

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

Tony Tam
ITEC 830 - Sprint 2023

Introduction

For students who want to learning 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, 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 further because are inspired.

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

three.js

From the GitHub page for three.js : "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.)

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.."

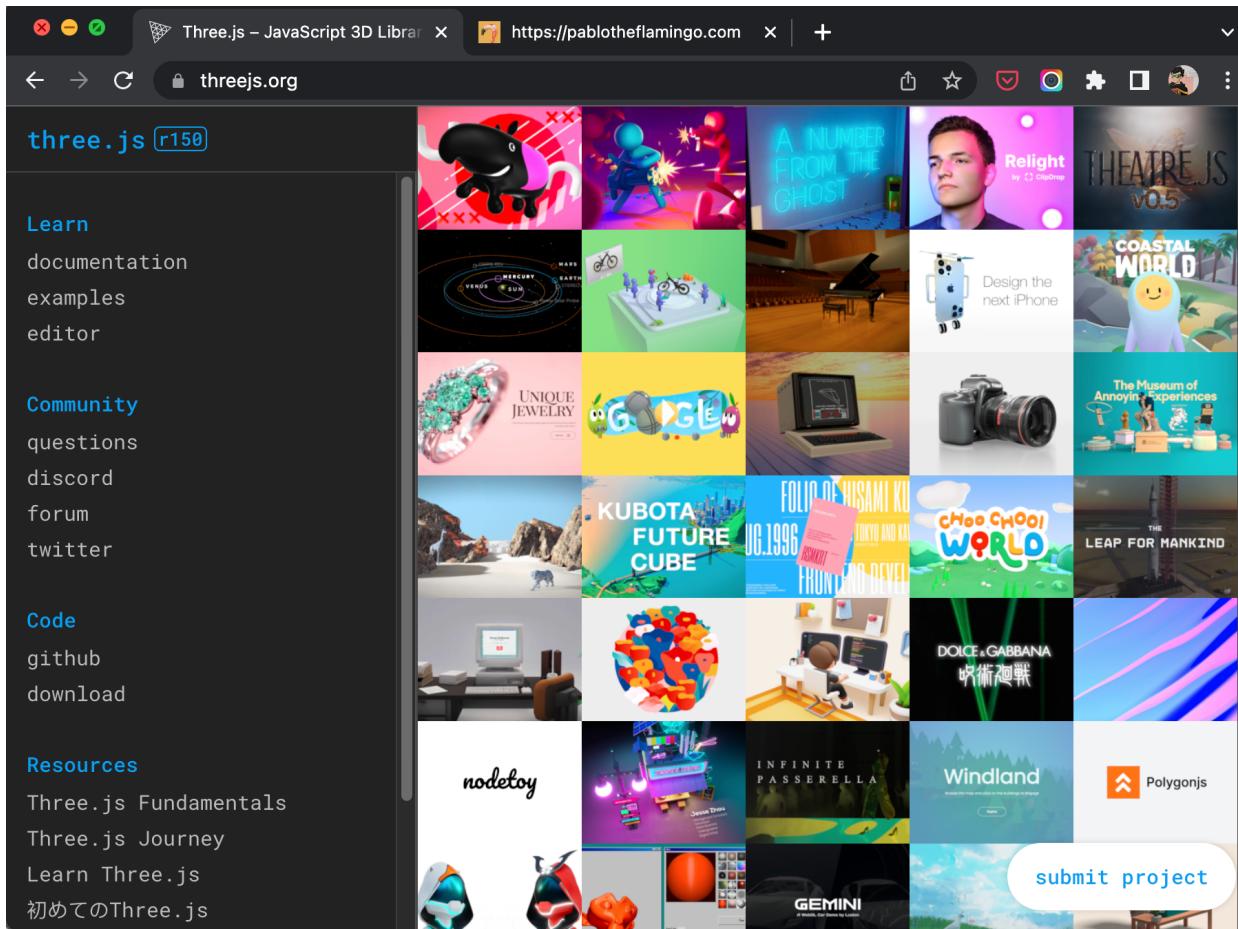
The source code has been 'forked' on Github by over 31,000 people, which means that many people have made copies of the source code and expressed some intent to "iterate on ideas or changes before they are proposed back to the upstream repository"

Open

The source code for three.js is open to the world to read and learn from. It's also open for other people to propose changes and contribute 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 showed 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

