Fists of Fury (an interactive three.js application)

By

Name: Krishna Gopalan, Arvind

Student ID: 122104361

Revisions

S.No	Author	Description	Date Completed
1	Krishna Gopalan, Arvind	Initial Version	30-MAR-2024

Readme

System Overview

"Fists of Fury" is an engaging and dynamic web application created with the three.js library, harnessing the power of WebGL to showcase captivating 3D animations directly within a web browser. Drawing inspiration from the action-packed movie "Real Steel", this application plunges users into the electrifying realm of robotic boxing.

Immersive Booking Experience
Delivering an immersive and interactive experience, "Fists of Fury" allows users to reserve seats for thrilling robot boxing tournaments. This application aims to provide users with an interactive solution for reserving stadium seats, offering a clear depiction of their anticipated experience. Users can navigate through a virtual representation of the venue, immersing themselves in a lifelike simulation of the seating area. The application randomly displays a seat for the user to book. Users can interact with the 3d model and book their favorite seats.

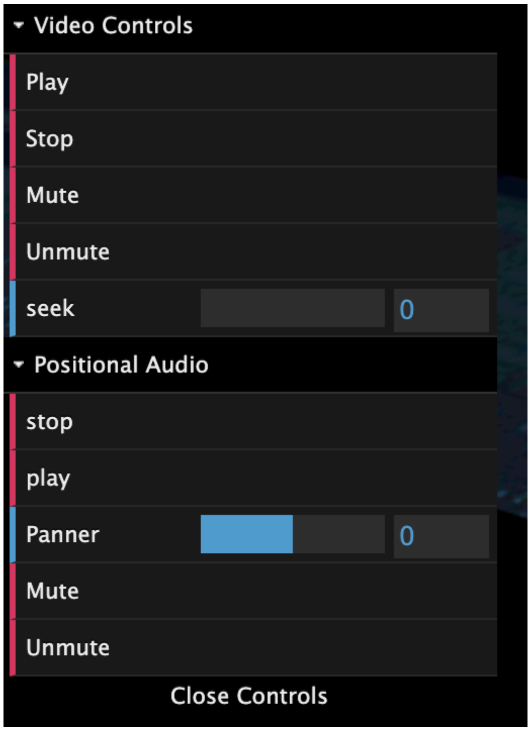
Key Features

Animated 3D Models
The application showcases a pair of dynamic 3D robots participating in intense boxing matches within the virtual arena, vividly capturing the thrill of combat through seamless animations.

The 3D models were sourced from cgtrader.com and sketchfab.com. They were then imported into Blender, rigged, and given basic animations to bring the characters to life.

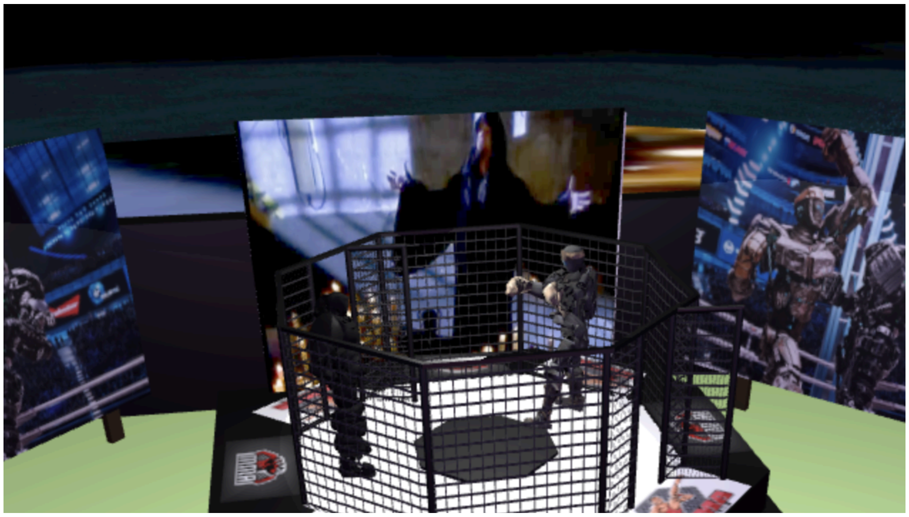
Spatial Sound

The application immerses the users further into the action with spatially positioned background music, heightening the sense of realism and immersion. Spacial sound is tied to an invisible cube added to the scene whose position can be controlled through the graphical user interface



