Trayecto Programador júnior

# 

# Misión 1: ¿Cómo crear con código 1?



# Descripción del trayecto

Diseñado para cualquier persona interesada en aprender a programar u obtener un rol de nivel inicial en Unity, este curso incluye un conocimiento básico de Unity y no tiene requisitos previos de matemáticas. Programador júnior te prepara para obtener un certificado de Unity para que puedas demostrar tu capacidad de trabajo a los empleadores.

**Detalles clave**

* Esta misión te tomará aproximadamente 14 horas para completarla. Hazla a tu propio ritmo. Recibirás experiencia a cada paso durante tu camino.
* Conecta con la comunidad de Unity mientras aprendes y consultas el calendario Learn Live para seguir sesiones prácticas con creadores consolidados de Unity.
* Cuando hayas finalizado la misión, obtendrás una insignia para lucir en tu perfil y tu portafolio.

**Habilidades abordadas en este curso**

Comprensión del código básico

* ¿Cómo interpretar código simple?
* Mejora el código simple al utilizar las características de un IDE.

Creación de *scripts* de aplicación básica

* ¿Cómo utilizar estructuras lógicas comunes para controlar la ejecución del código?
* ¿Cómo escribir código que utilice las distintas Interfaz de programación de aplicaciones (API) de Unity?
* ¿Cómo implementar tipos de datos adecuados?
* ¿Cómo escribir un código que se integre en un sistema existente?
* Implementa un estilo de código que sea eficiente y fácil de leer
* Nuevos conceptos de prototipos:

Depuración básica

* Diagnostica y corrige código que compila, pero no logra rendir según lo esperado.
* Diagnostica y corrige errores comunes de compilación.
* Diagnostica y corrige errores de compilación relacionados con la Interfaz de programación de aplicaciones (API) de creación de *scripts* de Unity
* Diagnostica y corrige la causa de una excepción

Creación de *scripts* de aplicación para principiantes

* ¿Cómo crear el flujo de la Escena en el estado de una aplicación?
* ¿Cómo implementar la continuación de datos entre escenas y sesiones de usuario?
* Nivel 1: Control de versiones
* ¿Cómo mantener un proyecto al implementar correctamente el control de versiones?
* ¿Cómo implementar las prácticas recomendadas del control de versiones con Unity Collaborate?

Optimización de código básico

* Maximiza la eficiencia del código al ejecutar de manera correcta las prácticas recomendadas de la creación de código
* Depura problemas de rendimiento

Teoría de la programación para principiantes

* Analiza los pilares fundamentales de la programación orientada a objetos
* Simplifica el código y hazlo reutilizable al implementar correctamente los principios de herencia y polimorfismo
* Aumenta la seguridad del código y su usabilidad al implementar de forma correcta los principios de abstracción y encapsulación, incluido el uso de interfaces
* Escribe código eficiente, organizado y comprensible al implementar de forma correcta los principios de la programación orientada a objetos

| **¿Cómo usar el trayecto?** | | |
| --- | --- | --- |
| El trayecto Aspectos básicos de Unity se divide en 3 «misiones», cada una con muchos tutoriales y evaluaciones. Las misiones siguientes forman todo el trayecto: | | |
|  | **Programador júnior: ¿Cómo crear con código 1?** | 13 horas 45 minutos |
|  | **Programador júnior: ¿Cómo crear con código 2?** | 24 horas 15 minutos |
|  | **Programador júnior: ¿Cómo administrar el flujo y los datos de la Escena?** | 2 horas |
|  | **Programador júnior: ¿Cómo aplicar los principios orientados a objetos?** | 1 hora 45 minutos |
| Instamos a los estudiantes a completar todas las misiones en el orden correcto para garantizar una mejor experiencia de aprendizaje. | | |

# 

# 

# 

# 

# 

# 

# 

# Índice

[Descripción del trayecto1](#_heading=h.1fob9te)

[Índice](#_heading=h.qsj6fbui9phm) [2](#_heading=h.qsj6fbui9phm)

[Misión 1: ¿Cómo crear con código 1?](#_heading=h.ipibja8mbovj) [3](#_heading=h.ipibja8mbovj)

[Primeros pasos](#_heading=h.3x0flmod2ua) [3](#_heading=h.3x0flmod2ua)

[Introducción al curso](#_heading=h.kxqari1xc7f6) [4](#_heading=h.kxqari1xc7f6)

