1 Introduction

1.1 Topic

The theme of this project is to simulate a scene in Kubrick's movie 2001: A Space Odyssey, where the astronauts walk through the long corridor of the space station, and the black stone pillar in the middle symbolizes the black stone pillar that appeared in the movie. The person sitting in front of the stone pillar is moving the skull of an ape, symbolizing the evolutionary path of human beings from apes to flying into space.
In this scene, there is a stationary T-shaped astronaut and two animated robots. One of them can be moved Used by the user (client) with the keyboard direction arrows, the other performs a fixed animation where The robot moves an ape skull between two pillars.

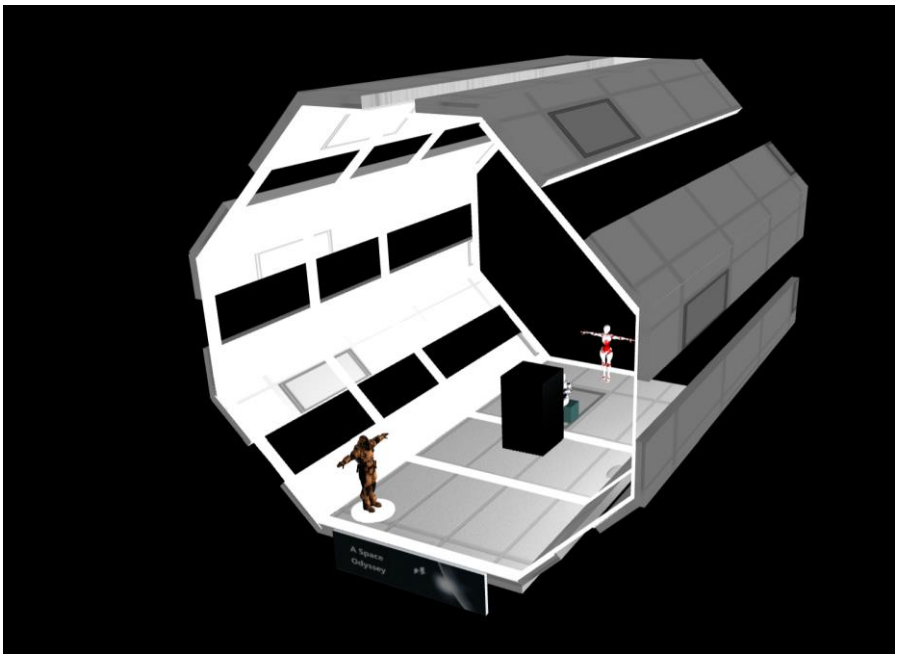


Figure 1: screen view

1.2 Project Requirements

The specific project requirements can be known from the PPT at the end of the course. For this project,
1.The composition of the two robots is complex enough and can be obtained from an external library.
2. Regarding textures and lights, there are three different types of lights in the scene: one is Ambient Light, and there are lights under the model astronaut's feet and surrounding lights that make up the corridor. Additionally, there are two types

of textures, depending on your needs: one for displaying a simple sign showing the project name, and another for the space station represented using color, normal, displacement, roughness and ambient occlusion maps Corridor surface texture.

3. User interaction: The user can interact with the scene by changing the view using the mouse, moving the white standing robot using the keyboard, and changing the color of the walking robot using the GUI in the upper right corner of the screen.

4. Animation: As needed, both animations are implemented in JavaScript and are not imported. In addition, they take advantage of the structure of the robot.

1.3 Hierarchical model

This model was found on the Mixamo website and consists of many parts. For example, the hand is made up of all the joints in humans: little finger, ring finger, middle finger, index finger and thumb, each of which is made up of four joints

