Creating a VR Game in Unity with Oculus Integration

TEAM MEMBERS

Atharva Chadha - SE20UARI025
Siddartha Rahul kumbagiri - SE20UARI084

Introduction:

This report documents the step-by-step process of developing a VR game in Unity, incorporating Oculus Integration, XR Interaction Toolkit, and various other elements to create a compelling virtual experience. The creator has prior experience in Unity, having worked on simple games and experimented with ragdoll physics.

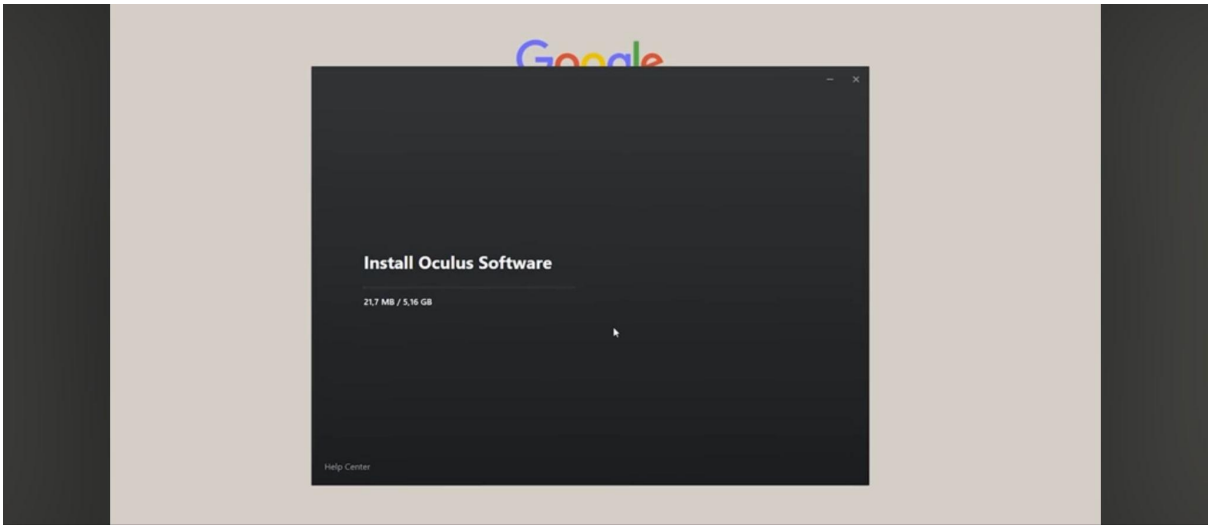
Prior Experience:

The project benefits from the creator's familiarity with Unity, where they have previously developed simple games and explored the intricacies of ragdoll physics. This existing knowledge base serves as a foundation for the more advanced features introduced in the VR game development process.

Steps in Detail:

1. Installing Oculus Software:

We Begin by installing the Oculus software on the development machine. This step is crucial for enabling Oculus headset compatibility with the Unity project.

