

Developer Tools for VR & AR

There are a number of tools available for those interested in developing their own native applications in VR, AR, or even mixed reality. Before you undertake any project, decide exactly which tools you will use based on your needs and objectives. What kind of reality do you want to create? What platform do you want to develop for? Who is your audience? Are you most focused on 3D graphics, on interactivity, on data collection? Perhaps, most importantly, what is the learning curve for mastering the authoring tool?

Virtual Reality

There are a number of game engines and development platforms for VR currently on the market. While all are capable of doing essentially the same things, there are some differences which can play a role in helping a user to decide which tool is right for them when learning to develop.

[Unity 3D](#) has long been considered the default platform for developing within a VR environment.

- Has a direct VR mode to preview your ongoing work in an HMD (head mounted device)
- Based in C# programming language
- Supports all major HMDs (Vive, Oculus, etc) and can export to almost any platform
- Large active community of users, forums, documentation and how-to videos

[Unreal Engine](#) is a gaming engine with VR integrations and is gaining popularity as an alternative to Unity.

- Arguably the most advanced, realistic graphics.
- Based in C++ programming language.
- Massive library of free assets including objects, environments, and textures.
- Exports to most platforms

[A-Frame](#) is not a game engine. It is a web framework for building WebVR experiences.

- Based on top of HTML code, A-Frame may be the most simple way to get started with VR development.
- Independent, open source project with a large developer community.
- Supports all major HMDs
- Ideal for a novice looking to build simple experiences.

Augmented Reality

Again, deciding what tools to use for AR development depends on what platforms you are creating for and what exactly you are looking to achieve.

- [Unity](#) has led the way with AR development as well. In fact, more than 90% of all AR applications for the Microsoft HoloLens have been made with Unity. So if the HoloLens is your device of choice, Unity is likely the way to go. It also supports mobile AR.
- [Unreal Engine](#) also provides a rich, unified framework for building AR experiences for both iOS and Android handheld devices as well as Magic Leap.
- [ARCore](#) is Google's platform for building AR experiences on Android and Google devices. Includes both facial tracking and world tracking. There's also integration with Google's VR building tools Tilt Brush and Blocks for creating 3D objects.
- [ARKit](#) is Apple's platform for building AR experiences on iOS devices. Includes both facial tracking and world tracking. Also make optimal use of the iPhone's camera sensor to estimate the total amount of light in a scene.

Other Tools

Although there is currently a great deal of both free and paid 3D content (objects, environments, textures, etc) available for use in your projects, there are times when you will want/need to create your own. A wide variety of tools are available for 3D modeling for VR and AR experiences.

- [Blender](#) - Probably the most popular modeler for VR developers. Free and open source with a large community and great deal of documentation and tutorials.
- [SketchUp](#) - Google's basic 3D modeling app. Very low learning curve with good tutorials to get you up and running.
- [Remix 3D & Paint 3D](#) - Free 3D paint tool and 3D model library from Microsoft. Create your own or browse and download for use in AR & VR applications.
- [Tilt Brush & Blocks](#) - More tools from Google which allow you to actually create models first in VR, then export them into a game engine such as Unity for use in projects.
- [3DS Max, Maya, etc](#) - While these modeling tools from Autodesk can probably produce top of the line content, they do not offer VR support by default and require paid plug-ins.