unity Certification

Unity Certified Associate Courseware

Instructor Resources

20180920R2

In the Instructor Resources, you will find Chapter Outlines and Chapter Rubrics for all twenty chapters of the Unity Certified Associate Courseware. These are resources intended to help both instructors and students to understand the contents of the Unity Certified Associate Courseware and to assess students' understanding of the Courseware material.

Table of Contents

Chapter 01

Welcome to Unity!

1. Certified Associate Courseware Introduction
    - Exploring the Zombie Toys Game Project

2. The Economics of Game Development
    - Assessing Game Markets and Platforms
    - Marketing Methods for Games
    - Monetizing Games and Upgrades

3. Introduction to Game Production
    - Video Game Platforms and Genres
    - Describing the Game Production Pipeline
    - Game Development Jobs and Roles
    - The Game Design Document
    - The Technical Design Document

4. Getting Started in Unity
    - Creating a New Unity Project
    - Using the Unity Asset Store
    - Source Control for Working in Teams

Chapter 01: Rubric

Welcome to Unity!

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of video game development industry.	Learners cannot list, define, nor differentiate common video game development industry terminology.	Learners can list common video game development industry terminology, but definitions and delineations of each term are minimal.	Upon request, learners can consistently list, define, and differentiate common video game development industry terminology.	Upon request, learners can consistently list, define, and differentiate common video game development industry terminology. Learners can cite practical examples of common industry terminology (e.g. AR, VR) and explain their significance in contemporary media today.
Knowledge and understanding of video game production pipeline.	Learners cannot list nor explain video game production processes.	Learners can list video game production processes with minimal explanations. Learners can sequence some production processes in linear order.	Learners can consistently list and explain video game production processes. Learners can sequence production processes in linear order, consistent with a typical game development pipeline.	Learners can consistently list and explain video game production processes. Learners can sequence production processes in linear order, consistent with a typical game development pipeline. Learners can state and explain common challenges and categorize industry standard documents used in game development productions (e.g. Game Design Document, Technical Design Document).
Knowledge and understanding of project management in video game development.	Learners cannot list nor explain project management practices and terminology.	Learners can list project management practices and terminology with minimal explanations.	Learners can consistently list and explain project management processes.	Learners can consistently list/label and explain project management processes. Learners can list and explain project management terminology (e.g. Source control, project ID).
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 02

The Unity User Interface

1. Introduction to the Unity Editor Interface
    - Analyzing the Unity Editor User Interface

2. Utilizing the Unity Editor User Interface
    - Navigating the Scene View Window
    - Utilizing the Game View Window
    - Navigating the Hierarchy Window
    - Using the Inspector Window
    - Managing Assets in the Project Window
    - Searching and Filtering in the Project Window
    - Organizing the Scene with Layers

Chapter 02: Rubric

The Unity User Interface

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of the Unity Editor User Interface.	Learners cannot identify, describe, nor differentiate primary Unity Editor features and terminology.	Learners can identify primary Unity Editor features and terminology, but descriptions and delineations of each are minimal.	Upon request, learners can consistently identify, describe, and differentiate primary Unity Editor features and terminology.	Upon request, learners can consistently identify, describe, and differentiate primary Unity Editor features and terminology. Learners can identify, describe, and differentiate ancillary Unity Editor features and terminology.
Ability to utilize the Unity Editor User Interface (e.g. navigate the UI, search & filter for project data, manipulate project data, organize scenes and assets).	Learners cannot demonstrate efficient utilization of the Unity Editor to achieve common tasks.	Learners can demonstrate utilization of the Unity Editor to achieve common tasks, but some tasks are inefficient or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate efficient utilization of the Unity Editor to achieve common tasks.	Learners can consistently demonstrate efficient utilization of the Unity Editor to achieve common tasks. Learners can demonstrate efficient utilization of the Unity Editor to achieve specialized (i.e. uncommon and more technically challenging) tasks.
Chapter assignment project completion (e.g. guidelines, deliverables, time management.)	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 03

Using GameObjects and Assets

1. Creating and Modifying GameObjects
    - Defining Unity Editor Units
    - Describing Assets in the Production Pipeline
    - Review: Defining an Asset
    - Organizing Assets in the Unity Editor
    - Defining a GameObject
    - Review: Defining a GameObject
    - Creating Unity-native GameObjects
    - Manipulating GameObjects in the Unity Editor
    - Describing What is a Unity-native GameObject
    - The Role of Components in the Unity Editor

2. Defining Prefabs and Scene Structure
    - Defining the Role of the Prefab in Unity
    - Creating and Saving a Scene
    - The Hierarchy of Scenes within a Game

3. Importing Assets into a Project
    - Importing and Configuring a 3D Model
    - Importing Textures for Use in Materials
    - Importing FBX Files with Animation

