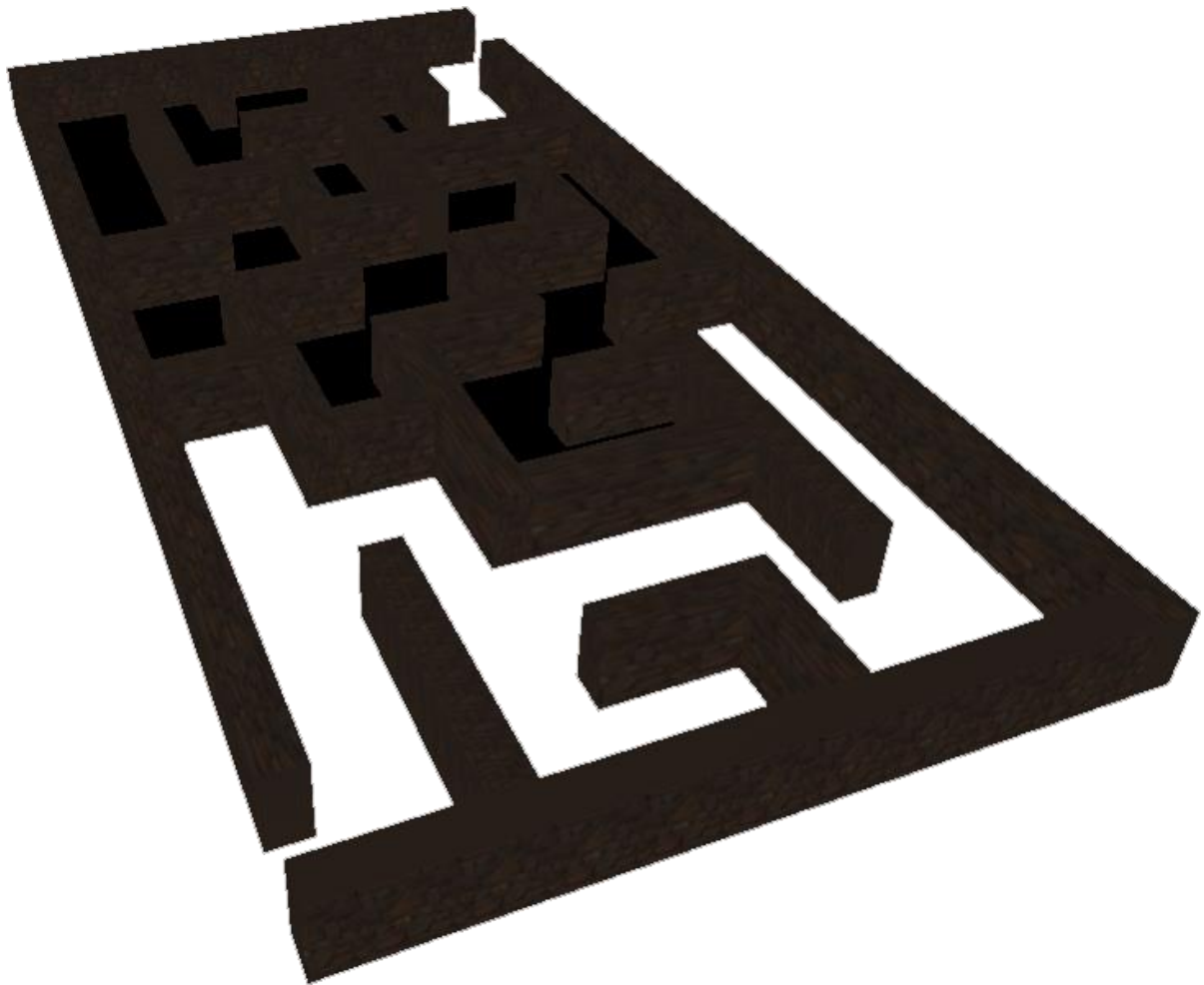


COSC3306

ASSIGNMENT3

REPORT



Teammates:

Kukreti, Manan
Poudel, Sandip
Kumar, Nikhil
Kong, Weixuan

mkukreti@algonau.ca
sapoudel@algonau.ca
nkumar@algonau.ca
wkong@algonau.ca

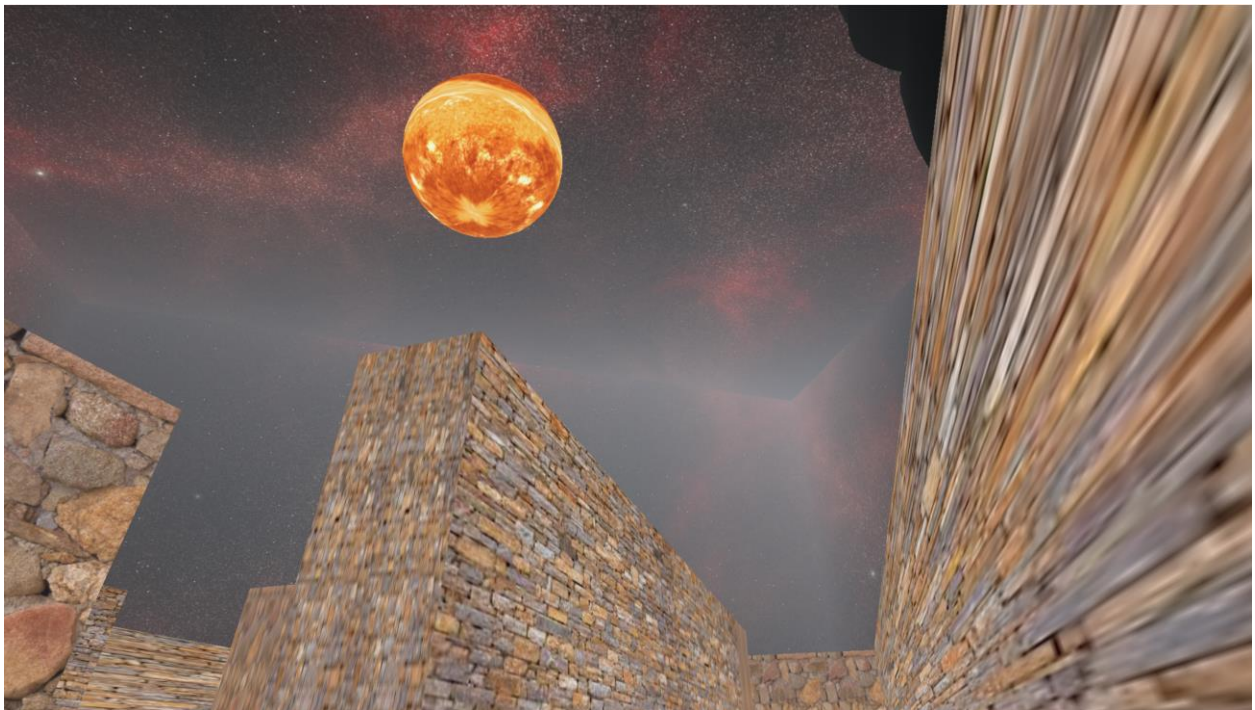
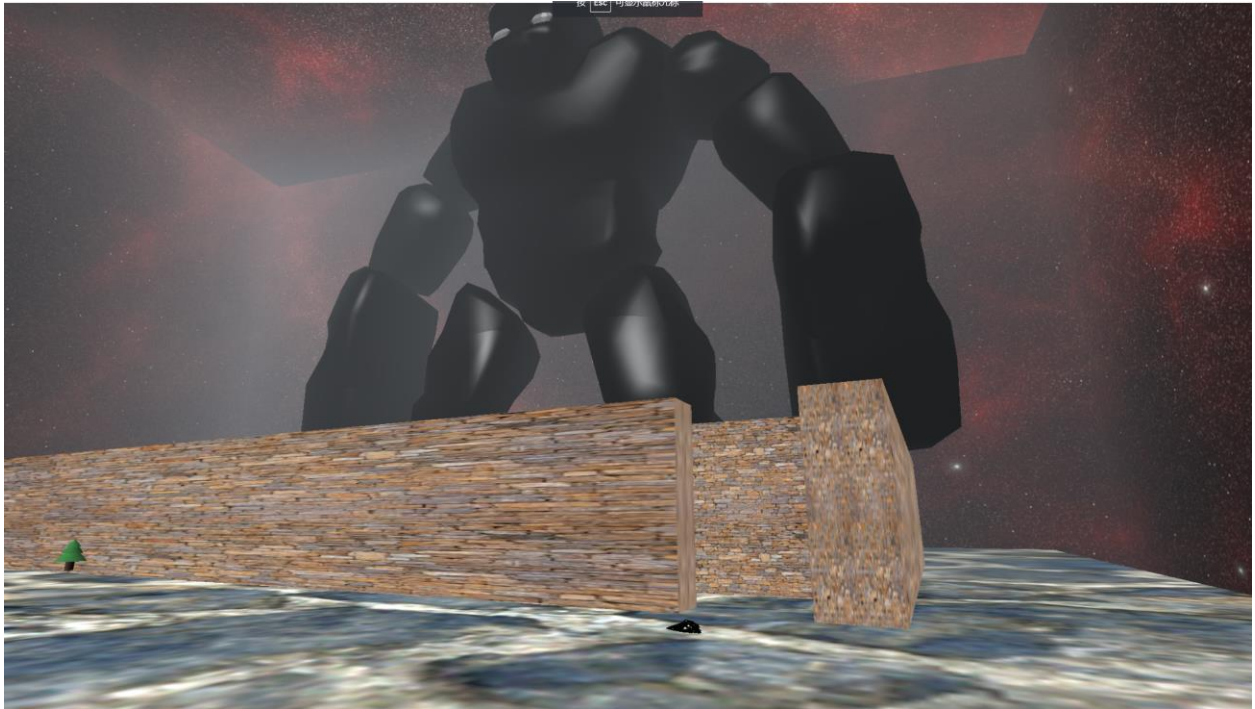
Tasks

