Caffeine Machine

Version 1.2

Prepared By

Sarah Miles

4/16/18

Table of Contents

List of Figures

Record of Changes

*A-Added M-Modified D-Deleted

CHANGE NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	*A M D	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER
1.0	4/15/18		A	Introduction, Functional Requirements and Materials and Resources	
1.1	4/30/18		A	Design Overview and Detail Design added	
1.2	4/30/18	Figures 1, 2, 3	A	Addition of figures	

Project Design Document

1 INTRODUCTION

This project presents an interactive data visualization of the amount of caffeine that a person consumes daily compared to the daily recommended maximum, or user-set maximum. Its intent is to inform the user of the scale of a person's caffeine intake and the relative amounts of caffeine in products, so that they may make more informed choices. It is visualized as a coffee cup filling up for each caffeine input the user selects. It is not intended to be an exhaustive list of all caffeinated products, but a learning tool based off common ones.

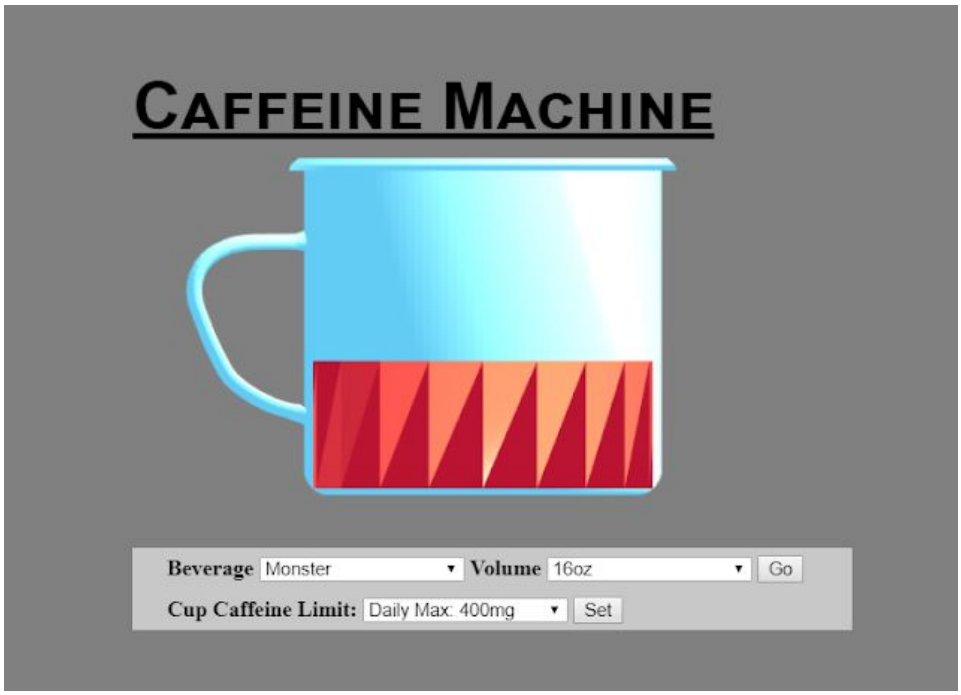

Figure 1: Example of display

2 FUNCTIONAL REQUIREMENTS

2.1 General

2.1.1 Coffee cup model is displayed in 3D

2.1.2 Caffeine amounts are visually displayed proportionate to cup model's volume

2.2 User Interaction

2.2.1 User may select beverages to add to cumulative caffeine amount

2.2.1.1 User may select beverage volume in ounces

2.2.2 User may set "volume" amount of cup model

2.2.2.1 All caffeine inputs will scale accordingly, while the model stays the same size

2.2.2.2 The default maximum (set to the maximum recommended daily caffeine intake) will serve as an upper volume limit

3 Materials and resources

3.1 Non-functional Requirements

3.1.1 The program will be written with JavaScript, HTML, and CSS

3.1.2 A 3D model of a coffee cup will be imported

3.1.3 Maya will be used to create a cylinder

3.1.4 The program will run on modern web browsers, including mobile platforms

4 Design Overview

4.1 Introduction

The program is based off of the WebGL library, and uses the additional imported libraries MV.js, webgl-utils.js, and initShaders.js to supply efficient rendering tools. A free cup model is used for the coffee cup, and the cylinder was originally created in maya. Caffeine amounts are hard coded per ounce to the beverage, and calculated upon "Go" being pressed. This amount corresponds to the height of the cylinder, to give different selections visually comparable volumes. The user may set the cup's volume through a dropdown menu, and select "Set" to update the value.

