Augmented Reality (AR) Pet Simulator

https://github.com/nguyenmx/ARVirtualPet

Anyah Arboine & Mary Nguyen

Introduction

The main goal of our final project is to create an interactable virtual pet using React Native and Three.js. In addition to the main functionalities of interacting with the virtual pet through rotating the model by the x, y, and z axis, we will integrate augmented reality (AR) features to enhance the user experience. This will enable users to bring the virtual pet into their real-world environment, similar to the AR experiences seen in popular mobile games like Pokémon Go.

React-Three-Fiber

React Native is a JavaScript library and framework that allows you to create apps that are compatible with both iOS and Android devices. Since React Native builds on top of JavaScript, the React-Three-Fiber was created to bridge React Native and Three.js. The canvas, ambient light, and camera that are normally within the Three.js library are created as separate components and rendered within the HTML code of a component, providing efficient rendering and easy readability.

Rendering the 3D Model

The most important step to completing the project was rendering the 3D model onto the app. This step happened to be the most challenging as testing and compiling 3D models within the Expo Go app has not been well-documented and the project could not recognize pathways that ended in .obj and .glb. The issue was solved by creating a meto.config.js file, that way the compiler could recognize these files and successfully run through the Expo Go app. At first, we wanted to use the OBJ loader, but we opted to for useGLTF function within the React-Three-Drei

library to render 3D models through GLB files because the packages for this library were compatible with Expo Go.

The project also includes several sliders that allow the user to adjust the model's scale, rotation, and contrast manually, to further customize the 3D model's position. These sliders are implemented by creating a custom slider component that will adjust the minimum and maximum of the the sliders based on the type and pass these values to the model's rotation on change. The model rotates by creating a reference to the model and utilizing the useEffect() method to render the new rotation and adjust the rotation sensitivity by multiplying a small integer. We've also included a particle system for visual appeal whether the user is in camera mode or taking care of their pet in the game mode. The ParticleSystem is a custom component that initializes a large number of particles, assigns initial positions and velocities to them, and updates their positions over time to create an animated particle effect by using BufferGeometry and BufferAttribute from the Three library in Expo Go. The user can turn on and off the particle effect by clicking on the 'Particles' button on the screen.

Implementing AR Features

Expo Go has ended support for AR libraries, therefore, we decided to extract the gyroscope data of the phone and adjust the rotation of the 3D model accordingly. The gyroscope data consists of a three-coordinate system (x, y, and z). The movement of the 3D model will depend on the phone's x and z position. If the phone is tilted to the right, the model will turn on the y-axis in the clockwise direction (-y) to see more of the model's right side. Having a positive z position of the phone will tilt the model in the downward (-x) direction to see the top of the model's head.

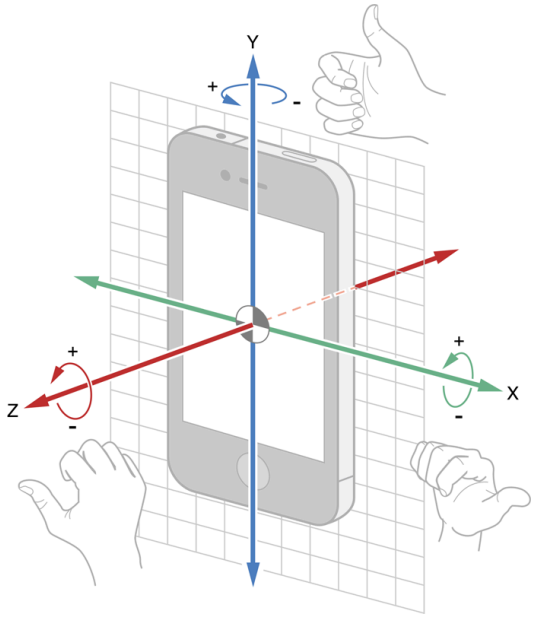

Figure 1. Represents how the phone's x, y, and z coordinates are mapped based on the movement and tilt of the phone

