Emerging Technology Demo - Creating 3D Animation with Three.js

Tony Tam

San Francisco State University

ITEC 830 - Sprint 2023

Instructor - Dr. Brian Beatty

March 15, 2023

Abstract

For students who want to learn computer programming, a simple google search yields plenty of free and paid courses. These courses teach motivated students the basics of programming languages such as Javascript, Python, and Java. Lessons from Khan Academy ("khanacademy.com/computing") blend the fun parts of programming with the basic syntax by offering students the possibility of building simple games such as a side scroller game Hoppy Beaver.

While a game with a hopping animal is entertaining for young children, that is hardly inspiring and does not match the creative potential that lives inside all of us. Unlocking that creative potential requires a tool or technology that has boundless possibilities to create. Creation is not just about starting from a blank slate, but also about looking at what other people

have created, how they created the work, and being inspired to reach beyond the boundaries of our own imagination when we see examples of what other creators have done.

I have been intrigued by an open-source project called three.js (Cabello) and I would like to show how this technology unlocks creativity by being free, open, discoverable, extensible, and inspiring.

three.js

From the GitHub page for three.js written by the founder: "The aim of the project is to create an easy to use, lightweight, cross-browser, general purpose 3D library." (*Mrdoob/three.js: JavaScript 3D Library.*, n.d.)

From a book on teaching three.js: "three.js is the world's most popular JavaScript framework for displaying 3D content on the web… This amazing library and the vibrant community that surrounds are all you need to create games, music videos, scientific and data visualizations, or pretty much anything else you can imagine, right in your browser, on your laptop, tablet, or smartphone!" (Blue, n.d.)

During my exploration of three.js, I found the vibrant community to be a key part of the success of the technology. They follow the footsteps of the three.js project by openly sharing the source code of what they have created. This is very valuable in learning because students can look under the hood of how 3D content was created. The founder's decision to build a cross-browser technology removes any barrier to entry for the 3D content consumer.

Free

The three.js source code is hosted on GitHub with the MIT license. (Cabello, n.d.) From the license file: "Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files..".

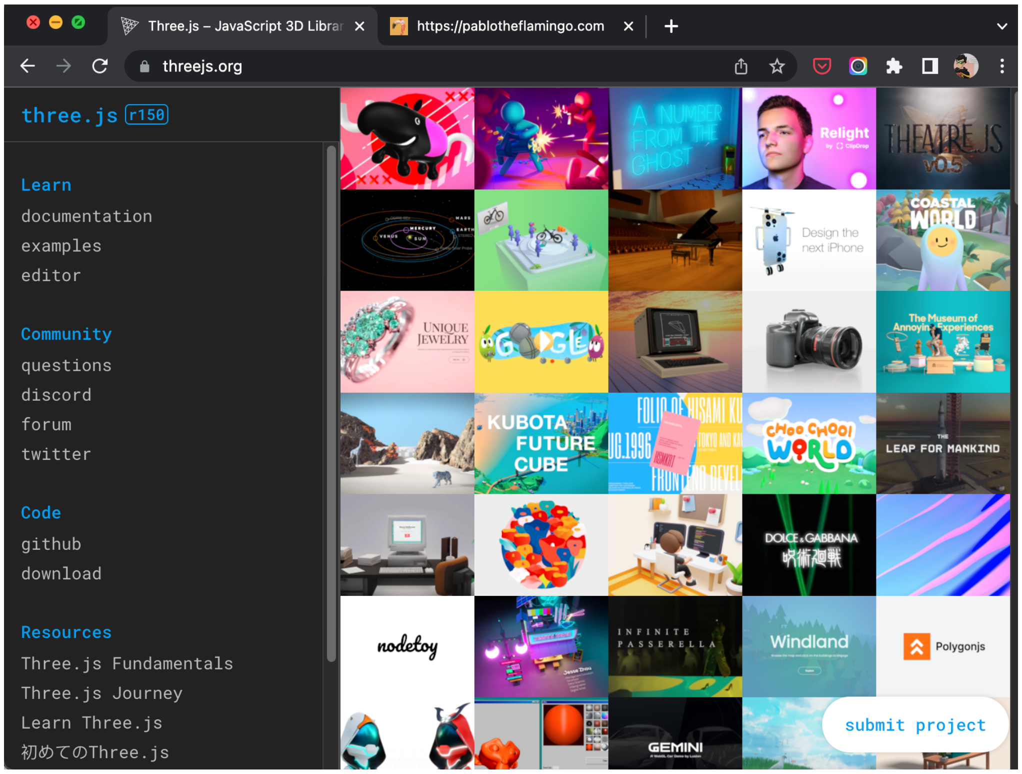
The source code has been 'forked' on GitHub by over 31,000 people, which means that many people have made copies of the source code and expressed some intent to "iterate on ideas or changes before they are proposed back to the upstream repository".

