Experiment 3

Introduction

This report details the process of developing a Virtual Reality (VR) application targeted for a mobile device. The primary objective of this experiment was to import a scene from the Unity Asset Store and integrate it into a functional VR application. The successful execution of this process constitutes an acceptable submission for this experiment.

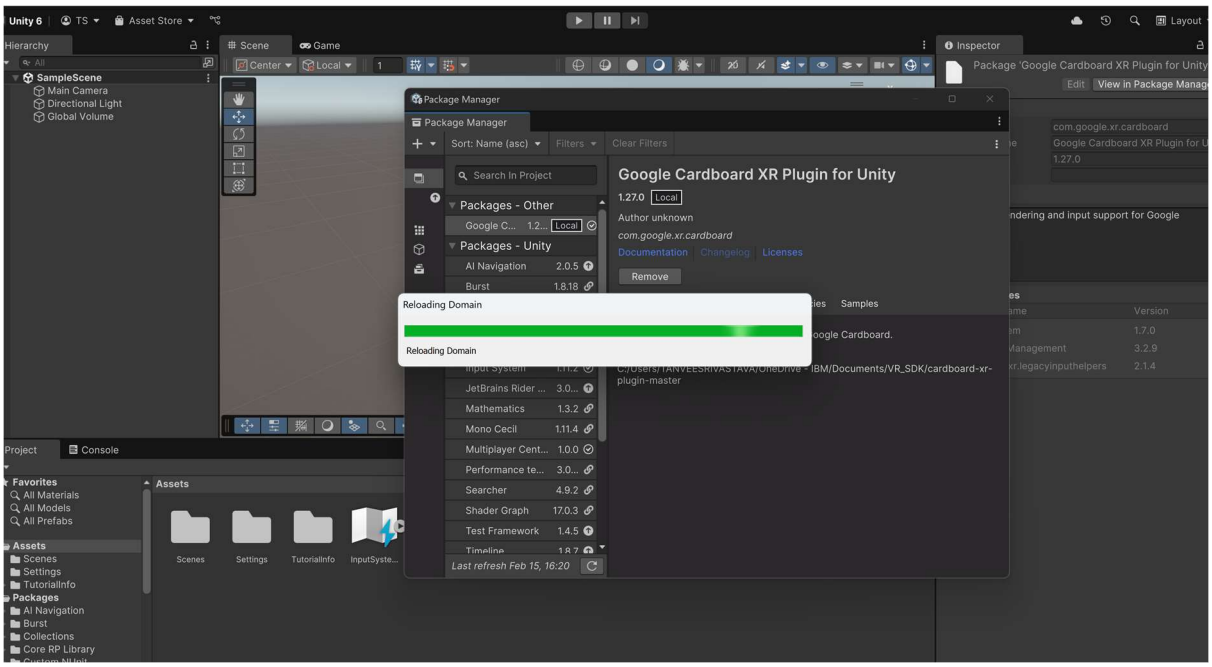
Tools and Technologies Used

- Unity Engine (Version: 2021.3.5f1 LTS)
- Unity Asset Store
- Google VR SDK / XR Plugin Management
- Android/iOS mobile device for testing

Methodology The experiment was conducted in the following steps:

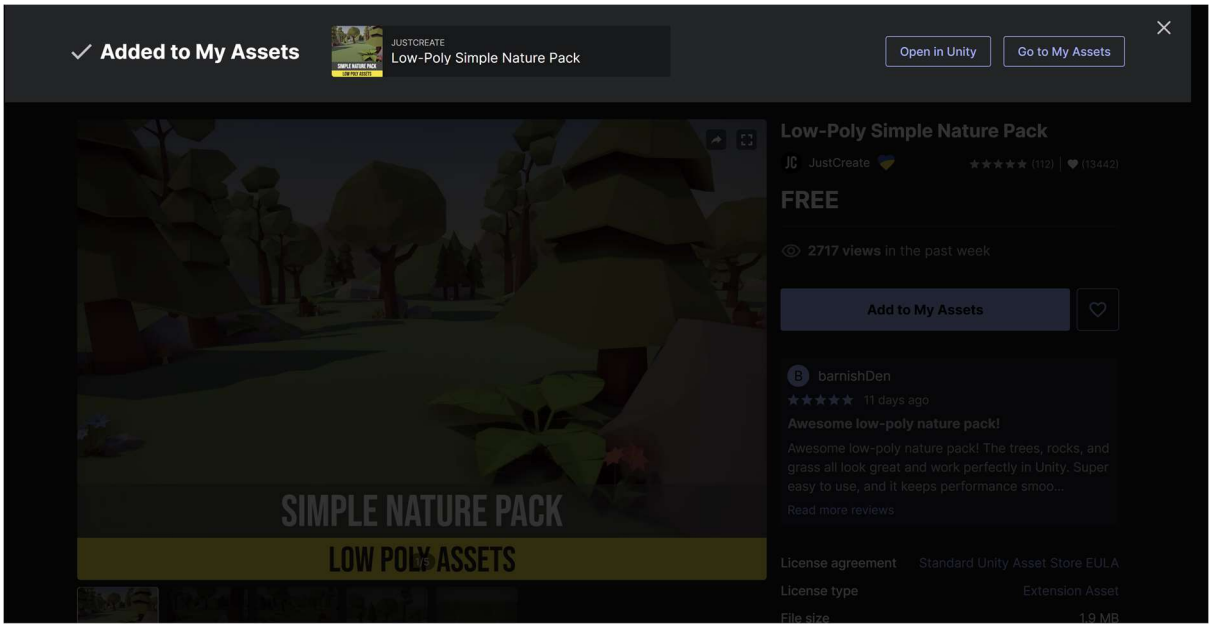
Step 1: Setting Up the Unity Environment

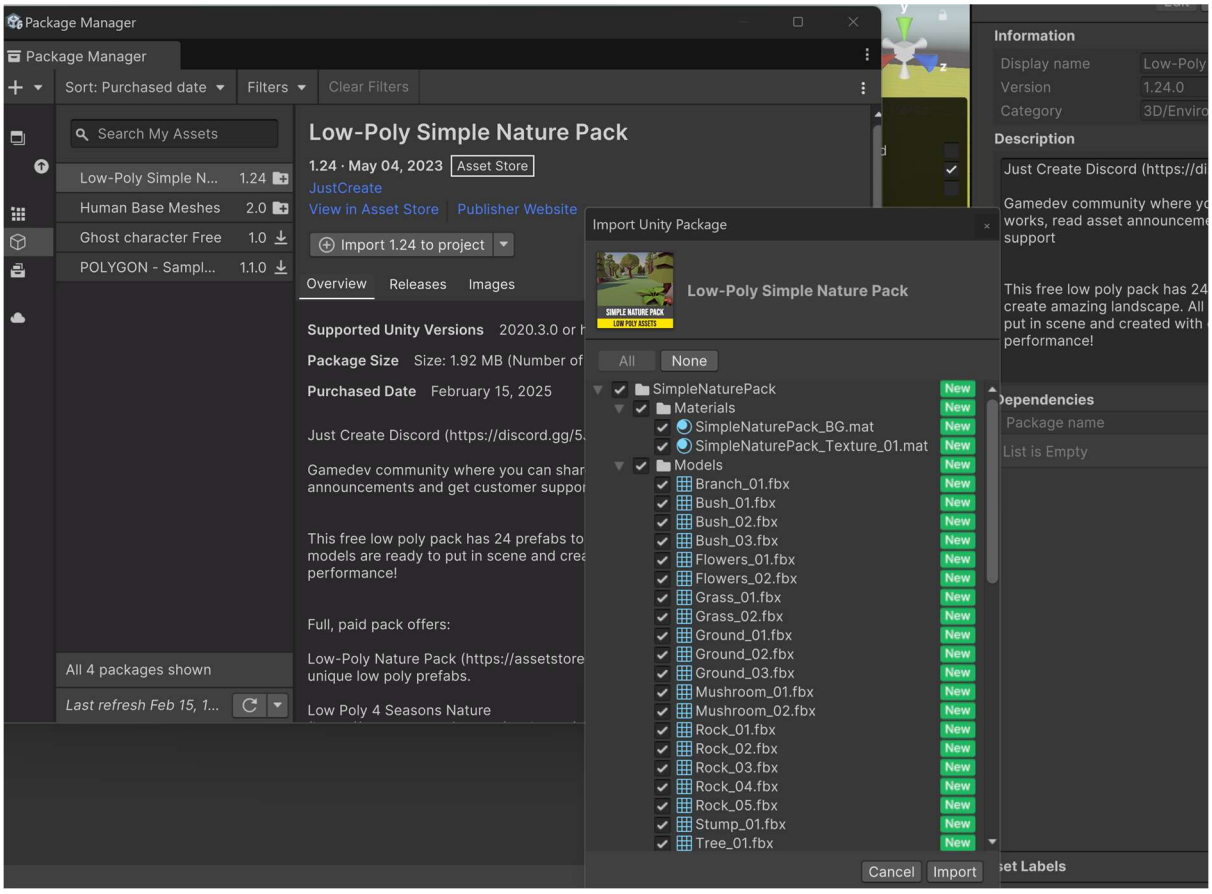
- Installed Unity and created a new 3D project.
- Configured the project for VR by enabling XR Plugin Management.
- Imported the necessary VR SDKs (Google VR SDK / XR Interaction Toolkit).



