Walk Like an Egyptian!

Freddy Mansour
freddy.mansour001@umb.edu
University of Massachusetts Boston

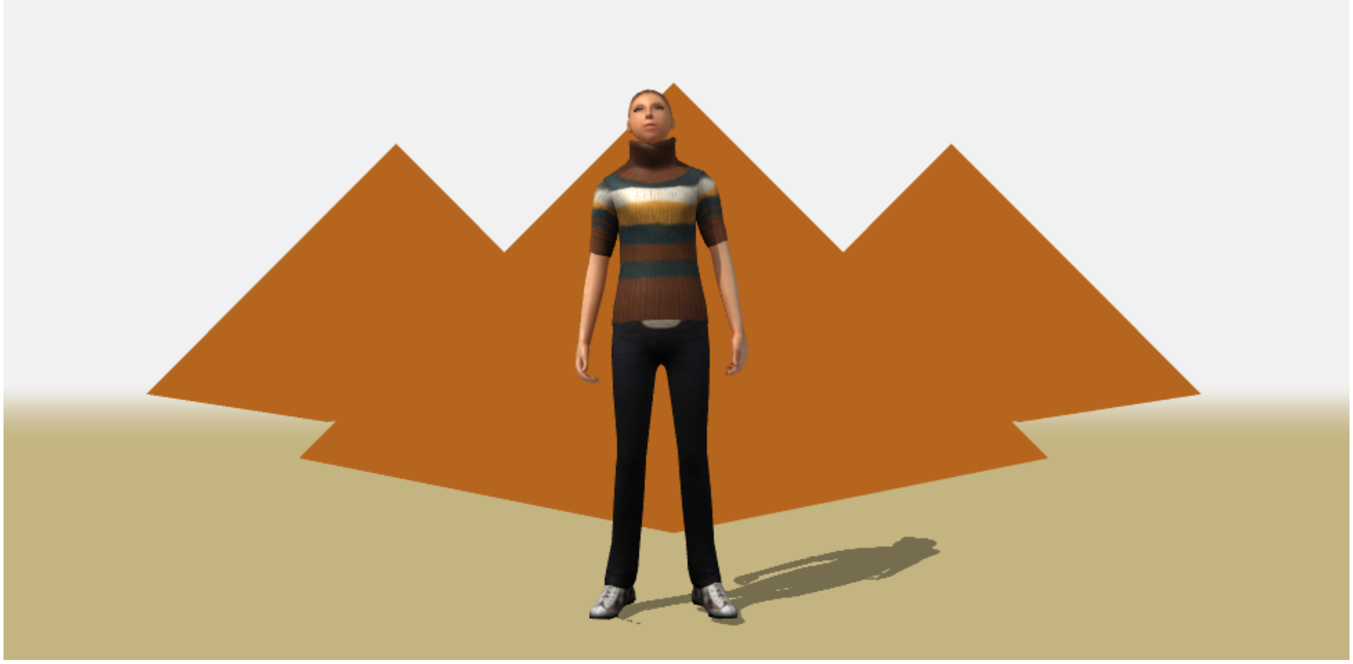

Figure 1: Scene upon loading. Sandy floor and pyramids in the background.

ABSTRACT

This project uses the Three.js framework to create a desert scene with a model that has several built in animations.

KEYWORDS

WebGL, Visualization

ACM Reference Format:
Freddy Mansour. 2020. Walk Like an Egyptian!. In *CS460: Computer Graphics at UMass Boston, Fall 2020.* Boston, MA, USA, 2 pages. https://CS460.org

1 INTRODUCTION

My contribution to this project was the pyramids and sandy floor.

2 RELATED WORK

Kyle Wetton's Three.js Character tutorial. [https://tympanus.net/codrops/2019/10/14]

3 METHOD

I found a source online detailing how to create an interactive 3D character with Three.js. I followed the tutorial and then edited the scene so it lined up with an Egyptian desert scene.

3.1 Implementation

I implemented this project mainly by following the tutorial and learning about the different features provided. The code below shows the use of a loader to load a gltf file which the human model in the scene.

```
let stacy_txt = new THREE.TextureLoader().load('https://s3-us-west

 stacy_txt.flipY = false; // we flip the texture so that its the rig

   const stacy_mtl = new THREE.MeshPhongMaterial({
     map: stacy_txt,
     color: 0xffffff,
     skinning: true
   });

   var loader = new THREE.GLTFLoader();

   loader.load(
       MODEL_PATH,
```

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

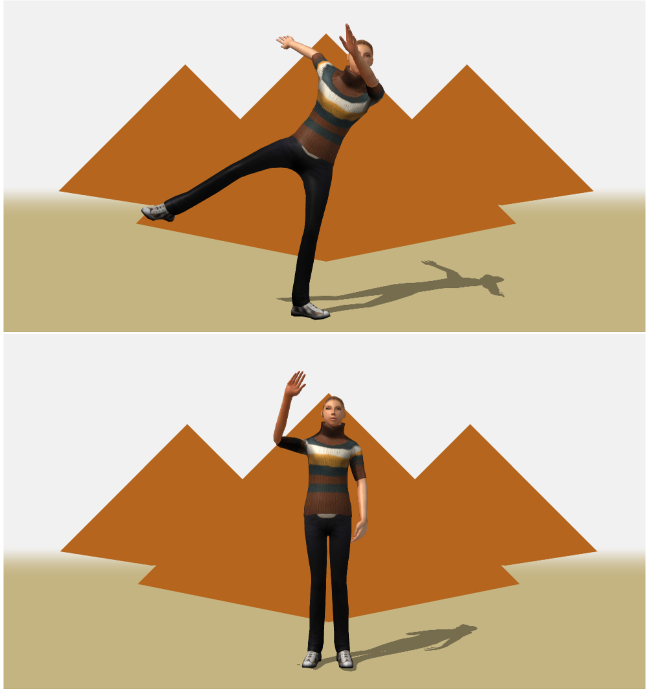

Figure 2: Some of the animations built in.

```
function(gltf) {
model = gltf.scene;
let fileAnimations = gltf.animations;

model.traverse(o => {
  if (o.isMesh) {
    o.castShadow = true;
    o.receiveShadow = true;
    o.material = stacy_mtl;
}
```

```
// Reference the neck and waist bones
  if (o.isBone && o.name === 'mixamorigNeck') {
        neck = o;
    }
 if (o.isBone && o.name === 'mixamorigSpine') {
        waist = o;
    }
});
```

3.2 Milestones

3.2.1 Milestone 1. Decided on project type from final projects list.

3.2.2 Milestone 2. Decided on implementation method (gltf model, animations).

3.2.3 Milestone 3. Developed scene features (background, floor).

3.3 Challenges

- Challenge 1: One challenge was loading an alternative gltf file. I wanted to use something else that resembled something Egyptian but when I downloaded the files, they would show up with different/incomplete.
- Challenge 2: Creating pyramids in the background. There was no "pyramid geometry" that existed so I used a cylinder geometry, editing the top and bottom radii until they resembled pyramids.

4 RESULTS

5 CONCLUSIONS

In conclusion, I learned about the gltf file format, loaders, and built in animations in three.js through this project. The tutorial walked me through a few different features and I was happy to implement them and see them working on my own. I wish I was able to figure out how to load an alternative gltf model.

REFERENCES