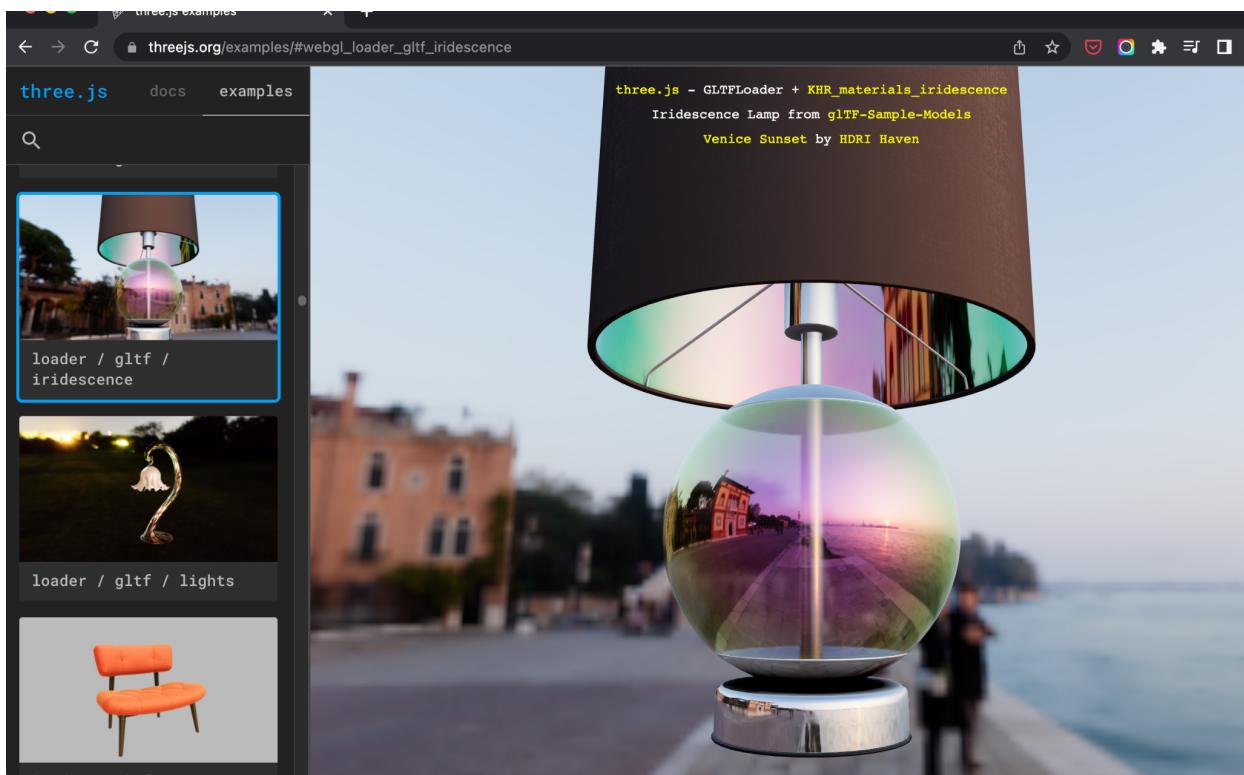
I clicked on a random tile and found a fun and [silly flamingo](#) that I can drag it's long neck and see the flamingo head bounce [Fig B]. If you leave the flamingo alone and turn on the music, it will bounce with rhythm of the music.



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

For the purposes of 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', '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 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 and make changes to remix someone else's work.

I browsed through at least 10 examples and would like to highlight the 2 examples. The first example is 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 glass surface that is transparent, how the light bends and also the inside surface of the lamp shade is reflective.