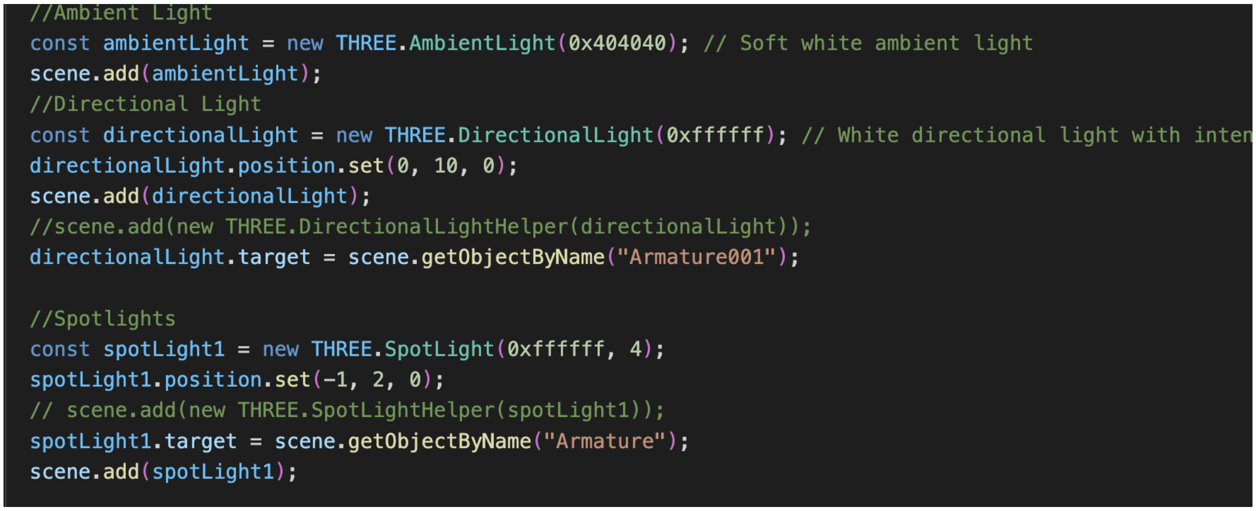
Video Playback

Application plays a video in the 3D environment that simulates stadium screens displaying exciting match footage. Video was added to the plane geometry as a texture



Lighting and Shadows

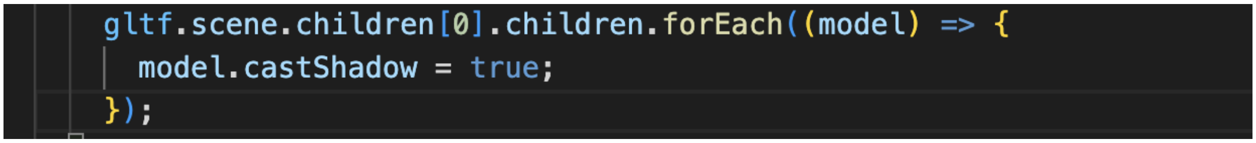
Using a variety of lighting techniques, the application showcases a visually stunning environment casting realistic shadows that enhance the overall ambiance and depth of the experience. The application uses ambient lights for the scene, directional lights and spotlights for individual models



```
//Ambient Light
const ambientLight = new THREE.AmbientLight(0x404040); // Soft white ambient light
scene.add(ambientLight);
//Directional Light
const directionalLight = new THREE.DirectionalLight(0xffffff); // White directional light with inten
directionalLight.position.set(0, 10, 0);
scene.add(directionalLight);
//scene.add(new THREE.DirectionalLightHelper(directionalLight));
directionalLight.target = scene.getObjectByName("Armature001");

//Spotlights
const spotLight1 = new THREE.SpotLight(0xffffff, 4);
spotLight1.position.set(-1, 2, 0);
// scene.add(new THREE.SpotLightHelper(spotLight1));
spotLight1.target = scene.getObjectByName("Armature");
scene.add(spotLight1);
```

The "cast shadow" property was set to true to activate the shadow effect on the models.