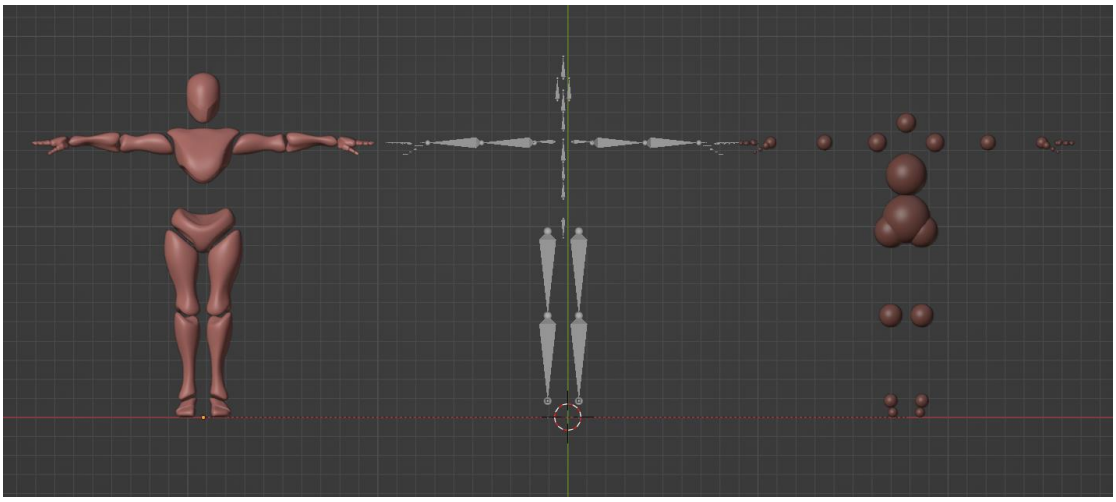


Figure 2: X Bot

2. Libraries Used

The project has been implemented using Three.js 2021 library, alongside, the following modules/libraries have been used:

```
import * as THREE from '../libs/three_js/three.module.js';
import { Environment } from './Environment.js'
import { X_Bot } from './X_Bot.js'
import { X_bot_Walk } from './X_bot_Walk.js'
import { X_Bot_Sphere_Animation } from './X_Bot_Sphere_Animation.js'
import Stats from '../libs/human_interface/stats.module.js';
import { GUI } from '../libs/human_interface/dat.gui.module.js';
import { OrbitControls } from '../libs/human_interface/OrbitControls.js';


let camera, scene, renderer, stats;

const X_BOT_PATH_MODEL = './src/models/x_bot_rotated.glb';
const ENV_PATH_MODEL = './src/models/room.glb';
```

3. Build the model and its associated libraries

3.1 Build a robot

In order to facilitate the management of the X Bot model, X Bot.js was developed. It is used to access each robot component and obtain the configuration of the robot component for further operations.

```
set_configuration(config){
    // HIPS ------------------------------------------
    this._assign_dict_xyz(this.parts.hips.position, config.hips_position);
    this._assign_dict_xyzw(this.parts.hips.quaternion, config.hips_quaternion);
    //_____________________________________________________

    // SPINES ----------------------------------------------------
    this._assign_dict_xyzw(this.parts.spine.quaternion, config.spine_quaternion);

    this._assign_dict_xyzw(this.parts.spine_1.quaternion, config.spine1_quaternion);

    this._assign_dict_xyzw(this.parts.spine_2.quaternion, config.spine2_quaternion);
    // ___________________________________________________________
```

3.2 X Bot Animation

X Bot Animation.js is a tool developed to easily implement X Bot model animation, walking animation and the animation of the robot sitting next to a pillar has been built using this module. This is its interface:

```
init(array_animation, loop){
    this.loop = loop;
    this.array_animation = array_animation;

    // HIPS ------------------------------------------------------------------------------------------
    this._init_hips_position(this.array_animation.hips_position, this.x_bot.parts.hips.position);
    // ____________________________________________________________________________________________

    // HIPS ------------------------------------------
    this._init_hips_quaternion(this.array_animation.hips_quaternion, this.x_bot.parts.hips.quaternion);
    // ___________________________________________________
```

3.3 Linear animation

The animation is necessary to move the X Bot model during walking,
The next position depends on the user, so the Linear animation.js module has been developed,

```
start(){
    this.distance_traveled = 0;
    this.elapsed = 0;
    this.clock.start();
}

update(){
    this.elapsed = this.clock.getElapsedTime();
    const y = this.angular_coefficient * this.elapsed - this.distance_traveled;
    this.distance_traveled += y;

    this.object_to_control[this.name_component_to_control] += y;
}
```

4. Space corridor construction

4.1 Build space environment

The space corridor was designed using blender software and configured in the environment