Open

The source code for three.js is open to the world to read and learn from. It is also open for other people to propose changes and contribute to changes. There are 11,245 closed issues which signify proposed changes that have been fixed. There are 13,712 closed merge requests, which are code contributions by the author or contributions from forks that have been merged into the source code. Over 100 people have contributed code as shown in the top 100 contributed dashboard on GitHub.

Discoverable

What makes three.js unique is being able to explore the awe-inspiring 3D examples that other people have built. There are 2 ways to explore. Going to the website https://threejs.org [Fig A] on the right side there are projects built by the community and you are encouraged to jump to a random tile



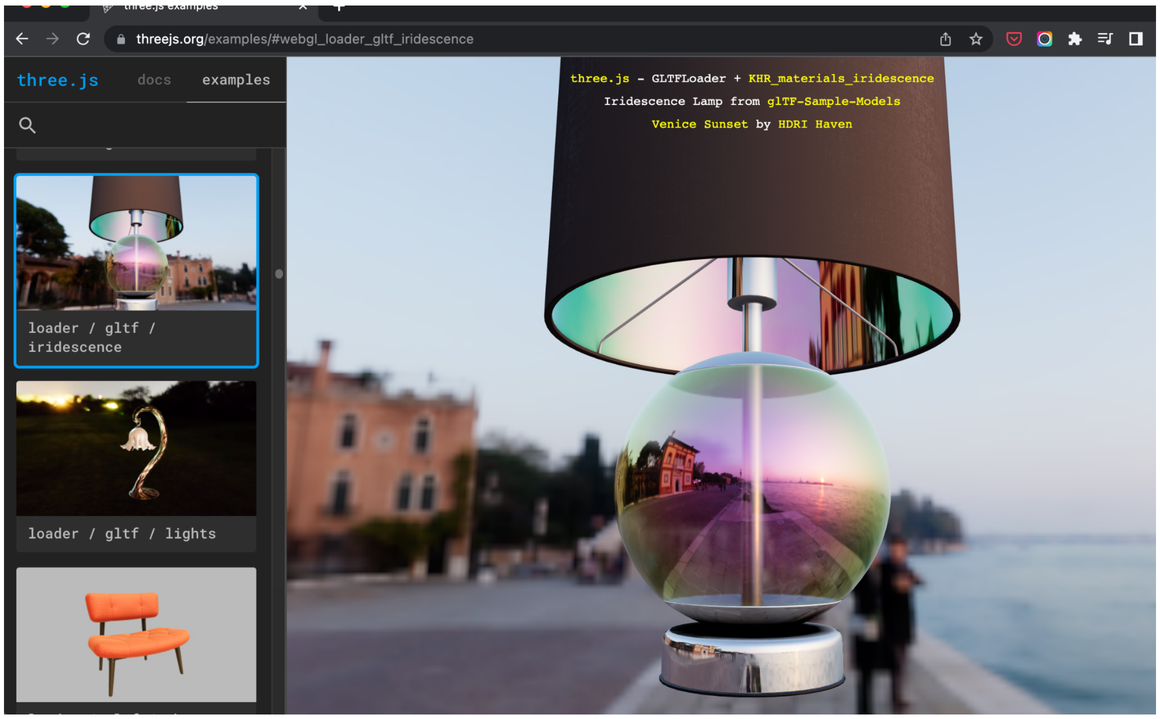
[Fig A] https://threejs.org Homepage

I clicked on a random tile and found a fun and [silly flamingo](https://pablotheflamingo.com) that I can drag its long neck and see the flamingo head bounce [Fig B]. If you leave the flamingo alone and turn on the music, it will bounce with the rhythm of the music.

