

Learn Unreal VR Syllabus

Create VR worlds with the powerful Unreal Engine 4



Before You Start

Welcome to Learn Unreal VR. In order to succeed in this course, we recommend that you have access to a VR-ready PC, and an HTC Vive or Oculus Rift with touch controllers. You do not need any programming experience for this course, and we will not be covering any C++, C#, Java or any programming language beyond Blueprints (Unreal Engine 4's powerful built-in node-based visual programming language). The steps are outlined in the lesson, but you can download the Epic Games Launcher (to get Unreal Engine 4) from their [website](#). Below you will find the topics that will be covered in the two courses and their associated projects.

Project 1: Kitchen Cleanup

This project is your first chance to make a game within Unreal Engine 4 for either standing or roomscale VR. You will utilize motion controllers to build a kitchen-themed interaction game, and you'll use functions, physics, Blueprint communication, and audio to create an immersive experience. Spawn messy dishes and get them into the sink as quickly as possible!

Supporting Lesson Content: Intro to Unreal Engine 4 VR

| Lesson Title | Learning Outcomes |
|---|---|
| THE DIFFERENCES BETWEEN UNREAL AND UNITY | <ul style="list-style-type: none">→ Compare differences between Unreal Engine and Unity such as development style, target audience, and ease-of-use so you can choose the engine most appropriate for your VR projects→ Learn the basics of the UI for the Unreal Engine, and its major features |
| DESKTOP VR | <ul style="list-style-type: none">→ Apply design theory for roomscale and standing scale VR→ Build a basic VR-ready playable character→ Create and launch a working Unreal Engine project |
| INTERACTING WITH MOTION CONTROLLERS | <ul style="list-style-type: none">→ Implement Motion Controllers→ Simulate basic physics interactions→ Create input events for use in Blueprints→ Create basic Blueprint scripting→ Use line traces to identify objects |
| AUDIO AMBIENCE | <ul style="list-style-type: none">→ Create sound cues→ Modify the properties of an audio cue→ Implement audio and use it within Blueprints |

| | |
|--------------------------------|--|
| BLUEPRINT COMMUNICATION | <ul style="list-style-type: none"> → Learn how to directly communicate with Blueprints → Learn how to communicate with a specific class of Blueprints → Learn how to communicate with an arbitrary number of different types of Blueprints → Learn how to use, implement, and call custom events |
| HANDY INTERACTIONS | <ul style="list-style-type: none"> → Learn to implement Blueprint interfaces → Use Blueprint functions → Learn to create physics interactions |

Project 2: Hide and Seek

This is your first chance to build an Unreal VR experience based around locomotion. You will build a find-the-object style of game, create a set of Blueprints that randomly hides an object, and develop a locomotion method that allows you to move around a provided apartment scene so you can find the object.

Supporting Lesson Content: Advanced Unreal Engine 4 VR

| Lesson Title | Learning Outcomes |
|-----------------------------|--|
| PERFORMANCE ANALYSIS | <ul style="list-style-type: none"> → Use Unreal Engine 4's built-in tools for analyzing performance → Review examples of proper art asset optimization → Analyze an unoptimized scene in Unreal Engine 4 |
| LOCOMOTION DESIGN | <ul style="list-style-type: none"> → Look at examples of different types of locomotion in VR → Look at the causes of motion sickness with locomotion methods → Create a projectile-based locomotion method → Generate more advanced physics simulations → Build a spline mesh |
| ANIMATION | <ul style="list-style-type: none"> → Create a time-based animation using a timeline → Use the Unreal Engine 4 animation pipeline → Create an animated motion controller that responds to input → Use Enumerators to store the state of the players interaction |
| LIGHTING | <ul style="list-style-type: none"> → Create different types of lights in a scene → Use the forward renderer → Examine the different types of shadows in UE4 → Use different types of lighting to create an optimized lighting scene → Use a lightmass importance volume |