To: CS312

From: Kayden Vicenti

Date: June 9th, 2024

Subject: Homework Assignment 1: Dice Challenge

This report covers the deliverables related to Homework Assignment 1: Dice Challenge, starting with the problem statement and requirements given, followed by supplementary research and additional supporting information, and finally the implementation and testing of the code. All these steps are laid out in this memorandum to highlight the value and usage of the engineering design process for this project. Attached are the relevant documents for this assignment:

- Dicee Challenge - Starting Files
    - Images Folder 1-6
    - .DS_Store
    - dicee.html
    - styles.css

Introduction:

The purpose of this assignment is to practice the implementation of HTML, CSS, and JavaScript basics. Utilizing the given assignment starter files, I developed a program that randomly selects a number for two dice, compares the values, and selects a winner.

Project Requirements and Constraints:

Requirements:

- Independent Attempt:
    - Solving challenge by creating own starter files
- Utilize different approaches to programming concepts
- Understand logic and use of implementation steps, apply to own code

Constraints

- None

Research/Methodology:

For this assignment, I had to research in-depth about different functions and their usage within JavaScript. Some of these functions include `querySelectorAll, setAttribute, querySelector.`

Simulation and Testing:

After doing some background research, it was time to implement some of the functions into my code. For this specific assignment, I realized that you could either do a <script> tag to implement the use of javascript in the given HTML file, however, I chose to create a .js file. The next step was to generate a random number between one and six, and I accomplished this by using this line of code:

```
Math.floor(Math.random()*6)+1;
```

Math.floor, rounds to the nearest whole number, Math.random returns a number between 0 and 1, however, I needed to multiply this by 6, and using math.floor round to the nearest whole number. With this approach, there was an issue because math.random includes the number 0, and there aren't any dice that use that number so finally I had to add one to the line of code so the final range would be between 0 and 6. I used this approach to generate a random number for dice one and two and finally compared the two numbers to generate the appropriate display message. To do this I used this other line of code:

```
document.querySelector("h1").innerHTML = "Player 1 Wins!";
```

QuerySelector searches for the "h1" tag in the HTML file to replace it with the appropriate message. Additionally, the QuerySelectorAll method is used to select elements with the "img" tag. Since there are two of those elements, I had to specify which element to use by doing this step:

```
querySelectorAll("img")[0]
```

The thought process behind this was for me to think of the elements selected, to be put into an array by the order they are found. After that line of code, I had to replace the img elements src with the appropriate file, which was determined by the random number generator. At the beginning of the .js document, I chose to create an array that included the image paths. Later on, when replacing the images with the randomized image, I ran into some issues, that being that since the range is between 1 and 6, the array cannot account for the extra value being 6. So, if the generated number was 6, the sixth element in the array does not exist. To fix this issue, I had to do additional research and learned that I can use NULL as an element in the array. This fixed my array's undefined value and I was able to generate the final product:

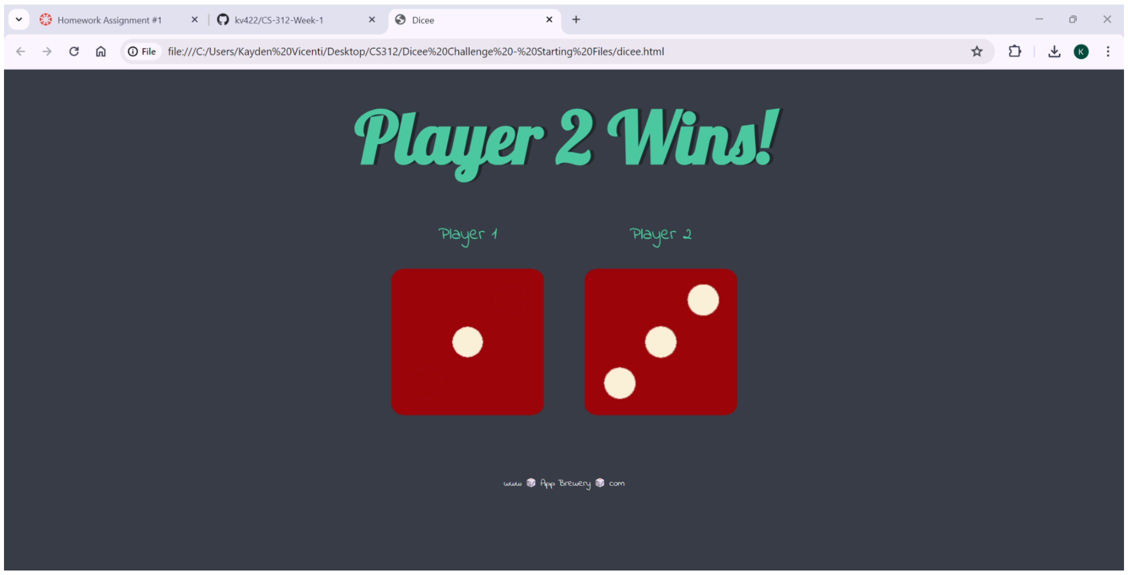
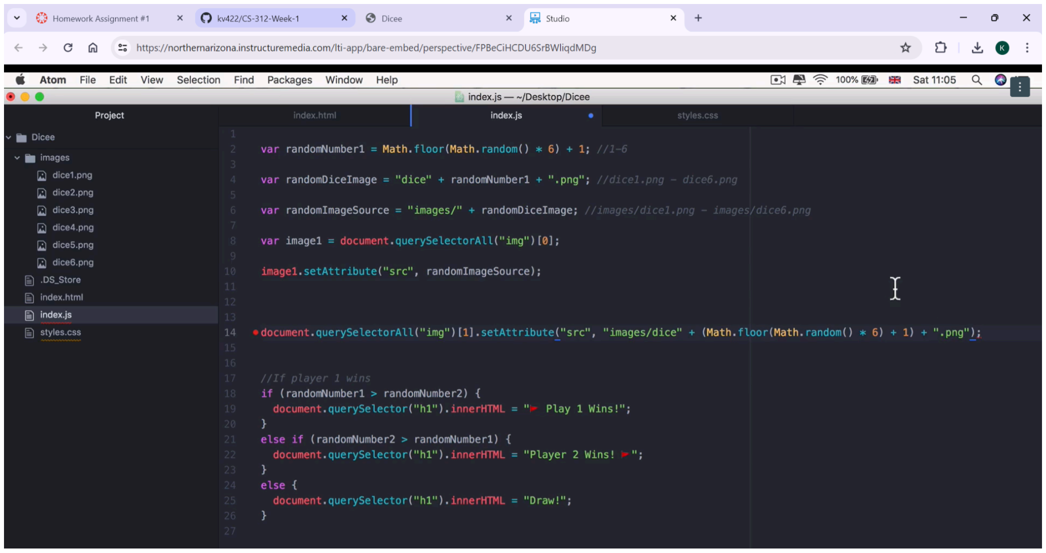

Homework Assignment #1
kv422/CS-312-Week-1
Dicee
file:///C:/Users/Kayden%20Vicenti/Desktop/CS312/Dicee%20Challenge%20-%20Starting%20Files/dicee.html
Player 2 Wins!
Player 1
Player 2
www App Brewery com

Figure 1: Local HTML File.