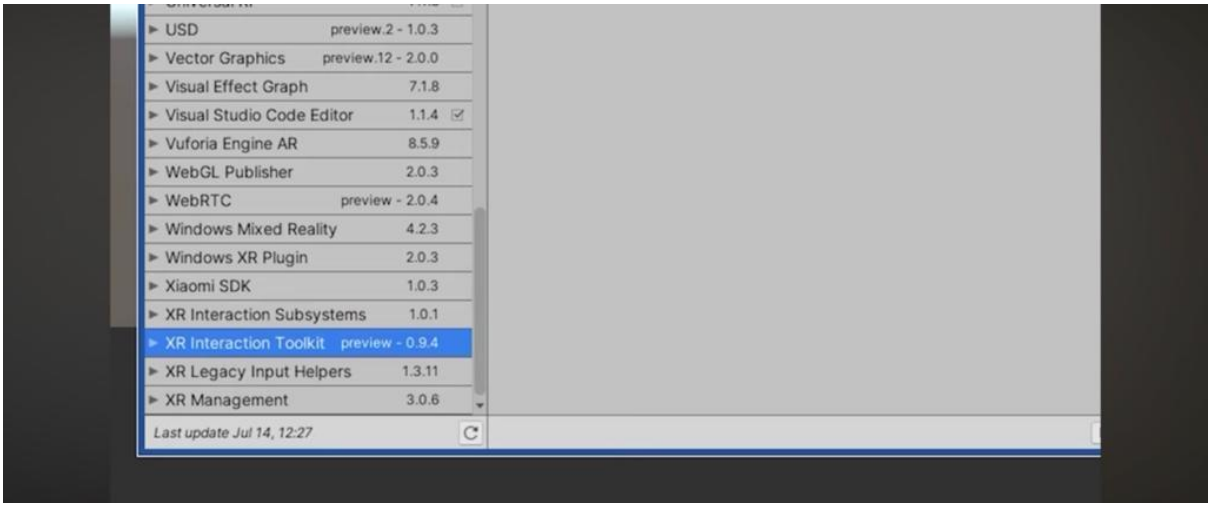


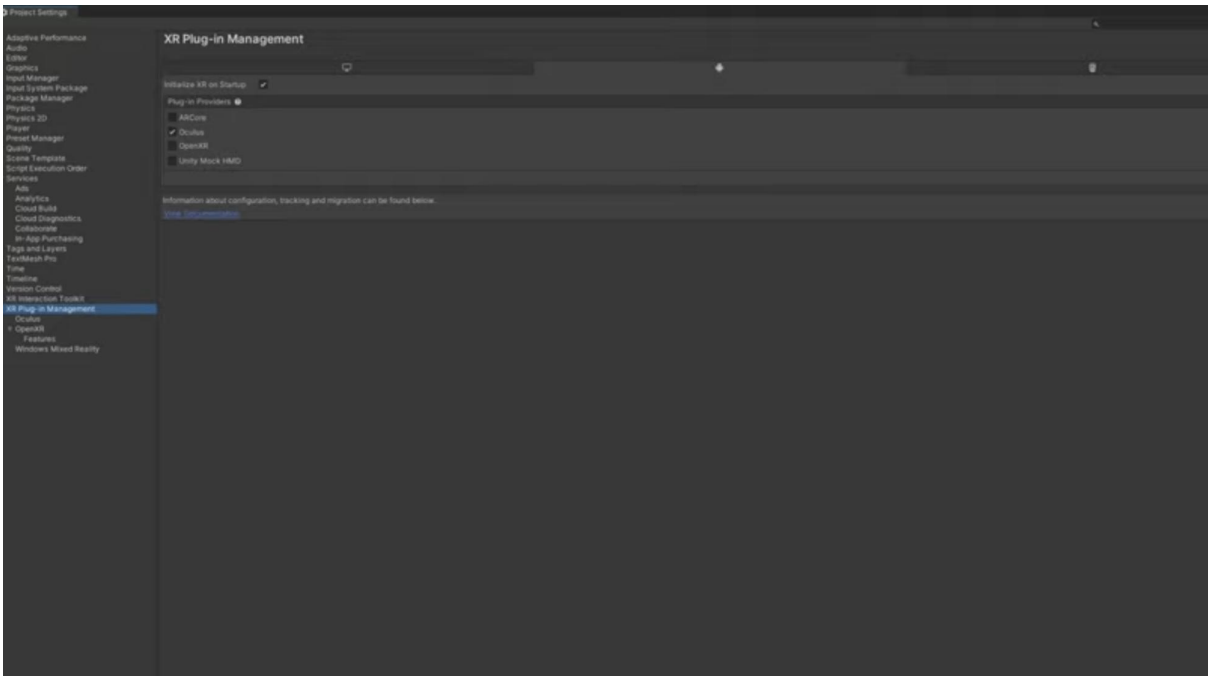
2. Creating a New 3D Project in Unity:

Launch Unity and initiate a new 3D project, ensuring compatibility with the XR Interaction Toolkit and Oculus Integration.