4. Working with Sprites
    - Introduction to Sprites in Game Development

Chapter 03: Rubric

Using GameObjects and Assets

	Unacceptable	Basic	Proficient	Distinguished
Ability to create and modify game objects and assets.	Learners cannot demonstrate the ability to effectively and efficiently modify game objects and assets.	Learners can demonstrate the ability to modify game objects and assets, but some tasks are inefficient or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently modify game objects and assets.	Learners can consistently demonstrate the ability to effectively and efficiently modify game objects and assets. Learners can demonstrate the ability to perform specialized tasks involving the modification of game objects and assets (e.g. building a custom inspector).
Knowledge and understanding of Prefabs and scene structure.	Learners cannot define nor explain Prefabs and scene structure.	Learners can define Prefabs and scene structure, but explanations of each are minimal.	Upon request, learners can consistently define and explain Prefabs and scene structure.	Upon request, learners can consistently define and explain Prefabs and scene structure. Upon the analysis of Unity scenes, learners can identify opportunities for scene structure optimization and explain how to achieve them.
Ability to import assets into a project.	Learners cannot demonstrate the ability to effectively and efficiently import assets into a project.	Learners can demonstrate the ability to import assets into a project, but some details are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently import assets into a project.	Learners can consistently demonstrate the ability to effectively and efficiently import assets into a project. Learners can anticipate technical challenges with asset project tasks and provide assistance to their peers.
Ability to work with sprites.	Learners cannot demonstrate the ability to effectively and efficiently work with sprites.	Learners can demonstrate the ability to work with sprites, but some tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently work with sprites.	Learners can consistently demonstrate the ability to effectively and efficiently work with sprites. Learners can anticipate technical challenges with sprite project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 04

Managing Projects and Assets

1. Project Management in Unity
    - Introduction to Game Project Management

2. Managing Assets
    - Using the Unity Asset Store (Reprise)
    - Importing Offline Content
    - Creating Project Structure Based on Assets
    - Sorting the Zombie Toys Prop Model Assets
    - Setting Resolution and Type of Texture Files

Chapter 04: Rubric

Managing Projects and Assets

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of game project management in Unity.	Learners cannot identify project management tools nor explain project management methodologies in Unity.	Learners can identify project management tools in Unity, but explanations of project management methodologies in Unity are minimal.	Upon request, learners can consistently identify project management tools and explain project management methodologies in Unity.	Upon request, learners can consistently identify project management tools and explain project management methodologies in Unity. Learners can anticipate project management challenges within Unity game projects and provide assistance to their peers.
Ability to organize the Zombie Toys project.	Learners cannot demonstrate the ability to effectively and efficiently organize the Zombie Toys project.	Learners can demonstrate the ability to organize the Zombie Toys project, but some results are obtained inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently organize the Zombie Toys project.	Learners can consistently demonstrate the ability to effectively and efficiently organize the Zombie Toys project. Upon analysis of various Unity scenes, learners can identify opportunities for organizational optimization and explain how to achieve them.
Ability to manage file structure and metadata files.	Learners cannot demonstrate the ability to effectively and efficiently manage file structure and metadata files.	Learners can demonstrate the ability to manage file structure and metadata files, but some tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can demonstrate the ability to effectively and efficiently manage file structure and metadata files.	Learners can demonstrate the ability to effectively and efficiently manage file structure and metadata files. Learners can anticipate technical challenges with file management project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 05

Preparing Assets for Implementation

1. Best Practices in 3D Content Creation
    - Modeling for Games
    - Animating for Games
    - UV Mapping and Texturing Techniques
    - Exporting to Unity
    - Importing into Unity

2. Materials in Unity
    - The Interaction of Lighting and Materials
    - Discovering the Standard Shader in Unity
    - Exploring other Material Types
    - Analyzing the Benefits of Customer Shaders
    - Creating the Materials for Zombie Toys Props
    - Duplicating and Modifying Materials
    - Case Studies in Material Creation

3. Managing and Using Textures in the Unity Editor
    - Texturing for Game Development
    - Optimization and Reuse of Textures

Chapter 05: Rubric

Preparing Assets for Implementation

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of best practices in production (e.g. modeling, UV mapping and texturing, animating, importing and exporting).	Learners cannot recite nor explain production best practices in Unity.	Learners can recite production best practices in Unity, but explanations of each practice are minimal.	Upon request, learners can consistently recite and explain production best practices in Unity.	Upon request, learners can consistently recite and explain production best practices in Unity. Upon analysis of Unity game projects, learners can identify opportunities for implementing production best practices and explain how to achieve them.
Knowledge and understanding of materials and textures in Unity.	Learners cannot identify, label, nor differentiate materials and textures in Unity.	Learners can identify and label materials and textures in Unity, but their ability to differentiate each is minimal.	Upon request, learners can consistently identify, label, and differentiate materials and textures in Unity.	Upon request, learners can consistently identify, label, and differentiate materials and textures in Unity. Learners can explain how to modify materials and textures in order to produce desired results.
Ability to create the materials and textures for Zombie Toys.	Learners cannot demonstrate the ability to effectively and efficiently create materials and textures for Zombie Toys.	Learners can demonstrate the ability to create materials and texture for Zombie Toys, but some tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create materials and textures for Zombie Toys.	Learners can consistently demonstrate the ability to effectively and efficiently create materials and textures for Zombie Toys. Applying their creativity, learners can demonstrate how to create interesting new materials and textures for their game project.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 06

