



Crear con código

Resumen

En este curso, aprenderás a crear con código mientras

programas tus propios

proyectos desde cero en C#. Al iterar con prototipos, abordar retos de programación, completar cuestionarios y desarrollar tu propio proyecto personal, pasarás de ser un principiante absoluto a ser un desarrollador de Unity capaz. Lo más importante, sin embargo, es que cuando complete este curso, tendrá la confianza de que puedes crear con código

Unidad 1

Resumen

En esta Unidad, usted programará un auto moviéndose de lado a lado en un camino flotante, tratando de evitar (o golpear) obstáculos en el camino. Además de familiarizarse con el editor y el flujo de trabajo de Unity, aprenderá a crear nuevos scripts C# y a realizar una programación sencilla. Al final de la Unidad, podrá llamar a las funciones básicas, luego declarar y ajustar nuevas variables para modificar los resultados de esas funciones.

Objetivo

Al final de esta Unidad, usted podrá ser capaz de hacer esto::

Navegar cómodamente por el espacio 3D y el Editor de Unity.

Añadir y manipular objetos en la escena para colocarlos donde quieras, incluida la

cámara.

Controlar el diseño del Editor de Unity para que se adapte a sus necesidades

Crear scripts C# y aplicarlos a objetos

Utilizar Visual Studio y algunas de sus funciones básicas

Añadir componentes Rigidbody y Collider para permitir que los objetos choquen de forma realista

Declarar variables apropiadamente e inicializarlas/asignarlas con modificadores de acceso apropiados.

Lección 1.1

Plan para Educadores

Al final de esta lección, usted podrá hacer lo siguiente:

Crear un nuevo proyecto a través de Unity Hub

Navegar cómodamente por el espacio 3D y el Editor de Unity.

Añadir y manipular objetos en la escena para colocarlos donde quieras

Colocar una cámara en un lugar ideal para su juego

Controlar el diseño del editor de Unity para que se adapte a sus necesidades

Lección 1.2

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Crear scripts C# y aplicarlos a objetos

Utilizar Visual Studio y algunas de sus funciones básicas

Escribir comentarios para que su código sea más legible

Utilizar métodos y clases fundamentales de C# como transform.Translate y Vector3

Agregar componentes de Rigidbody y Collider para permitir que los objetos choquen de forma realista.

Duplicar objetos en la jerarquía para poblar la escena

Lección 1.3

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Declarar variables correctamente y entender que las variables pueden ser de

diferentes tipos (float, Vector3, GameObject)

Inicializar/asignar variables a través del código o a través del inspector para establecerlas con valores apropiados.

Utilizar modificadores de acceso apropiados (públicos/privados) para sus variables con el fin de hacerlas más fáciles de cambiar en el inspector.

Lección 1.4

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Obtener la entrada del usuario con Input.GetAxis, lo que permite que el jugador se mueva de diferentes maneras

Usar la función Rotate para girar un objeto alrededor de un eje

Limpiar y organizar la jerarquía con objetos vacíos

Desafío 1

Plan para Educadores

En este desafío, usted reforzará las siguientes habilidades/conceptos:

Usar la clase Vector3 para mover y rotar objetos a lo largo/alrededor de un eje

Usar Time.deltaTime en el método Update() para mover objetos apropiadamente

Mover y girar objetos en la vista de escena para colocarlos de la forma deseada

Asignar variables en el inspector e inicializarlas en código

Implementación de variables de entrada para controlar el movimiento/rotación de objetos basándose en la entrada del usuario.

Laboratorio 1

Plan para Educadores

Al final de este laboratorio, serás capaz de hacer lo siguiente:

Proponer una idea para un proyecto con un alcance adecuado a su tiempo y a los recursos disponibles.

Reflexionar sobre el concepto de un proyecto para comprender mejor sus necesidades.

Planificar los hitos de un proyecto con fechas de vencimiento para entender mejor el ciclo de producción y para ser más responsable.

Crear un simple boceto/guión gráfico para comunicar mejor sus ideas.

Resumen

Este cuestionario evaluará su conocimiento de las habilidades y conceptos aprendidos en la Unidad 1.