3. Importing XR Interaction Toolkit 0.9.4:

Download and import the XR Interaction Toolkit 0.9.4 package into the Unity project. This toolkit provides essential components for creating immersive VR interactions.

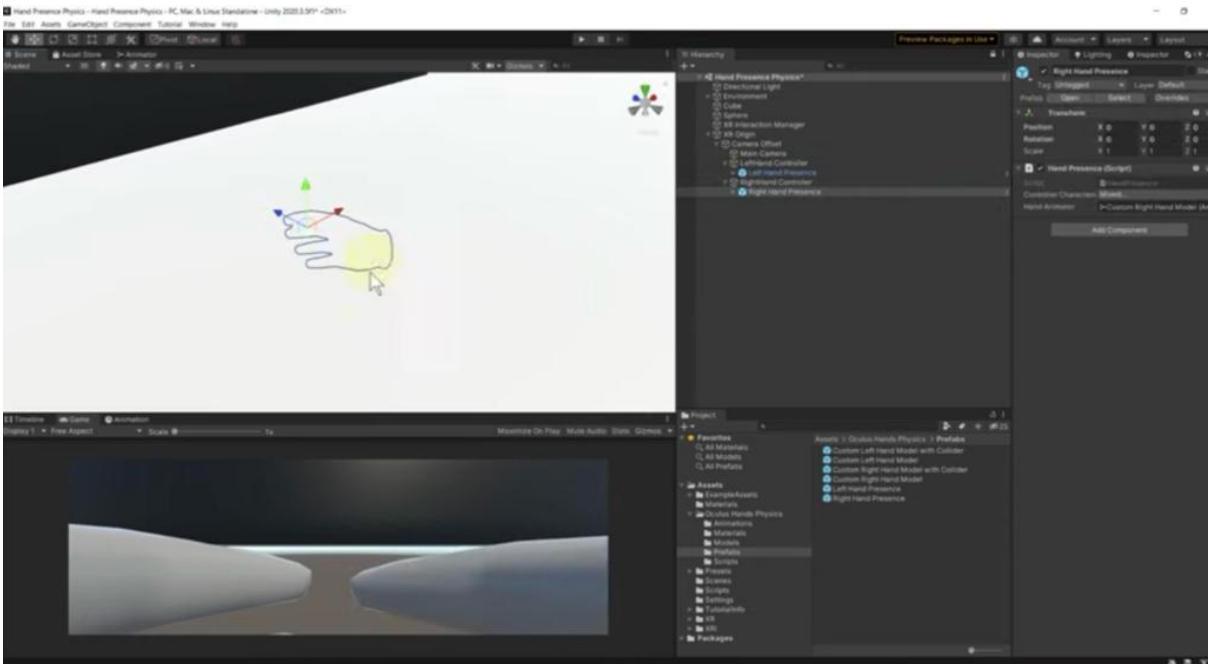




4. Importing Oculus Hand Models:

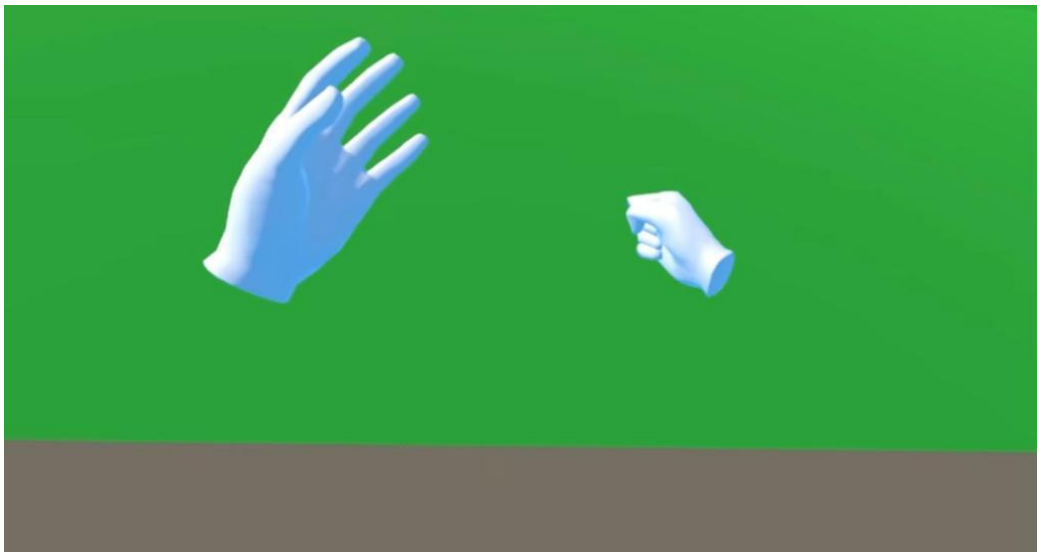
Integrate Oculus Hand Models into the project, this offering realistic representations of hands for VR interactions with preloaded physics.

