3D Development Essentials on NVIDIA Omniverse

Duration: 05 days (40 hours)

Course Prerequisites:

- Basic familiarity with Python is helpful but not required. Suggested materials to satisfy this prerequisite: The Python Tutorial.
- Basic understanding of computer graphics concepts such as vertices, meshes, and RGB values is useful but not required. Suggested materials to satisfy this prerequisite: Scratchapixel's Gentle Introduction to Computer Graphics (Programming).
- Windows or Linux computer with the ability to install Omniverse Launcher and Omniverse applications.
- Internet bandwidth sufficient to support the Isaac Sim client/server stream (performance will vary based on internet bandwidth).
- Familiarity with terminal commands, GitHub, and Visual Studio Code with the Python Extension is useful but not required.

Module 01: Essentials of Developing Omniverse Kit Applications

About this Module

NVIDIA Omniverse is a platform that enables you to build applications for complex 3D and industrial digitalization workflows. To build applications from scratch, NVIDIA offers Omniverse Kit SDK and free templates to build starter applications that can be easily customized and extended. This course provides the fundamentals of using Kit SDK to develop such applications. By the end of the course, you will know how to access the kit-app-template GitHub repository, implement the Kit SDK templates to create an application, and use simple techniques to customize the application with extensions.

Learning Objectives

By the end of this course, you will know how to:

- Clone a code repository
- Create and edit applications using Kit App Template and Kit SDK
- Explore and edit the Kit architecture for a custom application
- Add extensions to your application for additional functionality

Topics Covered

- Omniverse

- Kit SDK
- Extensions
- Application Development

Module Outline

This course is split into chapters, each of which covers a specific skill or topic:

- **Chapter 1:** Clone the Kit Repository – Copy the kit-app-template repository from GitHub to your local computer.
- **Chapter 2:** Create Applications with Templates – Use built-in kit commands to create, build, and launch a basic application.
- **Chapter 3:** Add Extensions – Extend an application's functionality by adding extensions to it.

===

Module 02: Develop, Customize, and Publish in Omniverse with Extensions

About this Course

Want to change the functionality and User Interface (UI) of Omniverse more to your liking? Customize the Omniverse experience with Extensions using Python code. Extensions can be used for a wide variety of modifications from spawning objects with a button press to applying custom physics on selected objects. Optimize a workflow by copying commonly repeated operations into an Extension or add a new way to manipulate objects in the UI.

Learning Objectives

In this course, participants will learn how to:

- Make a button to spawn a cube
- Alter the Omniverse UI with custom windows
- Debug and an publish Extensions

Topics Covered

Upon completion, you will be able to create and share your own custom Extension with the Omniverse community. Additional topics related to this that you can check out include:

Module Outline

Develop, Customize, and Publish in Omniverse with Extensions	• Installing Omniverse, Completed. • An Introductory Extension: Spawn an Object, Completed. • Extension Debugging, Completed.
Intermediate Extension Concepts	• Intermediate Extension Concepts. • How to Build a Command, Completed. • Extension Publishing, Completed.
Advanced Example: The Scatter Tool	• Model-view-controller, Completed. • Advanced Windows, Completed. • Scatter with Style, Completed.

===Module 03: Assemble a Simple Robot in Isaac Sim

Learning Objectives

In this lab, you'll learn how to:

- Connect a local streaming client to an Omniverse Isaac Sim server in the cloud
- Load a USD mock robot into the Isaac Sim environment
- Add joint drives and joint properties to the robot body
- Add articulations to the robot

Upon completion, you will have a basic understanding of the Isaac Sim interface and documentation needed to begin your own robot simulation projects.

Topics Covered

Tools, libraries, and frameworks used in this course include:

- NVIDIA Omniverse™ Isaac Sim

Module Outline

1. Getting Started
2. Import a Mock Robot
3. Add Joints
4. Add Joint Drive
5. Add Articulation
6. Save Your Robot

==Module 04: Build Beautiful, Custom UI for 3D Tools on NVIDIA Omniverse

About this Module

Experience the NVIDIA Omniverse development platform for builders and creators of virtual worlds. Become a master in UI with a deep dive into NVIDIA Omniverse Kit's powerful omni.ui suite of tools and frameworks. In this self-paced course you'll build your own custom UI for workflows in Omniverse with hands-on exercises.

Learning Objectives

By participating in this workshop, you will learn how to:

- Launch Omniverse Code.
- Install/Uninstall/Enable extensions in Omniverse Code.
- Navigate to the extensions panel, open the viewport, select a Prim).
- Navigate inside the USD stage hierarchy.
- Build a small form window including styling.
- Clear your test.
- Challenge yourself to match a window design built by the Omniverse UX team.
- Match the window design's form and function.

