

IUNO

XR | CULTURE | AI

Innovation Centre for
Intelligent and Immersive
Technologies in the
Independent Cultural Scene

Tutorials

Tutorial XR

Best Practice in Unity

Christoph Holtmann, HTW Berlin

June 2026



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



Best Practice in Unity

Welcome to our tutorial! Let's dive in...



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



June 2026, 2

Plan & set up your project

Decisions and groundwork that are hard to change later



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



June 2026, 3

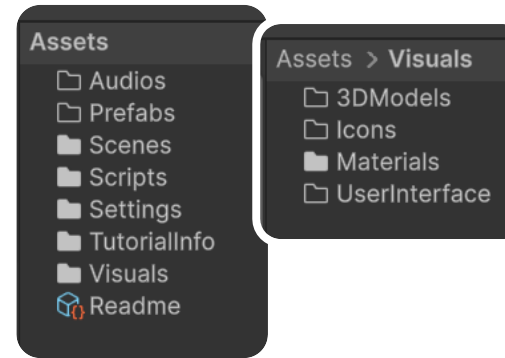
Project Structure

A clean project starts with a clean structure.

1. Use clear, consistent folder names: **Scenes, Scripts, Prefabs, Materials, Textures, Audios, Models.**
2. Agree on naming conventions and write them down e.g. **PascalCase** for scripts, a prefix per type.
 1. Unity specific conventions:
<https://unity.com/how-to/naming-and-code-style-tips-c-scripting-unity>
3. Know your file types: **.unity** scenes, **.prefab** prefabs, **.asset** data, plus imported media.

Tip: a structure everyone understands saves hours of searching later.

Note: keeping a clear structure over time takes discipline – revisit and tidy it regularly.



Pascal case (PascalCase)

Pascal case is a variation of camel case, where the initial letter is capitalized. Use this for class, public fields and method names in Unity development. For example:

`ExamplePlayerController`

`MaxHealthPoints`

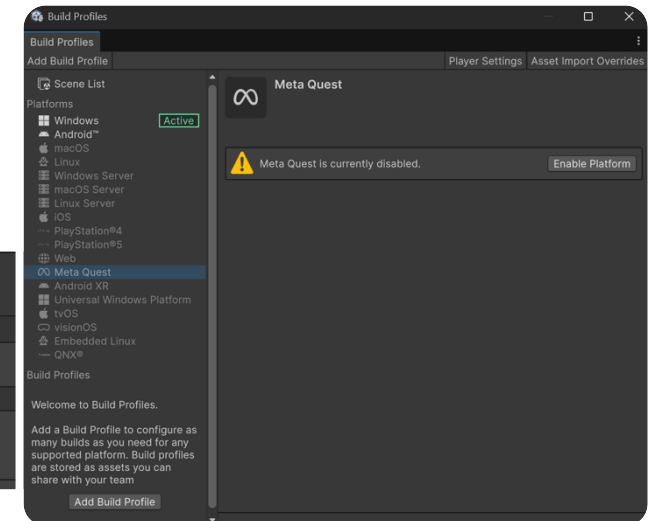
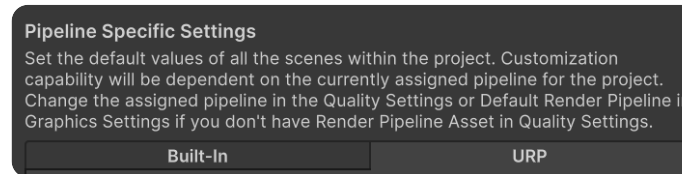
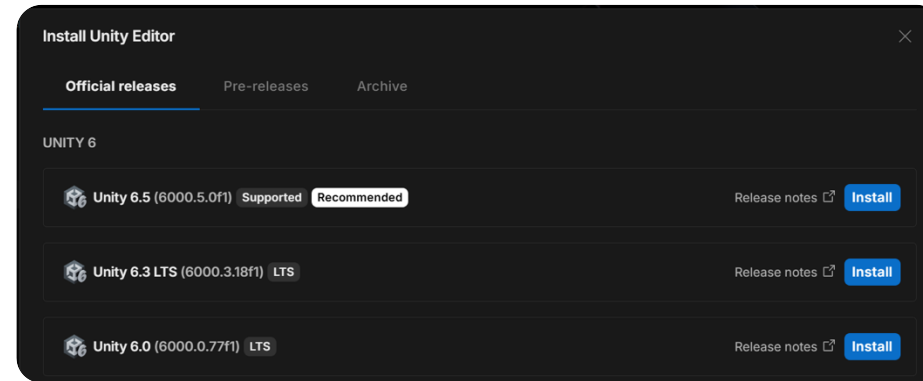
`EndOfFile`

Decide Early

Some decisions are very hard to change later. Make them once, as a team, at the very beginning:

1. **Unity version** – pick one LTS version and make sure everyone installs exactly the same one.
2. **Render pipeline** – Built-in, URP or HDRP. Switching later can break materials and lighting.
3. **Frameworks (if needed)** – e.g. OpenXR / the device toolkit you target.
4. **Target device(s)** – decide what you build for before you design for it.

Communicate them to the whole team and inform the team whenever versions change.