[Fig B] Pablo the Flamingo https://pablotheflamingo.com/

For learning, the curated examples on three.js are the most helpful to navigate via scrolling but also to see the category of 3D models such as 'skinning', 'animation', 'blending', and 'clipping'. Most of the examples are interactive which spurs 'what if' scenarios. The most powerful part of the examples is the button with the '< >' symbol which allows you to open the source code to see how each example is implemented. This allows the viewer to take the example, download and make changes to it to understand how each demo example was built.

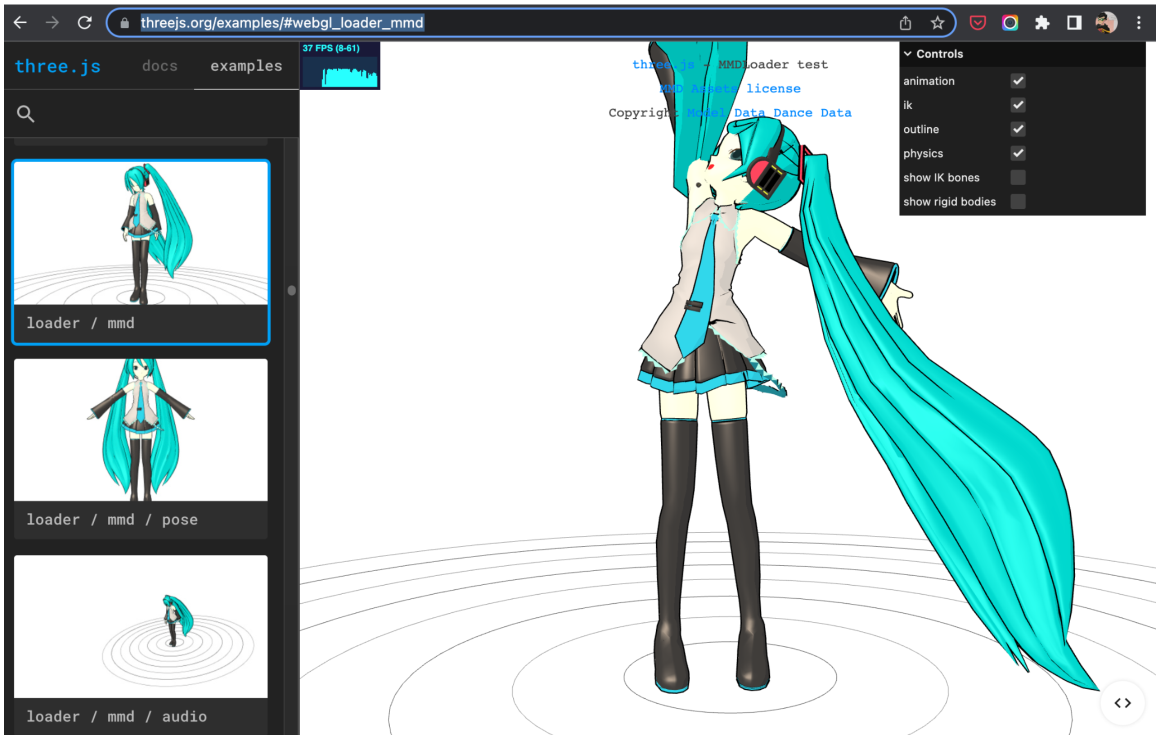
I browsed through at least 10 examples and would like to highlight the 2 examples. The first example is a very classical computer graphics rendering example, showing the power of lighting, material, and reflection models. It is called #webgl_loader_gltf_iridescence and can be viewed here [Fig C]. This demo shows off a transparent glass surface, how the light bends, and also the inside surface of the lampshade is reflective.