5 Detailed Design

5.1 Object Rendering

To obtain the cup displayed in the program, a free .obj format model was downloaded, then imported into MeshLab and exported as a .off file. This .off file was then run through a file parser program created particularly for the purpose of formatting the data into vertex and face arrays. These arrays were then uploaded to gitHub, then modified through rawgit.com to make the arrays cdn compatible for jsFiddle.

Once loaded into the fiddle as resources, the vertex and face arrays were run through createMesh(), a function that compiled them into a mesh and calculated their normals. These mesh and normal arrays are then allocated space in the GPU with bufferData(). Finally, render() is called on the model, and it is sent to the GPU to be displayed.

The cylinder was created in Maya and its vertex and face arrays extracted, buffered, and rendered through the same means.

The lightingModel() and transformation functions are called before the creation of both the cup and cylinder, as their values are modified separately for the objects. Both object's color is derived from the lighting material component.

5.2 Caffeine Calculation

The caffeine amounts are hard-coded into selection boxes in the HTML. For each beverage, the selection value is set to the pre-calculated amount of caffeine per ounce.

```
Beverage:
<select id="beverage">
    <option value="0">Select</option>
    <option value="16">Dark Roast</option>
    <option value="19">Medium Roast</option>
    <option value="22">Blond Roast</option>
    <option value="10">Iced Coffee</option>
    <option value="13">Cold Brewed Coffee</option>
    <option value="75">Espresso</option>
    <option value="2">Black Tea</option>
    <option value="2">Green Tea</option>
    <option value="6">Chai Latte</option>
    <option value="4">Green Tea Latte</option>
    <option value="3">Starbucks Refresher</option>
    <option value="6">Coffee-Based Frappucino</option>
    <option value="10">Redbull</option>
    <option value="10">Monster</option>
    <option value="3">Coca-Cola</option>
    <option value="100">5 Hour Energy</option>
</select>
```

Figure 2: HTML code outlining selection choices and values

When the "Go" button is pressed, the program calls the calculateCaffeine() function to multiply the volume by the caffeine per ounce of the selected beverage. This output value is then used to create a cylinder via the createProportionateCyl() function. If the calculated caffeine amount is over the set

maximum, no cylinder will be rendered, instead, a message will appear showing the amount of milligrams over the limit of the selection. If a cylinder was rendered previously, it will continue to display.

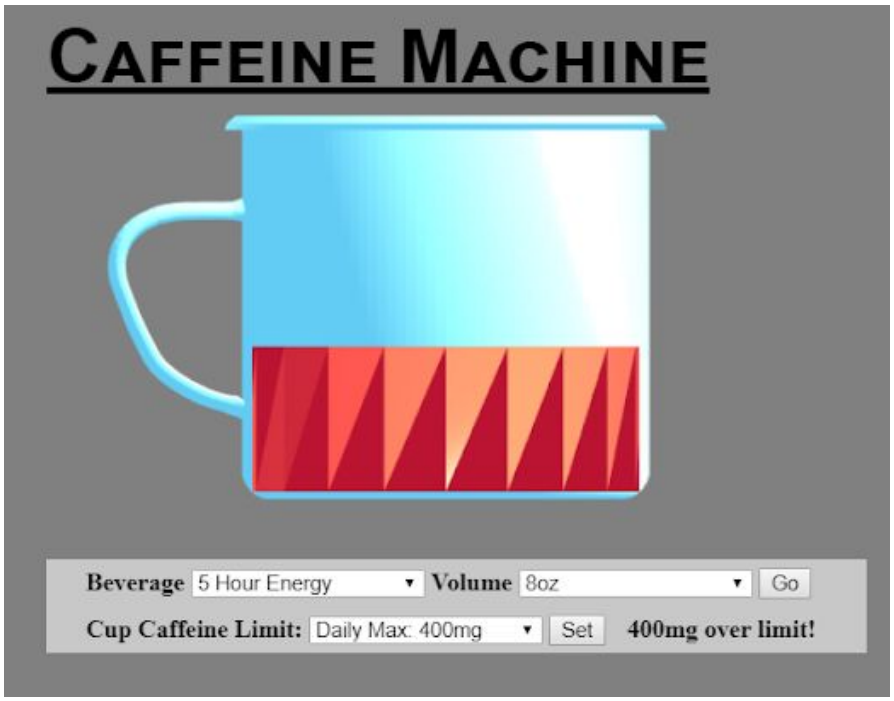

Figure 3: Demonstration of overflow of limit message

5.3 Interface

Dropdown menus were chosen for ease of use and understanding of function. A “Go” button functions to initialize the creation of a cylinder based on the specified inputs. Having the function execute on the press of a button enables the dropdowns to be selected in any order and perused before a decision is made. The “Set” button that follows the Cup Caffeine Limit dropdown does not have the same functional necessity but was included for consistency of interface.