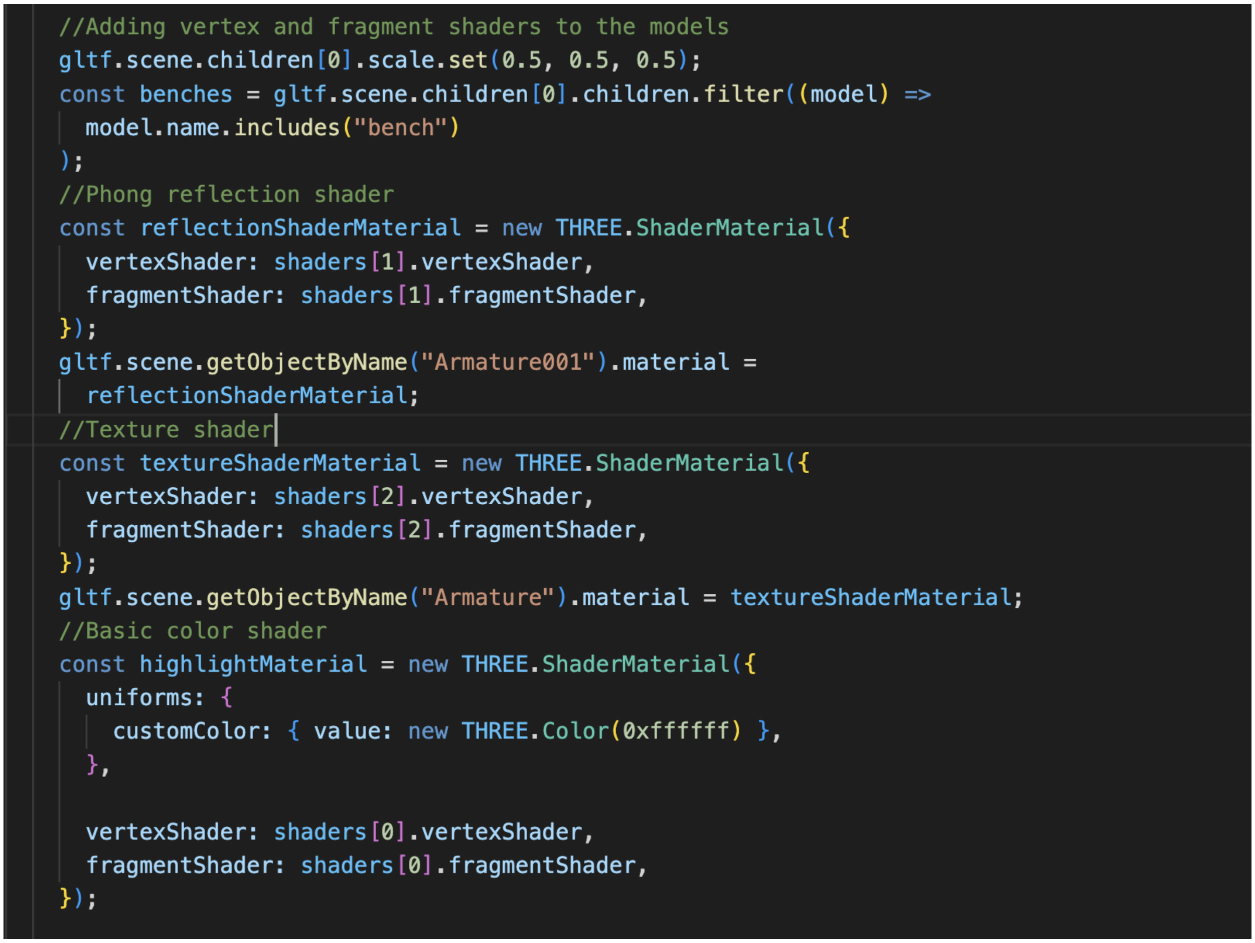


```
gltf.scene.children[0].children.forEach((model) => {
  model.castShadow = true;
});
```

Textures and Fragment Shaders

Intricately detailed characters feature diverse textures and vertex-fragment shader combinations, creating visually striking visuals that captivate the audience.

Used various vertex and fragment shaders pairs including Phong reflection shader, texture shader and basic color shader.