[¿Cómo instalar el software de Unity?](#_heading=h.le6dcay2xqt4) [4](#_heading=h.le6dcay2xqt4)

[Unidad 1: Control del jugador](#_heading=h.3yy6zwxscdu9) [5](#_heading=h.3yy6zwxscdu9)

[Unidad 1: Introducción](#_heading=h.wkiy08xp8yj9)  [6](#_heading=h.wkiy08xp8yj9)

[Lección 1.1: Inicia los motores en 3D](#_heading=h.iqp8p0xnqhoe) [6](#_heading=h.iqp8p0xnqhoe)

[Lección 1.2: Del pedal al metal](#_heading=h.gfxmcbli974s) [7](#_heading=h.gfxmcbli974s)

[Lección 1.3: Persecución a alta velocidad](#_heading=h.z6sli7d92rsm) [8](#_heading=h.z6sli7d92rsm)

[Lección 1.4: Al volante](#_heading=h.u96thgw8v1w) [9](#_heading=h.u96thgw8v1w)

[Desafío 1: ¿Cómo programar un avión?](#_heading=h.um4i052yha0k) [10](#_heading=h.um4i052yha0k)

[Laboratorio 1: Documento de diseño del proyecto](#_heading=h.39kcd3jq4rp3) [11](#_heading=h.39kcd3jq4rp3)

[Prueba 1](#_heading=h.ir5pxmkoj6pz) [11](#_heading=h.ir5pxmkoj6pz)

[Características adicionales 1: Comparte tu trabajo](#_heading=h.2i0wax34q9yz) [12](#_heading=h.2i0wax34q9yz)

[Introducción a la gestión de proyectos y al trabajo en equipo](#_heading=h.9kphnfwc5q5z) [13](#_heading=h.9kphnfwc5q5z)

[Unidad 2: Jugabilidad básica](#_heading=h.tbiu3rfd6ol1) [14](#_heading=h.tbiu3rfd6ol1)

[Unidad 2: Introducción](#_heading=h.gcgjrxbadqji) [14](#_heading=h.gcgjrxbadqji)

[Lección 2.1: Posición del jugador](#_heading=h.kh28ov3af01g) [15](#_heading=h.kh28ov3af01g)

[Lección 2.2: Comida voladora](#_heading=h.i95d84kkqt1k) [16](#_heading=h.i95d84kkqt1k)

[Lección 2.3: Estampida de animales aleatorios](#_heading=h.6qjscmc1az0b) [17](#_heading=h.6qjscmc1az0b)

[Lección 2.4: Decisiones de colisiones](#_heading=h.nxbtini9jrd4) [18](#_heading=h.nxbtini9jrd4)

[Desafío 2: Jugar a traer la pelota](#_heading=h.sm0ycgw5cv99) [19](#_heading=h.sm0ycgw5cv99)

[Laboratorio 2: Nuevo proyecto con elementos primitivos](#_heading=h.kvmv2jetvja6) [20](#_heading=h.kvmv2jetvja6)

[Prueba 2](#_heading=h.e62cuebjo417) [21](#_heading=h.e62cuebjo417)

[Características adicionales 2: Comparte tu trabajo](#_heading=h.swcxj4ku13no) [22](#_heading=h.swcxj4ku13no)

[Evaluación de la misión 122](#_heading=h.97i79jgfcodt)

# 

## Misión 1: ¿Cómo crear con código 01?



Parte del [trayecto Programador júnior](https://learn.unity.com/mission/programming-basics?pathwayId=5f7e17e1edbc2a5ec21a20af)

**Descripción general de la misión**

La primera misión del trayecto de Programador júnior te proporcionará la base necesaria para crear una amplia gama de experiencias digitales en Unity. Aprenderás los conceptos fundamentales de la programación, como las variables, las funciones y la lógica básica, mediante dos proyectos prácticos. También modificarás un *script* para personalizar una experiencia sencilla de Unity: el comienzo de tu portafolio personal.

**Datos clave**

* Esta misión te tomará aproximadamente 14 horas para completarla. Tómala a tu propio ritmo, obtendrás experiencia en cada paso del camino.
* Conecta con la comunidad de Unity mientras aprendes y consultas el calendario Learn Live para seguir sesiones prácticas con creadores consolidados de Unity.
* Cuando hayas finalizado la misión, obtendrás una insignia para lucir en tu perfil y tu portafolio.

