**Model**

The model location is backwards (3 sec) and left (1 sec)

**Camera Controls:**

WASD = Forward/Left/Backward/Right

R/F = Up/Down

Right Click (long press) = Change camera direction to the mouse pointer

**Jump Feature:**

In development version the jump feature is turned off by default. To enable this feature, uncomment the code inside the “START JUMP FUNCTIONALITY” and “END JUMP FUNCTIONALITY” comments.

**Canvas:**

* Update viewport when the browser window is resized:

function resizeViewport(renderer);

**Lighting:**

Point Light is being used to provide more light to objects next to it. Same functionality as a Light Bulb, provides light from a specific point, with its intensity weakening the further an object is.

* Create objects for all the lights to be loaded into the scene and create these lights

function loadAllLights();

* with the light object (created in loadAllLights) pass it to the function to create the light

function createLight(lightParam);

|  |  |
| --- | --- |
| **Default Light Objects** | |
| **Lights** | **Default properties and values** |
| PointLight | "type": "PointLight",  "color": 0xffffff,  "intensity": 1,  "distance": 0,  "decay": 1,  "size": 2,  'position': {  'x': 0,  'y': -20,  'z': 0  } |
| AmbientLight | 'type': 'AmbientLight',  'color': 0xffffff,  'intensity': 1 |
| DirectionalLight | 'type': 'DirectionalLight',  'color': 0xffffff,  'intensity': 1,  'size': 2,  'position': {  'x': 0,  'y': 1,  'z': 0  },  'target': {  'x': 0,  'y': 10,  'z': 10  },  'castShadow': false |
| HemisphereLight | 'type': 'HemisphereLight',  'skyColor': 0xffffff,  'groundColor': 0xffffff,  'intensity': 1,  'size': 2,  'position': {  'x': 0,  'y': 1,  'z': 0  } |
| SpotLight | 'type': 'SpotLight',  'color': 0xffffff,  'intensity': 1,  'distance': 100,  'angle': Math.PI/3,  'penumbra': 0,  'decay': 1,  'position': {  'x': 0,  'y': -20,  'z': 0  },  'castShadow': false,  'shadowDarkness': 0,  'shadowCameraVisible': false,  'position': {  'x': 0,  'y': 1,  'z': 0  } |

**Skybox:**

We create an enormous 3D cube and we are putting textures inside it. We have downloaded multiple images which when placed together can resemble a real 3D environment. Each picture is carefully placed in a specific interior side of the cube. Inside this cube we will place our normal objects, and because the cube is so large it can fool the user that this is the whole sky and not just a cube side. For the picture used in the skybox consult ‘resources’