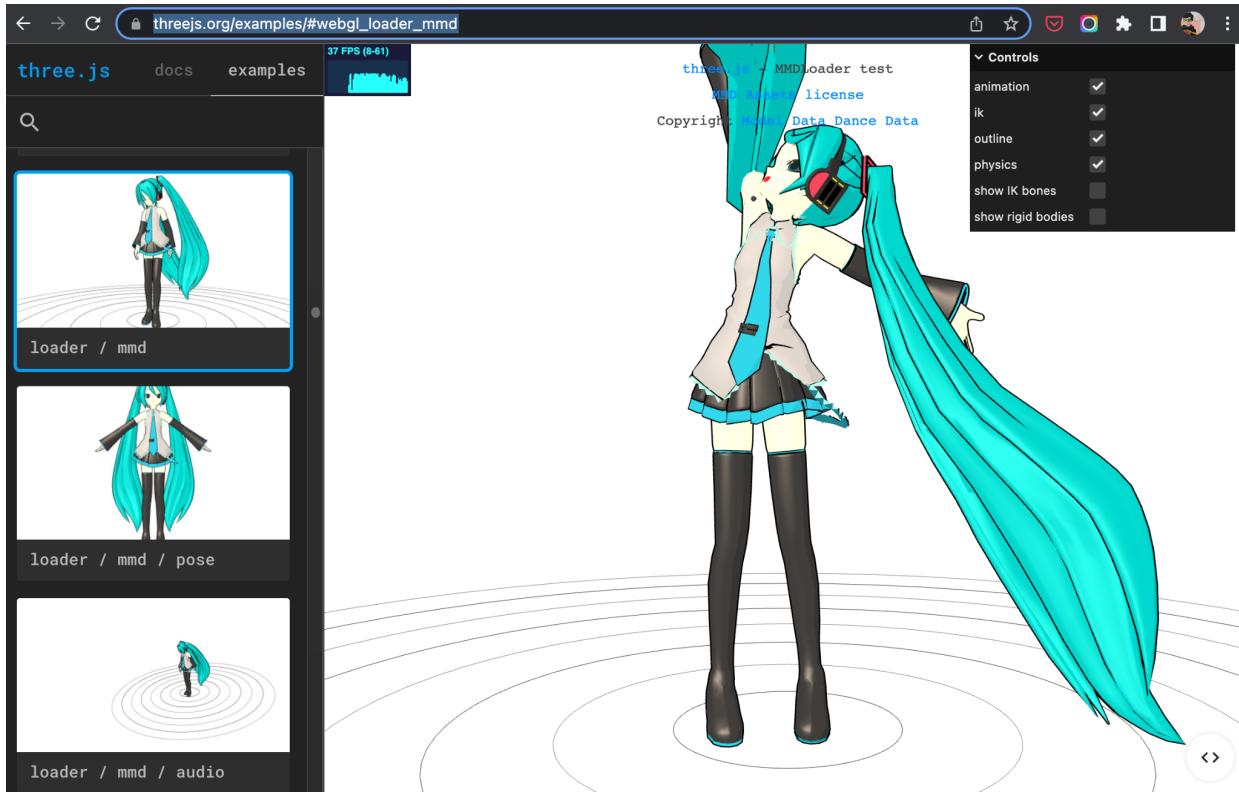


[Fig C] Lamp with glass base and reflective cover

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

The 2nd set of demos is of a Japanese anime dancer who is dancing to a 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 angle is a predefined angle while watching the anime character is dancing to a predefined set. The reader can open this link and interact with the demo example.

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



[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 engine 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 inspiring project that don't need words to explain what it is. The website is surprising short on words and the examples and gallery of projects built on three.js inspires the viewer to imagine what is possible. When the viewer is inspired and is motivated to create, the hard work begins.

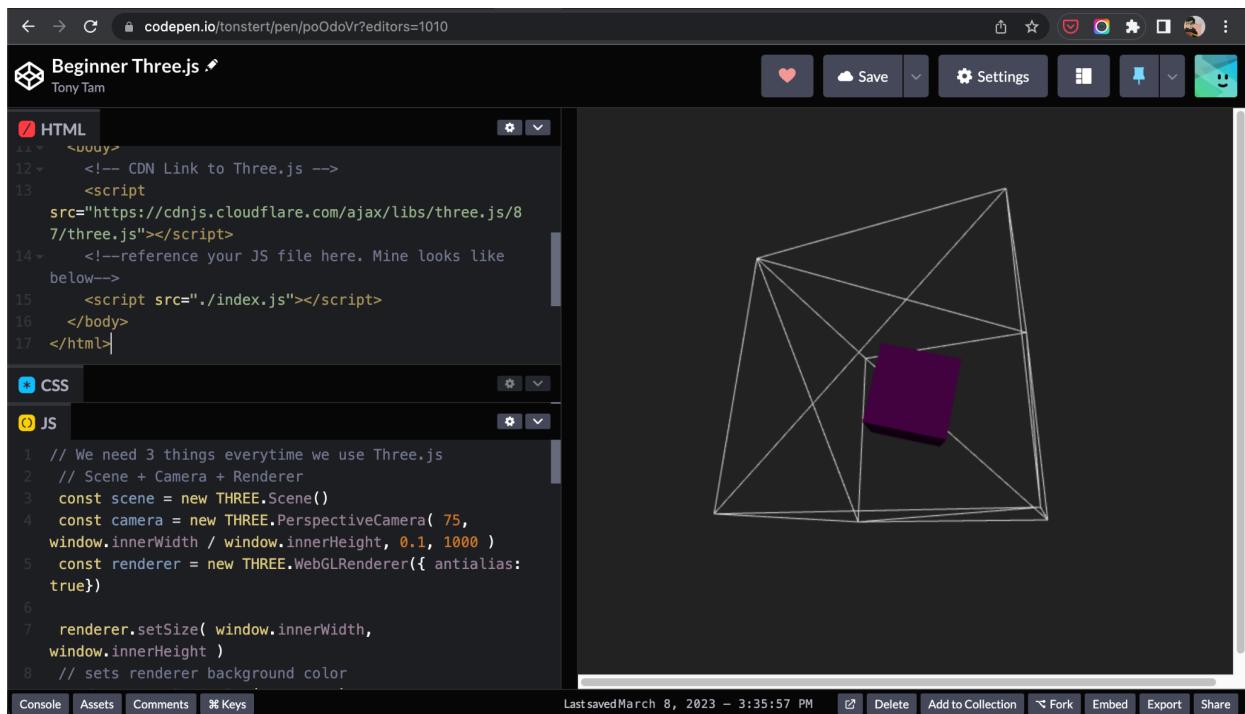
In the next section, I will summarize my journey to get started and make note of challenges to get started with an active project that is every evolving.