Assembling the Game Level

1. Branching and Hierarchies
    - Creating Hierarchies in Unity
    - Using Empty GameObjects as Pivots

2. Introduction to Physics in Unity
    - Understanding the Physics System in Unity
    - Introduction to the Rigidbody Component
    - Introduction to Colliders
    - Creating the Colliders for Zombie Toys Props

3. Introduction to Game Level Design
    - Introduction to Game Level Design
    - The Level Design in Zombie Toys

4. Placing Objects in a Scene
    - Importing the Prop Prefabs into the Scene
    - Cloning the Stars
    - Creating the Level Boundaries

Chapter 06: Rubric

Assembling the Game Level

	Unacceptable	Basic	Proficient	Distinguished
Ability to create branching and manage hierarchies.	Learners cannot demonstrate the ability to effectively and efficiently create branching and manage hierarchies.	Learners can demonstrate the ability to create branching and manage hierarchies, but some results are obtained inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create branching and manage hierarchies.	Learners can consistently demonstrate the ability to effectively and efficiently create branching and manage hierarchies. Upon analysis of Unity game projects, learners can identify opportunities for improvements with data hierarchies and demonstrate how to achieve them.
Knowledge and understanding of physics in Unity (e.g. Colliders, rigidbodies).	Learners cannot list nor define physics terminology in Unity.	Learners can list physics terminology in Unity, but definitions of each term is minimal.	Upon request, learners can consistently list and define physics terminology in Unity.	Upon request, learners can consistently list and define physics terminology in Unity. Upon analysis of contemporary Unity game projects, learners can identify and explain the significance that physics has within the game.
Knowledge and understanding of level design in game development and Zombie Toys.	Learners cannot describe the level design in Zombie Toys nor explain the role that level design plays in game development.	Learners can explain the role that level design plays in game development, but their ability to describe the level design in Zombie Toys is minimal or inaccurate.	Upon request, learners can consistently describe the level design in Zombie Toys and explain the role that level design plays in game development.	Upon request, learners can consistently describe the level design in Zombie Toys and explain the role that level design plays in game development. Learners can list and detail options for modifying the level design in Zombie Toys in order to produce a series of desired results.

Chapter 07
Lighting in Games

1. Introduction to Game Lighting
    - Introduction to Game Lighting
    - Differences in Lighting for Games and for Film

2. Placing and Adjusting Lights in a Scene
    - Analyzing the Different Lights and Properties
    - Light Types and Behaviors
    - Using Layers to Exclude Objects from Lighting

3. Casting and Modifying Shadows
    - Mesh Renderer Attributes for Shadows
    - Differentiating Shadow Types
    - Creating Cookies to Shape Lights

4. Faking Shadows for Better Performance
    - Benefits of Faking Shadows in Games
    - Utilizing Painted Shadows
    - Using Projectors to Project Shadow Cookies

5. Lighting the Zombie Toys Game
    - Lighting the Zombie Toys Scene
    - Lighting Variations for Changing the Mood

Chapter 07: Rubric

Lighting in Games

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of lighting in game development and Zombie Toys.	Learners cannot describe the lighting design in Zombie Toys nor explain the role that lighting plays in game development.	Learners can explain the role that lighting plays in game development, but their ability to describe the lighting design in Zombie Toys is minimal or inaccurate.	Upon request, learners can consistently describe the lighting design in Zombie Toys and explain the role that lighting plays in game development.	Upon request, learners can consistently describe the lighting design in Zombie Toys and explain the role that lighting plays in game development. Learners can list and detail options for modifying the lighting design in Zombie Toys in order to produce desired results.
Ability to place and adjust lights in a scene.	Learners cannot demonstrate the ability to effectively and efficiently place and adjust lights in a scene.	Learners can demonstrate the ability to place and adjust lights in a scene, but lighting results are obtained inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently place and adjust lights in a scene.	Learners can consistently demonstrate the ability to effectively and efficiently place and adjust lights in a scene. Upon the analysis of Unity game projects, learners can identify opportunities for improvements in lighting (e.g. placement and/or quality) and explain how to achieve them.
Ability to cast and modify shadows in a scene.	Learners cannot demonstrate the ability to effectively and efficiently cast and modify shadows in a scene.	Learners can demonstrate the ability to cast and modify shadows in a scene, but shadow results are obtained inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently cast and modify shadows in a scene.	Learners can consistently demonstrate the ability to effectively and efficiently cast and modify shadows in a scene. Upon the analysis of Unity game projects, learners can identify opportunities for improvements with shadows (e.g. placement and/or quality) and explain how to achieve them.
Ability to develop the lighting for Zombie Toys.	Learners cannot demonstrate the ability to effectively and efficiently develop the lighting for Zombie Toys.	Learners can demonstrate the ability to develop the lighting for Zombie Toys, but some tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently develop the lighting for Zombie Toys.	Learners can consistently demonstrate the ability to effectively and efficiently develop the lighting for Zombie Toys. Applying their creativity, learners can demonstrate how to create compelling new lighting design modifications for their game project.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 08

