



Exam Objectives

Unity Certified Associate: Game Developer

Unity Associate



Whether you want to get Unity Certified, or just want to learn to make games, we've got you covered!

Unity Associate certification and courseware are for anyone who wants a structured, hands-on, self-study program for learning Unity and game development. With the courseware, you'll build a working game from the ground up using the Zombie Toys assets provided and learn everything you need to be prepared to take the Unity Certified Associate exam.



Learning Outcomes

With Unity Certified Associate Courseware, you can learn the Unity platform and game development fundamentals from the trenches by following the production of a working game from concept all the way through to publishing. Unity Certified Associate courseware gives you a structured, self-study program that includes everything you and your students need to succeed:

- 20 chapters of video-rich learning content (199 videos, approximately 19 hours*)
- All Zombie Toys game project exercise files and assets you or your students will need to follow along in Unity

Focus on the essentials. Learn about the job roles and skills most essential to game production, and gain Unity experience that directly maps to preparation for the Unity Certified Associate Exam.

Build a working game. Develop an end-to-end understanding of game production with Unity by building Zombie Toys, a third-person, 3D, survival arcade game. Learn hands-on as you execute game development tasks along with the videos—from importing assets, to scripting behavior, to building the game for publication.



1. Welcome to Unity!

- 1.1. Differentiate Unity Services
- 1.2. Differentiate video game production practices
- 1.3. Distinguish console hardware
- 1.4. Distinguish production talent roles and responsibilities
- 1.5. Distinguish video game design elements
- 1.6. Distinguish video game genres
- 1.7. Distinguish video game production practices
- 1.8. Distinguish video game types
- 1.9. Examine Unity services
- 1.10. Explain Unity services
- 1.11. Identify production phases by criteria
- 1.12. Identify video game production practices
- 1.13. Implement Unity services
- 1.14. Know video game controls
- 1.15. Know video game design principles
- 1.16. Know video game industry terms
- 1.17. Recognize Unity services
- 1.18. Summarize Unity services
- 1.19. Understand model asset optimization
- 1.20. Understand the Unity Asset Store
- 1.21. Understand video game art principles
- 1.22. Understand video game industry practices
- 1.23. Understand video game industry terminology

2. Exploring the Unity User Interface

- 2.1. Distinguish the Hierarchy Window
- 2.2. Distinguish Unity editors
- 2.3. Distinguish Unity views
- 2.4. Distinguish Unity windows
- 2.5. Know the Project View Window
- 2.6. Manage Scene files
- 2.7. Navigate the Scene View Window
- 2.8. Reorganize the Unity interface
- 2.9. Understand Tags
- 2.10. Understand the Hierarchy Window
- 2.11. Understand the Inspector Window
- 2.12. Understand the Project View Window
- 2.13. Utilize the Inspector Window



3. Using Game Objects and Assets

- 3.1. Define Prefabs
- 3.2. Differentiate GameObjects
- 3.3. Distinguish components
- 3.4. Distinguish Models
- 3.5. Examine GameObject components
- 3.6. Identify GameObjects
- 3.7. Know the Project View Window
- 3.8. Manage GameObjects
- 3.9. Manage Prefabs
- 3.10. Understand GameObject components
- 3.11. Understand the Project View Window
- 3.12. Utilize Models
- 3.13. Utilize the Hierarchy Window
- 3.14. Utilize the Toolbar

4. Managing Projects and Assets

- 4.1. Distinguish Models
- 4.2. Distinguish Unity views
- 4.3. Implement project management settings
- 4.4. Organize Unity game projects
- 4.5. Understand project management features
- 4.6. Understand the Project View Window

5. Preparing Assets for Implementation

- 5.1. Create materials
- 5.2. Create textures
- 5.3. Evaluate materials and effects
- 5.4. Examine material and lighting features
- 5.5. Know materials and effects
- 5.6. Know model import and export best practices
- 5.7. Know modeling best practices
- 5.8. Manage materials
- 5.9. Manage textures
- 5.10. Optimize textures
- 5.11. Refine material properties
- 5.12. Understand animation best practices
- 5.13. Understand material and texturing best practices