Objetivo

Esta prueba evaluará su capacidad para:

Navegar cómodamente por el espacio 3D y el Editor de Unity.

Agregar y manipular objetos en la escena para colocarlos donde quieras, incluyendo la cámara.

Controlar el diseño del Editor de Unity para que se adapte a sus necesidades

Crear scripts C# y aplicarlos a objetos

Agregar componentes de cuerpo rígido y colisionadores para permitir que los objetos choquen de manera realista.

Declarar variables apropiadamente e inicializarlas/asignarlas con modificadores de acceso apropiados.

Unidad 2

Objetivo

Al final de esta unidad, usted será capaz de hacer lo siguiente:

Cree una sentencia if-then para implementar la lógica básica en su proyecto, incluyendo el uso de operadores mayor que (>) y menor que (<).

Transformar un objeto de juego en un prefabricado que puede ser instanciado en la escena.

Trabajar con los prefabricados para aumentar la eficiencia de sus flujos de trabajo

Obtener la entrada del usuario con GetKey y KeyCode para probar las pulsaciones específicas del teclado.

Usar matrices para crear una lista accesible de objetos o valores y seleccionar aleatoriamente un objeto de esa matriz.

Generar valores aleatoriamente para aleatorizar las posiciones de aparición (spawn)

Cambiar la perspectiva de la cámara para que se adapte mejor a tu juego

Repetir funciones en un temporizador con InvokeRepeating

Escribir métodos personalizados para que su código sea más legible

Detectar colisiones y destruir objetos que colisionan entre sí
Mostrar mensajes en la consola con Debug Log

Lección 2.1

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Ajustar la escala de un objeto proporcionalmente para obtener el tamaño deseado.

Utilizar más cómodamente la función `GetInput` para utilizar la entrada del usuario para controlar un objeto.

Crear una sentencia `if-then` para implementar la lógica básica en su proyecto, incluyendo el uso de operadores mayor que (`>`) y menor que (`<`).

Usar comentarios y formato automático para que su código sea más limpio y legible para otros programadores.

Lección 2.2

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Transformar un objeto de juego en un prefabricado que se puede utilizar como plantilla.

Instanciar los prefabricados para generarlos en la escena

Sobrescribir los prefabricados para actualizar y guardar sus características

Obtener la información de entrada del usuario con `GetKey` y `KeyCode` para probar las pulsaciones específicas del teclado.

Aplicar componentes a múltiples objetos a la vez para trabajar de la forma más eficiente posible.

Lección 2.3

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Crear un objeto vacío con un script adjunto

Utilizar matrices para crear una lista accesible de objetos o valores

Usar variables enteras para determinar el índice de un array

Generar valores aleatoriamente con `Random.Range` para aleatorizar objetos en

matrices y posiciones de aparición.

Cambiar la perspectiva de la cámara para que se adapte mejor al juego

Lección 2.4

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Repetir funciones temporizadas con `InvokeRepeating`

Escribir funciones personalizadas para que su código sea más legible

Editar los Box Colliders para que se ajusten a sus objetos correctamente

Detectar colisiones y destruir objetos que colisionan entre sí

Mostrar mensajes en la consola con Debug Log

Desafío 2

Plan para Educadores

En este desafío, usted reforzará las siguientes habilidades/conceptos:

Asignación de variables y matrices en el inspector

Edición de colisionadores al tamaño apropiado

Probar posiciones xyz con operadores mayor/menos que en sentencias if-else

Generación aleatoria de valores y selección de objetos de matrices

Laboratorio 2

Plan para Educadores

Al final de este laboratorio, serás capaz de hacer esto::

Crear un plano simple como fondo para su proyecto

Colocar la cámara, el fondo y el jugador de forma adecuada en función del tipo de proyecto que se está creando.

Crea formas primitivas que sirvan como comodines para tus objetos de juego.

Crear nuevos materiales de colores y aplicarlos para distinguir objetos de juego

Resumen

Este cuestionario evaluará su conocimiento de las habilidades y conceptos aprendidos en la Unidad 2.

