We need a debug UI to be able to use and adjust things quickly on the screen

Librarys to use

Dat.GUI

Control-panel

controlKit

Guify

Oui

We are instead going to use <https://www.npmjs.com/package/lil-gui>

Is looks pretty cool.

There are different types of elements you can add in the panel: (tweeks)

Range – for nums w min and max value

Color – for colors with various formats

Text – for simple texts

Checkbox – for Booleans

Select – for choice from list of values

Button – triggers function

Folder – to organize

To Add elements we are going to use gui.add(…) to add an element to tweek.

The first parameter is an object and the second is the property you want to change.

gui.add(mesh.position, "y");

have to do this after creating variable! You also have to tweek an OBJECTs properties

The next parameters handle min and max or step

gui.add(mesh.position, "y", -3, 3, 1);

This really helps you manually change the values and keep is within certain steps.

gui.add(mesh.position, "x", -3, 3, 0.1);

gui.add(mesh.position, "y", -3, 3, 0.1);

gui.add(mesh.position, "z", -3, 3, 0.1);

here to change all variable to move things around. Can also add these properties like so:

gui.add(mesh.position, "z").min(-3).max(3).step(0.1);

This is a good way to do it, to make things more semantic

Can also change the name or property with .name(‘elevation’)

This is a good idea since you will eventually have a bunch of things to potentially change.

mesh.visible = false;

can even handle things like this:

gui.add(mesh, "visible");

this adds a check box to make the cube visible or invisible

what about making the material wireframe

gui.add(material, "wireframe");

makes checkbox to do this.

What about color now? Well things get complicated. We can not use add, we have to use gui.addColor

Gui does not know it is a color, so we have to use this function. Given that we are not using lil.gui we can use addColor directly on the material.

gui.addColor(material, "color");

this is pretty cool honestly and can change colors on the screen now!