



Certified
3D Artist

Exam Objectives

Unity Certified
3D Artist

The role

Unity 3D artists focus on the implementation of 3D art for interactive software rendered in real-time using the Unity Engine. The 3D artist brings visual assets into Unity to populate the “world” of the game or application, and adds or manipulates layers of visual information to execute a key part of the creative vision for the project. The 3D artist is a generalist, adept at imbuing objects, characters, and environments with a distinctive style and mood. They contribute to a broad range of tasks in service of the application’s aesthetics, including applying Materials and shaders to 3D models, setting up and managing 3D environments, configuring and controlling cameras, lighting scenes, and using particle effects.

Unity Certified 3D Artist certification is a professional certification for entry-level to mid-level artists and graduating post-secondary students seeking roles in a variety of industries. This certification shows potential employers that the holder:

- Employs a mix of artistic and technical skills within the context of professional software development processes to turn design assets into fully-realized worlds.
- Is creative and expressive, but comfortable with the technical aspects of their work and can employ a foundational knowledge of programming terminology to communicate with more technical team members.
- Can achieve the required “look and feel” for real-time 3D application, and may draw on a foundation in 2D art and animation to help them prototype user interfaces and object movements.

Job titles for this role

- 3D Artist
- 3D Generalist
- Game Artist
- Level Designer
- Environment Artist
- 3D Visualization Artist

Prerequisites

Artists who are ready to begin a professional career as real-time 3D artists on games, design visualizations, or other types of applications that are enabled by the Unity Engine. These artists may be recent college graduates in game art, computer graphics, or related fields; independent learners who have completed 2 or more years of college-equivalent study or work experience in 3D modeling, or early-career professional who have already started their first jobs. Regardless of background, these artists must have hands-on experience implementing 3D objects and environments with Unity by themselves or as part of a cross-functional team, resulting in a completed prototype or demo reel. They come to the exam in order to test and evaluate their skills with Unity and to make themselves more attractive on the job market.

Prerequisite experience:

- Practical experience in implementing 3D art and environments for video games or other real-time 3D applications built with Unity
- Experience in importing, configuring, and lighting 3D objects and environments in Unity for a variety of platforms including PCs, mobile devices, and XR
- Experience in the full software development lifecycle working from early concept through completion
- Experience prototyping 3D environments and application UI
- Foundational understanding of animation and 2D rendering
- Foundational understanding of Unity programming workflows and terminology
- Understanding of professional software development practices including design documentation and version control
- Experience in game development or other real-time 3D application development such as simulation and design visualization

Note: This certification was developed for Unity version 2017.3.

Core skills

Rendering 3D Objects

- Select the appropriate import settings for importing 3D assets into Unity
- Troubleshoot common issues with imported 3D assets
- Add Materials to objects, and use advanced Material settings to achieve a desired effect
- Add textures to Materials, and use advanced texture settings to achieve a desired effect
- Manipulate the settings of the Unity Standard Shader to achieve a desired effect
- Adjust camera properties to achieve a desired effect
- Use Level of Detail (LOD) groups and objects to optimize scenes

Lighting Objects and Environments

- Adjust environment lighting settings to create the desired lighting and reflection effects
- Determine scene lighting modes and their uses for balancing fidelity and performance
- Add and configure lights to create a desired lighting effect
- Recognize processes for creating custom Skyboxes

Working with Particles and Effects

- Use particle systems to achieve a variety of effects such as explosions, emissions, and trails
- Use the Post-Processing Stack to achieve cinematic rendering effects and apply color grading

Prototyping Basic Application Elements

- Identify techniques to prototype scenes and maintain Prefabs throughout the production cycle
- Recognize uses of UI components and configure UI prototypes

Working with 2D Assets

- Configure settings for importing 2D assets into Unity
- Use the Sprite Editor to slice Sprite Sheets
- Recognize processes for creating 2D animations from Sprite Sheets

Working with Animation

- Determine methods for creating simple 3D and 2D keyframed animation sequences using Unity's Animation Editor
- Use basic state machines and blend trees to create and manage multiple animations.
- Identify methods to procedurally control camera views and movement using Cinemachine
- Determine methods to sequence and control animations and camera movements with Timeline

Engaging in Software Development teams

- Evaluate design requirements to make decisions about project graphic and animation configurations
- Demonstrate understanding of the programming workflow and related terminology for work with Unity programmers
- Demonstrate an understanding of professional practices for utilizing version control software including Unity Collaborate

Sample Questions

Question 1

An Artist is working on an automotive visualization. The model itself is a high-detail Mesh with 100,000 vertices. The visualization requires accurate rendering and detail of the Mesh. When the model is brought into the Scene view window, it appears to be broken into several sub-Meshes. This causes rendering artifacts and improper smoothing.

How should the Artist change the import settings to fix these issues?

- A** Set Index Format to 32 bit.
- B** Enable Weld Vertices.
- C** Set Mesh Compression to Low.
- D** Enable Optimize Mesh.

Question 2

An Artist is working on an architectural visualization of a retail store. The Artist needs to create a camera setup to allow the user to see the store through the eyes of a player-controlled customer. The customer model has no head Mesh and only animates from the neck down.

How should the Artist configure a camera's settings and Transform properties to achieve this effect?

- A** Make the camera a child of the character and place at the character's eye level.
- B** Make the character a child of the camera and reset the camera's Transform values.
- C** Make the camera a child of the character and place above and behind the character's shoulder.
- D** Create multiple cameras and position them throughout the environment at eye-level positions.

Question 3

An Artist needs to create a fire effect for a continuously burning torch. The torch is carried by the player's character, who can move at variable speeds. There are torches in the environment that the character can light to illuminate the path. The player's character moves through interior and exterior environments that have dynamic weather and wind zones.

Which settings should the Artist use in the particle system to achieve this result?

- A** Limit Velocity over Lifetime, Noise, External Forces
- B** Looping, Gravity Modifier, Velocity over Lifetime
- C** Force over Lifetime, Start Speed, Burst Count
- D** Looping, Inherit Velocity, External Forces

Question 4

An Artist is optimizing an explosion effect for a mobile platform. The explosion consists of a flash, fire, sparks, debris that lands on the ground, and lingering smoke.

Which changes should the Artist make to optimize the effect?

- A** Reduce Max Particles
Reduce Start Size
Reduce Sorting Fudge
- B** Reduce Max Particles
Switch to mobile Particle shaders
Reduce Collision Quality
- C** Reduce Start Size
Switch to mobile Particle shaders
Reduce Simulation Speed
- D** Reduce Particle Lifetime
Reduce Start Speed
Reduce Collision Quality

Question 5

An Artist needs to create a flythrough of an architectural visualization using Timeline. Users must be able to control the camera until they reach the building's foyer, at which point a flythrough sequence of the rest of the building automatically begins.

Which approach should the Artist select to start the sequence?

- A** Call an OnTriggerEnter event when the Camera's Collider reaches the foyer's Collider.
- B** Set the PlayableDirector component to Play On Awake.
- C** Use an Activation Track to trigger the sequence.
- D** Set the PlayableDirector component's initial time to match the time when the Camera reaches the desired position.

Correct Answers: A, A, D, B, A