PUI Assignment 8:
Implementing a Web Portfolio/Tool Project

December 6th, 2019
Nathan Barnhart

Part 1: About This Website

For assignment 8, I chose to create a website that gives a brief introduction to the Three.js JavaScript library / API.

The purpose of my website is to introduce people to the capabilities of this library, and the power it derives from its relationship with the WebGL API.

In addition to demonstrating the API through a 3D model rendered on the website, information regarding its origins as well as locations for further information are provided.

The primary interest and engagement will come from the model itself. The model is already rotating, and if the visitor clicks/touches the model, they will discover they are capable of orbiting around the model, zooming in and out, as well as panning.

The target audience might include people such as PUI students just beginning to learn about the capabilities of JavaScript.

It might also be equally interesting to those interested in any type of web applications to present 3D content for marketing, video games, interactive art, etc.

Part 2: Website Interactivity

- Upon arriving at the website, users will most likely immediately notice the slowly rotating 3D cube in the center of the page.
- If they click or touch the cube, and they will discover they can orbit their view around the cube.
- They will also notice a button towards the top of the page that says, “Learn About Three.js”
- When they click on this button, a modal window appears.

- The window presents information about Three.js and also some suggestions for how to interact with the rotating cube.
- The modal window also offers a number of links that if clicked on will open a new browser window to take them to helpful resources for learning more about Three.js.
- Lastly this modal window can be closed either with the “X” at the top right, or by clicking the “close” button at the bottom right.

Part 3: External Tools Used In Addition to HTML/CSS/JS

- Bootstrap CSS Library
  - I chose to use Bootstrap primarily to help facilitate a responsive mobile-first design. I also wanted the convenience of only needing to add class modifiers in order to make changes to the CSS.
  - Bootstrap was especially helpful with styling and locating the “Learn About Three.js” button. I also relied on Bootstrap to create a modal window.
  - Bootstrap added a consistent and clean look to my website. That uniformity will lend to the usability of the website.
- Three.js JS Library / API
  - I chose to use Three.js because I wanted to create and present a high quality 3D rendered object as the central focus of the website. Three.js offers numerous options for built-in geometries, cameras, rendering methods, and more.
  - I used Three.js to create a 3D scene, as well as to animate renderings of that scene. The primary mesh, a cube, was set to incrementally rotate on the x, y, and z axes for each frame. I also used Three.js’s built-in script to set up a control that allows the visitor to orbit the camera.
  - In my opinion, interactive 3D rendered objects are one of the most compelling types of interactions you can have in a web browser. I think anyone who sees this small website will be impressed with the quality of the rendering, and the smoothness of the animation. The ability to intuitive control the cube will add another level of delight to the experience.

Part 4: Iterations and Other Changes

In the original concept for my website, I proposed a functionality that would allow the site visitor to toggle between different ways of rendering views in Three.js, including WebGL, Canvas, SVG, CSS 2D, and CSS3D. After learning more about Three.js and WebGL, I realized that in the vast majority of cases WebGL is most likely to be the renderer of choice, and there is really no need to highlight the other options. Instead, I focused on creating an engaging and informative interaction with a simple 3D model, that would be accessible by browser on both a desktop computer as well as a mobile phone.

Part 5: Challenges Experienced

One specific challenge came up when I tried to add a model to my Three.js scene from a local directory and learned there are security restrictions on browsers loading files from local file systems to prevent people from stealing your personal information through malevolent scripts. (See: MDN: same-origin policy) As a result, I decided to focus on a form available through Three.js built-in primitive shapes and a texture map available through the internet.

Another challenge was how do I demonstrate an affordance for interacting with the 3D model without resorting to simply telling people what to do. I decided to try giving the model a subtle steady rotation in order to demonstrate its ability to move, however I also added some instructions in the text of the modal window.

Appendix: text from the website

Three.js

A Powerful 3D Creation Tool For Web On Any Device

More Info:
Three.js is a JavaScript library and Application Programming Interface (API). It was developed as a way to create and display animated 3D models in a web browser. This is most often done through another API called WebGL (Web Graphics Library). Because WebGL can make use of a computers graphics card (GPU), it offers much more rendering and animation power than typically available through HTML and CSS alone, however WebGL can be very complex. Three.js takes the power of renderers like WebGL, and minimizes their complexity, making them much easier to use.

The cube you see rendered on this website was created with Three.js. Try manipulating the camera's orbit by right and left click+drag, or with touch+swipe and pinch-to-zoom.

Three.js was originally released by Ricardo Cabello, April 24 2010, and its most current release is r111 made available November 27 2019.

The official website is: threejs.org
The GitHub repository is: github.com/mrdoob/three.js

Additional References:
Discover Threejs / web book by Lewy Blue
Wikipedia: Three.js
Wikipedia: WebGL
Cube Texture from Wikipedia: "Missing square puzzle"
Texture Map Image by Joaquim Alves Gaspar, and used according to CC BY 2.5