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 [cite] offered the flexibility of setting three.js 3 different ways, it didn't offer instructions that leads 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](#) [cite] combined with codepen.io. The tutorial required only a 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 is <https://codepen.io/tonstert/pen/poOdoVr?editors=1011> [Fig E]

A screenshot of the codepen.io interface. The title bar says "codepen.io/tonstert/pen/poOdoVr?editors=1011". The main area shows a 3D scene with a wireframe cube and a single purple rectangular face. On the left, there are three tabs: "HTML", "CSS", and "JS". The "HTML" tab contains the following code:

```
<!-- CDN Link to Three.js -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/r77/three.js"></script>
<!--reference your JS file here. Mine looks like below-->
<script src=".index.js"></script>
```

The "CSS" tab is empty. The "JS" tab contains the following code:

```
// 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
```

At the bottom, there are buttons for "Console", "Assets", "Comments", and "⌘ Keys". On the right, there are buttons for "Delete", "Add to Collection", "Fork", "Embed", "Export", and "Share".

Last saved March 8, 2023 – 3:35:57 PM

[Fig E] Simple cube rotating with a mesh

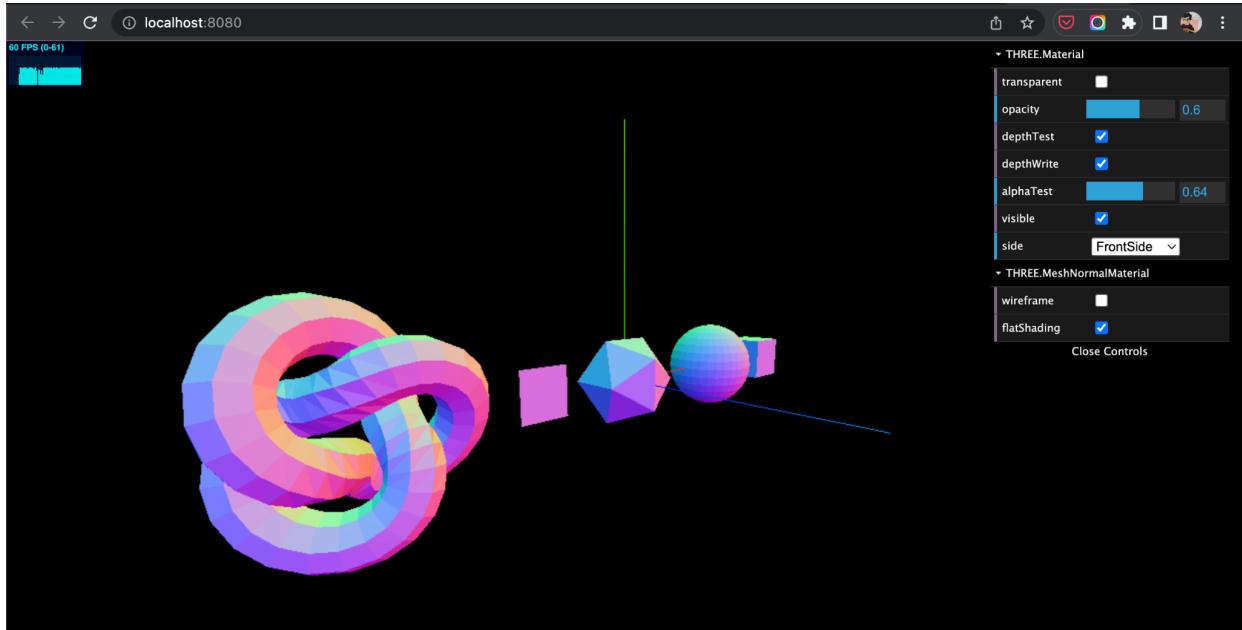
Getting Started : Locally On The Computer

An non primary source to setup three.js was much more helpful to get it setup locally. The website <https://sbcodes.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 advantages 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]



[Fig F] More complicated example working locally

A screenshot of the CodePen interface. The title bar says 'Broken: Three.js example' by Tony Tam. The left sidebar shows the HTML, CSS, and JS code. The JS code includes an import statement for 'three/examples/jsm/libs/stats.module'. A tooltip message from Skypack asks if the user wants to replace invalid imports. The right side of the screen shows the browser console with an error message: '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'. Below the console are standard CodePen sharing and export buttons.

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

Conclusion

After loosing myself during several afternoons engrossed 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.