

3D Avatar and Animation

Shruti Ghelani
s.ghelani001@umb.edu
University of Massachusetts Boston



Figure 1: 3D avatar and animation on portfolio.

ABSTRACT

This project shows an integration of Blender and WebGL. A realistic 3D avatar model is created in blender and then rendered in WebGL. It also include sounds, animations, and lights with Three.js.

KEYWORDS

WebGL, Visualization, Blender, Three.js, glTF

ACM Reference Format:

Shruti Ghelani. 2020. 3D Avatar and Animation. In *CS460: Computer Graphics at UMass Boston, Fall 2020*. Boston, MA, USA, 2 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 2020, Boston, MA

© 2020 Copyright held by the owner/author(s).

ACM ISBN 1337.

<https://CS460.org>

1 INTRODUCTION

I always wanted to make my portfolio fascinating and eye-catching that showcase my skills, so I took this opportunity to develop the same. It includes my recorded voice to give a brief idea of each module, 3D avatar, and body instigated in the blender and rendered with texture and animation with three.js.

2 RELATED WORK

Three.js []. https://threejs.org/examples/?q=obj#webgl_loader_obj_mtl

3 METHOD

Blender

[1] In blender I created avatar with body and bone and gave little animation to it. Also I created head by scanning my face with phone to give realistic view.

[2] I extract .glb file and UV texture image file.

WebGL, Three.js

[1] Designed scene by adding camera, rendered and lights.

[2] Used THREE.CircleGeometry and gave texture with THREE.TextureLoader() along with position, light and scale adjustment.
 [3] Created cube with THREE.BoxBufferGeometry and gave texture to it.
 [4] For loading .glb file I used THREE.GLTFLoader() and THREE.TextureLoader() but with different approach for .glb file.
 [5] I integrated mouse movement with model that follows the cursor.
 [6] I have sound to make it feel better using THREE.AudioListener().

3.1 Implementation

Here you can see implementation of GLTFLoader() and AudioListener().

GLTFLoader:

```
const loader1 = new THREE.GLTFLoader();
const textureface = new THREE.TextureLoader()
.load('texture.png' );
textureface.flipY = false;
const mtl = new THREE.MeshStandardMaterial({
  map: textureface,
  color: 0xffffff,
  skinning: true });
loader1.load(
  // resource URL
  'notexture.glb',
  // called when the resource is loaded
  function ( gltf ) {
    model = gltf.scene;
    model.traverse(o => {
      if (o.isMesh) {
        o.castShadow = true;
        o.receiveShadow = true;
        o.material = mtl;
      }
    });

    model.scale.set(15, 15, 15);
    model.position.set(90,-27,0);

    scene.add(model);
```

AudioListener:

```
const listener = new THREE.AudioListener();
camera.add( listener );
const sound = new THREE.Audio( listener );

// load a sound and set it as the Audio object's buffer
const audioLoader = new THREE.AudioLoader();
audioLoader.load( 'pro.m4a', function( buffer ) {
  sound.setBuffer( buffer );
  sound.setLoop( false );
  sound.setVolume( 0.4 );
  sound.play();
});
```

Table 1: table

Device	Performance
Macbook	60 FPS

3.2 Milestones

3.2.1 *Milestone 1.* Working on Blender as a beginner was difficult and time consuming considering creating a 3D human model with bones and animations.

3.2.2 *Milestone 2.* Integration of blender and WebGL was complicated, Also I feel lack of online resources.

3.3 Challenges

- Challenge 1: Again working with the camera on the blender, also it does not have inbuilt libraries for a model builder and .glb file loader. It took some time to figure everything out.
- Challenge 2: Chrome sometimes fails to refresh all changes as in result does not display the correct result and needs to clear the cache.

4 RESULTS

I did achieve good results on creating a model. I believe it could be better with more time and accuracy. I managed to get 60 FPS on Macbook.



Figure 2: Final Output

5 CONCLUSIONS

To wrap up, I have achieved my goal in creating 3D avatar. Working on animation with Blender and on WebGL at a same time was prime factor here which you can see working model in output files. I believe Blender is a diverse platform from which you can gain realistic and high performance. In this project as I have focused on managing neck, waist and hand movements in future I will be working on facial expressions.

REFERENCES

- [1] <https://threejs.org/> [2] <https://tympanus.net/codrops/category/tutorials/>
 [3] <https://stackoverflow.com>