

# 3D INTERACTIVE CITY

DATHRIKA SAICHARITHA

s.dathrika001@umb.edu

University of Massachusetts Boston

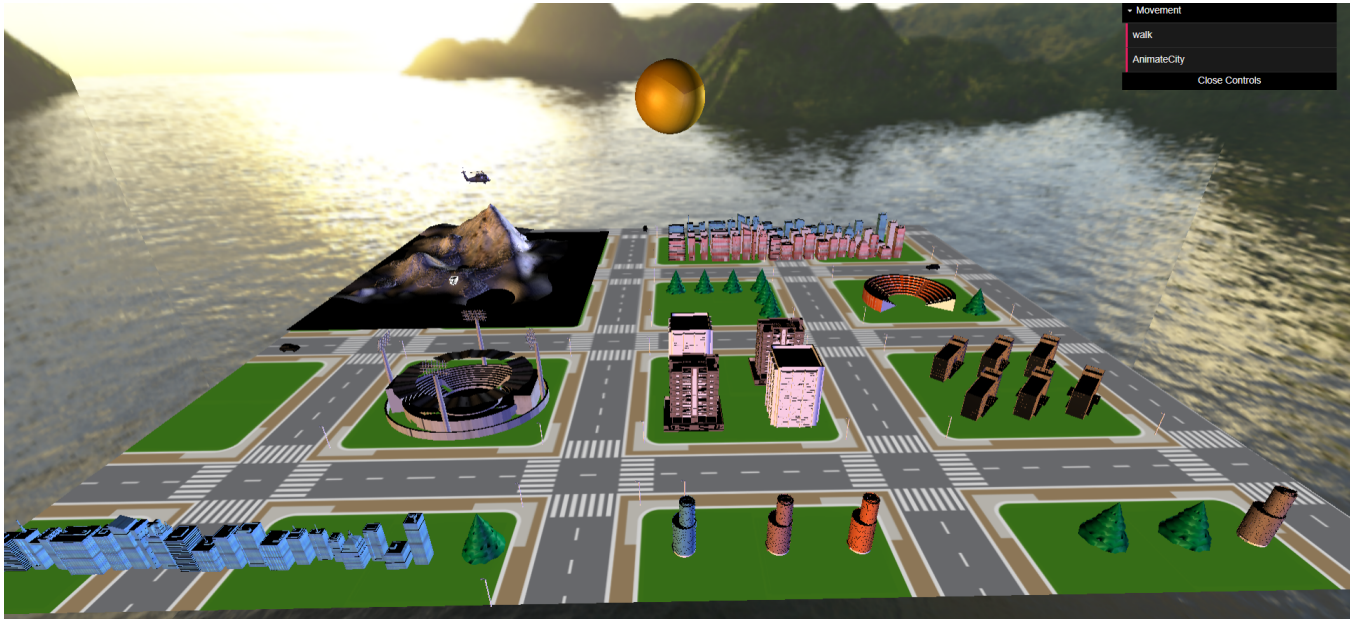


Figure 1: snapshot of the project

## ABSTRACT

My project is a 3D Animation Model of a city. To view the 3D Effects, we need to be on the python server. The Project has been implemented using THREE JS. Various objects have been loaded using OBJ Loaders. The cars and helicopter move around once you click on AnimateCity in the GUI. The Robot also moves around once you click on walk. The cars have been modified to move around in the designated road area.

## KEYWORDS

WebGL, Three.js, Visualization

### ACM Reference Format:

DATHRIKA SAICHARITHA. 2022. 3D INTERACTIVE CITY. In *CS460: Computer Graphics at UMass Boston, Fall 2022*. Boston, MA, USA, 3 pages. <https://CS460.org>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CS460, Fall 2022, Boston, MA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 1337.

<https://CS460.org>

## 1 INTRODUCTION

Drawing is my favorite hobby and I used to draw many scenes like the layout present here and always wanted to model it in 3d....this is my chance to those art scenes visually in 3d. The project helped me learn how THREE JS animation is built.

## 2 RELATED WORK

Low Poly City created on Blender.

## 3 METHOD

A sky box was created to give a 3d effect of the sky.

Further, a plane was created as the floor of the city.

An image which has roads pre-drawn was selected as the floor to depict the roads of the city.

The Project has various objects loaded in it.

Also, some of the objects have been created using Helper.js. Buildings, Trees, Mountains, helicopter, car, water are some of the objects loaded.

Robots and street lights have been created using Helper.js All objects have been loaded with various textures.

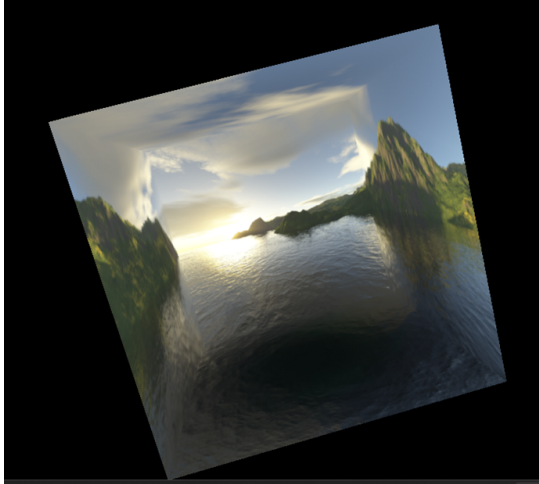
A video effect was added to the water to show moving waves.