## Primeros pasos

| **Enlace de la lección** | [Primeros pasos](https://learn.unity.com/project/getting-started?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0) | |
| --- | --- | --- |
| **Duración** | **1 hora 30 minutos** | |
| **Objetivo de la lección**  En esta **unidad** introductoria, se te presentará el curso, luego descargarás e instalarás el software de Unity.  Esta lección es parte del [trayecto Programador júnior](https://learn.unity.com/pathway/junior-programmer?uv=2020.3). | |  |

### 

## Introducción al curso

| **Enlace de la lección** | [Introducción al curso](https://learn.unity.com/tutorial/welcome-and-install-the-software?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5d0921d5edbc2a001fca4661) | |
| --- | --- | --- |
| **Duración** | **10 minutos** | |
| **Resumen**  En esta lección, se te presentará el curso, los objetivos, el contenido y cómo puedes sacarle el máximo provecho. | |  |
| **Pasos**   1. Te damos la bienvenida 2. Observa y luego aplica lo que aprendiste | | |

## ¿Cómo instalar el software de Unity?

| **Enlace de la lección** | [¿Cómo instalar el software de Unity?](https://learn.unity.com/tutorial/lesson-0-1-install-unity-software?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5d0921d5edbc2a001fca4661) | |
| --- | --- | --- |
| **Duración** | **25 minutos** | |
| **Resumen**  Si aún no tienes la versión recomendada de Unity instalada en tu computadora, lo primero que debes hacer antes de comenzar el curso es instalarla. Para ello tendrás que crear una cuenta de Unity, si aún no tienes una. Cuando instales el software, instalarás el Unity Hub, que te permite gestionar tus instalaciones y proyectos, el propio motor de Unity y Visual Studio, el entorno de desarrollo integrado (IDE) que utilizarás para programar en C#.  **Resultado del proyecto**  El Unity Hub, el Editor de Unity y Visual Studio se instalarán en tu computadora. | |  |
| **Pasos:**   1. Antes de comenzar 2. ¿Cómo descargar e instalar el Unity Hub? 3. ¿Cómo instalar una nueva versión de Unity? 4. ¿Cómo iniciar sesión o crear una nueva cuenta de Unity? 5. Resumen | | |
| **Nuevo avance**   * El Unity Hub, la versión correcta del Editor de Unity y Visual Studio instalados. * Sesión en tu cuenta de Unity iniciada   **Nuevos conceptos**   * El Unity Hub y sus características * Versiones del Editor, incluidas las versiones de LTS * Visual Studio * Cuenta de Unity | | |

## Unidad 1: Control del jugador

| **Enlace de la lección** | [Unidad 1: Control del jugador](https://learn.unity.com/project/unit-1-driving-simulation?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0) | |
| --- | --- | --- |
| **Duración** | **7 horas 30 minutos** | |
| **Resumen**  En esta unidad, programarás un automóvil que se mueve de lado a lado en una carretera flotante tratando de evitar (o golpear) los obstáculos en el camino. Además de familiarizarte con el Editor y el flujo de trabajo de Unity, aprenderás a crear nuevos Scripts de C# y a realizar un poco de programación sencilla. Al final de la unidad, podrás llamar a funciones básicas, y luego definir y ajustar nuevas variables para modificar los resultados de esas funciones.  **Habilidades**  Creación de *scripts* de aplicación básica   * ¿Cómo implementar tipos de datos adecuados? * Implementa un estilo de código que sea eficiente y fácil de leer * Nuevos conceptos de prototipos:   Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE.   Depuración básica   * Diagnostica y corrige código que compila, pero no logra rendir según lo esperado. | |  |

## Unidad 1: Introducción

| **Enlace de la lección** | [Unidad 1: Introducción](https://learn.unity.com/tutorial/unit-1-introduction?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5caccdfbedbc2a3cef0efe63#) | |
| --- | --- | --- |
| **Duración** | **5 minutos** | |
| Un video introductorio a la Unidad 1 en el que aprenderás a implementar el Player Control (Control del jugador). | |  |
| **Pasos**   1. Introducción | | |

