Programmer

Highlight your programming skills to employers

Demonstrate core skills and competencies across programming, UI, debugging and asset management to help you obtain your first professional programming role with Unity.

Prerequisites

- 2-3 semesters of post-secondary Unity classwork or equivalent independent study
- Experience with a diverse range of Unity projects
- Importing assets or code, including from the Unity Asset Store or Unity Package Manager, and addressing conflicts that arise as a result
- Performing debugging of non-complex problems
- Interpreting pre-existing, well-documented code
- Integrating and modifying pre-existing well-documented code
- Building basic scene management, including loading scenes
- Creating, editing, and using Prefabs
- Deploying a basic build

Exam details

The exam is based on Unity 6.

This exam is available in the following languages:

- English
- Chinese - Simplified & Traditional
- Japanese

- Korean
- Spanish - Latin America

What's on the exam?

Unity Programming

- Evaluate code for integration into an existing system created/architected by a lead
- Apply coding standards and best practices as guided by senior programmers
- Determine code that would accomplish a specified interaction or programming logic
- Determine the process to implement transitions between scenes
- Save data between scenes and between sessions using approaches such as static variables and PlayerPrefs
- Obtain a defined result by using Unity API methods, given Unity's API documentation
- Select the appropriate properties, scripts, and components of GameObjects for required tasks
- Explain the differences between basic inheritance and interfaces
- Choose the appropriate commonly used data structures for a specific situation including but not limited to lists, arrays, and dictionaries
- Choose the appropriate data types for a specific situation including but not limited to floats, bools, and strings
- Build an application to WebGL or a personal computer

UI

- Arrange UI components on the canvas according to a defined layout using anchors, pivots, and groups
- Identify the process required to display data in various UI elements
- Explain how to use the UnityEvent system to respond to User Input

Debugging

- Program debug messages to identify the possible causes of code failing to execute as expected
- Identify the cause of a compilation error, given a block of code

- Identify errors caused by a null variable
- Identify techniques required to refactor and improve code to fit defined coding standards
- Select appropriate profiling tools to identify the sources of performance problems

Asset Management

- Explain how to use prefabs in a scene
- Describe the process and outcomes for changing a nested prefab or prefab variant
- Explain the primary purposes of version control when working in Unity