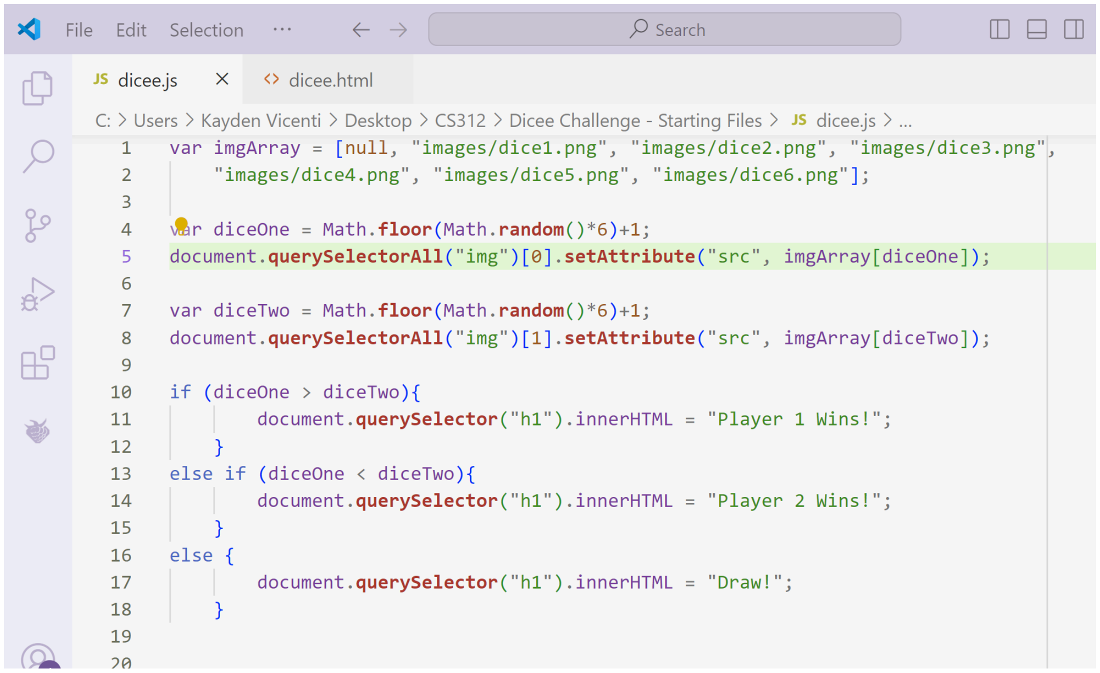
<u>The Design Process:</u>

From this assignment, I was able to complete the challenge. After doing so, I decided to compare my code with the provided solution video. I had not thought about using the concatenation approach to selecting a dice, I think this would work out well if there was a dice with a large number range. Additionally, this would allow images to be excluded from an array and to fix and ultimately eliminate the use of NULL within the array. Finally, the solution video also had code to be written in one line of code, however, I like to break down my code until I understand fully what my code is doing.



```
var randomNumber1 = Math.floor(Math.random() * 6) + 1; //1-6

var randomDiceImage = "dice" + randomNumber1 + ".png"; //dice1.png - dice6.png

var randomImageSource = "images/" + randomDiceImage; //images/dice1.png - images/dice6.png

var image1 = document.querySelectorAll("img")[0];

image1.setAttribute("src", randomImageSource);



document.querySelectorAll("img")[1].setAttribute("src", "images/dice" + (Math.floor(Math.random() * 6) + 1) + ".png");


//If player 1 wins
if (randomNumber1 > randomNumber2) {
  document.querySelector("h1").innerHTML = "🚩 Play 1 Wins!";
}
else if (randomNumber2 > randomNumber1) {
  document.querySelector("h1").innerHTML = "Player 2 Wins! 🚩";
}
else {
  document.querySelector("h1").innerHTML = "Draw!";
}
```

Figure 2: Video Solution Code



C: > Users > Kayden Vicenti > Desktop > CS312 > Dicee Challenge - Starting Files > dicee.js > ...

```
var imgArray = [null, "images/dice1.png", "images/dice2.png", "images/dice3.png",
    "images/dice4.png", "images/dice5.png", "images/dice6.png"];

var diceOne = Math.floor(Math.random()*6)+1;
document.querySelectorAll("img")[0].setAttribute("src", imgArray[diceOne]);

var diceTwo = Math.floor(Math.random()*6)+1;
document.querySelectorAll("img")[1].setAttribute("src", imgArray[diceTwo]);

if (diceOne > diceTwo){
        document.querySelector("h1").innerHTML = "Player 1 Wins!";
    }
else if (diceOne < diceTwo){
        document.querySelector("h1").innerHTML = "Player 2 Wins!";
    }
else {
        document.querySelector("h1").innerHTML = "Draw!";
    }

```

Figure 3: My Coded Solution.