https://www.patreon.com/posts/free-oculus-vr-46544401



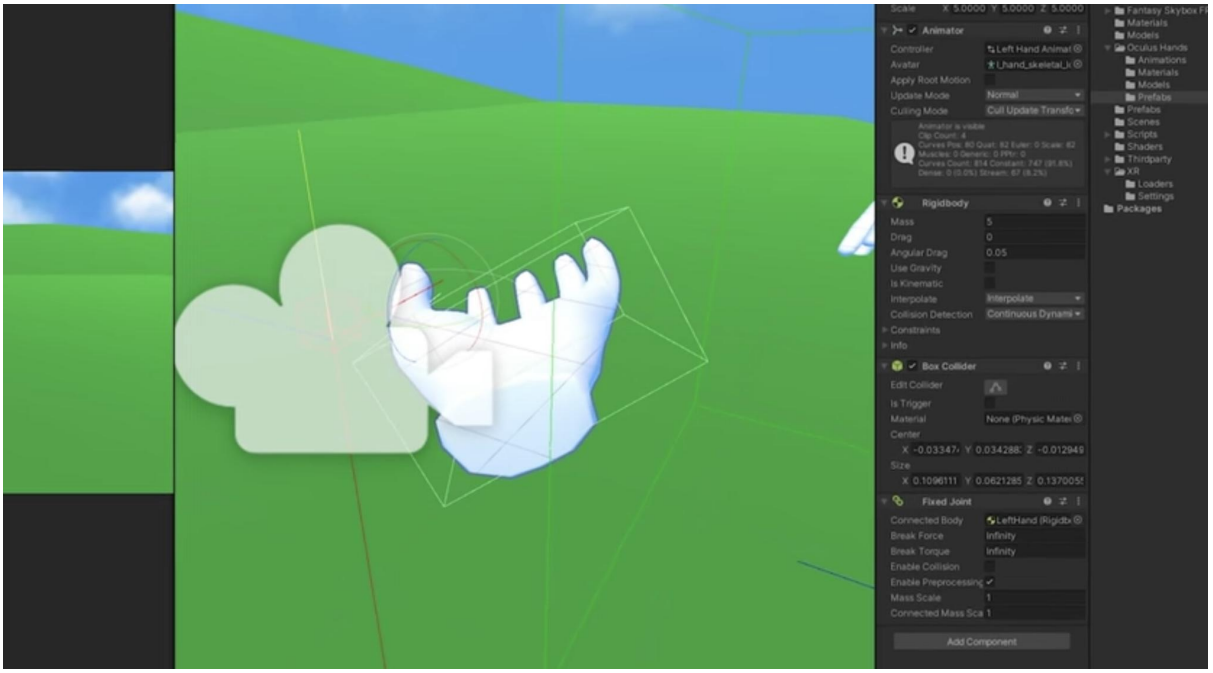
5. Setting Up Hands in Unity VR:

We followed a tutorial by VALEM on YouTube for guidance on configuring hands in Unity VR, ensuring accurate and responsive hand interactions.



