

Three.js beyond the playground

Updated by Cameron 13/07/2018

Getting your code working outside the playground

Up to now, all of your experimentation has been inside the playground site at:

<http://threejsplaygnd.brangerbriz.net/>

This is a great way to get started in three.js, but it is not the only resource. In this quick tutorial, we will make three.js function outside of the playground.

Also, a big part of three.js is interactivity - this is not possible within the playground framework. Below, we'll learn to add some Mouse Inputs to your code.

The strength of Three.JS is that it can be stored and run from an HTML file, this means that you can store the HTML files locally and/or upload these to the web, and they will display in commonly available browsers.

What you will need

1. A Text Editor
2. A Browser - Chrome, Safari, Edge and Firefox all are fine

The Text editors

You can write your three.js code in many different editors, including some basic text editors, but we recommend a text editor intended for writing code. Here are some of the common ones.

The text editor is where you will write your code. Think of it as the coding screen in Processing, and your Browser window as the viewer/presentation area.

Brackets (Windows/OS X/Linux)

<http://brackets.io/>

Notepad++ (Windows)

<https://notepad-plus-plus.org/>

Atom (Mac OSX/Windows/Linux)

<https://atom.io/>

Sublime Text (Mac OSX/Windows/Linux)

<https://www.sublimetext.com/>

The extraction process

1. Navigate the Archive in <http://threejsplaygnd.brangerbriz.net/> to find a code that you have written previously. Or, jump back in the Gui and editor to make something from scratch.
2. Show Code, Select every line in your code, and **copy**
3. Open your new text editor, and paste the code
4. Save the file as an HTML file somewhere locally
5. Navigate to that file, and Open the HTML file in your browser.

You should see the same animation as in the playground. Now editing and feedback process is looks like this:

- Make and edit to the HTML document in your text editor, then save
- Refresh the browser to see the change

How is this working (or what is three.min.js)

If you look at the top of your code, you will see the following line of code.

```
<script src="http://brangerbriz.net/labs/threejs_playGnd/js/three.min.js"></script>
```

The awesome code that you have written, is relying on this source - called a library, in order to make everything happen in your code. (For other libraries, see

https://en.wikipedia.org/wiki/List_of_JavaScript_libraries)

The file "three.min.js" was stored on the playground.

The updated builds of these files are actually available as source code from:

<https://github.com/mrdoob/three.js/tree/dev/build>

As source code, these can be updated often. You can store the three.js or three.js.min files locally or on another site, and refer to their locations in your script. For now, you can do nothing, and just keep the code referring to the playground. For building bigger and more sophisticated Three.js scripts, it is a good idea to store your own copy to ensure any changes to future builds do not affect your project.

Adding mouse Input

The following is a snippet for mouse input in Three.JS. Please note that this will not work in the Playground, you need to take your code outside the playground, as outlined above.

```
// Add the following code above function setup(), and below <script>.
```

```
    mouseX = 0, mouseY = 0;
```

```
// Add the following code inside function init(), below document.body
```

```
document.addEventListener( 'mousemove', onMouseMove, false );
```

// Add the following **as a separate function**. This should ideally go between *function setup()* and *function draw()* This is

```
function onMouseMove(event)
{
  mouseX = event.clientX;
  mouseY = event.clientY;
  mouse = true;
}
```

// Now you can refer to mouseX and mouseY in the *function draw()* section of your script. For example:

```
mesh.position.x = (mouseX-(window.innerWidth/2));
mesh.position.y = -(mouseY-(window.innerHeight/2));
```