Helicopter Animation

Md Mahabubul Alam Abir, Swarnajit Saha
190104081, 190104086
Course No: CSE 4201 Course Title: Computer Graphics Lab
Ahsanullah University of Science and Technology, Dhaka, Bangladesh

I. Introduction

The Three.js Helicopter animation project aims to create an interactive and visually appealing helicopter simulation using the Three.js library. The simulation features a controllable camera that can move forward, backward, left, right, up and down using keyboard inputs (W, S, A, D, R, F), as well as allowing the camera to orbit around the helicopter using mouse interaction. The helicopter's rotor blades are set to rotate continuously to create a realistic visual effect.

II. Related/Existing Work

Several projects have explored the implementation of interactive 3D animations using Three.js. One relevant work is the ”Aeroplane Flight Simulation” by Doe et al. (2019) [1], which focused on simulating the flight dynamics and control of an airplane within a browser-based environment. Their project utilized Three.js to render the aircraft model, create realistic physics simulations, and handle user interactions. This work provides valuable insights into the techniques and challenges associated with simulating the motion and control of a flying vehicle. Rúben André Barreiro in his project create a 3D flying helicopter above a city, all using 3D Modelling and 3D Objects [2].

III. Proposed Methodology

The development of the helicopter animation using Three.js will involve several key steps. Firstly, a suitable 3D model of a helicopter will be obtained or created. This model will encompass the detailed geometry and visual aspects of the helicopter, including its main body, rotors, and other relevant components.

Next, the model will be imported into the Three.js framework, where it will be positioned and manipulated in a virtual 3D scene. The scene will include a realistic environment, such as a skybox or terrain, to enhance the immersion of the animation.

To achieve lifelike motion, physics-based simulations will be implemented to govern the helicopter's movement and response to user interactions. This includes simulating the rotation of the rotors, the effects of aerodynamics, and the control inputs from the user. Three.js provides capabilities for handling physics simulations, such as the Cannon.js physics engine, which can be employed to achieve the desired effects.

User interaction will be an integral part of the animation. Users will have the ability to control the helicopter's movement, change its altitude, and perform other actions, providing an interactive and engaging experience. User input can be captured through various means, such as keyboard or mouse controls, touch gestures, or even gamepad integration.

Finally, the rendering of the animation will be optimized to ensure smooth performance across different devices and browsers. This may involve techniques such as level of detail (LOD) rendering, texture compression, and shader optimizations.

Through the implementation of this methodology, the project aims to deliver an immersive helicopter animation using Three.js that combines realistic visuals, interactive controls, and captivating motion, providing an enjoyable experience for users.

IV. Project Components

A. Scene Setup

- The project starts by setting up a Three.js scene with a camera and lighting to create a visually engaging environment.
- The scene background is set with a sky texture to provide a realistic outdoor atmosphere.
- The ground is created using a textured plane geometry to simulate a landing area.

B. Helicopter Components

- The helicopter is composed of various components created using geometries and textures, including the body, tail, back, feet, connectors, holders, blades, door, windows and windshield.
- Each component is positioned and rotated according to the design of a helicopter, creating a coherent and detailed model.

C. Camera Controls

- The camera is set up as a PerspectiveCamera to provide a realistic field of view.
- FirstPersonControls from the Three.js library are integrated to allow camera movement using keyboard inputs.
- - The camera can move forward (W), backward (S), and strafe left (A) or right (D) based on user input.
- Mouse interaction enables the camera to orbit around the helicopter, providing different viewing angles.

D. Continuous Animation

- The helicopter's rotor blades are animated to rotate continuously to simulate their movement during flight.
- The rotation of the blades is achieved by incrementing the rotation angle at each animation frame.

V. Future Work

The helicopter can be flown and controlled by user interaction. Implementing collision detection to prevent the helicopter from passing through objects or the ground. Enhancing the helicopter model and adding more details to improve visual realism. Adding additional interactive elements, such as buttons to toggle features like light rotation or helicopter engine sound.

VI. Conclusion

The Three.js Helicopter Simulation project successfully implements a visually appealing and interactive helicopter simulation. Users can control the camera's movement, explore the environment, and enjoy the realistic rotation of the helicopter's rotor blades. This project provides a foundation for further development, offering the potential for adding advanced features and refining the simulation for a more immersive experience

References

[1] https://www.jakobmaier.at/posts/flight-simulation/

[2] https://github.com/rubenandrebarreiro/3d-object-modelling-flying-helicopter.git.