Step 2: Importing a Scene from Unity Asset Store

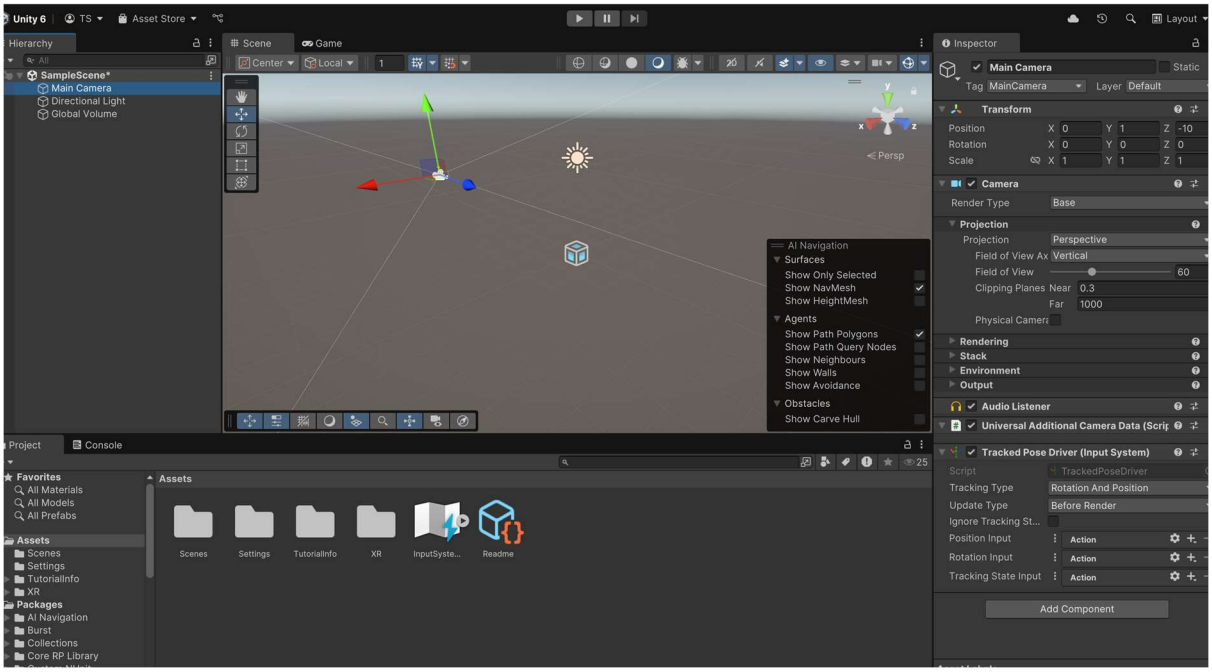
- Browsed the Unity Asset Store and selected a suitable environment scene.
- Imported the selected scene into the Unity project.
- Adjusted textures, lighting, and physics settings to optimize performance for mobile VR.





