CS 606 / Computer Graphics
Assignment 4/mini-project

Aman Kumar
IM2018006
Aman.Kumar@iiitb.ac.in

Dev Patel
IM2018021
Dev.Patel@iiitb.ac.in

Prateek Kamboj
IMT2018057
Prateek.Kamboj@iiitb.ac.in

I. PROBLEM STATEMENT

3D rendering with animation, camera setup, and textures. Learning Objectives:

- Procedural animation of a scene
- Modelling a dynamic scene with a Scene Graph and relative transforms and enabling animation through the scene graph
- Generating texture mappings for objects of different shapes.
- Managing lights and camera
- Handling collisions
- Basics of a VR application

II. APPROACH

1. Scene Graph Organization:

Our scene consists of 2 sets of objects (bus and train) moving in a defined path, an avatar that can move around using w/a/s/d keys, can attach to one the leader from 2 sets of objects using key 'p', and can jump around using the spacebar. Below image is our scene graph organization of our project. As we can see, our scene mainly consists of three segments. First is Group1 that is one bus following the leader and the other bus following the first bus. Group1 also includes streetlights which account for 10 street light mesh and corresponding 10 spotlights.

The second segment is Group2 that includes three trains, 1st train follows the leader, 2nd train follows 1st train and 3rd train follows the 2nd train. And the last segment is our Avatar which consists of 3d mesh for the Avatar and some animation that simulates walking, jumping, and standing Idle. Finally, we have some basic functionalities that are part of the scene like groundMesh on which the whole scene is happening, MainCamera that is observing the whole scene, a Third Person camera that simulates the Avatar’s view of the scene, a Track Light that follows the Avatar, and some elementary light that illuminates the whole scene.

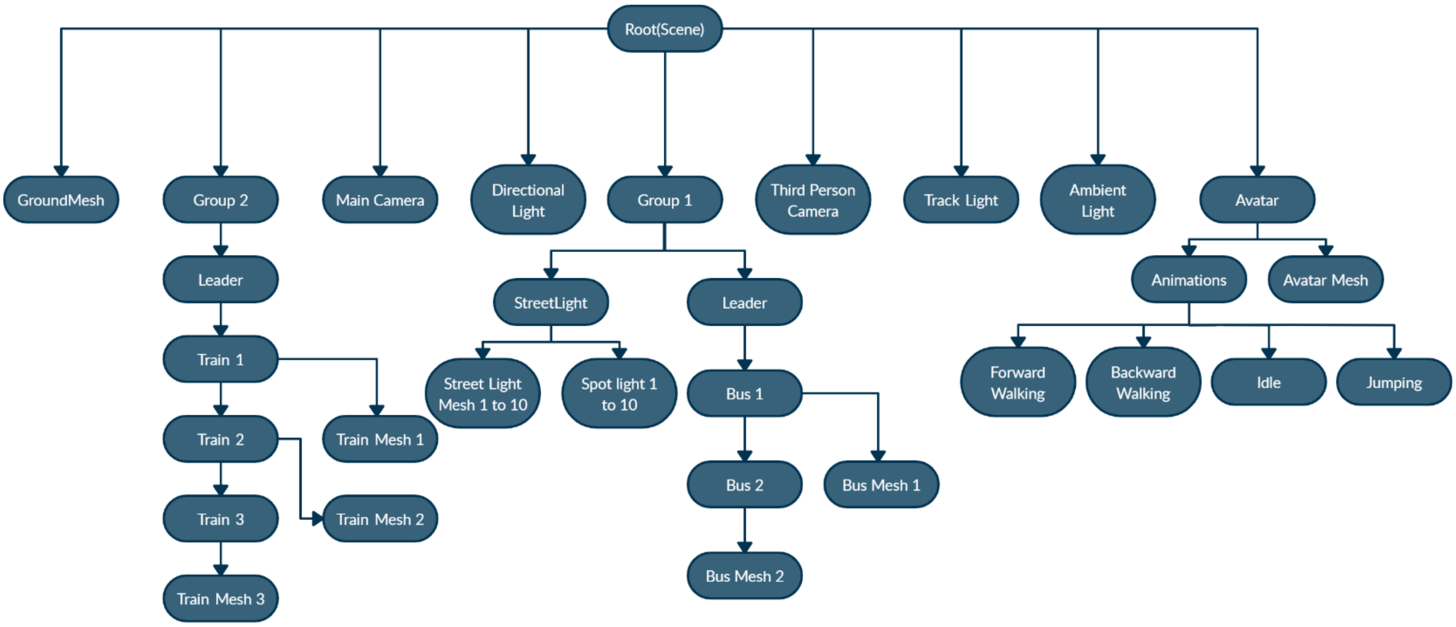

Figure 1. Scene Graph

2. Position and Orientation of Lights and Cameras

There are two elementary lights in the scene: directional light and ambient light. Directional light was aligned perpendicular to the groundMesh, and its color was white. The ambient light was there to simulate natural light. The main purpose of these lights is to illuminate the whole scene. Then we had a set of lights fixed to the ground which acts as a streetlight. We had used a spotLight for this purpose and to align this spotLight with the streetlamp mesh, we had used spotLight helper. Then finally we had track light, whose position is fixed but its target is set as Avatar, so basically it will follow the avatar such as spotlight who tracks actors around the stage.

We had two different types of cameras: the main camera and the third Person camera. The main camera is the default camera that is observing the whole scene. OrbitControls is implemented into the mainCamera that allows it to orbit around the whole scene, zoom, and pan using the mouse. The third-person camera simulates the Avatar’s view of the scene. It follows the avatar movement by updating its position and lookAt according to the avatar's current position and orientation in the render loop.

3. Collision detection/avoidance

In our project avatar acts as a dynamic obstacle in our scene. Both the groups (bus and train) are moving on a predefined path and when the avatar comes close to the leader from one of the groups then that group halts and wait for the avatar to pass by. To check for the collision between leaders and avatar, we had used a simple technique of intersection bounding box. Our

Avatar movement is made up of tiny steps i.e in every animate loop our avatar takes a tiny step that adds up to a complete animation for the movement for the avatar. Now to check for the collision, we pass the bounding box of both the leaders as an argument to the animate loop of the Avatar. In every animate loop, we check if these two BBox are intersecting with the avatars BBox using function intersectsBox(). If they are then we revert back the last tiny step taken by the Avatar.

III. CONCLUSION

We had got to learn so many things such as scene graph is used to organize complex scene and animations. How to apply animation on 3D models to make its movements realistic. How to use images for texture mapping. How to implement simple actions like jumping and avoid a collision. How to incorporate multiple lights in the scene. And lastly how to manage different views of the scene using different cameras.

IV. REFERENCES

[1] https://threejsfundamentals.org/threejs/lessons/threejs-scenegraph.html

[2] https://threejsfundamentals.org/threejs/lessons/threejs-textures.html

[3] https://threejsfundamentals.org/threejs/lessons/threejs-game.html

[4] https://www.mixamo.com/#/?page=1&type=Character

[5] https://threejs.org/examples/#webgl_animation_skinning_blending

[6] https://threejs.org/examples/#webgl_animation_keyframes

[7] https://www.youtube.com/watch?v=EkPfhzIbp2g&t=545s

[8] https://www.youtube.com/watch?v=UuNPHOJ_V5o&t=556s