



CERTIFIED  
User

# Exam Objectives

## Unity Certified User Artist

The **Unity Certified User Artist** certification exam will test the basics of 2D and 3D digital artistry within Unity software to create interactivity in games, apps, AR/VR, and other experiences. The exam objectives are aligned with current industry standards set by professionals and educators. Individuals will be expected to have at least 150 hours of Unity software use and training to obtain this certification.

Individuals who have earned the Unity Certified User Artist certification have demonstrated mastery of the following skills:

### 1. Asset Management

- 1.1. Import assets including but not limited to settings for FBX, OBJ and associated textures.
- 1.2. Import and configure assets from the Unity Asset Store.
- 1.3. Slice spritesheets for use in a 2D scene including but not limited to using the default Sprite Editor and 9-slicing.
- 1.4. Identify mesh components including vertices, polygon faces and edges.
- 1.5. Create key frames and change tangents in the Curve Editor using the Animation window.
- 1.6. Create, modify and utilize Prefabs.

### 2. Scene Content Design

- 2.1. Utilize Transform tools and the Transform component in the Inspector.
- 2.2. Create prototype scenes using Unity primitives and/or low poly meshes utilizing white box/grey box techniques.
- 2.3. Create and edit a landscape with materials utilizing the Terrain tool including but not limited to mask maps, texture painting, and diffuse properties.

### 3. Lighting, Cameras, and Materials Implementation

- 3.1. Modify materials using the Standard Shader and editing properties including but not limited to specular, transparency, normal, and albedo.
- 3.2. Identify basic lighting including but not limited to shadows, light settings, and light shapes such as directional, area, spot, and point.
- 3.3. Utilize single camera set up including but not limited to isometric vs. standard, camera component, background, culling masks, clipping planes, field of view (FOV), etc.
- 3.4. Given a scenario, determine the appropriate rendering pipeline that should be used.