threejs.org/examples/#webgl_loader_gltf_iridescence
three.js
docs
examples
three.js - GLTFLoader + KHR_materials_iridescence
Iridescence Lamp from glTF-Sample-Models
Venice Sunset by HDRI Haven
loader / gltf / iridescence
loader / gltf / lights

[Fig C] Lamp with glass base and reflective cover

https://threejs.org/examples/#webgl_loader_gltf_iridescence

The 2nd set of demos is of a Japanese anime dancer who is dancing to preset music and movement. There are 3 demos of this character. The first is of her dancing and the viewer can control the viewing angle of the camera. The 2nd is where the viewer can move each part of her body and facial movement individually and the last demo is where the camera is at a predefined angle while watching the anime character dancing to a predefined set. The reader can open this link and interact with the demo example. https://threejs.org/examples/#webgl_loader_mmd



animation ✓
ik ✓
outline ✓
physics ✓
show IK bones
show rigid bodies

[Fig D] Dancing anime character rendered in 3D

https://threejs.org/examples/#webgl_loader_mmd

Extensible

The author of the three.js library realizes that what he created cannot anticipate everything that is needed in a 3D model library. He provides a plugin framework that has already has many extension points. Those plugins are listed here. Some notable extensions are Game AI, File Formats to be able to import and export to different 3D modeling tools, and Physics engines such as connon.js (The rigid body physics engine includes simple collision detection, various body shapes, contacts, friction, and constraints.)

Inspiring

three.js is a unique and visually inspiring project that doesn't need words to explain what it is. The website (threejs.org) is surprisingly short on words and the examples and gallery of projects built on three.js inspire the viewer to imagine what is possible. When the viewer is inspired and motivated to create, the hard work begins.

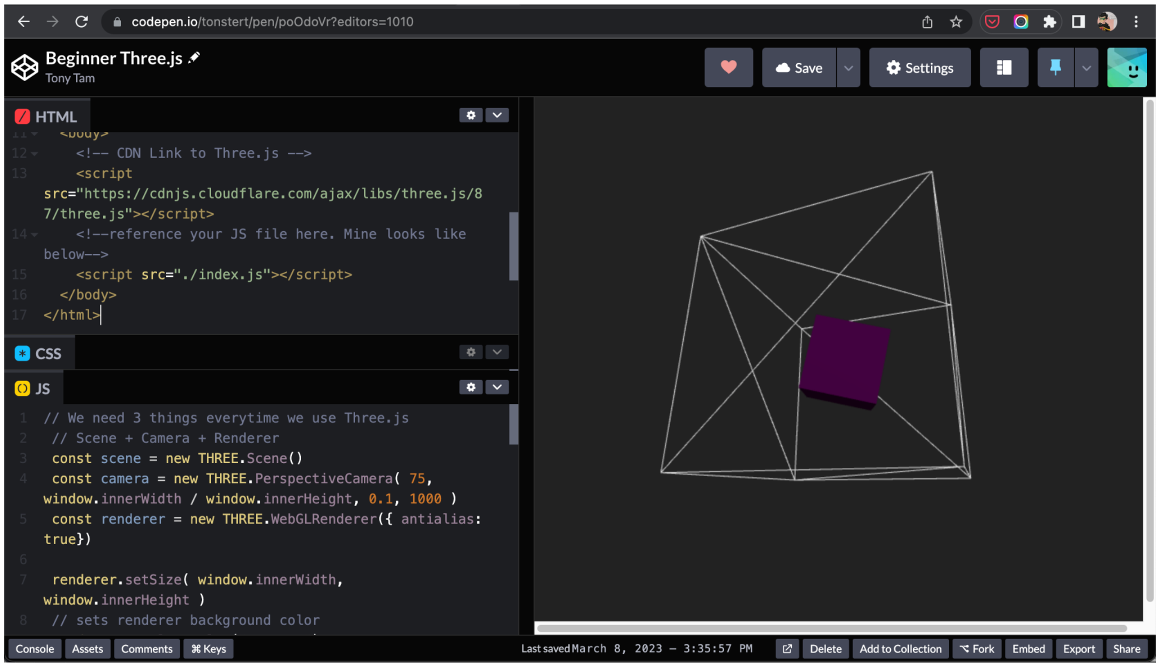
In the next section, I will summarize my journey to get started and make note of the challenges to get started with an active project that is constantly changing.