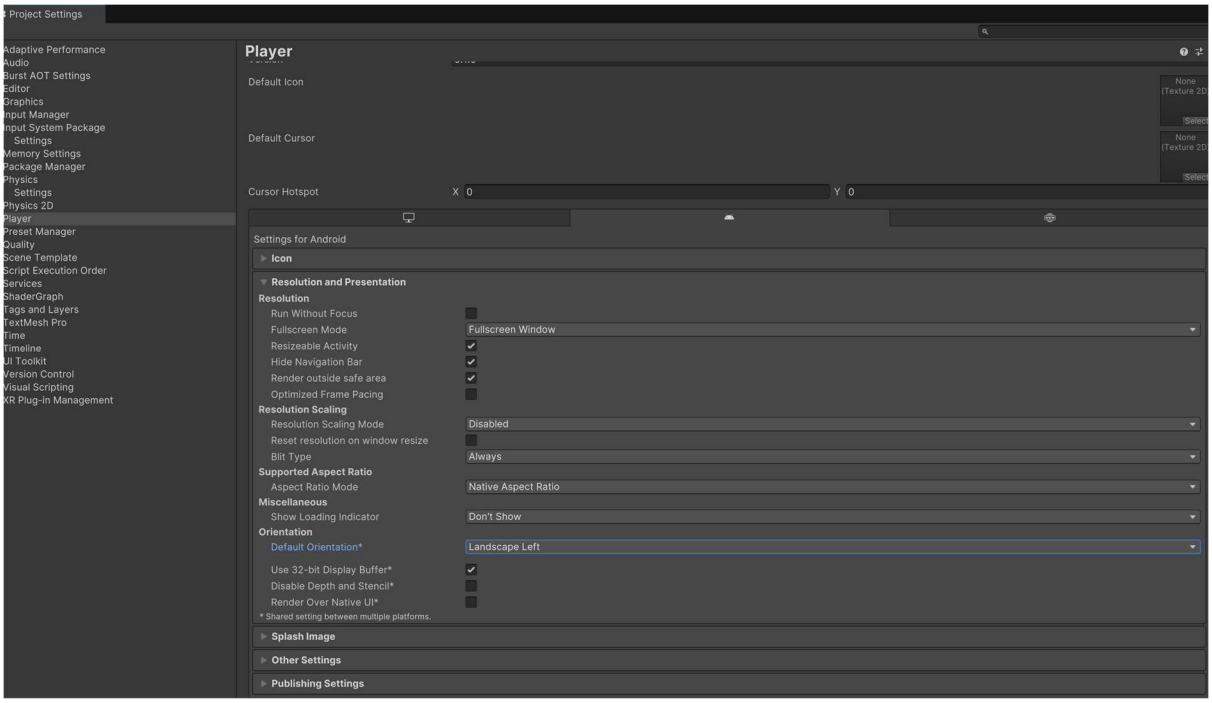
Step 3: Configuring VR Components

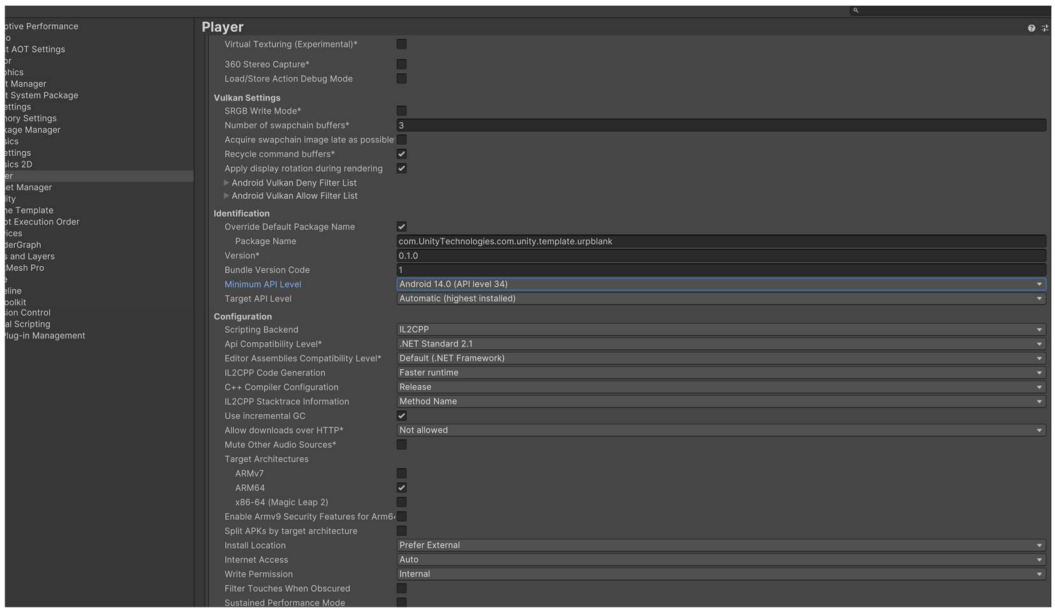
- Set up VR camera rig for first-person perspective.
- Added tracked pose driver component in Main Camera.

