



CERTIFIED
User
Programmer

Objetivos del Examen

Usuario certificado en Programación Unity

El examen de certificación de Programador de usuario certificado de Unity probará los conceptos básicos de la programación C# dentro del software Unity para crear interactividad en juegos, aplicaciones, AR/VR y otras experiencias. Los objetivos del examen están alineados con los estándares actuales de la industria establecidos por profesionales y educadores. Se espera que las personas tengan al menos 150 horas de uso y capacitación del software Unity para obtener esta certificación.

Las personas que han obtenido la certificación de Programador de Usuario Certificado de Unity han demostrado dominio de las siguientes habilidades:

1.

Depuración, resolución de problemas e interpretación de la API

- 1.1. Dado un ejemplo de un mensaje de registro de depuración, cree el código que creó el mensaje de registro.
- 1.2. Dado un clip de código y sus mensajes de error asociados, determine qué objetos son nulos.
- 1.3. Dada una tarea de programación específica que requiere el uso de una clase particular en la API, determine el método y/o las propiedades, argumentos u otra sintaxis apropiados a usar.

2. Creando código

- 2.1. Indique cuándo y cómo inicializar y usar variables, incluido, entre otros, el uso apropiado de todos los modificadores de variables y colecciones de datos, como matrices, listas y diccionarios.
- 2.2. Dada una lista de palabras clave y elementos de sintaxis, construya una función viable declaración.
- 2.3. Dado un clip de código y una descripción del resultado deseado, identifique el función adecuada para controlar o desencadenar un estado que incluye, entre otros, el Animator Controller.
- 2.4. Dado un escenario donde se requiere un tipo específico de entrada y el edificio Se proporcionan los bloques necesarios, construya el oyente de entrada necesario, incluido, entre otros, el teclado y la entrada táctil.
- 2.5. Demostrar cuándo y/o cómo utilizar las distintas lógicas y controles de flujo. Operadores utilizados en C# y Unity.
- 2.6. Dado un escenario, identifique las acciones apropiadas a tomar cuando un elemento de la interfaz de usuario informa un cambio.

3. Evaluación del código

- 3.1. Dado un escenario sobre la necesidad de administrar una función de evento, determine la acción adecuada a tomar, que incluye, entre otras, el teclado y la entrada táctil.



- 3.2. Dado un clip de código que produce un error debido a una variable cuyo tipo de datos se declara incorrectamente, identifique el error.
- 3.3. Dado un clip de código que produce un error porque una función o variable se declara o se usa incorrectamente (desajuste público/privado), identifique el error, incluido, entre otros, el uso de eventos de animación.
- 3.4. Dado un clip de código que contiene una definición de clase, distinga si la clase es una clase ECS o algún otro tipo de clase.
- 3.5. Dado un conjunto de clips de código, reconozca el clip que utiliza convenciones de nomenclatura que respetan los estándares de nomenclatura de Unity.
- 3.6. Dado un clip de código (o un conjunto de clips de código), reconozca los comentarios que describen con precisión lo que está haciendo el código.

4. Navegando por la interfaz

- 4.1. Describir el propósito, las características y las funciones de los distintos Unity IDE. ventanas.
- 4.2. Demuestre cómo cambiar el IDE de secuencias de comandos predeterminado.

Dado un escenario que incluye lo siguiente, cree un estado funcional máquina.
 - a. una porción limitada de un escenario de juego
 - b. un conjunto de clips de animación
 - c. una lista de configuraciones de propiedades
- 4.4. Cree y programe una máquina de estado de función dentro de Unity Animator Controlador que incluye, entre otros, el uso de la sintaxis de las funciones de Animator.