Objetivo

Esta prueba evaluará su capacidad para:

Crear una sentencia if-then para implementar la lógica básica en su proyecto, incluyendo el uso de operadores mayor que (>) y menor que (<).

Transformar un objeto de juego en un prefabricado que pueda ser instanciado en la escena.

Trabajar con los prefabricados para aumentar la eficiencia de sus flujos de trabajo

Obtener la información de entrada del usuario con GetKey y KeyCode para probar las pulsaciones específicas del teclado.

Usar matrices para crear una lista accesible de objetos o valores y seleccionar aleatoriamente un objeto de esa matriz.

Generar valores aleatoriamente para aleatorizar las posiciones de aparición

Cambiar la perspectiva de la cámara para que se adapte mejor a tu juego

Repetir funciones en un temporizador con InvokeRepeating

Escribir métodos personalizados para que su código sea más legible

Detectar colisiones y destruir objetos que colisionan entre sí

Mostrar mensajes en la consola con Debug Log

Unidad 3

Resumen

En esta unidad, programarás un juego de desplazamiento lateral sin fin y de gran velocidad en el que el jugador necesita saltar para superar los obstáculos y evitar choques. Al crear este prototipo, aprenderás a añadir música y efectos de sonido, transformando completamente la experiencia de tus proyectos. También aprenderá a crear fondos dinámicos e interminables que se repiten, que son críticos para cualquier juego de desplazamiento lateral. Finalmente, aprenderás a incorporar efectos de partículas como salpicaduras y explosiones, que hacen que tus juegos sean mucho más satisfactorios.

Objetivo

Al final de esta unidad, usted será capaz de hacer esto::

Usar GameObject. Find y GetComponent para manipular los componentes de los objetos del objeto de juego actual o de otros objetos del juego.

Ajustar la gravedad de su proyecto con Physics.gravity y utilizar ForceModes para aplicar fuerzas de diferentes maneras

Utilizar nuevos operadores como "&&" y variables booleanas para controlar mejor la lógica del juego.

Congelar o restringir el componente Rigidbody para detener el movimiento en

determinados ejes.

Usar etiquetas para etiquetar los objetos del juego y llamarlos en el código

Utilizar la comunicación de scripts para acceder a los métodos y variables de otros scripts

Administrar estados de animación en el Controlador del Animador, incluyendo el ajuste de los parámetros de los estados y el estado por defecto.

Utilizar los métodos SetTrigger, SetBool y SetInt para activar transiciones entre estados de animación.

Detener y reproducir efectos de partículas para que se correspondan con los estados de animación de los personajes

Trabajar con fuentes de audio y con oyentes para reproducir música de fondo

Añadir efectos de sonido para pulir el proyecto

Lección 3.1

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Usar GetComponent para manipular los componentes de GameObjects

Influenciar la física de los objetos del juego con ForceMode.Impulse

Ajustar la gravedad de su proyecto con Physics.gravity

Utilizar nuevos operadores y variables como &&&.

Usar variables booleanas para controlar el número de veces que se puede hacer algo

Congelar o restringir el componente Rigidbody para detener el movimiento en determinados ejes.

Lección 3.2

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Usar etiquetas para etiquetar los objetos del juego y llamarlos en el código

Utilizar la comunicación de scripts para acceder a los métodos y variables de otros scripts

Lección 3.3

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Gestionar estados básicos de animación en el Animator Controller

Ajustar la velocidad de las animaciones para que se adapten al personaje o al juego

Establecer una animación predeterminada y activar otras con `anim.SetTrigger`

Establecer un estado permanente para "Game Over" con `anim.SetBool`

Lección 3.4

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Adjuntar efectos de partículas como si fueran hijos a objetos del juego

Detener y reproducir efectos de partículas para que se correspondan con los estados de animación de los personajes

Trabajar con fuentes de audio y oyentes para reproducir música de fondo

Añadir efectos de sonido para pulir el proyecto

Desafío 3

Plan para Educadores

En este desafío, usted reforzará las siguientes habilidades/conceptos:

Declarar e inicializar variables con el método `GetComponent`

Uso de booleans para desencadenar estados de juego

Visualización de los efectos de partículas en un lugar determinado en relación con un objeto de juego

Desplazamiento sin problemas de un fondo que se repite

Laboratorio 3

Plan para Educadores

Al final de este laboratorio, serás capaz de hacer esto::

Programar el tipo de movimiento del jugador que desea en función de la entrada del usuario

Restringir el movimiento de los jugadores de la manera apropiada, dependiendo de las necesidades del proyecto.