6. Assembling the Game Level

- 6.1. Distinguish properties
- 6.2. Distinguish Rigidbody properties
- 6.3. Examine forces on Rigidbody
- 6.4. Integrate colliders
- 6.5. Know colliders
- 6.6. Know Rigidbodies
- 6.7. Manipulate colliders
- 6.8. Understand Rigidbodies
- 6.9. Utilize the Hierarchy Window

7. Lighting in Games

- 7.1. Analyze lighting tools and processes
- 7.2. Distinguish light types
- 7.3. Examine lighting situations
- 7.4. Know lighting tools and processes
- 7.5. Understand lighting tools and processes
- 7.6. Understand UI components
- 7.7. Understand video game art principles
- 7.8. Utilize the Sprite Editor

8. Baking Lighting in Game Production

- 8.1. Distinguish light types
- 8.2. Examine lighting situations
- 8.3. Examine lighting tools and processes
- 8.4. Know lighting tools and processes
- 8.5. Understand lighting tools and processes

9. Animating Game Objects in the Unity Editor

- 9.1. Animate game objects
- 9.2. Distinguish character animation options
- 9.3. Manage animation settings
- 9.4. Refine the animation of game objects
- 9.5. Understand character animation processes

10. Bringing Animations into the Game



- 10.1. Assess Animator Controllers
- 10.2. Examine Animation Types
- 10.3. Examine States Examine Transitions
- 10.4. Know States
- 10.5. Manage Animator Controllers
- 10.6. Understand Transitions
- 10.7. Utilize States
- 10.8. Utilize the Animator Window
- 10.9. Utilize Transitions

11. Scripting in Game Development

- 11.1. Assess program code
- 11.2. Distinguish programming terms
- 11.3. Distinguish variables in code
- 11.4. Examine program code
- 11.5. Examine raycasts within a scene
- 11.6. Execute programming tasks
- 11.7. Identify script types
- 11.8. Understand layers
- 11.9. Understand programming terms
- 11.10. Understand raycast parameters
- 11.11. Understand raycasts

12. Implementing Navigation and Pathfinding

- 12.1. Understand a NavMesh
- 12.2. Understand a NavMesh baking
- 12.3. Understand Max Slope
- 12.4. Understand obstacle avoidance

13. Building the Player and Allies

- 13.1. Create allies
- 13.2. Create players
- 13.3. Implement a game manager
- 13.4. Implement a player controller
- 13.5. Manage cameras

14. Building the Enemies



- 14.1. Create enemies
- 14.2. Design enemy behaviors
- 14.3. Evaluate enemy behaviors
- 14.4. Integrate enemies into a game
- 14.5. Manage enemies

15. Creating Particle Systems

- 15.1. Distinguish Image
- 15.2. Effects Distinguish particle options
- 15.3. Evaluate materials and effects
- 15.4. Produce particle effect results
- 15.5. Understand materials and effects

16. Adding Audio to Game Levels

- 16.1. Control Audio Properties
- 16.2. Enable Audio Properties
- 16.3. Examine Audio Properties
- 16.4. Identify Audio Clips
- 16.5. Identify Audio Effects
- 16.6. List Audio Clips
- 16.7. Manage Audio Clips
- 16.8. Understand Audio Properties

17. Building the Camera and Player Selection System

- 17.1. Configure cameras
- 17.2. Evaluate camera choices
- 17.3. Evaluate player behaviors
- 17.4. Examine player behaviors
- 17.5. Integrate character selections
- 17.6. Refine player settings

18. Designing User Interfaces for Games

- 18.1. Administer pivots and anchors
- 18.2. Demonstrate text properties



- 18.3. Distinguish button properties
- 18.4. Distinguish render modes
- 18.5. Distinguish UI components
- 18.6. Evaluate UI features
- 18.7. Examine Rect Transforms
- 18.8. Know anchor points
- 18.9. Understand button properties
- 18.10. Understand UI components

19. Building and Deploying the Game

- 19.1. Administer Unity Cloud Build tools
- 19.2. Build a game
- 19.3. Distinguish console hardware
- 19.4. Refine build settings
- 19.5. Understand the build process

20. Preparing for Mobile Deployment

- 20.1. Distinguish build platforms
- 20.2. Evaluate mobile publishing choices
- 20.3. Examine mobile publishing options
- 20.4. Manage game settings for mobile publishing
- 20.5. Understand mobile development procedures



Core Skills

(Certification exam topics)

The **Unity Certified Associate exam** is made up of 100 questions over 16 topic areas. Question formats include multiple choice, hot-spot, drag-and-drop, and matching. The following pages include a detailed outline of the topics covered on the exam.

