**VR-Webb Cluster**

**Steps –**

* **Star Rendering –**
* **Faster loading of stars –** Usage of BufferGeometry class was considered as a good solution. (**Not Implemented**).

**Links:**

* + **Spec -** [**http://threejs.org/docs/#Reference/Core/BufferGeometry**](http://threejs.org/docs/#Reference/Core/BufferGeometry)
  + **Example:** <http://threejs.org/examples/#webgl_buffergeometry_custom_attributes_particles>
* **Star appearance –** 
  + Texture Loaderclass to load an image texture on the material of the stars**. (Implemented).**

**Issues:**

* + - **Transparent images –** Transparent images enabled transparency of stars but did not map image texture. So, in order to have stars rendered with the texture of the image non-transparent images with a black background were used to render stars
    - **Size Attenuation –** Size Attenuation parameter of the material applied to the stars enabled the stars that were closer to the camera to look bigger and the stars that were far off to look smaller proportional to their distance from the camera. This made the visible set of stars to disappear as those were comparatively further away than the stars from the cluster. The size attenuation parameter is a Boolean and can be easily turned off and on. This was enabled in the code to make the scene look less cluttered.
    - **Size-** The size of stars became an issue because of their appearance. After applying an image texture the size needed to be increased to greater values than before to achieve visibility.
    - **Scale -** Some combination of the two functions- **setScale() and multiplyScalar(),** was tried to achieve complete visibility of both sets of stars after the setting of Size Attenuation property. This did not work as a feasible solution and so was discarded.
    - **Overlapping of star positions with objects on the screen -** The properties - **depthTest and renderer.sortObjects** were enabled to ensure the render order of the stars and the objects on the scene which also was a feasible approach to the elimination of the overlapping of stars with other scene objects.
* **WebVR –** WebVR is an experimental Javascript API that provides access to Virtual Reality devices, such as the Oculus Rift or Google Cardboard, in your browser. This was used to facilitate mobile device detection and virtual reality modes for the application and has the potential to be used for multiple VR headsets besides the Google Cardboard. The WebVR boilerplate was used as a foundation as it takes care of features such as VR controls(**VRControls.js**) such as enter and exit mode controls, effects for VR (**VREffects.js**) such as stereoscopy and support for other headset devices (**webvr-polyfill**).
* **Menu Icons –** Menu Icons were rendered as image textures on box geometries with no depth values.
* **Remainder of the scene**  - ThreeJS was a key player in rendering the remainder of the scene and examples on the official three js webpage were referred to help incorporate features in the application.