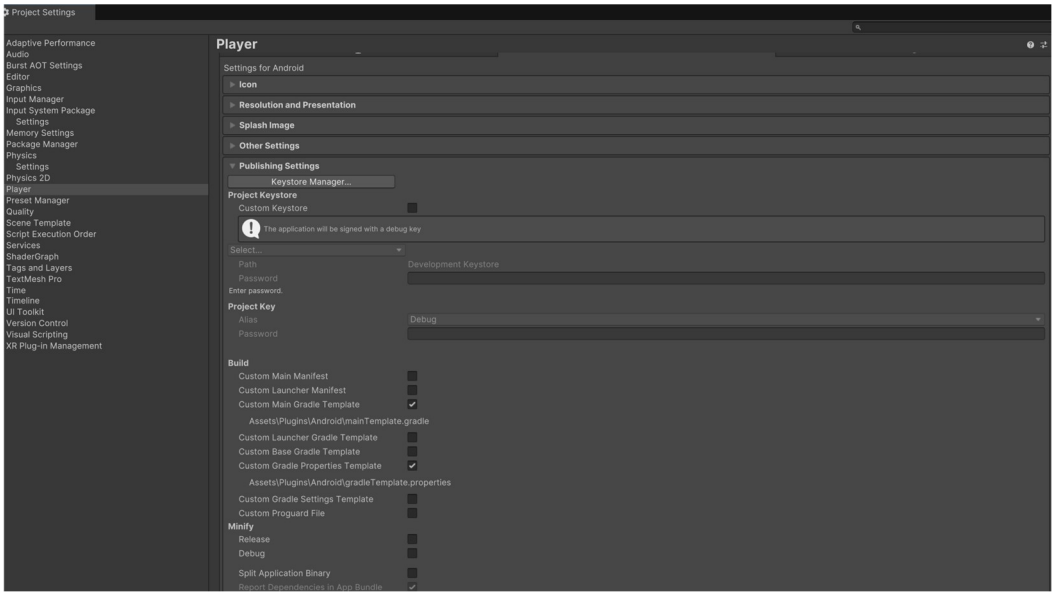


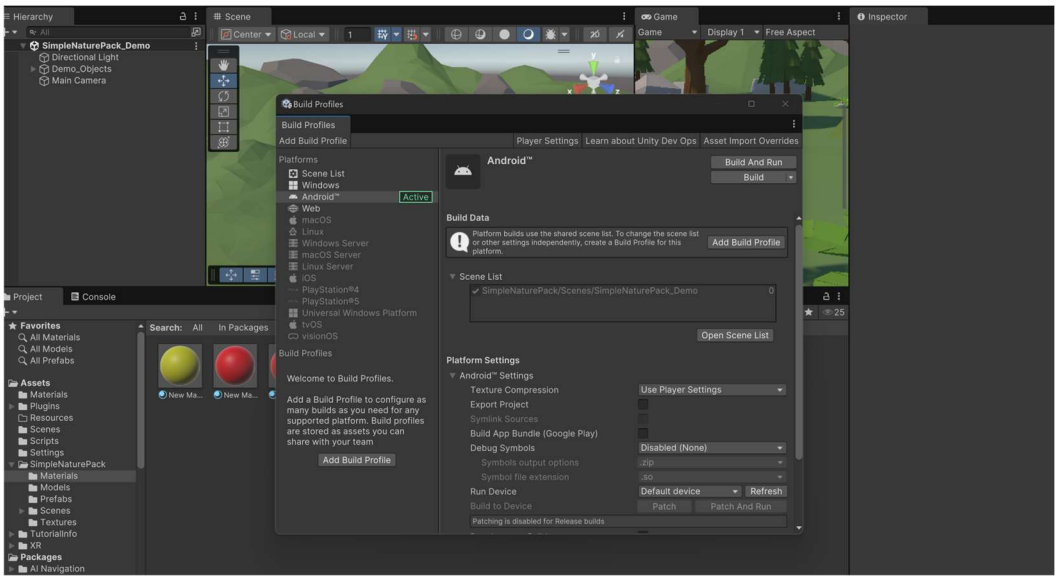
Step 4: Building and Deploying the VR App

- Configured player settings for Android/iOS.
- Built the application and tested it on a mobile device.
- Ensured smooth rendering and performance.