Baking Lighting in Game Production

1. Light Baking in Video Games
    - Introduction to Light Baking in Video Games

2. Setting Objects to Participate in Light Baking
    - Marking Objects as Static for Light Baking
    - Creating UV Coordinates for Light Baking

3. Baking Lighting
    - Continuous and Manual Light Baking
    - Placing Light Probes for Moving Objects
    - Creating Reflection Probes

4. Baking the Lighting in Zombie Toys
    - Creating the Light Probes in Zombie Toys

Chapter 08: Rubric

Baking Lighting in Game Production

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of light baking in Unity game development (e.g. role, process, advantages).	Learners cannot describe the light baking process nor explain the role that light baking plays in Unity game development.	Learners can explain the role that light baking plays in Unity game development, but their ability to describe the light baking process is minimal or inaccurate.	Upon request, learners can consistently describe the light baking process and explain the role that light baking plays in Unity game development.	Upon request, learners can consistently describe the light baking process and explain the role that light baking plays in Unity game development. Upon the analysis of Unity game projects, learners can list and detail options for improving light baking (e.g. setup and/or quality) and explain how to achieve them.
Ability to implement light baking within a scene.	Learners cannot demonstrate the ability to effectively and efficiently implement light baking in a scene.	Learners can demonstrate the ability to implement light baking in a scene, but results are obtained inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement light baking in a scene.	Learners can consistently demonstrate the ability to effectively and efficiently implement light baking in a scene. Learners can anticipate technical challenges with light baking project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 09

Animating GameObjects in the Unity Editor

1. Introduction to Animation in Game Development
    - Introduction to Animation in Game Development
2. Animating in the Unity Editor
    - Creating Animation in the Unity Editor
    - Refining Animation in the Unity Editor

Chapter 09: Rubric

Animating GameObjects in the Unity Editor

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of animation in game development (e.g. role, types, terms, process , advantages).	Learners cannot state animation terms nor explain animation processes in game development.	Learners can state animation terms in game development, but their ability to explain animation processes is minimal.	Upon request, learners can consistently state animation terms and explain animation processes in game development.	Upon request, learners can consistently state animation terms and explain animation processes in game development. Upon analysis of Unity game projects, learners can identify opportunities for animation improvements (e.g. setup and/or quality) and explain how to achieve them.
Ability to create animation within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently create animation within a Unity scene.	Learners can demonstrate the ability to create animation within a Unity scene, but results are obtained inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create animation within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently create animation within a Unity scene. Learners can anticipate technical challenges with animation project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 10

Bringing Animations into the Game

1. Importing Animated Characters
    - Introduction to Rigging and Imported Animation
    - Recognizing Asset Data when Importing
    - Differentiating Available Rig Animation Types
    - Creating and Naming Animation Clips

2. Creating an Animator Controller
    - Introduction to the Animator Controller
    - Creating and Modifying Animation States
    - Creating Parameters to Control Transitions
    - Creating an Animator Override Controller

Chapter 10: Rubric

Bringing Animations into the Game

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of best practices for importing animated characters (e.g. rig types, animation clip management).	Learners cannot identify nor describe best practices for importing animated characters in Unity game development.	Learners can identify best practices for importing animated characters in game development, but their descriptions are minimal.	Upon request, learners can consistently identify and describe best practices for importing animated characters in Unity game development.	Upon request, learners can consistently identify and describe best practices for importing animated characters in Unity game development. Upon analysis of Unity game projects, learners can identify opportunities for character data improvements (e.g. configuration, clip naming) and explain how to achieve them.
Ability to create an Animator Controller within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently create an Animator Controller within a Unity scene.	Learners can demonstrate the ability to create an Animator Controller within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create an Animator Controller within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently create an Animator Controller within a Unity scene. Learners can anticipate technical challenges involving Animator Controller project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 11
Scripting in Game Development

1. Intro to Scripting in Game Development
    - Intro to Scripting in Game Development

2. Creating Scripts in Unity
    - Creating and Saving a Script in Unity
    - Analyzing the Default Script Methods

3. Scripting Primer and Best Practices
    - Attaching a Script to a GameObject
    - Declaring Variables
    - List of Variable Types

4. Creating Conditions in Scripting
    - Introduction to Conditions
    - The “if” Condition
    - The “if else” Condition
    - Complex Conditions

