battleship game using HTML, CSS, and JavaScript. The game will feature a two-player mode: one player will be the user, and the other will be the computer (PC). Your task is to create an interactive, browser-based version of this timeless game, implementing game logic and user interactions. This lab must be done individually.

Sample games (for reference only):

- http://en.battleship-game.org/
- https://www.battleshiponline.org/
- https://cliambrown.com/battleship/play.php

Minimum requirements:

1. Game board: Two grids (10x10) representing the ocean, one for the user and one for the PC.
2. Ship Placement with Drag and Drop: Players should be able to place ships on the grid.
3. Valid Ship Placement Check: The system must verify that ships can be placed in the chosen position (e.g., ensuring the space is not already occupied).
4. Real-time Feedback for Bomb Placement: Provide immediate feedback on whether a bomb placement is valid or invalid.
5. Hit or Miss Indication: Indicate whether a bomb hit a ship or missed, such as by changing the cell color or adding icons.
6. Updating Player's Board: After each turn, update the player's board to show which areas have been bombed, both hits and misses.
7. Updating PC's Board: Similarly, update the PC's board with the player's bombing actions.
8. Game Mechanics: Enable turn-based gameplay where the user and the PC take turns selecting a target cell in the opponent's grid to attack.
9. Destroyed Ship Highlighting: When a ship is completely destroyed, highlight it distinctly on the board to indicate its destruction.
10. Game Progress: Keep track of and display the number of hits and misses. The game ends when all the ships of a player are sunk.
11. End-of-Game Bombing Phase Logic: Include logic to prevent further bombing once a winner is declared (perhaps with a message box on the page).
12. After the game ends, there should be a button to start a new game without refreshing the browser.
13. User Alerts for Invalid Actions: Provide clear alerts or messages for invalid actions, like attempting to place a bomb in an already bombed location.

Bonus marks:

1. Optional Sound Effects for Bombing Actions: Add distinct sound effects for bombing actions, with different sounds for hits and misses.
2. Enhanced UI with Animations and Imagery: Implementing an aesthetically pleasing UI with animations for bombing actions, images for ships, and other visual enhancements will qualify for additional marks. This is an opportunity to demonstrate creativity and technical skills in UI design.
3. Note: Lab grade cannot exceed 100%.

Deliverables:

1. A complete set of HTML, JavaScript, CSS, and any other used files.
2. A video demonstrating the game (five minutes in duration maximum) in .mp4 format.
3. Zip all your files (source code + video demo) in a folder named as follows: YourLastName_YourFirstName_YourStudent# and upload to D2L.
4. **Your submission must be complete according to the guidelines above. Failure to do so will result in a mark of Zero on the lab.**

Evaluation Criteria:

1. Functionality: Does the game work as expected without bugs?
2. Code Quality: Is the code well-organized, commented, and following best practices?
3. Creativity and Design: How visually appealing and user-friendly is the game interface?

Submission Deadline:

4. **Monday, February 12, 2024 @ 11:59 pm (no extensions)**