Solucionar problemas y encontrar soluciones relacionadas con el movimiento de los jugadores

Cuestionario 3

Resumen

Este cuestionario evaluará su conocimiento de las habilidades y conceptos aprendidos en la Unidad 3.

Objetivo

Esta prueba evaluará su capacidad para:

Ajustar la gravedad de su proyecto con `Physics.gravity` y utilizar `ForceModes` para aplicar fuerzas de diferentes maneras

Utilizar nuevos operadores como `&&` y variables de `bool` para controlar mejor la lógica del juego.

Congelar o restringir el componente `Rigidbody` para detener el movimiento en determinados ejes.

Usar etiquetas para etiquetar los objetos del juego y llamarlos en el código

Utilizar la comunicación de scripts para acceder a los métodos y variables de otros scripts

Administrar estados de animación en el Controlador del Animador, incluyendo el ajuste de los parámetros de los estados y el estado por defecto.

Usar los métodos `SetTrigger`, `SetBool` y `SetInt` para activar transiciones entre estados de animación.

Detener y reproducir efectos de partículas para que se correspondan con los estados de animación de los personajes

Trabajar con fuentes de audio y oyentes para reproducir música de fondo

Añadir efectos de sonido para pulir el proyecto

Unidad 4

Resumen

En esta Unidad, programarás una batalla de Sumo estilo arcade con el objetivo de derribar olas cada vez más difíciles de enemigos de una isla flotante, usando potenciadores para ayudar a derrotarlos. Al crear este prototipo, aprenderás a implementar nuevas mecánicas de juego en tus proyectos, que son nuevas reglas o sistemas que hacen que el juego sea más interesante. Por un lado, aprenderás a programar un powerup, que le da al jugador una ventaja temporal. Por otro lado, aprenderás a programar olas enemigas cada vez más difíciles, lo que hace que la supervivencia sea más difícil para el jugador. Un buen equilibrio entre los

potenciadores y la dificultad creciente hacen que la experiencia de juego sea mucho más interesante.

Objetivo

Al final de esta unidad, usted será capaz de hacer esto::

Aplicar Materiales de Física para hacer que los objetos del juego reboten más o menos.

Calcular nuevos vectores para dirigir objetos en direcciones personalizadas

Escribir funciones y variables personalizadas más avanzadas para que su código sea limpio y profesional

Escribir mensajes informativos de depuración con concatenación.

Usar IEnumerator y Coroutines para repetir y retardar funciones

Uso de bucles for para ejecutar código de forma eficiente y dinámica varias veces

Usar SetActive para hacer que los objetos del juego aparezcan y desaparezcan de la escena

Use FindObjectsOfType para rastrear el número actual de objetos en la escena

Lección 4.1

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Aplicar Texturas a los objetos

Fijar una cámara a su punto focal utilizando las relaciones padre-hijo

Transformar objetos basados en valores XYZ locales

Lección 4.2

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Aplicar Materiales de Física para hacer que los objetos del juego reboten

Normalizar vectores para apuntar al enemigo en la dirección del jugador

Generar aleatoriamente con Random.Range en dos ejes

Escribir funciones y variables personalizadas más avanzadas para que su código sea limpio y profesional

Lección 4.3

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Escribir mensajes informativos de depuración con Concatenación y variables

Repetir funciones con la potencia de IEnumerator y Coroutines

Usar SetActive para hacer que los objetos del juego aparezcan y desaparezcan de la escena

Lección 4.4

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Repetir funciones con bucles For

Incrementar valores enteros en un bucle con el operador ++

Encontrar objetos en una escena con FindObjectsOfType

Devolver la longitud de un array como un entero con .Length

Desafío 4

Plan para Educadores

En este desafío, usted reforzará las siguientes habilidades/conceptos:

Definición de vectores mediante la sustracción de una ubicación en el espacio 3D de otra

Rastrear el número de objetos de un determinado tipo en una escena para desencadenar determinados eventos.