5. Looping
    - Introduction to Looping
    - The “while” Loop
    - The “for” Loop

6. Creating Custom Methods
    - The Benefits of Using Custom Methods
    - Utilizing Arguments
    - Utilizing Method Return Types

7. Coroutines
    - Introduction to Coroutines

8. Accessing Components via Script
    - Utilizing the GetComponent() Function

9. Common Code Cases
    - Common Pieces of Zombie Toys Code

Chapter 11: Rubric

Scripting in Game Development

	Unacceptable	Basic	Proficient	Distinguished
Understanding of scripting in Unity game development.	Learners cannot describe the role of scripting in Unity game development.	Learners can describe the role of scripting in Unity game development, but description is minimal.	Learners can consistently describe the role of scripting in Unity game development.	Learners can consistently describe the role of scripting in Unity game development. Learners can cite examples of (and explain) scripting advantages in contemporary Unity game development projects.
Knowledge and understanding of data types in Unity (e.g. int, float, bool, string, Vector3, Quaternion, Transform).	Learners cannot enumerate, identify, define, nor provide examples (within code) of at least three (3) data types in Unity.	Upon request, learners can consistently enumerate, identify, define, and provide examples (within code) of at least five (5) data types in Unity.	Upon request, learners can consistently identify, define, and provide examples (within code) of at least seven (7) data types.	Upon request, learners can consistently identify, define, and provide examples (within code) of at least seven (7) data types in Unity. Learners can enumerate, identify, define, and provide examples (within code) of additional data types not covered in the Unity Certified Associate Courseware.
Ability to evaluate conditional statements in C#. Instructor will provide statements for learners to evaluate and answer. (e.g. (2+2) * 2 < (2*2) + 2)	Learners cannot demonstrate the ability to correctly evaluate more than three (3) conditional statements in C#.	Learners can demonstrate the ability to correctly evaluate at least five (5) conditional statements in C#.	Learners can demonstrate the ability to correctly evaluate at least seven (7) conditional statements in C#.	Learners can demonstrate the ability to correctly evaluate at least ten (10) conditional statements in C#.

Ability to evaluate loop cycles.
 	
Instructor will provide code snippets for learners to evaluate and answer,
e.g. Given the following code snippet, how many times will the message be printed to the console?
 	
void Update()
{
int counter = 35;
for(int i = 25, i < 100; i++)
}
	Learners cannot demonstrate the ability to correctly evaluate more than three (3) statements.	Learners can demonstrate the ability to correctly evaluate at least five (5) statements.	Learners can demonstrate the ability to correctly evaluate at least seven (7) statements.	Learners can demonstrate the ability to correctly evaluate at least ten (10) statements.

Chapter 11: Rubric

Scripting in Game Development (cont)

	Unacceptable	Basic	Proficient	Distinguished
Ability to complete programming assignments. Instructor will provide programming assignments for learners to complete. (e.g. <<Write a script that scales a gameobject 10 units larger than its current size, then scales the gameobject back to its original size>>)	Learners cannot demonstrate the ability to correctly complete the programming assignments.	Learners can demonstrate the ability to complete the programming assignments, but some code is unclear or inefficient.	Learners can consistently demonstrate the ability to correctly complete all programming assignments.	Learners can consistently demonstrate the ability to correctly complete all programming assignments. Learners can provide technical guidance to peers, increasing their likelihood and/or ability to complete their programming assignments.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 12

Implementing Navigation and Pathfinding

1. Introduction to Navigation and Pathfinding
    - Introduction to Navigation in Unity
    - Describing a NavMesh
    - Defining a NavMesh Agent
    - Describing a NavMesh Obstacle

Chapter 12: Rubric

Implementing Navigation and Pathfinding

	Unacceptable	Basic	Proficient	Distinguished
Knowledge of navigation and pathfinding concepts and terminology (e.g. NavMesh, NavMesh Agent, NavMesh Baking, Obstacle Avoidance).	Learners cannot list nor define at least three (3) navigation and pathfinding terms.	Learners can list at least three (3) navigation and pathfinding terms, but definitions of each are minimal.	Learners can consistently list and define at least three (3) navigation and pathfinding terms.	Learners can consistently list and define all pertinent navigation and pathfinding terms. Learners can also explain additional navigation and pathfinding concepts (e.g. pathfinding algorithms).
Navigation Assignment Instructor should ask learners to make a simple game project, with assets provided from Zombie Toys. (e.g. create a maze with primitive objects and make the sheep character navigate and solve the maze).	Navigation project is not complete or does not function as planned.	Navigation project is complete and functions as planned, however the environment is simple or lacks individuality.	Navigation project is complete and functions as planned. Additional features and/or more characters are included in the navigation project, adding complexity and individuality.	Navigation project is complete and functions as planned. Additional features and/or more characters are included in the navigation project, adding complexity and individuality. Production best practices are also implemented in the project, such as naming conventions for assets.
Assignment Creativity Instructor should assess the creativity displayed in the navigation project.	The navigation project is not creative. The project does not offer new approaches or ideas beyond the original concept.	The navigation project is somewhat creative, offering some new ideas. However, most approaches or ideas presented are typical (i.e. industry standard) in nature.	The navigation project is creative, offering new ideas and approaches to change the game in a unique way. The navigation project takes some risks in ideas suggested for the revision.	The navigation project is highly creative, offering multiple alternative approaches to changing the game from its original design. Familiar methods and approaches have been combined to generate new and innovative results.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 13
Building the Player and Allies