6. Making Hands Their Own Physics Object:

Configure Oculus hands as physics objects within Unity, enabling realistic physical interactions with the virtual environment.



7. Adding Rocks and Combining Motion:

Import rock models into Unity and synchronise their motion by combining the movement of both hands. We utilised Unity's particle system for animations and effects.

8. Importing Blender Model of a Caveman:

We Downloaded a Blender model of a caveman and imported it into Unity. We again set it up as a separate physical object, incorporating ragdoll physics for a lifelike response.

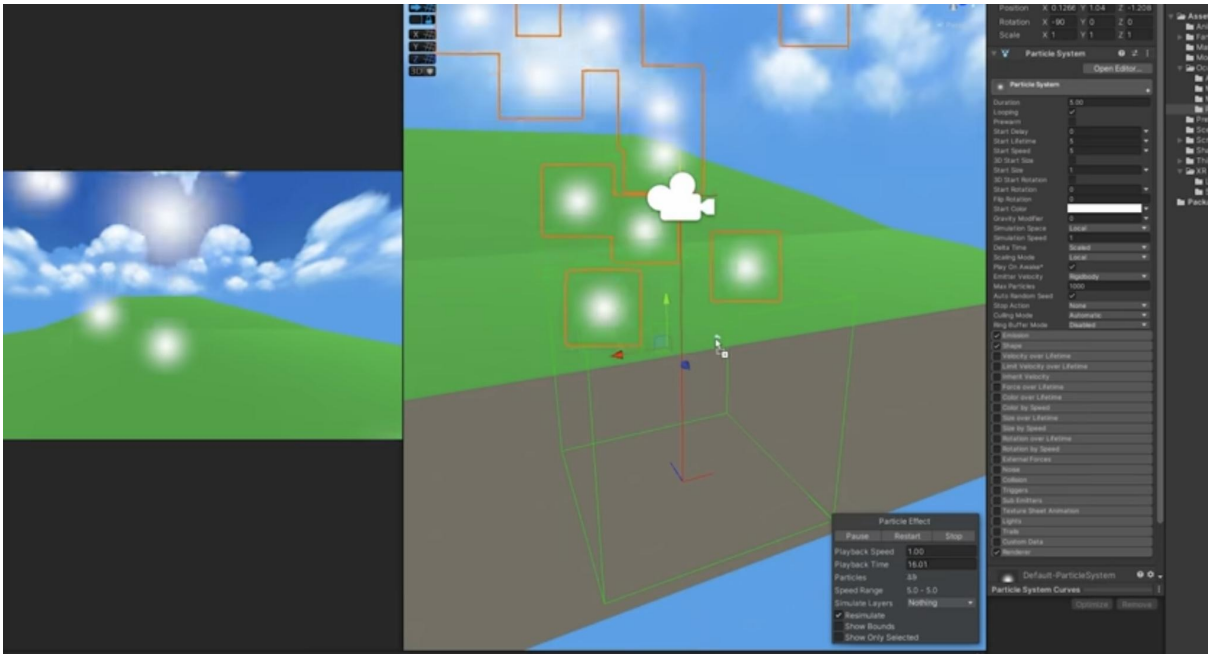
9. Using CEL Shader for Edge Outlining:

We implemented the CEL Shader for edge outlining effects, enhancing visual appeal. We referred to the provided GitHub link for detailed instructions and help.

https://github.com/daniel-ilett/cel-shading

10. Running Animation for Ragdolls and Using Unity Particle System:

Develop animations for the caveman's ragdoll physics, ensuring seamless and responsive movements based on user interactions.



11. Creating a Scene with Randomly Generated Terrains:

We then built a diverse scene, such as a small island, and introduced randomly generated terrains with trees and other elements. We Leveraged Unity's terrain tools for a dynamic and immersive environment.

Conclusion:

This report highlights progression from basic Unity game development to crafting a VR experience with Oculus Integration. The combination of prior experience, new tools, and detailed steps illustrates the iterative and expanding nature of the creator's Unity development skills