

In this course we will be using Unity as the official development environment. Unity consists of both a game engine and an IDE (Integrated Development Environment). The IDE provides a graphical interface for the construction of game worlds that use the Unity engine and allows the developer to immediately preview and play any changes they make.

In addition to Unity, another popular freely-available game engine is Unreal Engine. Between the two, some people feel that Unreal Engine is more powerful, but Unity is easier to learn. Unity requires the purchase of a license for games that earn over \$100 000, while Unreal Engine requires payment of a 5% royalty on games that earn over \$3 000 in any three-month period.

There is much more information available on Unity than can be covered in this lecture, and different groups are likely to use different capabilities in their projects. Also, I find that programming APIs are better learned by use than from a lecture. This lecture will, therefore, not provide many details on development in Unity, but instead point you to resources that you can use to learn Unity as you do your assignments and projects.

The official Unity web site is <http://unity3d.com>. This site contains a large amount of tutorials (<http://unity3d.com/learn/tutorials>) and documentation (<http://docs.unity3d.com/Manual/index.html>) and is an excellent resource for learning Unity. A good place to start learning Unity is the Roll-a-ball tutorial, which forms the basis of your first assignment. Due to Unity's popularity, there are also a large amount of 3rd-party tutorials available online.

The Unity web site also contains the Asset Store (<https://www.assetstore.unity3d.com/en/>), which offers both free and paid assets such as 3D models, animations, audio, etc.. A number of other sites that provide free assets also exist, such as:

- <http://opengameart.org/>
- <http://freesound.org/>
- <http://soundbible.com/>
- <http://archive3d.net/>
- <http://www.blendswap.com/>
- <http://my-textures.com/>

Unity has built-in support for several types of VR devices, including Stereo, Split-Screen Stereo, Oculus, and Sony's Project Morpheus. The projects in this class will be focused on the Oculus Rift. Enabling VR support in Unity is very simple; full instructions are given in the tutorial at <http://unity3d.com/learn/tutorials/topics/virtual-reality>.

Unity has been used for a wide variety of games, both 3D and 2D. Some examples are:

- Dreamfall Chapters
(<http://redthreadgames.com/games/chapters/>)
VR in development. Developers say DK1 was working but broke. Full 3D, third-person.
- InMind
(<http://inmind.nivalvr.com/>)
DK2 works, available on Steam. Full 3D, first-person. Rail-shooter where you target neurons by focusing on them for a certain amount of time.

- Kerbal Space Program
(<https://kerbalspaceprogram.com/en/>)
No VR support. Full 3D, first- and third-person.
- Pillars of Eternity
(<http://eternity.obsidian.net/>)
No VR support. 3D assets in a 2D world (“2.5D”), third-person.
- Subnautica
(<http://unknownworlds.com/subnautica/>)
DK2 works in Experimental version, available on Steam. Full 3D, first-person.
- Sunless Sea
(<http://www.failbettergames.com/sunless/>)
No VR support. Full 2D, third-person.
- Surgeon Simulator 2013
(<http://www.surgeonsim.com/surgeon-simulator-2013/>)
VR support. “Interesting” VR control scheme. Full 3D, first-person.

Some examples of non-game programs made with Unity are:

- Ferrari AR Showroom App
(<http://www.pleribus.com/portfolio-item/ferrari-augmented-reality-ipad-app/>)
AR on a tablet.
- Guided Meditation
(<http://guidedmeditationvr.com/>)
VR support. Full 3D, first-person.
- Universe Sandbox²
(<http://universesandbox.com/>)
VR support. Full 3D, third-person.
- Virtual Zoll Defibrillator Trainer
(<http://sitel.org/digital-media-services/>)
No VR support.

More examples can be found on the Unity website (<http://unity3d.com/showcase/gallery>).