1. Creating a Player Controller
    - Examining Why to Use a Custom Controller
    - Creating the Player Controller GameObject

2. Adding a Game Manager
    - Explaining the Purpose of the Game Manager

3. Making the Controller Functional
    - Adding Scripts for Behavior
    - Configuring the Camera

4. Creating the Sheep Ally
    - Building the Sheep Ally From a Model

5. Creating the Dog Ally
    - Building the Dog Ally From a Model

Chapter 13: Rubric

Building the Player and Allies

	Unacceptable	Basic	Proficient	Distinguished
Ability to create a player within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently create a player within a Unity scene.	Learners can demonstrate the ability to create a player within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create a player within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently create a player within a Unity scene. Learners can anticipate technical challenges involving player creation project tasks and provide assistance to their peers.
Ability to create a Game Manager within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently create a Game Manager within a Unity scene.	Learners can demonstrate the ability to create a Game Manager within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create a Game Manager within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently create a Game Manager within a Unity scene. Learners can anticipate technical challenges involving Game Manager project tasks and provide assistance to their peers.
Ability to implement the Zombie Toys character allies.	Learners cannot demonstrate the ability to effectively and efficiently implement the Zombie Toys character allies.	Learners can demonstrate the ability to implement the Zombie Toys character allies, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement the Zombie Toys character allies.	Learners can consistently demonstrate the ability to effectively and efficiently implement the Zombie Toys character allies. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 14
Building the Enemies

1. Creating an Enemy
    - Designing the Enemy Behaviors
    - Creating the First Enemy Character
    - Creating the Enemy Animator Controller

2. Creating Additional Enemies
    - Creating the Zombear Enemy
    - Creating the Zombie Duck Enemy
    - Creating the Other Enemies

3. Integrating Enemies into the Game
    - Placing the Spawn Points
    - Spawning the Enemies

Chapter 14: Rubric

Building the Enemies

	Unacceptable	Basic	Proficient	Distinguished
Ability to create an enemy within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently create an enemy within a Unity scene.	Learners can demonstrate the ability to create an enemy within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create an enemy within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently create an enemy within a Unity scene. Learners can anticipate technical challenges involving enemy creation project tasks and provide assistance to their peers.
Ability to implement enemy behaviors within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently implement enemy behaviors within a Unity scene.	Learners can demonstrate the ability to implement enemy behaviors within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement enemy behaviors within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently implement enemy behaviors within a Unity scene. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Ability to spawn enemies within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently spawn enemies within a Unity scene.	Learners can demonstrate the ability to spawn enemies within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently spawn enemies within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently spawn enemies within a Unity scene. Learners can anticipate technical challenges involving enemy spawning project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 15

Creating Particle Systems

1. Intro to the Particle Systems in the Unity Editor
    - Examples of Unity Particles in Video Games
    - The Role of the Effects Artist in Video Games
    - Comparing Game Effects with Other Media

2. Production Best Practices for Particle Systems
    - Introduction to Unity's Particle System
    - Analyzing Existing Particle Effects
    - Setting Up the Interface for Effects

3. Case Study: Developing the Lightning Attack
    - Overview of the Lightning Attack
    - Building the Lightning Attack Hit
    - Building the Lightning Attack Emitter
    - Building the Lightning Bolt
    - Integrating the Lightning Attack into the Game

4. Case Study: Developing the Frost Attack
    - Introduction to the Frost Attack
    - Building the Frost Debuff
    - Building the Frost Attack Emitter
    - Building the Frost Cone Effect
    - Integrating the Frost Attack into the Game

5. Case Study: Developing the Stink Bomb Attack
    - Introduction to the Stink Bomb Attack
    - Creating the Stink Bomb Hit Effect
    - Creating the Stink Bomb Attack Reticle
    - Building the Stink Bomb Attack Emitter
    - Building the Stink Projectile
    - Integrating the Stink Bomb into the Game

6. Case Study: Developing the Slime Attack
    - Introduction to the Slime Attack
    - Creating the Slime Hit Effect
    - Creating the Slime Debuff
    - Creating the Slime Attack Reticle
    - Building the Slime Attack Emitter
    - Building the Slime Projectile
    - Integrating the Slime Attack into the Game

