# Comp. Graphics report - w18010435

How the application works / is layed out:

* I installed **three.js** using the command **npm install three –save three** as detailed on the three.js installation guide.
* To run the program in vs code I made an **app.js** file which creates the **express** server using **node** which is needed to run the program. This is detailed in the package.json file **on line 13**. **Using the command node app.js in the vs code terminal runs the program in the web browser**. **Note:** Once extracted from the zip open folder **Basic\_3D\_World** in vs code then run in the terminal from there.
* Within the **main.js** file the scene camera and renderer are set up along with all the animations and effects included in the project.

External files used in project:

* Mmi file is a simple **utility class** which lets you add **event listeners** to meshes so the user can execute functions using **raycast,** for example as demonstrated in the scene when the user clicks one of the box meshes located in the middle, the lightbulb is switched on along with an animation playing as well as turning on and off the blizzard effect too.
* **Cannon.js** is a file which includes physics within a scene. For example, within my scene I have two cubes that are affected by physics and can be manipulated using the **lil-gui** to change the force of gravity in the **x, y, z** directions.

Provided files included with **three.js installation command** used in project**:**

* Included **Stats** to display framerate of scene.
* **Lil-gui** isused for easy implementation of UI for interactions for physics manipulations of two cubes present in the scene.
* **GLTF and FBX Loaders** are used for loading models and animations into the scene for the two knights where one is dancing and the other is jumping.
* **Pointer lock controls** are used to simulate the first person camera controls.
* **Font loader and Text Geometry** are used to display explanatory text within the scene to help the user know what is going on.
* **Water2** is used to provide the water effect in the scene with a realistic water texture. The effect itself is a modified wave function from the workshops to make the simulation more real.

Other:

* The **solar system** is just very basic grouping using **THREE.Group()** to create a solar system group adding each individual planet group and setting them to orbit the sun at different speeds along the **Y axis**.

References:

* *Building the Ocean with Three’s*. (n.d.). Retrieved January 20, 2022, from <https://www.liquid.fish/current/threejs>
* *danielblagy/three\_mmi: A utility class that enables to easily set up a simple callback for mouse interacting with threejs mesh.* (n.d.). Retrieved January 20, 2022, from [https://github.com/danielblagy/three\_mm](https://github.com/danielblagy/three_mmi)i
* *express - npm*. (n.d.). Retrieved January 20, 2022, from <https://www.npmjs.com/package/express>
* *FBX Loader - Three.js Tutorials*. (n.d.). Retrieved January 20, 2022, from <https://sbcode.net/threejs/loaders-fbx/>
* *FontLoader – three.js docs*. (n.d.). Retrieved January 20, 2022, from <https://threejs.org/docs/?q=font#examples/en/loaders/FontLoader>
* *GLTFLoader – three.js docs*. (n.d.). Retrieved January 20, 2022, from [https://threejs.org/docs/index.html?q=gltf#examples/en/loaders/GLTFLoade](https://threejs.org/docs/index.html?q=gltf#examples/en/loaders/GLTFLoader)r
* *How To Get Started with Node.js and Express | DigitalOcean*. (n.d.). Retrieved January 20, 2022, from <https://www.digitalocean.com/community/tutorials/nodejs-express-basics>
* *Installation – three.js docs*. (n.d.). Retrieved January 20, 2022, from <https://threejs.org/docs/#manual/en/introduction/Installation>
* *Mixamo*. Paladin J Nordstrom (n.d.). Retrieved January 20, 2022, from <https://www.mixamo.com/#/?page=1&type=Character>
* *PointerLockControls – three.js docs*. (n.d.). Retrieved January 20, 2022, from <https://threejs.org/docs/#examples/en/controls/PointerLockControls>
* *schteppe/cannon.js @ GitHub*. (n.d.). Retrieved January 20, 2022, from <https://schteppe.github.io/cannon.js/>
* *Solus - The Knight - Low Poly Character free VR / AR / low-poly 3D model animated rigged | CGTrader*. (n.d.). Retrieved January 20, 2022, from <https://www.cgtrader.com/free-3d-models/character/man/solus-the-knight-low-poly-character>
* *TextGeometry – three.js docs*. (n.d.). Retrieved January 20, 2022, from <https://threejs.org/docs/?q=textg#examples/en/geometries/TextGeometry>