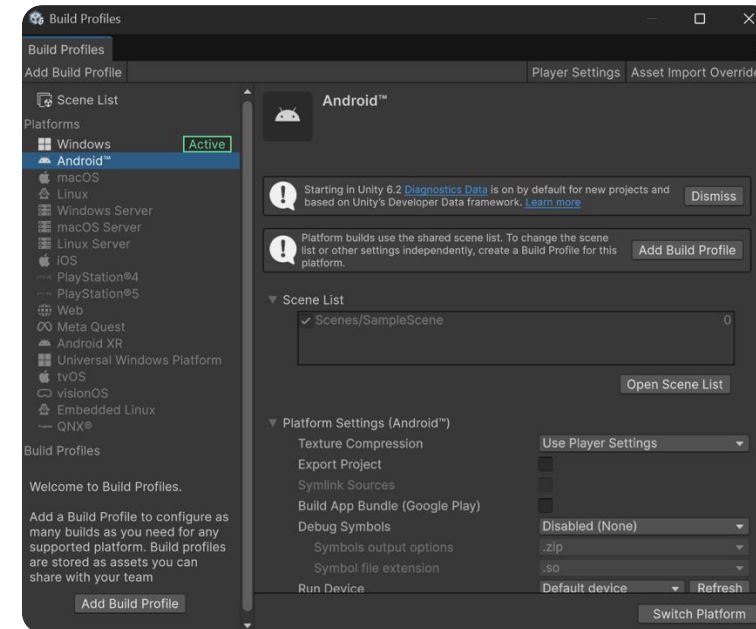


Set Up the Build Pipeline

Set up a working build for your target device as early as possible and not only at the end.

1. Make sure every team member can **build** and **run** the project themselves.
2. **Test on the real device** continuously, not just in the Editor.
3. Automate or document the build steps so a build is never a mystery.

*Why: problems found on day 2 are cheap.
Problems found the night before a deadline are not.*



Organise the team & the work

Working together without stepping on each other



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



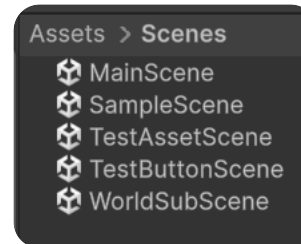
June 2026, 7

Structure the Workflow

Not everyone should work chaotically on the same scene. Organise the work instead:

1. **Distribute tasks** clearly – everyone knows what they own.
2. **One person owns the master scene** and merges changes into it.
3. **Use an issue board** (e.g. GitHub Projects, Trello) to track tasks and progress.
4. **Hold a regular jour fixe** to sync, unblock and re-plan.

Clear ownership prevents merge conflicts.



Google Gemini

Detailed Storyboards

Before building, make the idea concrete with storyboards, both visual and written.

1. **Sketch** what the user sees.
2. **Describe in text** what happens, what the user does and how the app reacts.
3. **Cover interactions and edge cases**, not only the happy path.

A shared storyboard keeps the whole team building the same experience.



Google Gemini

Prototype First

Test risky features in isolation before integrating them.

1. **Build small, functioning prototypes** of a feature, interaction or visual concept first.
2. **Validate the idea** in a separate test scene or project and on the target device.
3. **Only then integrate** the proven version into the main project.



Google Gemini

***Never experiment directly in the main scene.
Keep it stable!***

Build, test & optimize

Prototype, test on device and keep it performant



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



June 2026, 11

Test on the Target Device

What feels right in the Editor can feel very different on the real device. Test early and often.

1. **Test regularly on the actual target device**, as part of an iterative workflow.
2. **Involve external people**, not just the team that built it.
3. **Watch how they use it**, note problems, then improve and test again.

Each test → fix → test cycle makes the experience noticeably better.



<https://www.gizchina.com/>

USER TESTING

Do users need my app?



USABILITY TESTING

Can users use my app?



WEAREBRAIN.COM

<https://dovetail.com/ux/user-testing/>



<https://www.meta.com/>

Performance from the Start

Performance is part of the concept, not an afterthought (especially in XR or on mobile devices).

1. **Keep digital media small** (textures, models, audio).
2. **Use prefabs** to reuse and manage objects efficiently.
3. **Be careful with lighting and heavy postprocessing**: few lights, avoid real-time shadows.
4. **Limit particles** and other expensive effects.

Build reduction into the aesthetic concept from the start.



Google Gemini

Finish & hand over

Document, feature-freeze, clean up and leave it usable



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



June 2026, 14

Continuous Documentation

Document while you work and not from memory months later.

1. **Write down how things work** as you build them.
2. **Keep it close to the project** (repo, wiki) so the team actually finds it.
3. **Update it** whenever something important changes.



Good documentation is what lets someone else (or future you) continue the work.

Set a Feature Freeze

Agree on a date after which no new features are added.

1. **Set a clear feature-freeze date** together, in advance.
2. **After the freeze**, only fix bugs, polish and stabilize. No new ideas!
3. **Protect time** for testing and clean-up before the deadline.

A feature freeze is how a project actually gets finished.



Clean Up & Write a Readme

Leave the project in a state someone else could pick up.

1. **Clean up the project:** remove unused assets, test scenes and dead code.
2. **Check structure and naming** one more time.
3. **Write a readme file:** what is the project about, how to open, build and run it.

Your future self, and anyone who works on this next, will thank you.



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



You're ready to build well

A quick recap of the best practices:

1. **Plan & set up:** structure, early decisions, build pipeline.
2. **Organise the team:** tasks, master scene, issue board, jour fixe.
3. **Design & prototype:** storyboards first, prototypes before the main scene.
4. **Build & optimize:** test on device, performance from the start.
5. **Finish & hand over:** document, feature freeze, clean up, readme.

Good habits beat heroics. Happy creating!



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



Thank you!

There is much more to discover about building in Unity.

Take these practices into your next project for happy creating!



Co-funded by
the European Union



Senate Department
for Culture
and Social Cohesion

BERLIN



htw
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences



June 2026, 19