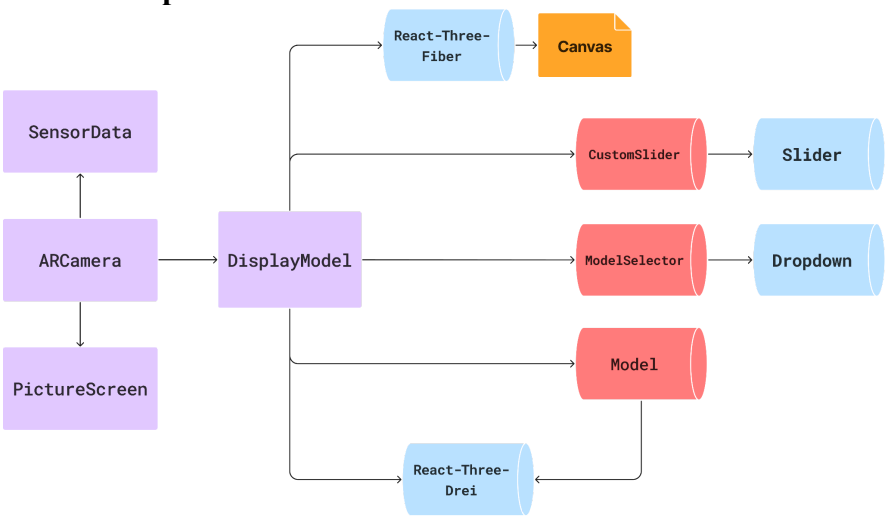

Figure 2. Represents the overarching design and structure of the program. The purple boxes represent the UI components that are necessary for rendering the interfaces and passing inputs to the backend. The red represents the components that handle the backend and functionality of the project. The blue represents the helper libraries.

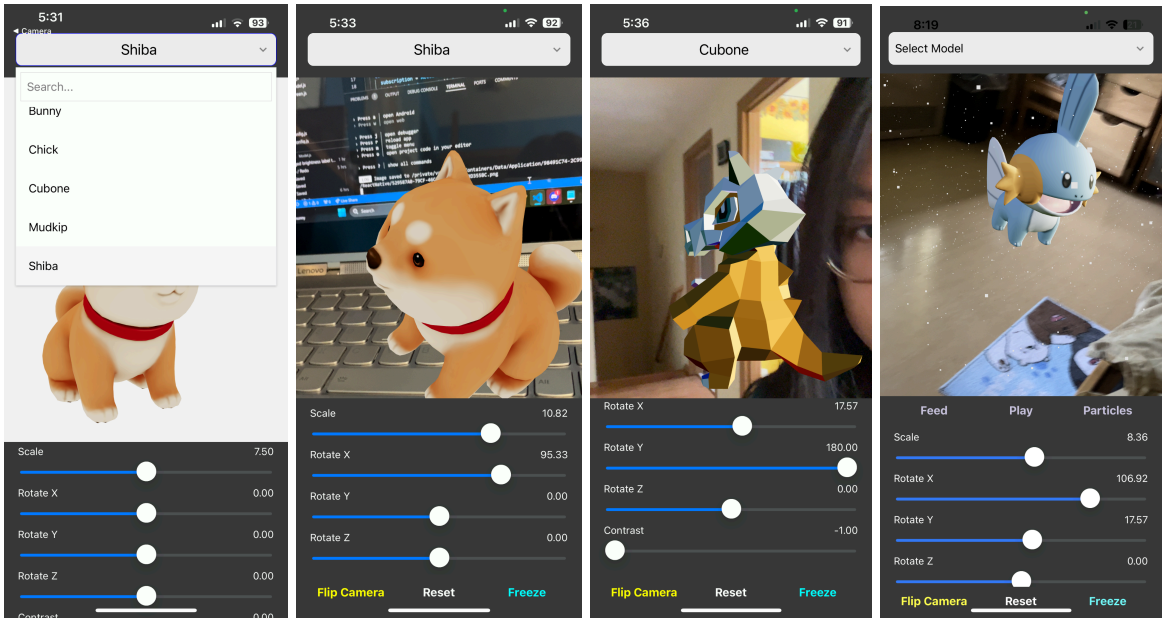

Figure 3. Shows the final results of the project. The user can search through various 3D models, view particles, play different animations, and view the model within their environment by accessing the phone's back or front camera.