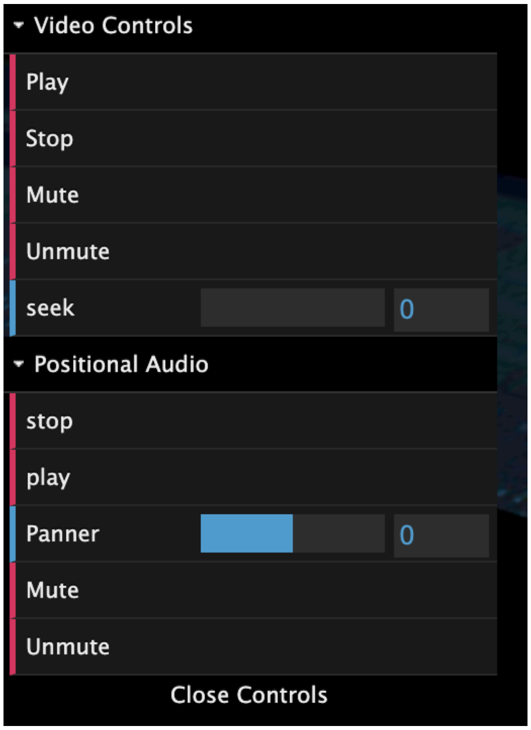
```
//Adding vertex and fragment shaders to the models
gltf.scene.children[0].scale.set(0.5, 0.5, 0.5);
const benches = gltf.scene.children[0].children.filter((model) =>
  model.name.includes("bench")
);
//Phong reflection shader
const reflectionShaderMaterial = new THREE.ShaderMaterial({
  vertexShader: shaders[1].vertexShader,
  fragmentShader: shaders[1].fragmentShader,
});
gltf.scene.getObjectByName("Armature001").material =
  reflectionShaderMaterial;
//Texture shader
const textureShaderMaterial = new THREE.ShaderMaterial({
  vertexShader: shaders[2].vertexShader,
  fragmentShader: shaders[2].fragmentShader,
});
gltf.scene.getObjectByName("Armature").material = textureShaderMaterial;
//Basic color shader
const highlightMaterial = new THREE.ShaderMaterial({
  uniforms: {
    customColor: { value: new THREE.Color(0xffffff) },
  },

  vertexShader: shaders[0].vertexShader,
  fragmentShader: shaders[0].fragmentShader,
});
```

Graphical User Interface

The application allows users to seamlessly navigate through the application's features and settings with a user-friendly graphical interface by utilizing “dats.gui" library, ensuring a smooth and enjoyable user experience for all.

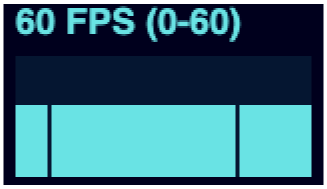
The GUI offers the users the capability to control various aspects of the video and audio. Users can perform basic actions such as play, pause, seek, pan, mute and unmute



seek 0

Panner 0

Real-time Performance Statistics:
The application provides real time statistics about the web application’s performance with the help of a library called stats.js, providing valuable insights into frame rates, memory usage, and rendering times.



WebXR:
Incorporated a VR Button and configured accordingly to have the scene rendered in WebXR.

Raycasting:
Used three.js Raycaster for detecting mouse interactions over the 3D models

Tweening:
Tween.js library was utilized to create seamless camera animations, ensuring smooth transitions between different views or positions.

References

Blender models were sourced from the below sites

https://www.cgtrader.com/free-3d-models/exterior/other/amphitheater-51b361ad-dc65-468c-8eb2-b3123b409869

https://www.cgtrader.com/items/3657374

https://sketchfab.com/3d-models/real-steel-hd-original-atom-model-3b5a8967cdef475b97c7bc9390d291b3

https://www.cgtrader.com/free-3d-models/furniture/chair/bench-1314c54a-4e15-4179-a443-ee86222b7eb1

https://creazilla.com/nodes/63412-boxer-bot-3d-model

Texture Images used in blender models were sourced from the below sites

https://in.pinterest.com/pin/486599934742616776/

https://dribbble.com/shots/10823052-MMA-Mascot-logo

Application plays https://www.youtube.com/watch?v=8dcY9LDdRIk&ab_channel=WWE in the 3D environment that simulates stadium screens

User flow diagrams were created using https://www.canva.com/