

# Final Deliverable

Rose Garden XR

## Unity Game (XR Version)

**Github:** <https://github.com/br-zhou/Rose-Garden-VR>

### Development Environment:

Unity            6000.0.37f1 LTS  
Computer      ROG Strix G16 (RTX 4060 Laptop GPU, I9-1400HX, 16GB ram)  
OS            Windows 11 Home 24H2 (26100.3775)

### Project Setup:

Step 1. Clone the repository

Step 2. Open the project in Unity

- Make sure to install 6000.0.37f1 LTS with android build support

Step 3. Download the Firebase Unity SDK from here:

- <https://firebase.google.com/download/unity>

Step 4. Import the **FirebaseDatabase.unityEngine** into the project

Step 5. Import FireBase API keys into Assets\StreamingAssets

- Ignore this step if you don't have a Firebase backend set up yet.
- This tutorial gives a visual for steps 3-5: [Firebase Unity Integration Tutorial](#)
- If you want to skip the Firebase Unity Integration, ignore this step

Step 7. Meta Quest Developer Mode Setup

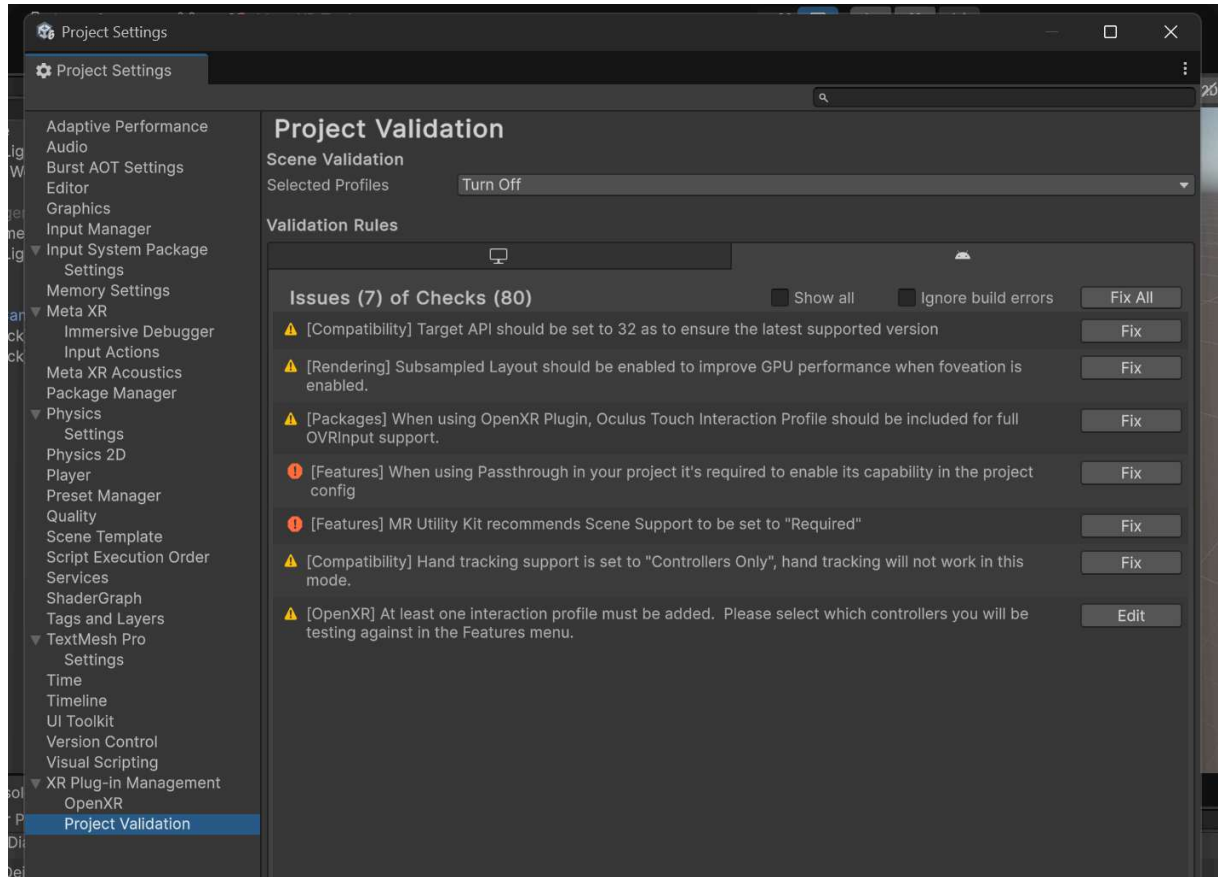
- Follow this tutorial to install the required settings to develop for Meta Quest: [Meta Quest Unity Setup](#)