Getting Started

When I tried to follow along the tutorials from the official documentation, I ran into multiple issues with how complex it was to run a web server and get it to work locally on my computer. The official setup documentation (threejs.org) offered the flexibility of setting up three.js multiple different ways, it didn't offer instructions that lead to a successful installation for me even though I am a professional backend software engineer who is not well versed with frontend technologies.

Getting Started: Web Browser

The simplest and quickest way to get started is via this tutorial The Beginner's Guide to Beginning Three.js (Blue, n.d.) combined with codepen.io. The tutorial required only an HTML file and a Javascript file and can be done in codepen.io without using a file editor on your computer. The codepen.io URL which I copied the source code from the tutorial is at https://codepen.io/tonstert/pen/poOdoVr?editors=1011 [Fig E]



```html
  <body>
    <!-- CDN Link to Three.js -->
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/three.js/8
7/three.js"></script>
    <!--reference your JS file here. Mine looks like
below-->
    <script src="./index.js"></script>
  </body>
</html>
```

```javascript
// We need 3 things everytime we use Three.js
 // Scene + Camera + Renderer
 const scene = new THREE.Scene()
 const camera = new THREE.PerspectiveCamera( 75,
window.innerWidth / window.innerHeight, 0.1, 1000 )
 const renderer = new THREE.WebGLRenderer({ antialias:
true})

 renderer.setSize( window.innerWidth,
window.innerHeight )
 // sets renderer background color
```

[Fig E] Simple cube rotating with a mesh

Student Learner Exercise 1

As an instructor, a student exercise could be as follows for examples they find

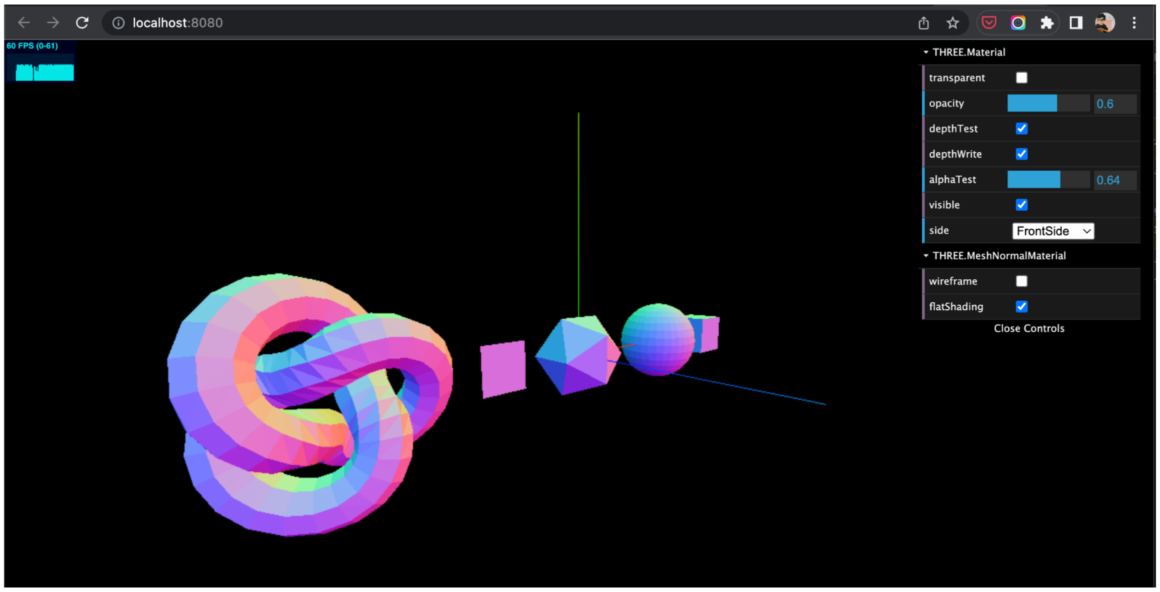
1) Follow the steps of the tutorial from (Blue, n.d.), and set up a codepen.io so that the students see the spinning cube.
2) Make at least 5 comments in the HTML and the Javascript files describing what each line and section is doing. Also, make comments where you don't understand what the code is doing and do some research online in the reference manual.
3) Make 3 changes to the Javascript files to cause some visual changes. These are 3 examples: Make the cube spin faster. Make the cube transparent, or the mesh is not transparent. Change the sizes of the mesh relative to the cube. Move the light source. Can the student identify the coordinate system and explain to the class where in space is the light source relative to the cube?