## Lección 1.1: Inicia los motores en 3D

| **Enlace de la lección** | [Lección 1.1: Inicia los motores en 3D](https://learn.unity.com/tutorial/set-up-your-first-project-in-unity?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5caccdfbedbc2a3cef0efe63#) | |
| --- | --- | --- |
| **Duración** | **1 hora 20 minutos** | |
| **Resumen**  En esta lección, crearás tu primer proyecto de juego en el Unity Hub. Elegirás y colocarás un vehículo para que el jugador lo conduzca y un obstáculo que pueda evadir o chocar. Además, colocarás una cámara por la que verá el jugador, ofreciéndole una perspectiva perfecta de la Escena. A lo largo de este proceso, aprenderás a navegar dentro del Editor de Unity y a sentirte cómodo al moverte en un espacio en 3D. Por último, personalizarás tu propio diseño de ventanas para el Editor de Unity.  **Resultado del proyecto**  Tendrás un vehículo y un obstáculo colocados en la carretera, y una cámara posicionada a la perfección detrás del vehículo. También tendrás un nuevo Layout personalizado de Unity, perfectamente optimizado para la edición.  **Materiales**  [Prototipo 1: Archivos de inicio](https://connect-prd-cdn.unity.com/20210507/6c595e73-affa-42f7-ae53-9d7e33e7247a/Prototype%201%20-%20Starter%20Files.zip?_gl=1*k46cd9*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTczNzEyMi42Ny4xLjE2Mjk3Mzg5MTIuMjA.&_ga=2.38767264.242656276.1629722835-1660102106.1600431951) (.zip) | |  |
| **Pasos**   1. ¿Cómo crear una carpeta para el curso y un nuevo proyecto? 2. ¿Como importar los recursos y abrir Prototype 1? 3. ¿Cómo añadir tu vehículo a la Escena? 4. ¿Cómo añadir un obstáculo y reubicarlo? 5. ¿Cómo ubicar la cámara y ejecutar el juego? 6. ¿Cómo mover la cámara detrás del vehículo? 7. ¿Cómo personalizar el Layout de la interfaz? 8. Resumen de la lección | | |
| **Nueva funcionalidad**   * Configuración del proyecto con los recursos importados * Vehículo posicionado al inicio de la carretera * Obstáculo posicionado frente al vehículo * Cámara posicionada detrás del vehículo   **Nuevos conceptos y habilidades**   * Crear un proyecto nuevo * Importar recursos * Agregar objetos a la Escena * Juego versus vista de Escena * Ventanas de Proyecto, Hierarchy e Inspector * Navega en el espacio en 3D * Mover y girar herramientas * Personalizar el diseño | | |

## Lección 1.2: Del pedal al metal

| **Enlace de la lección** | [Lección 1.2: Del pedal al metal](https://learn.unity.com/tutorial/1-2-move-the-vehicle-with-your-first-line-of-c/?contentId=5f7229b2edbc2a001f834db7&missionId=5f71fe63edbc2a00200e9de0&pathwayId=5f7e17e1edbc2a5ec21a20af&projectId=5caccdfbedbc2a3cef0efe63&tab=skills&uv=2020.3) | |
| --- | --- | --- |
| **Duración** | **1 hora 10 minutos** | |
| **Resumen**  En esta lección haremos que tu simulador de conducción cobre vida. Primero escribirás tus primeras líneas de código en C#, cambiando la posición del vehículo y permitiéndole avanzar. A continuación, añadirás componentes de física a tus objetos para que puedan encontrarse entre sí. Por último, aprenderás a duplicar objetos en Hierarchy y a ubicarlos en la carretera.  **Resultado del proyecto**  Dispondrás de un vehículo en movimiento con su propio Script de C# y una carretera llena de obstáculos, donde todos podrán encontrarse entre sí mediante componentes de física.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE.   Creación de *scripts* de aplicación básica   * ¿Cómo implementar un estilo de código que sea eficiente y fácil de interpretar? | |  |
| **Pasos**   1. ¿Cómo crear y aplicar tu primer *script*? 2. ¿Cómo agregar un comentario en el método Update()? 3. ¿Cómo darle al vehículo movimiento hacia adelante? 4. ¿Cómo usar Vector3 para moverte hacia adelante? 5. ¿Cómo modificar la velocidad del vehículo? 6. ¿Cómo agregar un componente RigidBody a los objetos? 7. ¿Cómo duplicar y ubicar los obstáculos? 8. Resumen de la lección | | |
| **Nueva funcionalidad**   * El vehículo se mueve por la carretera a una velocidad constante. * Cuando el vehículo colisiona con los obstáculos, estos vuelan por los aires.   **Nuevos conceptos y habilidades**   * Scripts de C# * Inicio versus Update * Comentarios * Métodos * Parámetros de paso * Time.deltaTime * Operador Multiply (\*) * Componentes * Collider y RigidBody | | |