1. Animation

1.1 Animator System

- a. Animator Controller Asset
 - 1. Examine the Animator Controller
 - 2. Apply an Animator Controller to a GameObject
 - 3. Create an Animator Controller
- b. States
 - 1. Define parameter types
 - 2. Differentiate animation states
 - 3. Create a new animation state
 - 4. Implement the Any State
- c. Transitions
 - 1. Explain transition conditions
 - 2. Differentiate transition properties
 - 3. Create transitions

2. Asset Management

1.1 Assets

- a. Audio Clips
 - 1. List compression formats
- b. Default GameObjects
 - 1. Differentiate GameObjects by their appearance
 - 2. Identify GameObjects within a scene
 - 3. Identify script types
- c. Models
 - 1. Differentiate import file formats
- d. Prefabs
 - 1. Define a Prefab
 - 2. Create a Prefab



- e. Scene File
 - 1. Load a Scene
 - 2. Save a Scene

2.1 Sprites

- a. Sprite Editor
 - 1. Modify Sprites

3. Audio

1.1 Audio Mixer

- a. Audio Effects
 - 1. Describe various Audio Effects

2.1 Audio Reverb Zone

- a. Presets
 - 1. Differentiate audio properties

3.1 Audio Source

- a. Audio Properties
 - 1. Explain audio options
 - 2. Explain the Doppler effect
 - 3. Activate audio source looping
 - 4. Modify the volume of an audio source
 - 5. Locate Audio clips

4. Editor Interface

1.1 Editor Customization

- a. Layouts
 - 1. Customize the Unity interface
 - 2. Differentiate Unity Editors

2.1 Views

- a. Asset Store
 - 1. Explain the benefits of the Asset Store
- b. Console
 - 1. Differentiate the Console Window
- c. Hierarchy
 - 1. Explain the purpose of the Hierarchy Window
 - 2. Differentiate the Hierarchy Window
 - 3. Utilize the Hierarchy Window
 - 4. Create empty GameObjects
 - 5. Parent objects
- d. Inspector
 - 1. Explain the functionality of the Inspector Window
 - 2. Reset components
- e. Project
 - 1. Explain the functionality of the Project View Window



2. Identify UI functionality in the Project View Window
 3. Identify an empty Prefab
 4. Focus the Scene View Camera
- f. Scene
1. Differentiate the Project View Window
 2. Use the Zoom Tool
 3. Orbit the Camera
- g. Toolbar
1. Modify Gizmos

5. Employment Preparedness

- 1.1 Collaboration Skills
 - a. Providing Critique
 1. Define "critique" in the context of video game development
- 2.1 Employment Responsibilities
 - a. Company Confidentiality
 1. Explain the purpose of the Non-Disclosure Agreement (NDA)
 2. Explain the concept of "intellectual property" (IP)

6. Game Art principles

- 1.1 Character Design
 - a. Non-Player Characters (NPCs)
 1. Describe methods to optimize model assets
- 2.1 Concept Design
 - a. Color palette
 1. Summarize how art choices affect mood
 - b. Look-and-feel imagery
 1. Explain the purpose of concept art
- 3.1 Environment Design
 - a. Color palette
 1. Explain the concept of "unifying color"

7. Game Design Principles

- 1.1 Game Mechanics
 - a. Casual Games
 1. Differentiate video game types
 - b. Third Person
 1. Identify video game mechanics
 2. Differentiate video game participants
- 2.1 Genre
 - a. Realtime Strategy
 1. Differentiate video game genres
- 3.1 Platform



- a. Controller
 - 1. Identify video game controls

8. Industry Awareness

- 1.1 Employment Trends
 - a. Common Job Titles
 - 1. Differentiate production talent roles and responsibilities
- 2.1 Industry Trends
 - a. Hardware Products
 - 1. Differentiate console hardware based by feature
- 3.1 Production Trends
 - a. Production Cycles
 - 1. Match production phases by criteria