Getting Started: Locally On The Computer

A nonprimary source to set up three.js was much more helpful to get it set up locally. The website https://sbcode.net/threejs/introduction/ had a set of prescriptive steps if followed explicitly, helped me to get successfully up and running locally where I could use VSCode to make changes and previous my working code.

The advantage of working locally is to be able to use version control so we can save our work and come back to any known state. The other advantage is to be able to use local rendering files that are not available in tools such as codepen.io. I have also found more complex examples on the web only work locally without a lot of troubleshooting and changes.

For example, https://sbcode.net/threejs/meshnormalmaterial/ will not work in codepen.io https://codepen.io/tonstert/pen/JjaOowG?editors=1010 but it works locally on my computer without changes. [Fig F]



▾ THREE.Material

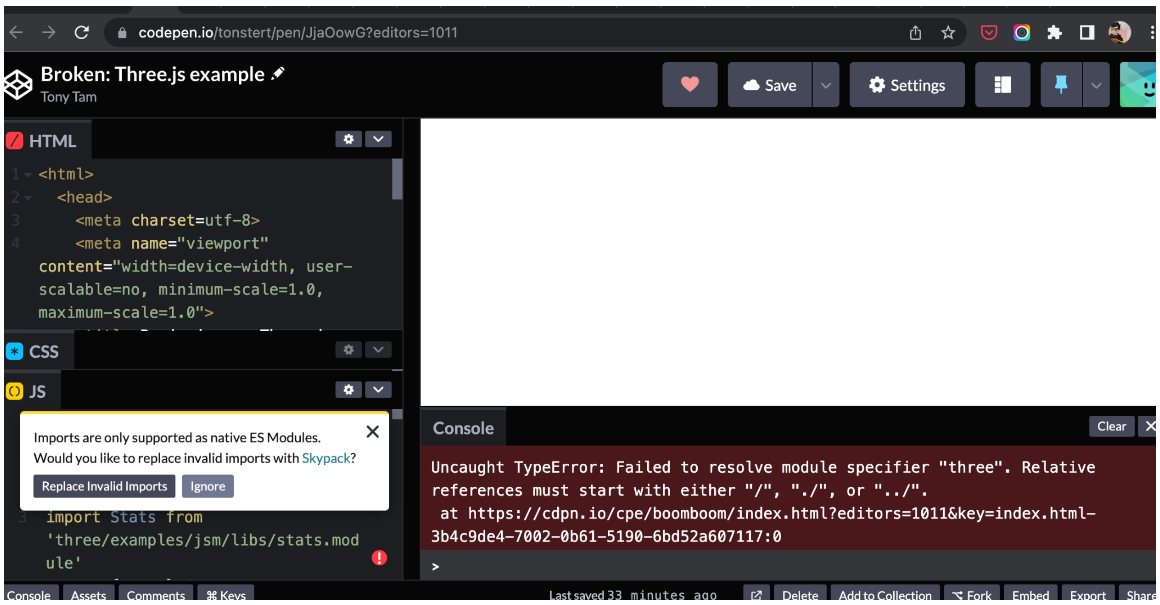
transparent	☐
opacity	0.6
depthTest	✓
depthWrite	✓
alphaTest	0.64
visible	✓
side	FrontSide