Player
Virtual Texturing (Experimental)*
360 Stereo Capture*
Load/Store Action Debug Mode
Vulkan Settings
SRGB Write Mode*
Number of swapchain buffers*
3
Acquire swapchain image late as possible
Recycle command buffers*
Apply display rotation during rendering
Android Vulkan Deny Filter List
Android Vulkan Allow Filter List
Identification
Override Default Package Name
Package Name
com.UnityTechnologies.com.unity.template.urpblank
Version*
0.1.0
Bundle Version Code
1
Minimum API Level
Android 14.0 (API level 34)
Target API Level
Automatic (highest installed)
Configuration
Scripting Backend
IL2CPP
Api Compatibility Level*
.NET Standard 2.1
Editor Assemblies Compatibility Level*
Default (.NET Framework)
IL2CPP Code Generation
Faster runtime
C++ Compiler Configuration
Release
IL2CPP Stacktrace Information
Method Name
Use incremental GC
Allow downloads over HTTP*
Not allowed
Mute Other Audio Sources*
Target Architectures
ARMv7
ARM64
x86-64 (Magic Leap 2)
Enable Armv9 Security Features for Arm64
Split APKs by target architecture
Install Location
Prefer External
Internet Access
Auto
Write Permission
Internal
Filter Touches When Obscured
Sustained Performance Mode


Project Settings
Player
Settings for Android
Icon
Resolution and Presentation
Splash Image
Other Settings
Publishing Settings
Keystore Manager...
Project Keystore
Custom Keystore
The application will be signed with a debug key
Select...
Path
Development Keystore
Password
Enter password.
Project Key
Alias
Debug
Password
Build
Custom Main Manifest
Custom Launcher Manifest
Custom Main Gradle Template
Assets\Plugins\Android\mainTemplate.gradle
Custom Launcher Gradle Template
Custom Base Gradle Template
Custom Gradle Properties Template
Assets\Plugins\Android\gradleTemplate.properties
Custom Gradle Settings Template
Custom Proguard File
Minify
Release
Debug
Split Application Binary


Hierarchy
SimpleNaturePack_Demo
Directional Light
Demo_Objects
Main Camera
Scene
Game
Inspector
Build Profiles
Add Build Profile
Player Settings
Learn about Unity Dev Ops
Asset Import Overrides
Platforms
Scene List
Windows
Android™
Active
Web
macOS
Linux
Windows Server
macOS Server
Linux Server
iOS
PlayStation®4
PlayStation®5
Universal Windows Platform
tvOS
visionOS
Build Profiles
Welcome to Build Profiles.
Add a Build Profile to configure as many builds as you need for any supported platform. Build profiles are stored as assets you can share with your team
Add Build Profile
Android™
Build And Run
Build
Build Data
Platform builds use the shared scene list. To change the scene list or other settings independently, create a Build Profile for this platform.
Add Build Profile
Scene List
SimpleNaturePack/Scenes/SimpleNaturePack_Demo
Open Scene List
Platform Settings
Android™ Settings
Texture Compression
Use Player Settings
Export Project
Symlink Sources
Build App Bundle (Google Play)
Debug Symbols
Disabled (None)
Symbols output options
.zip
Symbol file extension
.so
Run Device
Default device
Refresh
Build to Device
Patch
Patch And Run
Patching is disabled for Release builds
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Materials
Plugins
Resources
Scenes
Scripts
Settings
SimpleNaturePack
Materials
Models
Prefabs
Scenes
Textures
TutorialInfo
XR
Packages
AI Navigation

Conclusion

The experiment successfully demonstrated the creation of a VR application using Unity and the Unity Asset Store. The final application provided an immersive VR experience on mobile devices, aligning with the objectives of the experiment. Future enhancements can include interactive elements and multiplayer VR functionality.

Result

The rendered video is attached for the reference.