## Lección 1.3: Persecución a alta velocidad

| **Enlace de la lección** | [Lección 1.3: Persecución a alta velocidad](https://learn.unity.com/tutorial/1-3-make-the-camera-follow-the-vehicle-with-variables?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5caccdfbedbc2a3cef0efe63) | |
| --- | --- | --- |
| **Duración** | **50 minutos** | |
| **Resumen**  Mantén la vista en el camino. En esta lección vas a programar un nuevo Script de C# para tu cámara, que le permitirá seguir al vehículo por la carretera y ofrecer al jugador una vista adecuada de la Escena. Para hacer esto, debes utilizar un concepto muy importante en la programación: las variables.  **Resultado del proyecto**  La cámara seguirá al vehículo por la carretera en la Escena, que permitirá que el jugador sepa qué sucede.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejorar el código simple al utilizar las características de un IDE.   Creación de *scripts* de aplicación básica   * ¿Cómo implementar tipos de datos adecuados? * ¿Cómo implementar un estilo de código que sea eficiente y fácil de interpretar? | |  |
| **Pasos**   1. ¿Cómo agregar una variable de velocidad al vehículo? 2. ¿Cómo crear un nuevo *script* para la cámara? 3. ¿Cómo agregar un desplazamiento a la posición de la cámara? 4. ¿Cómo convertir el desplazamiento en una variable Vector3? 5. ¿Cómo suavizar la cámara con LateUpdate? 6. ¿Cómo editar el color de tinte del modo de juego? 7. Resumen de la lección   **Nueva funcionalidad**   * La cámara sigue al vehículo por la carretera a una determinada distancia de desplazamiento   **Nuevos conceptos y habilidades**   * Variables * Tipos de datos * Modificadores de acceso * ¿Cómo declarar e inicializar variables? * Siguiente lección | | |

### 

## Lección 1.4: Al volante

| **Enlace de la lección** | [Lección 1.4: Al volante](https://learn.unity.com/tutorial/lesson-1-4-use-user-input-to-control-the-vehicle?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5caccdfbedbc2a3cef0efe63) | |
| --- | --- | --- |
| **Duración** | **50 minutos** | |
| **Resumen**  En esta lección, tenemos que salir a la carretera y controlar el vehículo. Para ello, tenemos que detectar cuándo el jugador pulsa las teclas de dirección, y luego acelerar y girar el vehículo según esa entrada. Al emplear nuevos métodos, vectores y variables, harás que el vehículo avance o retroceda y gire de izquierda a derecha.  **Resultado del proyecto**  Cuando el jugador pulse las flechas arriba o abajo, el vehículo se moverá hacia delante y hacia atrás. Cuando el jugador pulse las teclas izquierda o derecha, el vehículo va a girar.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE. | |  |
| **Pasos**   1. ¿Cómo permitir que el vehículo se mueva hacia la izquierda o derecha? 2. ¿Cómo hacer que el jugador controle el movimiento de izquierda a derecha según lo que presione? 3. ¿Cómo tomar el control de la velocidad del vehículo? 4. ¿Cómo hacer que el vehículo rote en vez deslizarse? 5. ¿Cómo limpiar el código y la Hierarchy? 6. Resumen de la lección | | |
| **Nueva funcionalidad**   * Cuando el jugador pulse las flechas arriba o abajo, el vehículo se moverá hacia delante y hacia atrás. * Cuando el jugador pulse las teclas izquierda o derecha, el vehículo va a girar.   **Nuevos conceptos y habilidades**   * Objetos vacíos. * ¿Cómo conseguir la entrada de datos del usuario? * Desplazar frente a rotar | | |