7. Finalizing Player Attacks
    - Adding the Ally Manager

Chapter 15: Rubric

Creating Particle Systems

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of particle systems in Unity game development (e.g. emission, shape, inherit velocity, line renderer, duration).	Learners cannot list nor define at least five (5) particle system terms in Unity game development.	Learners can list at least five (5) particle system terms in Unity game development, but definitions of each are minimal.	Learners can consistently list and define at least five (5) particle system terms in Unity game development.	Learners can consistently list and define at least five (5) particle system terms in Unity game development. Learners can also list and define additional particle system terminology (e.g. emitters, speed, force, color, collision).
Ability to implement the lightning attack within the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently implement the lightning attack within the Zombie Toys game project.	Learners can demonstrate the ability to implement the lightning attack within the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement the lightning attack within the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently implement the lightning attack within the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Ability to implement the freeze attack within the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently implement the freeze attack within the Zombie Toys game project.	Learners can demonstrate the ability to implement the freeze attack within the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement the freeze attack within the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently implement the freeze attack within the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Ability to implement the stink bomb attack within the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently implement the stink bomb attack within the Zombie Toys game project.	Learners can demonstrate the ability to implement the stink bomb attack within the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement the stink bomb attack within the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently implement the stink bomb attack within the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.

Chapter 15: Rubric

Creating Particle Systems (cont'd)

	Unacceptable	Basic	Proficient	Distinguished
Ability to implement the slime attack within the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently implement the slime attack within the Zombie Toys game project.	Learners can demonstrate the ability to implement the slime attack within the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement the slime attack within the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently implement the slime attack within the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 16

Adding Audio to Game Levels

1. Introduction to Audio in Game Development
    - Introduction to Audio in Game Development

2. Importing Audio into Unity
    - Introduction to Importing Audio in Unity
    - Supported Audio Formats in Unity

3. Playing Audio in the Unity Editor
    - Audio Listeners and Audio Sources
    - Testing Audio Sources in the Scene

4. Mixing Audio in Unity
    - Using Audio Mixers and Audio Mixer Groups
    - Setting up the Zombie Toys Audio Mixers

5. Creating Audio Effects
    - Introduction to Audio Effects
    - Adding Audio to the Zombie Toys Characters
    - Implementing Audio Ducking

Chapter 16: Rubric

Adding Audio to Game Levels

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of audio in Unity game development (e.g. sources, listeners, reverb zones, doppler effect).	Learners cannot list nor define at least five (5) audio terms in Unity game development.	Learners can list at least five (5) audio terms in Unity game development, but definitions of each are minimal.	Learners can consistently list and define at least five (5) audio terms in Unity game development.	Learners can consistently list and define at least five (5) audio terms in Unity game development. Learners can also list and define additional audio terminology (e.g. Audio Profiler, audio clips, Audio Mixer, ducking).
Ability to import and control audio in the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently import and control audio in the Zombie Toys game project.	Learners can demonstrate the ability to import and control audio in the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently import and control audio in the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently import and control audio in the Zombie Toys game project. Learners can anticipate technical challenges involving audio project tasks and provide assistance to their peers.
Ability to implement audio effects within the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently implement audio effects within the Zombie Toys game project.	Learners can demonstrate the ability to implement audio effects within the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement audio effects within the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently implement audio effects within the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 17

Building the Camera and Player Selection System

1. Intro to the Camera and Player Selection Behaviors
    - Analyzing the Player Selection System

2. Creating Another Player Option
    - Making the Player Selectable
    - Adding Another Player

3. Finalizing the Camera
    - Adding Camera Animations
    - Configuring the Camera Animator Controller
    - Applying Behavior to the Camera
    - Adding Character Selection Spotlights

Chapter 17: Rubric

Building the Camera and Player Selection System

	Unacceptable	Basic	Proficient	Distinguished
Ability to duplicate a player within a Unity scene.	Learners cannot demonstrate the ability to effectively and efficiently duplicate a player within a Unity scene.	Learners can demonstrate the ability to duplicate a player within a Unity scene, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently duplicate a player within a Unity scene.	Learners can consistently demonstrate the ability to effectively and efficiently duplicate a player within a Unity scene. Learners can anticipate technical challenges involving player creation project tasks and provide assistance to their peers.
Ability to integrate character selection within the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently integrate character selection within the Zombie Toys game project.	Learners can demonstrate the ability to integrate character selection within the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently integrate character selection within the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently integrate character selection within the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Ability to implement camera animation into the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently implement camera animation into the Zombie Toys game project.	Learners can demonstrate the ability to implement camera animation into in the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently implement camera animation into the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently implement camera animation into the Zombie Toys game project. Learners can anticipate technical challenges involving audio project tasks and provide assistance to their peers.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 18

Designing User Interfaces for Games

1. Introduction to Designing the User Interface
    - Assessing User Interface Design Needs
    - Examining the UI Tools in the Unity Editor

