Objetivos del examen

Unity Certified Associate Game Developer

1. Trabajar en la industria de los videojuegos

1.1. Demostrar conocimiento del sector y preparación para el empleo

1.1.1. Puestos de trabajo habituales, hardware de consumo actual y heredado, fases de producción, desarrollo AAA, desarrollo indie, editoras (publishers), desarrolladoras, uso de IA

1.1.2. Acuerdo de confidencialidad (NDA), derechos de autor (Copyright), marca registrada (Trademark), propiedad intelectual (IP)

1.1.3. Crítica constructiva, colaboración, programación por parejas (pair programming)

1.2. Identificar los principios fundamentales del arte en videojuegos, incluyendo entre otros los principios de diseño de personajes, la optimización de assets de modelos, el concept art, la teoría del color unificada y el diseño de entornos. Recortar (slice) spritesheets para su uso en una escena 2D, incluyendo entre otros el uso del Sprite Editor predeterminado y el 9-slicing.

1.2.1. Principios de diseño de personajes, optimización de assets de modelos, LOD, paleta de colores, cómo influye el arte en la atmósfera, concept art, teoría del color unificada, diseño de entornos (incluyendo el grayboxing), 2D, 3D, 2.5D, ortográfico, isométrico, storyboarding, animatics

1.3. Identificar los principios fundamentales del diseño de videojuegos, incluyendo entre otros la perspectiva del jugador, las mecánicas de juego, los géneros y el gameplay loop. Crear, modificar y utilizar Prefabs.

1.3.1. Perspectiva del jugador (primera persona, tercera persona, isométrica, cenital/top down, etc.), mecánicas de juego, géneros, subgéneros, tipos de controles (digital frente a analógico), gameplay loop

2. Fundamentos de Unity

2.1. Demostrar la capacidad de utilizar Unity Hub

2.1.1. Instalaciones del Unity Editor, gestión de proyectos, gestión de licencias e instalación de módulos

2.2. Utilizar la interfaz del Editor

2.2.1. Personalizar la interfaz del Editor, ventana Console, ventana Hierarchy, ventana Inspector, ventana Project, ventana Scene, Toolbar, Gizmos, package manager, Rendering Pipelines, Build Profiles, ventana de búsqueda, ventana de configuración del proyecto (project settings)

2.3. Crear y modificar GameObjects

© 2026 Unity Technologies Incorporated. Todos los derechos reservados. “Unity” es una marca comercial o marca registrada de Unity Technologies o sus filiales en EE. UU. y otros países. Documento traducido al español (España) a partir del original en inglés; se conservan en inglés los nombres del examen, del programa y la r

2.3.1. Layers, tags, transforms, components, tipos de GameObjects integrados, emparentado de objetos (parenting)

2.4. Crear Materials y modificar sus propiedades

2.4.1. Universal Render Pipeline: material Lit. Comprender los efectos de modificar las Surface Options, Surface Inputs y Detail Inputs

2.5. Gestionar prefabs, ScriptableObjects, escenas y assets

2.5.1. Configuración de importación (import settings)

2.5.2. Formatos de archivo, prefabs, prefabs anidados (nested prefabs), variantes de prefab (prefab variants), Sprite Editor, Unity Asset Store, slicing/9-slicing, trabajo con escenas

2.6. Crear y modificar input actions mediante el Input System

2.6.1. Input action map, input actions, activación de un método mediante una input action

2.7. Identificar el Unity Gaming Service adecuado para lograr un objetivo

2.7.1. Unity Ads, Unity Build Automation, Unity DevOps Version Control, LiveOps, Multiplayer

3. Sistemas esenciales de Unity

3.1. Implementar física en un proyecto. Identificar la iluminación básica, incluyendo entre otros las sombras, la configuración de luces y los tipos de luz como directional, area, spot y point.

3.1.1. Colliders, Triggers, Raycast, Rigidbodies, Collision Layer Matrix, física 2D frente a 3D

3.2. Identificar las herramientas y propiedades de Navigation y Pathfinding

3.2.1. NavMesh agents, NavMesh surface, obstacle avoidance, Max Slope, Bake Settings

3.3. Implementar la configuración de audio en un proyecto

3.3.1. Audio effects, propiedades de audio, opciones de audio, reverb zone, audio source, looping de la fuente, control de volumen, audio clips, tipos de archivo, Audio Listener, Audio Mixers

3.4. Configurar comportamientos de animación mediante el Animator Controller

3.4.1. States (incluyendo Any State y Default State), tipos de parámetros de estado, transitions, propiedades de las transitions, condiciones de las transitions

3.5. Utilizar elementos o componentes de interfaz de usuario

3.5.1. Anchors, Rect Transform, Canvas y UI Components, incluyendo entre otros buttons, TextMesh Pro, Slider, Image.

3.5.2. UI Toolkit

© 2026 Unity Technologies Incorporated. Todos los derechos reservados. “Unity” es una marca comercial o marca registrada de Unity Technologies o sus filiales en EE. UU. y otros países. Documento traducido al español (España) a partir del original en inglés; se conservan en inglés los nombres del examen, del programa y la r

Objetivos del examen

Unity Certified Associate Game Developer

4. Composición de escenas y renderizado

4.1. Utilizar herramientas de construcción de escenas

4.1.1. Probuilder, Terrain tools, tilemaps

4.1.2. Terrain settings

4.2. Aplicar efectos visuales mediante partículas y post-procesado

4.2.1. Configuración del particle system, opciones de partículas según su resultado, emitters, post-procesado (post processing), volume profiles

4.3. Aplicar iluminación en una escena

4.3.1. Global Illumination, Baked GI, Lightmaps, Realtime, configuración de Lightmap UV, tipos de sombra, Culling Mask, Light Intensity, tipos de luz, cookies, light probes, reflection probes, light volumes, reflection volumes

5. Programación en Unity

5.1. Determinar el propósito del código C#

5.1.1. Classes, Monobehaviour, eventos del ciclo de vida de MonoBehaviour (Awake, Fixed Update, etc.), Methods, Variables, Public frente a Private, atributos de variable (Serialized Field, Tooltip, Header), Transform, Quaternion, Vector3, Delta Time, GetComponent y la documentación de Unity

5.1.2. Llamar a métodos mediante consultas de la Input System Action API, incluyendo entre otros WasPressedThisFrame(), IsPressed()

5.2. Depurar código C#

5.2.1. Mensajes de error, mensajes de advertencia, debug log, Console y la configuración de Console

© 2026 Unity Technologies Incorporated. Todos los derechos reservados. “Unity” es una marca comercial o marca registrada de Unity Technologies o sus filiales en EE. UU. y otros países. Documento traducido al español (España) a partir del original en inglés; se conservan en inglés los nombres del examen, del programa y la r