Environment Setup:

For the WebGL Workshop on October 5, 2017, the only two things we will require are Git and Node. If you have these two set up, you can skip to the pink section below!

If you are using a…

Mac:

1. Download and install Git (Select Mac OSX from given options)
   https://git-scm.com/downloads
2. Download and install Node (Select Mac OSX installer)
   https://nodejs.org/en/download/
3. Open Terminal, navigate to your intended folder using 'cd' and 'ls'
4. Enter command 'git clone https://github.com/dmalataeva/uw-workshops'

Windows:

1. Download and install Git (Select Windows from given options)
   https://git-scm.com/downloads
   Make sure you select to install Git bash - this is the command line interface (CLI) we will be using primarily!
2. Download and install Node (Select Windows installer)
   https://nodejs.org/en/download/
3. Open Git Bash, navigate to your intended folder using 'cd' and 'ls'
4. Enter command 'git clone https://github.com/dmalataeva/uw-workshops'
   ALTERNATIVELY, you can just download the repository from this Google Drive: X

Final step:
Try running 'git status'' after the repository is cloned, does it say 'up to date'? If it does, you are good to go!

Repo Build Instructions:

There are two existing script commands for this repository:
npm run build - transpiles and outputs your code in *src/script.js* into *src/public/webgl.js*
npm run dev - watches for changes and automatically transpiles/builds your code, i.e. continuous version of 'npm run dev'.

First, run npm install in '../uw-workshops/webgl-workshop' to download all dependencies needed for your project. (if you're curious, you can find all your dependencies in *package.json*)

There is also one python script we will be using once we get to images. Run python -m SimpleHTTPServer 8080 (the Windows equivalent would be 'python -m http.server 8080') in '../uw-workshops/webgl-workshop' to run a mini-server which will let webGL use your image files.
Once you have successfully installed all dependencies, you can start running the watcher. While the watcher is running, you only have to refresh the browser window to see changes, and the Terminal (or Git Bash) will give you very useful information if it encounters a simple bug in your code.

You can also view errors in the developer console of your browser.
For **Chrome**, go to View->Developer->Javascript Console.
For **IE**, press F12-> CTRL + 2
For **Firefox**, press CTRL + Shift + K on Windows, or CTRL + Command + K on Mac OSX.
For other browsers, google 'how to open console on <YOUR BROWSER>' and follow steps.
Keep in mind that if your console is open in the browser, you may have to select 'hard reset' or 'clear cache' every time you refresh your page to see changes.

The file *script.js* is where you will be writing all of your code. **All of it**.
Your friend in writing good webGL code will be the documentation for Three.js:
https://threejs.org/
Refer to it for examples, specifications and FAQ. Or ask us!