This effect is only visible when you turn on the python server.

Animation has been added to the helicopter, car and robots. They all move within the city roads.

### 3.1 Implementation

1. Created a sky box and a plane with directional lighting.



```
// create skybox
let materialArray = [];
let tl = new THREE.TextureLoader();
let tx_ft = tl.load("terrain_front.png");
//let tx_ft = tl.load("terrain_bk.png");
let tx_bk = tl.load("terrain_bk.png");
//let tx_up = tl.load("terrain_up.png");
let tx_up = tl.load("terrain_up.png");
let tx_dn = tl.load("terrain_down.png");
//let tx_rt = tl.load("terrain_right.png");
let tx_rt = tl.load("terrain_left.png");
let tx_lf = tl.load("terrain_right.png");

materialArray.push(new THREE.MeshBasicMaterial
  ({ map: tx_ft }));
materialArray.push(new THREE.MeshBasicMaterial
  ({ map: tx_bk }));
materialArray.push(new THREE.MeshBasicMaterial
  ({ map: tx_up }));
materialArray.push(new THREE.MeshBasicMaterial
  ({ map: tx_dn }));
materialArray.push(new THREE.MeshBasicMaterial
  ({ map: tx_rt }));
materialArray.push(new THREE.MeshBasicMaterial
  ({ map: tx_lf }));
for (var i = 0; i < 6; i++)
  materialArray[i].side = THREE.BackSide;
let skyboxGeo = new THREE.BoxGeometry
( 100000000, 100000000, 100000000 );
skybox = new THREE.Mesh(skyboxGeo, materialArray);
scene.add(skybox);
```

2. Created Robots and Street Lights using Helper.js

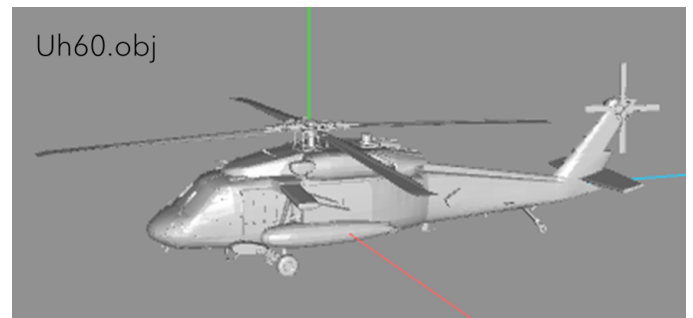


### 3. LOADING 3D MODELS

3D models are available in hundreds of file formats, each with different purposes, assorted features, and varying complexity.

A common format is the .OBJ 3D format to load and display 3d models.

I downloaded the .blend file of Helicopter and loaded it in blender and exported it as .OBJ file



### 4. Adding Materials

we haven't given the scene any materials and (.OBJ) files don't have material parameters.

(.OBJ) files come with a (.MTL) file that defines materials

When .MTL file finished loading, we add the just-loaded materials onto the OBJ Loader then load the .OBJ file.

```
// create helicopter UH60
loader.load('UH60/uh60.obj', function (objectheli) {
  objectheli.position.x = -3000;
  objectheli.rotateX(80);
  objectheli.rotateZ(200);
  objectheli.position.y = 1000;
  objectheli.position.z = -600;
  objectheli.translateZ( 1000);
  objectheli.scale.set(25,25,50);

  objectheli.traverse(function (child) {
    if (child instanceof THREE.Mesh) {
      child.material.map = heliTexture;
    }
  });
  model = objectheli;
  scene.add(objectheli);
});
```

Now we object created in the scene.

## 5. ANIMATION

```
function Flyhelicopter() {  
    y = model.position.x;  
    renderer.render(scene, camera);  
    model.position.x = y + 5;  
    requestAnimationFrame(Flyhelicopter);  
};
```

In this way I loaded many different structures along with their textures and created this whole 3d interactive scene.

### 3.2 Milestones

3.2.1 *Milestone 1.* Create a sky box and a choose a floor to suit the graphic needs of the project.

3.2.2 *Milestone 2.* Finding suitable obj files. Loading and placing these obj files at different positions on the floor

3.2.3 *Milestone 3.* Adding textures

3.2.4 *Milestone 4.* Added the animation to the objects and populated the city with interactive elements

### 3.3 Challenges

- Challenge 1: Adding animation to the cars was a challenging task. The cars had to be made moving only in the designated road area. Further, one of the cars required turning to its right.
- Challenge 2: Adding various materials to the objects was also a challenging task.

## 4 RESULTS

I am able to load multiple objects and make objects interactable.



## 5 CONCLUSIONS

To conclude I would say we enjoyed working on the project and this project helped me learn various THREE JS elements and how to combine these together to create an animated experience.

GITHUB URL:

<https://dathrika13.github.io/cs460student/3dCityProject/>

## REFERENCES

<https://threejs.org/docs/#manual/en/introduction/Loading-3D-models>  
<https://threejs.org/docs/#examples/en/loaders/GLTFLoader>