1. Create a sky
2. Create a ground
3. Create walls
4. Find good entities we can place in the maze from the internet
5. Add lights
6. Movement of the player
7. Collision detection

Difficulties & Challenges

1. When the team decides to reuse our assignment² work to save time for the assignment³ project, we must keep the old code working correctly in our new project. Fortunately, our codes are low coupling that we only need a few adjustments to apply to make the “sky” and “ground” fit the new project.
2. Creating “walls” is the most critical and complex part to implement. It requires much work to create Meshes and other Objects and finally combine them.
3. Another challenge with creating the maze was converting the 2d top-down view of the maze into a 3D model and ensuring the size and the separation of the walls is accurate.
4. The entities that will make the “Maze” look better; in this part, it is good to have a skilled member with Blender. The team also found a few pre-made objects; they could fit the “Maze” well by applying a few adjustments.
5. Lights could be another essential part, that by applying lights could make the final product more realistic. However, it is not easy to find suitable places and parameters.
6. Regarding using the built-in Class, PointerLockControls significantly reduces the difficulty and challenge of “movement.” The team finally decided on PointerLockControls instead of other controls because PointerLockControls could provide the “Lock” action, which locks the pointer in the web page.
7. “Collision detection “ is the bonus mark part that the team works on at the last stage of the development phase. It is recommended to use the bounding box on the assignment page. However, the team has already made the “Movement” part of controlling the camera and finished the build of Maze “Walls” so that at the least the Raycaster method is chosen to keep the collision detection with only the camera.

Screenshots



REFERENCES

1 & 2.

<https://r105.threejsfundamentals.org/threejs/lessons/threejs-backgrounds.html>

3

<https://stackoverflow.com/questions/31113572/create-3d-perpendicular-walls-using-three-js>

5.

<https://threejs.org/manual/?q=light#en/lights>

6 & 7.

https://threejs.org/examples/#misc_controls_pointerlock

<https://sbcode.net/threejs/raycast2/>