Uso de las Coroutines para realizar acciones basadas en un intervalo de tiempo

Usar bucles for y variables dinámicas para ejecutar código un número determinado de veces

Resolución de errores relacionados con referencias nulas de variables no asignadas

Laboratorio 4

Plan para Educadores

Al final de este laboratorio, serás capaz de hacer esto::

Programar más cómodamente el movimiento básico

Manejar más cómodamente las colisiones de objetos

Engendrar más cómodamente prefabricados de objetos en intervalos de tiempo

Unidad 5

Resumen

En esta Unidad, programarás un juego para probar los reflejos del jugador, donde el objetivo es hacer clic y destruir objetos lanzados al aire al azar antes de que se salgan de la pantalla. Al crear este prototipo, aprenderás a implementar una Interfaz de Usuario – o UI – en tus proyectos. Añadirás una pantalla de título con un menú de selección de dificultad que controlará el nivel de dificultad del juego, añadirás una pantalla de puntuación que registrará cuántos puntos ha ganado el jugador, y añadirás una pantalla de "Game Over", que permitirá al jugador reiniciar e intentarlo de nuevo. Al aprender estas habilidades, seras capaz de crear una experiencia completamente "jugable" que el usuario puede disfrutar de principio a fin sin tener que reiniciar la aplicación para intentarlo de nuevo.

Objetivo

Al final de esta unidad, usted será capaz de hacer esto::

Crear un objeto de Game Manager que controle los estados del juego

Detectar dónde el usuario ha hecho clic con el ratón para crear un programa basado en clics.

Utilice el Canvas para crear elementos de la interfaz de usuario como una pantalla de Título, Botones o marcadores.

Bloquear elementos y objetos en su lugar en la interfaz de usuario con anclas

Usar variables y comunicación de script para actualizar elementos en la interfaz de usuario

Hacer que los elementos de la interfaz aparezcan y desaparezcan con .SetActive

Utilizar la comunicación de scripts y los estados de juego para implementar la pantalla funcional "Game Over".

Reiniciar el juego usando un botón de interfaz de usuario y la clase SceneManager.

Agregar oyentes para detectar cuando se ha hecho clic en un botón de la interfaz de usuario y activar la funcionalidad.

Establecer la dificultad del juego desde la pantalla de título pasando parámetros entre scripts

Lección 5.1

Resumen

Sinopsis:

¡Es hora de la última unidad! Empezaremos por crear un nuevo proyecto e importar los archivos de inicio, y luego cambiaremos la vista del juego a 2D. A continuación haremos una lista de objetos de blanco para que el jugador haga clic en ellos: Tres objetos "buenos" y uno "malo". Los objetivos se lanzarán girando en el aire después de reproducirse en una posición aleatoria en la parte inferior del mapa. Por último, permitiremos que el jugador los destruya con un clic!

Resultado del proyecto:

Una lista de tres objetos de objetivo buenos y un objeto de objetivo malo se generará en una posición aleatoria en la parte inferior de la pantalla, empujándose en el aire con fuerza y torque aleatorios. Estos objetivos serán destruidos cuando el jugador haga clic en ellos o se salgan de los límites.

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Cambiar el juego a la vista 2D para una perspectiva diferente

Agregar torque a la fuerza de un objeto

Crear un objeto de Game Manager que controle los estados del juego así como el engendramiento.

Crea una lista de objetos y devolver su longitud con Count

Usar While Loops para repetir código mientras algo es verdadero

Utilice OnMouseDown para permitir que el jugador haga clic en las cosas

Lección 5.2

Resumen

Sinopsis:

Los objetos entran volando en la escena y el jugador puede hacer clic para destruirlos, pero no pasa nada. En esta lección, mostraremos una puntuación en la interfaz de usuario que rastrea y muestra los puntos del jugador. Le daremos a cada objeto de blanco un valor de punto diferente, sumando o restando puntos al hacer clic. Por último, añadiremos explosiones geniales cuando cada objetivo sea destruido.