9. Lighting

- 1.1 Global Illumination (GI)
 - a. Baked GI
 - 1. Explain lighting settings
 - b. Baking
 - 1. Explain Light Baking
 - 2. Explain Lightmaps
 - 3. Explain Generate Lightmap UV settings
- 2.1 Light Component
 - a. Shadows
 - 1. Assess Shadow types
 - b. Type
 - 1. Explain Light Intensity
 - 2. Define Culling Mask
 - 3. Differentiate light types by feature
 - 4. Infer lighting settings by visual indicators

10. Materials and Effects

- 1.1 Effects
 - a. Image Effects
 - 1. Differentiate Image Effects by their result
 - 2. Predict particle option results
- 2.1 Materials
 - a. Standard Shaders
 - 1. Assess Rendering Modes
 - 2. Match Standard Shader properties by their description
 - 3. Define the Albedo of a material
- 3.1 Particle System
 - a. Emitters



1. Explain particle system settings
2. Differentiate particle options by their result

11. Navigation and Pathfinding

1.1 Navigation Agents

a. Obstacle Avoidance

1. Explain obstacle avoidance using NavMesh agents

2.1 Navigation Baking

a. Bake Settings

1. Explain the function of Max Slope

b. Navigation

1. Explain the function of a NavMesh
2. Explain NavMesh baking

12. Physics

1.1 Colliders

a. 3D Capsule

1. Identify Colliders by their appearance
2. Differentiate properties
3. Transform Colliders
4. Utilize Colliders

2.1 Optimization and Debugging

a. Raycast

1. Explain raycast parameters
2. Describe the function of raycasts
3. Assess raycast trajectories

3.1 Rigidbodies

a. Components

1. Explain Rigidbodies
2. Locate Rigidbodies
3. Differentiate Rigidbody properties
4. Assess the impact of forces on Rigidbodies

13. Programming

1.1 Camera API

a. ScreenPointToRay

1. Recognize the purpose of existing code

2.1 GameObject

a. Components

1. Finalize specific lines of code

3.1 Methods/Functions

a. Declaration and Use

1. Explain the purpose of methods



2. Differentiate methods by their result

4.1 MonoBehaviour API

- a. Awake
 1. Differentiate methods by their result
 2. Recognize methods by their desired result
- b. Fixed update
 1. Evaluate the effectiveness of specific methods

5.1 Object Oriented Programming

- a. Objects
 1. Recognize class definitions from provided code

6.1 Quaternion

- a. Use of
 1. Explain Quaternions

7.1 Time

- a. DeltaTime
 1. Explain DeltaTime

8.1 Unity Interface

- a. File Management
 1. Differentiate public variable within code
 2. Create a new script

9.1 Variables

- a. Floating Point
 1. Recognize and replace variables within code
- b. Integer
 1. Recognize and replace variables within code
- c. Vector3
 1. Define variables

14. Project Management

1.1 Game Objects

- a. Layers
 1. Explain the function of Layers
- b. Tags
 1. Identify the function of Tags
- c. Transform
 1. Explain the function of GameObject components
 2. Recognize GameObject components
- d. Components
 1. Differentiate components by their properties

15. Services

1.1 Ads

- a. Ad Types



- 1. Differentiate Unity Services by a set of features
 - b. Function
 - 1. Recognize Unity Services from a description
- 2.1 Analytics
 - a. Function
 - 1. Summarize the benefits of Unity Analytics
 - b. Project ID
 - 1. Implement Unity Analytics within a game
- 3.1 Cloud Build
 - a. Platforms
 - 1. Examine the features of Unity Cloud Build
- 4.1 Collaborate
 - a. Function
 - 1. Describe the requirements for Unity Collaborate

16. User Interface

- 1.1 Button
 - a. Function
 - 1. Differentiate Button properties
 - b. Interaction
 - 1. Predict methods called by action
 - c. Transition
 - 1. Explain Button properties
- 2.1 Canvas
 - a. Coordinates
 - 1. Differentiate render modes
- 3.1 Image
 - a. Sprites
 - 1. Differentiate UI components
- 4.1 Rect Tool
 - a. Anchor
 - 1. Identify anchor points
 - 2. Utilize pivots and anchors
 - b. Rect Transform
 - 1. Differentiate UI components
 - 2. Assess Rect Transform features
 - 3. Utilize text properties
- 5.1 Slider
 - a. Slider
 - 1. Describe the function of UI components