**CSE2WDX Assessment 2: Objectives Report**

Student ID: 19390245

Full name: Justin Patterson

|  |  |  |  |
| --- | --- | --- | --- |
| Objective | Example | Location | Description (1-2 sentences) |
| A.1: Create the document structure by using HTML | 1 | yahtzee.html:14,29,234 | **Structure the UI by using semantic markup such Header, Footer:**  Used <header>, <main>, <footer> tag for some basic semantic structure. |
|  | 2 | yahtzee.html:70,143 | **Create a layout container in HTML:** The <div> tag is a layout container for the two <table> tags within. |
| A.3: Apply styling to HTML elements programmatically | 1 | yahtzee.js:106 | **Apply a transform:** the die images go from 1 to 6 then the go to the die image that comes from an array of random numbers |
|  | 2 | yahtzee.js:26 | **Show alert after dice have been rolled 3 times:** an alert pops if dice are tried to be rolled more than 3 times asking the user to submit a score |
| A.5: Establish the scope of objects and variables | 1 | yahtzee.js:11 | **Keep objects out of the global namespace:** The JavaScript file uses an immediately invoked function to keep objects out of the global namespace. |
|  | 2 | yahtzee.js:70,142, 221,  301,303,304,330 | **Use the “this” keyword to reference an object that fired an event:** The “this” keyword is used to reference the button that fired the “click” event. |
| A.6: Create and implement objects and methods | 1 | yahtzee.js:80, 119 | **Implement native objects:** The native “Math” object has been implemented to support random number generation for choosing the outcomes of dice rolls. |
|  | 2 | yahtzee.js 11,21,59,77,  94,102,118,125,130,138  149,156,164,174,187,197  229,237,243,254,277,300,  310,324,336,344,351 | **Implement native methods and create custom methods:** Custom methods have been created many times to modularise the JavaScript code. |
| B.1: Implement program flow | 1 | yahtzee.js:33 | **Iterate across collections and array items:** The jQuery “foreach()” method is used to iterate across a collection of int to create a random number for each int in the collection |
|  | 2 | yahtzee.js:246 | **Evaluate expressions:** A lengthy expression is used to evaluate each of the “small straight” and “large straight” dice combinations. |
| B.2: Raise and handle an event | 1 | yahtzee.js:23,139,198,345 | **Handle common events exposed by DOM (.on()):** The “click” event is handled in the code for each of the buttons. |
|  | 2 | yahtzee.js:23,139,198,345 | **Handle an event by using an anonymous function:** An anonymous function has been used to respond to the click event. |
| D.1: Style HTML text properties | 1 | yahtzee.css:8,9,25,26 | **Apply styles to text font:** The font family and font weight has been changed for the whole page. |
|  | 2 | yahtzee.css:218,223 | **Apply styles to text alignment, spacing, and indentation:** The alignment of the “points” data has been set to right alignment. It is good practice to have numeric fields aligned to the right. |
| D.2: Style HTML box properties | 1 | yahtzee.css:120 | **Apply styles to alter appearance attributes, including size, rounded corners, padding, and margin:** The size and margins of the dice pictures have been adjusted. |
|  | 2 | yahtzee.css: 114,120,  133,140,148, 317,322 | **Apply styles to establish and change an element’s position:** The dice have the “relative” position set so that they can be moved around via an animation when maximum rolls have been reached. |
| D.3: Create a flexible content layout | 1 | yahtzee.css:187,192,  197 | **Implement a layout using a flexible box model:** The flex model has been enabled for the container hosting the “upper score” and “lower score” tables allowing them to sit side by side. |
|  | 2 | yahtzee.css: 96 | **Implement a layout using grid alignment:** The grid model has been used for the footer validation images to have them side by side and centre of the page |
| D.4: Create an animated and adaptive UI | 1 | yahtzee.css:263,273 | **Animate objects by applying CSS transitions:** A transition to the hover and active style has been implemented for all buttons. |
|  | 2 | yahtzee.css: 59, 67 | **Apply 2-D transformations:** The die in the header rotate 360 degrees via an animation. |
| D.5: Find elements by using CSS selectors and jQuery | 1 | yahtzee.js: 338 | **Choose the correct selector to reference an element:** Using the jQuery global function “$("#total-score ")” correctly references the HTML tag with the ID of “total-score”. |
|  | 2 | yahtzee.js: 79 | **Find elements by using pseudo-elements and pseudo-classes:** The he pseudo- class “:checked” is used in the jQuery code to find die that have not been checked so they can be rerolled |
| D.6: Structure a CSS file by using CSS selectors | 1 | yahtzee.css: 209, 217,  202, 245 | **Reference elements correctly:** By saying “td, th”, we are referencing all <td> and <th> tags at the same time. |
|  | 2 | yahtzee.css: 279 | **Style an element based on pseudo-elements and pseudo-classes:** The :disabled pseudo-class is used on submit buttons once they have been clicked |