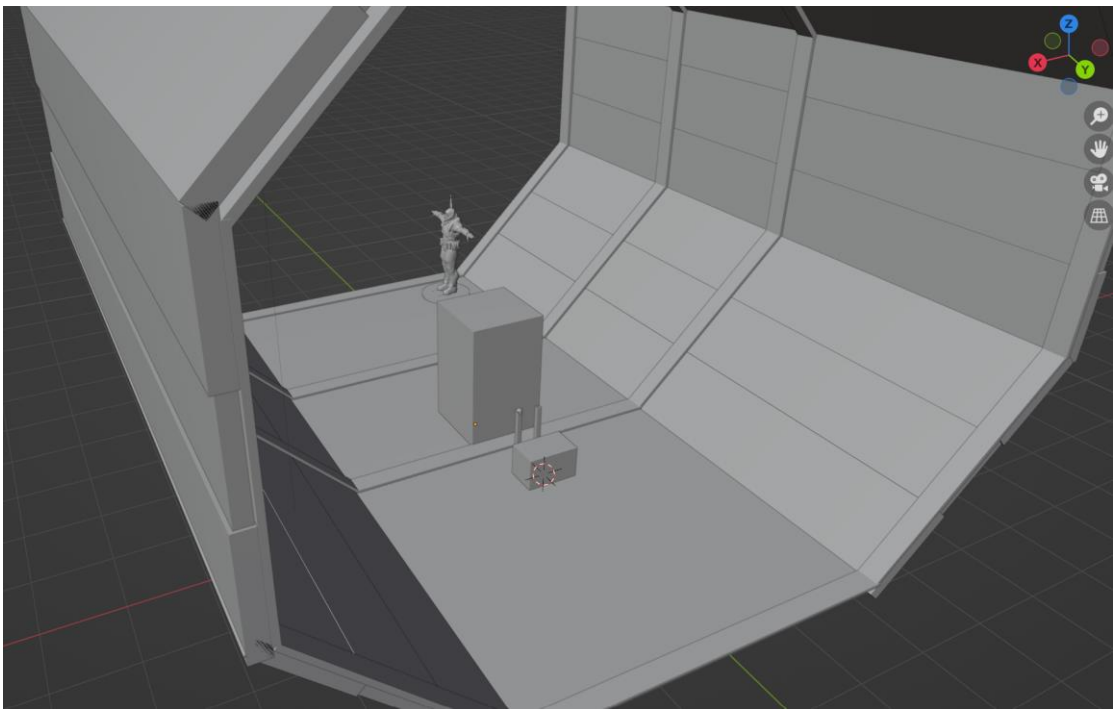
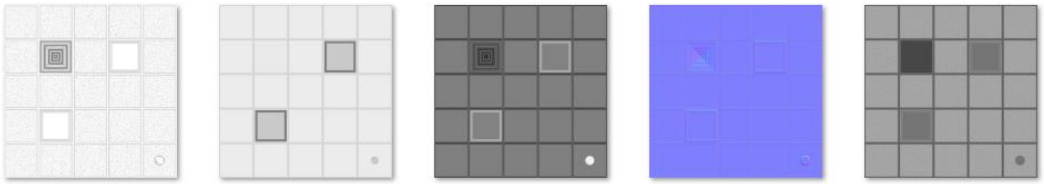


Figure 3: 3D Modelling in Blende

4.2 Materials and textures

All objects in the scene except the walls use MeshPhongMaterial with diffuse and specular components, while the corridor walls use MeshPhongMaterial, MeshStandardMaterial to use PBR textures.

The wall uses a special material that supports Physically Based Rendering (PRB), the applied texture contains: color, normal, displacement, roughness and ambient occlusion maps



.

4.3 Travel limit

In order to avoid obstacles during walking, and to avoid falling into space, in order to do this,The walking space is represented by eight rectangles.

```
_setup_walk_space(walk_animation){
    let boxes = [];

    let box_1 = walk_animation.get_box_template();

    box_1.right_bottom.x = -400;
    box_1.right_bottom.z = -318;

    box_1.left_up.x = 400;
    box_1.left_up.z = -244;


    let box_2 = walk_animation.get_box_template();

    box_2.right_bottom.x = -400;
    box_2.right_bottom.z = 574;

    box_2.left_up.x = 400;
    box_2.left_up.z = 705;
```

4.4 light

There are four surrounding light circles in the corridor, and there is a circular light source under the astronaut model. In Three.js, to represent these lights, use the MeshPhongMaterial material type with an emissive component.

```
this.light_1 = this.scene.getObjectByName('Light_1');
this.light_1.material = new THREE.MeshPhongMaterial( parameters: {emissive: 0xFFFAF0});

this.light_2 = this.scene.getObjectByName('Light_2');
this.light_2.material = new THREE.MeshPhongMaterial( parameters: {emissive: 0xFFFAF0});
```