2. Creating a User Interface
    - Investigating the Canvas Functionality
    - Utilizing the Power of the Rect Transform
    - Creating a UI Button
    - Creating a UI Image
    - Creating UI Text
    - Creating Interaction in the UI with Events

Chapter 18: Rubric

Designing User Interfaces for Games

	Unacceptable	Basic	Proficient	Distinguished
Knowledge of User Interface design in Unity game development (e.g. GUI, Rect Transform, UI Button, UI Image, UI Text).	Learners cannot list nor define at least five (5) UI terms in Unity game development.	Learners can list at least five (5) UI terms in Unity game development, but definitions of each are minimal.	Learners can consistently list and define at least five (5) UI terms in Unity game development.	Learners can consistently list and define all pertinent UI terms in Unity game development. Learners can also detail additional UI concepts (e.g. designing for multiple resolutions, creating UI elements from scripting).
Ability to create the User Interface in the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently create the UI in the Zombie Toys game project.	Learners can demonstrate the ability to create the UI in the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently create the UI in the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently create the UI in the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 19
Building and Deploying the Game

1. Building the Game
    - Introduction to the Build Process
    - Adjusting the Player Settings
    - Building the Game

2. Protecting Your Creation
    - Legal Considerations for Your Game
    - Unity Services
    - Unlocking the Unity Platform Potential
    - Surveying Unity Services

Chapter 19: Rubric

Building and Deploying the Game

	Unacceptable	Basic	Proficient	Distinguished
Knowledge and understanding of the build process in Unity game development (e.g. terminology, procedures, challenges).	Learners cannot list key terms nor describe the build process in Unity game development.	Learners can list key terms in Unity game development, but their description of the build process is minimal.	Learners can consistently list key terms and describe the build process in Unity game development.	Learners can consistently list key terms and describe the build process in Unity game development. Learners can also detail additional build considerations (e.g. graphics optimizations, version control options, Unity Cloud Build).
Ability to build the Zombie Toys game project.	Learners cannot demonstrate the ability to effectively and efficiently build the Zombie Toys game project.	Learners can demonstrate the ability to build the Zombie Toys game project, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently build the Zombie Toys game project.	Learners can consistently demonstrate the ability to effectively and efficiently build the Zombie Toys game project. Learners can implement modifications to the project that adds creativity, complexity, and/or individuality.
Knowledge of the available Unity Services, and their access requirements.	Learners cannot list the available Unity Services, nor describe their functions.	Learners can list the available Unity Services but not adequately differentiate them.	Learners can list and describe each of the Unity Services, including Cloud Build, Ads, Analytics, Multi-Player, Everyplay, Certification, and Performance Reporting. Additionally, learners can list the access requirements for each where applicable.	Learners can consistently identify and categorize the Unity Services, and describe access requirements for each. Learners can show how to access Unity Services, and demonstrate implementation of Ads and Cloud Build.
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).

Chapter 20

Preparing for Mobile Deployment

1. Modifying Zombie Toys for Mobile
    - Introduction to Mobile Development in Unity
    - Changing the Build Platform to Mobile
    - Adding the Mobile Interface UI
    - Implementing Mobile Input Behaviors

Chapter 20: Rubric

Preparing for Mobile Deployment

	Unacceptable	Basic	Proficient	Distinguished
Understanding of mobile game deployment preparation in Unity (e.g. processes, considerations, challenges).	Learners cannot explain mobile game deployment preparation processes and considerations in Unity.	Learners can provide an overview of mobile game deployment preparation processes and considerations in Unity, but some explanations lack detail or are inaccurate.	Learners can consistently explain mobile game deployment preparation processes and considerations in Unity.	Learners can consistently explain mobile game deployment preparation processes and considerations in Unity. Learners can provide technical details and considerations that are specific to popular mobile platforms (e.g. iOS, Android).
Ability to prepare the Zombie Toys game project for mobile deployment.	Learners cannot demonstrate the ability to effectively and efficiently prepare the Zombie Toys game project for mobile deployment.	Learners can demonstrate the ability to prepare the Zombie Toys game project for mobile deployment, but tasks are performed inefficiently or could be achieved with a better workflow (i.e. process).	Learners can consistently demonstrate the ability to effectively and efficiently prepare the Zombie Toys game project for mobile deployment.	Learners can consistently demonstrate the ability to effectively and efficiently prepare the Zombie Toys game project for mobile deployment. Learners can implement technical modifications to the project that adds complexity,
Chapter assignment project completion (e.g. guidelines, deliverables, time management).	Chapter assignment(s) was not delivered, was delivered late, is sloppy, and/or includes many errors.	Chapter assignment(s) was delivered on-time, but may include minor errors.	Chapter assignment(s) was delivered on-time, works as requested, and includes no errors.	Chapter assignment(s) was delivered on-time, works as requested, contains no errors, and contains additional, unique features designed and implemented by the learner (e.g. gameplay, assets, etc.).