Step 8. Additional Extensions

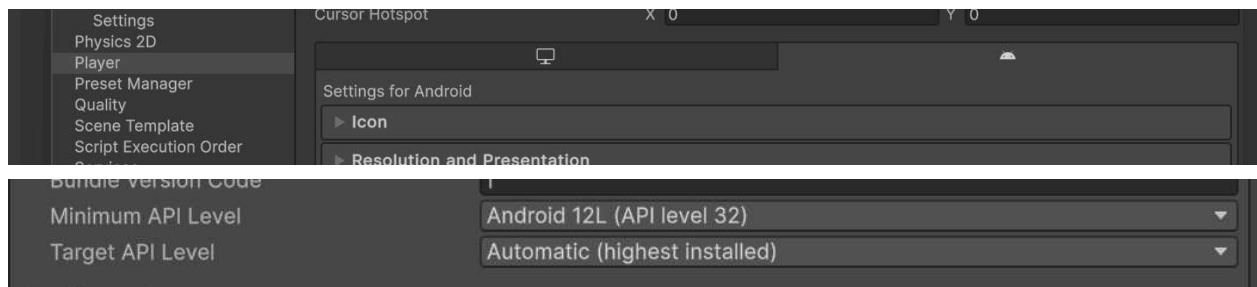
- These extensions are bundled with the project repository, but I've linked them just in case.
- [Unity Meta Quest SDK](#)

Step 9. Project Validation

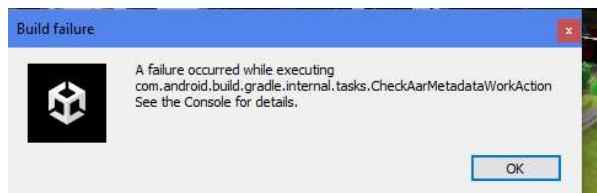
- Make sure to fix any build issues in the project settings!
- Example Screenshots on Next Page



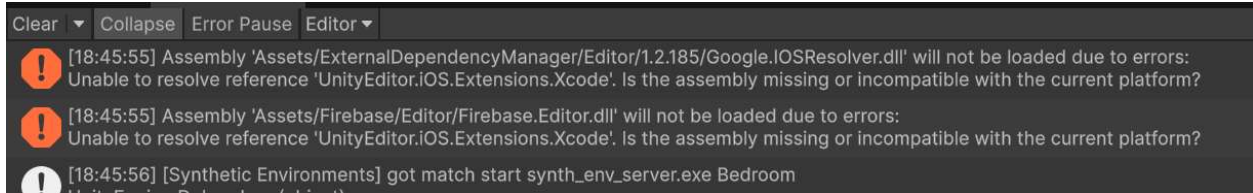
**Tip:** Make sure to resolve any red errors.



**Build Issues:** When building for the meta Quest, you need to set your Android API level to the ones above. Builds will fail otherwise.



(You will see this error if you don't set the correct API level)



**Expected Errors:** You will see two errors, when running don't worry! They're related to the IOS builds for Firebase. Since the Quest 3 is an android platform, we can safely ignore them.

**Note:** A full copy of the XR project files has also been cloned onto one of the EML desktops's external drives.

**Practical Warnings:** Make sure to update your drivers when developing! The newer 40 series graphics cards have lots of graphical issues and crashes often. These issues are not present in older 20 series cards. It is good to develop on a more stable system, but it is better to not develop on Unity at all. Meta silently pushes updates which have conflicted with existing drivers, breaking development builds.

## Unity Game (FPS Version)

This is a FPS (first-person shooter) version of the viewfinder project. We have two versions of our game we keep in synchronization because developers on Mac can not directly run VR. The only difference between the two projects is that XR uses a VR headset to run and FPS is controlled using mouse and keyboard. Usually, changes are first made in FPS then copied to XR.

### Code & Assets

Github: <https://github.com/Amon3141/rose-garden-viewfinder-fps>

Below are the locations of important assets:

/Assets/Objects	3D objects
/Assets/Material	Materials for the 3D objects
/Assets/Scripts	C# scripts
/Animation	Animation Controllers
/Firebase	Database-related

More detailed descriptions of each script are written in the [README.md](#) file.

### Development Environment

Unity	6000.0.36f1 LTS
Computer	MacBook Pro (2020), Apple M1 chip, 16GB
OS	macOS Sequoia 15.4

## Project Setup

Step 1. Clone the repository

Step 2. Open the project in Unity

- Make sure to install 6000.0.36f1 LTS with android build support

Step 3. Download the Firebase Unity SDK from here:

- <https://firebase.google.com/download/unity>

Step 4. Import the **FirebaseDatabase.unpackage** into the project

Step 5. Import FireBase API keys into Assets\StreamingAssets

- Ignore this step if you don't have a Firebase backend set up yet.
- This tutorial gives a visual for steps 3-5: [Firebase Unity Integration Tutorial](#)
- If you want to skip the Firebase Unity Integration, ignore this step.

## Mobile Web App

Framework & Libraries: React 19.0.0, Vite, TypeScript

### Testing Environment

- Browser: Chrome
- Screen Size: Responsive layout tested on desktop and mobile viewports

Github: <https://github.com/zhanginc/message-app-mdia-470>

- Link to the github for the frontend application
- Assets for the web application are within the public folder in the github repository

Create Firebase Project: <https://support.google.com/appsheet/answer/10104995?hl=en>

- Comprehensive tutorial on how to setup a Firebase project

### Env File for Vite

Place credentials in a `.env` file like this:

```
VITE_FIREBASE_API_KEY=  
VITE_FIREBASE_AUTH_DOMAIN=  
VITE_FIREBASE_PROJECT_ID=  
VITE_FIREBASE_STORAGE_BUCKET=  
VITE_FIREBASE_MESSAGING_SENDER_ID=  
VITE_FIREBASE_APP_ID=  
VITE_FIREBASE_MEASUREMENT_ID=
```

More info on securing API keys: [Securing Firebase Keys in Vite](#)