Resultado del proyecto:

Se mostrará una sección "Score:" en la interfaz de usuario, comenzando por cero. Cuando el jugador hace clic en un objetivo, la puntuación se actualizará y las partículas explotarán a medida que el objetivo sea destruido. Cada objetivo "bueno" añade un valor de punto diferente a la puntuación, mientras que el objetivo "malo" resta de la puntuación.

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Crear elementos de interfaz en el canvas

Bloquear elementos y objetos en su lugar con Anclas

Usar variables y comunicación de script para actualizar elementos en la interfaz de usuario

Lección 5.3

Resumen

Sinopsis:

Hemos añadido un excelente contador de puntuaciones al juego, pero hay muchos otros elementos de la interfaz de usuario que podrían añadirse. En esta lección, crearemos un texto de "Game Over" que se muestra cuando un objetivo "bueno" cae por debajo del sensor. Durante el final de la partida, los objetivos dejarán de reproducirse y la puntuación se reiniciará. Por último, añadiremos un botón "Restart Game" que permite al jugador reiniciar la partida después de haber perdido.

Resultado del proyecto:

Cuando un objetivo "bueno" cae por debajo del sensor en la parte inferior de la pantalla, los objetivos dejarán de reproducirse y aparecerá un mensaje de "Game Over" en toda la pantalla. Justo debajo del mensaje "Game Over" aparecerá un botón "Reset Game" que reiniciará el juego y restablecerá el marcador, para que el jugador pueda disfrutar de él de nuevo.

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Hacer que los elementos de la interfaz aparezcan y desaparezcan con `.SetActive`

Usar los estados de juego y la comunicación de scripts para tener una pantalla funcionando de "Game Over".

Reiniciar el juego usando un botón de la interfaz de usuario y el administrador de escenas.

Lección 5.4

Resumen

Sinopsis:

¡Es hora de la lección final! Para terminar nuestro juego, añadiremos una suerte de pantalla de menú y título. Creará su propio título y estilizará el texto para que luzca bien. Creará tres nuevos botones que establecerán la dificultad del juego. Cuanto mayor sea la dificultad, más rápido se reproducirán los objetivos!

Resultado del proyecto:

Comenzando el juego se abrirá un hermoso menú, con el título en un lugar destacado y tres botones de dificultad en la parte inferior de la pantalla. Cada dificultad afectará a la tasa de reproducción de los objetivos, aumentando la habilidad necesaria para evitar que los objetivos "buenos" caigan.

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Almacenar elementos de interfaz en un objeto padre para crear menús, interfaz de usuario o HUD

Agregar oyentes para detectar cuando se ha hecho clic en un botón de la interfaz de usuario

Establecer la dificultad mediante el paso de parámetros en las funciones del juego como SpawnRate

Desafío 5

Sumario

Resumen del desafío:

Pon a prueba tus habilidades con la Interfaz de Usuario con este desafío similar a whack-a-mole en el que tienes que conseguir toda la comida que aparece en una cuadrícula mientras evitas los cráneos. Tendrás que depurar los botones, los clics del ratón, el seguimiento de la puntuación, el reinicio de las secuencias y la configuración de la dificultad para llegar al fondo de este problema.

Resultado del desafío:

Todos los botones se ven bien con su texto correctamente alineado

Cuando se selecciona una dificultad, la tasa de reproducción cambia en consecuencia

Al hacer clic en un alimento, éste se destruye y la puntuación se actualiza en la parte superior izquierda.

Cuando pierdes el juego, aparece un botón de reinicio que te permite volver a jugar.