Upon completion, you will have created a custom UI and become familiar with Omniverse Code.

Topics Covered

- Omniverse
- Omniverse Extensions
- OpenUSD

Module Outline

Chapter I	• Open the Workshop Stage. • Install the Scatter Tool Extension. • Enable Physics. • Undo Scatter. • Undo Scatter.
Chapter II	• Add another Source to the UI. • Set and Scatter Prim A and Prim B. • Set Scale Parameters in the UI.
Chapter III	• Make your Scene. • How many marbles can you get in the jars and bowls?

===Module 05: Easily Develop Advanced 3D Layout Tools on NVIDIA Omniverse

About this Module

Get hands-on experience with NVIDIA Omniverse - the platform for connecting and creating physically accurate, 3D virtual worlds. See how easy it is to create your own custom scene layout tools in Omniverse Code with a few lines of Python script. In this self-paced course, you'll build your own custom scene layout in Omniverse with hands-on exercises in Omniverse Code and Python.

Learning Objectives

By participating, you will learn how to:

- Launch Omniverse Code.
- Install/Uninstall/Enable extensions in Omniverse Code.
- Navigate to the extensions panel, open the viewport, select a Prim).
- Navigate inside the USD stage hierarchy.
- Clear your test.
- Create a feature to Scatter from a Primative's (Prim) origin.
- Add randomization to the scatter extension.

Upon completion you will have learned about Prims and how to customize the scatter extension.

Topics Covered

- Omniverse
- Omniverse Extensions
- OpenUSD

Module Outline

Section I	• Open the Workshop Stage. • Adding the Extension. • Using the Extension. • What Else can you do with the Scatter Extension.
Section II	• Change the Scatter functionality to Handle any Given Origin • Get the Location of the Source Prim • Use the Selected Prim's Location as the Scatter Origin
Section III	• Scatter a Marble • Watch the Scene Play • Scale Scatter Prims based on Provided Scale

==Module 06: How to Build Custom 3D Scene Manipulator Tools on NVIDIA Omniverse

About this Module

See how you can build advanced tools on the modular, easily extensible Omniverse platform. You'll learn from the Omniverse developer ecosystem team how you can extend and enhance the 3D tools you know and love today. Become a developer for the next wave of physically accurate 3D virtual worlds and learn to build on NVIDIA Omniverse. In this self-paced course, you'll build your own custom scene manipulator tools in Omniverse with hands-on exercises writing a few lines of Python code.

Learning Objectives

By participating, you will learn how to:

- Launch Omniverse Code.
- Install/Enable extensions in Omniverse Code.
- Navigate to the extensions panel, open the viewport, select a Prim).
- Navigate inside the USD stage hierarchy.
- Create a widget manipulator (to control scale).
- Fix a broken manipulator (slider).
- Build a scale manipulator that only scales in one axis direction.

Upon completion you will have created and fixed a manipulator and become familiar with Omniverse Code.

Topics Covered

- Omniverse
- Omniverse Extensions
- OpenUSD

Module Outline

Chapter I	• Open the Workshop Stage. • Install the Widget Extension. • Find the Play Button.
Chapter II	• Find your Work Files • Fix the Broken Slider • Did it Work? • Scale Larger • What Other Properties Might You Want to Control with the Widget?
Chapter III	• Create your Scene • Scale on One Axis • Turn on the Light Manipulator

==
======

Module 07: Universal Scene Description for Collaborative 3D Workflows

About this Module

The Universal Scene Description (OpenUSD) framework was developed by Pixar to foster collaboration between technical and creative world builders, moving from a sequential workflow to a collaborative workflow.

In this self-paced course, participants will learn how to generate a scene using human-readable Universal Scene Description ASCII (.USDA) files. Upon completion, participants will be able to create their own scenes within the USD framework and will have a strong foundation to use it in applications such as Omniverse, Maya, Unity, and Unreal Engine.

There are two sections to this training:

- Chapter I – USD Fundamentals: An introduction to the basics of OpenUSD, no programming knowledge required.
- Chapter II – Advanced USD: A demonstration of how to use Python to generate USD files.

Learning Objectives

Learn the basics of how OpenUSD scenes are structured and get hands-on experience with OpenUSD Composition Arcs by:

- Overriding an asset's base properties with Sublayers
- Combining multiple assets with OpenUSD References
- Creating different states of an asset with OpenUSD Variants

Topics Covered

- OpenUSD

Chapter I – USD Fundamentals	• USD Format Overview. • How to use USD Sublayers. • How to use USD References. • How to use USD Variants.
Chapter II – Advanced USD	• Opening USD Stages. • Composition Arcs. • Prims, Attributes and Metadata.

	- Hierarchy and Traversal.

Module Outline