▾ THREE.MeshNormalMaterial

wireframe	☐
flatShading	✓

Close Controls

[Fig F] More complicated example of working locally



```
<html>
  <head>
    <meta charset=utf-8>
    <meta name="viewport"
content="width=device-width, user-
scalable=no, minimum-scale=1.0,
maximum-scale=1.0">
```

Imports are only supported as native ES Modules.
Would you like to replace invalid imports with Skypack?

Replace Invalid Imports | Ignore

```
  import Stats from
  'three/examples/jsm/libs/stats.mod
  ule'
```

```
Uncaught TypeError: Failed to resolve module specifier "three". Relative
references must start with either "/", "./", or "../".
 at https://cdpn.io/cpe/boomboom/index.html?editors=1011&key=index.html-
3b4c9de4-7002-0b61-5190-6bd52a607117:0
>
```

[Fig G] Complicated example not working in codepen.io

Conclusion

A flipped classroom model would work very well for a three.js 3D animation class. For students to begin to learn three.js, a foundational knowledge of Javascript is necessary as basic high school geometry. An instructional plan for students to learn 3D animation using three.js can be based on the Discover three.js free online book (Blue, n.d.) which has 14 sections. Each week, the student would be expected to complete one section of the book and record a screencast demo of what they have accomplished before coming to class. If they run into problems, the teacher is available online or in-person lab hours every day for 2 hours to help the student troubleshoot. Students would also be able to ask for help from fellow students during the week. Before class begins, each student is expected to write a reflection post about what they learned, and what questions they had to be discussed in class. During class, the teacher will go through the section, show the visual results when the section is completed, and talk about the questions the students came up with during the week. At the end of the term, each student is expected to build an original 3D world that can be inspired by any three. js-built project. Some example projects could be a recreation of a 3D world in a movie, an original anime character that moves to music, a re-creation of a real-world object with ambient and point light sources making use of reflections, or lastly, anything the students want to imagine and can be recreated visually in three.js

After engrossing myself during several afternoons in learning three.js and trying to modify some of the inspirational examples, I believe three.js is a valuable tool to inspire learners to want to use software coding to create fun, inspiring content for themselves, and more importantly to inspire others. Working with three.js sparks joy, curiosity, and creativity. Because three.js is free, open, discoverable, extensible, and inspiring, it is a worthwhile long-term investment both as a computer science educator and student.

References

Blue, L. (n.d.). Discover three.js! Retrieved March 8, 2023, from https://discoverthreejs.com/

Cabello, R. (n.d.). *three.js/LICENSE at dev · mrdoob/three.js*. GitHub. Retrieved March 8, 2023, from https://github.com/mrdoob/three.js/blob/dev/LICENSE

Cabello, R. (2010). *Creator*. Three.js – JavaScript 3D Library. Retrieved March 8, 2023, from https://threejs.org/

Coleman, B. (2017, September 18). *The Beginner's Guide to Beginning Three.js | by Ben Coleman*. Medium. Retrieved March 15, 2023, from https://medium.com/@benjamin.c.coleman/the-beginners-guide-to-beginning-three-js-c36b8947c2aa

khanacademy.com/computing. (n.d.). Khan Academy. Retrieved March 8, 2023, from https://www.khanacademy.org/computing/computer-programming

mrdoob/three.js: JavaScript 3D Library. (n.d.). GitHub. Retrieved March 8, 2023, from https://github.com/mrdoob/three.js/#threejs