Plan para Educadores

En este desafío, usted reforzará las siguientes habilidades/conceptos:

Trabajar con objetos de texto y de botones para que tengan el aspecto que usted desea

Uso adecuado de los diversos métodos relacionados con el ratón de Unity

Visualización correcta de variables en objetos de texto mediante concatenación

Activación y desactivación de objetos en función de los estados del juego

Pasar información entre scripts usando métodos y parámetros personalizados

Laboratorio 5

Resumen

Sinopsis:

En este laboratorio, finalmente reemplazarás esos aburridos objetos primitivos con otros bellos y dinámicos. Podrás utilizar los recursos de la biblioteca del curso o buscar en la tienda de recursos otros completamente nuevos para darle a tu juego exactamente el aspecto que desees. Luego, pasarás por el proceso de intercambiar esos nuevos activos en lugar de los primitivos de los marcadores de posición. Al final de este laboratorio, tu proyecto se verá mucho mejor.

Resultado del proyecto:

Todos los objetos primitivos son reemplazados por modelos 3D reales, conservando la misma funcionalidad básica del juego.

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente:

Navegar por el Tienda de activos para encontrar los activos perfectos para el proyecto.

Usar Prefabricados Anidados para intercambiar objetos de marcadores de posición con activos reales

Ajustar la configuración del material para obtener la resolución y el aspecto que desea

Cuestionario 5

Resumen

Esta prueba evaluará su conocimiento de las habilidades y conceptos aprendidos en la Unidad 5.

Objetivo

Esta prueba evaluará su capacidad para:

Crear un objeto de Game Manager que controle los estados del juego

Detectar dónde el usuario ha hecho clic con el ratón para crear un programa basado en clics.

Utilizar el lienzo para crear elementos de la interfaz de usuario como un título, botones o pantalla de puntuación.

Bloquear elementos y objetos en su lugar en la interfaz de usuario con anclas

Usar variables y comunicación de script para actualizar elementos en la interfaz de usuario

Hacer que los elementos de la interfaz aparezcan y desaparezcan con .SetActive

Utilizar la comunicación de scripts y los estados de juego para implementar la pantalla operativa "Game Over".

Reiniciar el juego usando un botón de la interfaz de usuario y la clase

SceneManager

Agregar oyentes para detectar cuando se ha hecho clic en un botón de la interfaz de usuario y activar la funcionalidad.

Establecer la dificultad del juego desde la pantalla de título pasando parámetros entre scripts

Lección 6.1

Resumen

Sinopsis:

En esta lección, aprenderá sobre una variedad de técnicas diferentes para optimizar sus proyectos y hacerlos más eficientes. Puede que no note una gran diferencia en estos pequeños proyectos de prototipos, pero cuando exporte un proyecto más grande, especialmente uno para móviles o web, cada pequeña mejora en el rendimiento es crítica.

Resultado del proyecto:

Varios de sus proyectos de prototipos tendrán una mejor optimización, sirviéndole de ejemplo para implementar en sus proyectos personales.

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Reconocer y utilizar nuevos atributos de variables para mantener los valores privados, pero aún así editables en el inspector.

Usar las Funciones de Evento de Unity apropiadas (por ejemplo, Update vs. FixedUpdate vs. LateUpdate) para hacer que su proyecto se ejecute de la manera más fluida posible.

Comprender el concepto de Object Pooling, y apreciar cuando puede ser utilizado para optimizar su proyecto.

Lección 6.2

Resumen

Sinopsis:

En esta lección, usted intentará agregar un velocímetro y una pantalla de RPM para su vehículo en el Prototipo 1. Al hacerlo, aprenderá el proceso de investigación en línea cuando intente implementar nuevas funciones y solucionar problemas en sus proyectos. Como verá, añadir una nueva característica rara vez es tan simple como parece inicialmente – inevitablemente se encuentra con complicaciones inesperadas y errores que normalmente requieren un poco de investigación en línea. En esta lección, aprenderá cómo hacerlo para que pueda hacerlo con sus propios proyectos.

Resultado del proyecto:

Al final de esta lección, el vehículo se comportará con una física más realista, y habrá un velocímetro y una pantalla de revolución por minuto (RPM)..,

Plan para Educadores

Al final de esta lección, podrá hacer lo siguiente::

Usar los foros de Unity, las respuestas de Unity y la documentación de Unity

Scripting en línea para implementar nuevas funciones y solucionar problemas con sus proyectos.