## Desafío 1: ¿Cómo programar un avión?

| **Enlace de la lección** | [Desafío 1: ¿Cómo programar un avión?](https://learn.unity.com/tutorial/challenge-1-steer-a-plane-through-obstacles-in-the-sky/?contentId=5f7229b2edbc2a001f834db7&missionId=5f71fe63edbc2a00200e9de0&pathwayId=5f7e17e1edbc2a5ec21a20af&projectId=5caccdfbedbc2a3cef0efe63&tab=skills&uv=2020.3) | |
| --- | --- | --- |
| **Duración** | **50 minutos** | |
| **Descripción general del desafío**  Utiliza las habilidades que has aprendido en el simulador de conducción para pilotar un avión y esquivar obstáculos en el cielo. Para controlar el ángulo de inclinación del avión hacia arriba y hacia abajo, tendrás que utilizar las teclas de arriba y abajo. También tendrás que hacer que la cámara siga al avión para poder mantenerlo a la vista.  **Resultado del desafío**   * El avión avanza a una velocidad constante. * Las flechas arriba y abajo inclinan la punta del avión hacia arriba y hacia abajo. * La cámara sigue al avión mientras vuela. | |  |
| **Materiales**  [Desafío 1: Archivos de inicio](https://connect-prd-cdn.unity.com/20210506/913574fa-af65-4d95-abe8-d90282b27a83/Challenge%201%20-%20Starter%20Files.zip?_gl=1*4h0mzb*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgwNzI2Ni43MC4xLjE2Mjk4MDcyODMuNDM.&_ga=2.51349162.242656276.1629722835-1660102106.1600431951) (.zip) | | |
| **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE.   Depuración básica   * Diagnostica y corrige código que compila, pero no logra rendir según lo esperado. | | |
| **Pasos**   1. Descripción general del desafío 1 2. Advertencia 3. El avión se mueve hacia atrás. 4. El avión se mueve demasiado rápido. 5. El avión se inclina automáticamente 6. La cámara está frente al avión 7. La cámara no sigue al avión 8. Bono: la hélice del avión no gira | | |

## Laboratorio 1: Documento de diseño del proyecto

| **Enlace de la lección** | [Desafío 1: ¿Cómo programar un avión?](https://learn.unity.com/tutorial/challenge-1-steer-a-plane-through-obstacles-in-the-sky/?contentId=5f7229b2edbc2a001f834db7&missionId=5f71fe63edbc2a00200e9de0&pathwayId=5f7e17e1edbc2a5ec21a20af&projectId=5caccdfbedbc2a3cef0efe63&tab=skills&uv=2020.3) | |
| --- | --- | --- |
| **Duración** | **50 minutos** | |
| **Resumen**  En esta primera sesión de laboratorio, comenzarás el trabajo preliminar necesario para crear con éxito un proyecto personal en este curso. En primer lugar, aprenderás lo que es un proyecto personal, cuáles son sus objetivos y cuáles son sus posibles limitaciones. A continuación, te tomarás el tiempo necesario para elaborar una idea y la describirás detalladamente en tu documento de diseño, incluido un calendario sobre cuándo esperas completar ciertas características. Por último, te tomarás un tiempo para dibujar un boceto de tu proyecto que te ayude a visualizarlo y a compartir tu idea con los demás.  **Resultado del proyecto**  Se rellenará el documento de diseño, incluido el concepto, la escala de tiempo y un boceto preliminar del producto mínimo viable. | |  |
| **Materiales**  [Documento de diseño del proyecto](https://docs.google.com/document/d/1FR-GYr2hL67d6MleWTTP-mXfCHVZTM1Mko77MFodxFg/copy) (documento de Google)  [Documento de diseño del proyecto](https://connect-prd-cdn.unity.com/20200401/a0ca9f73-ec70-4024-bc42-256d14ada1ce/Project%20Design%20Doc%20%5BWORD%5D.docx?_gl=1*1k739pu*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgwNzI2Ni43MC4xLjE2Mjk4MDg5MTAuNTk.&_ga=2.17204794.242656276.1629722835-1660102106.1600431951) (documento de Word)  [Documento de diseño del proyecto](https://connect-prd-cdn.unity.com/20190524/19ad3c2b-506e-46c9-9700-07180536a9d2_Project_Design_Doc__PDF_.pdf?_gl=1*1k739pu*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgwNzI2Ni43MC4xLjE2Mjk4MDg5MTAuNTk.&_ga=2.17204794.242656276.1629722835-1660102106.1600431951) (PDF) | | |
| **Habilidades**  Creación de *scripts* de aplicación básica   * Nuevos conceptos de prototipos: | | |
| **Pasos**   1. ¿Qué es un proyecto personal? 2. ¿Cómo revisar ejemplos de documentos de diseño? 3. ¿Cómo completar el concepto de proyecto versión 1? 4. ¿Cómo completar el cronograma del proyecto? 5. ¿Cómo completar el boceto del PMV? 6. Resumen | | |

## Prueba 1

| **Enlace de la lección** | [Prueba 1](https://learn.unity.com/quiz/quiz-1-9s?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5caccdfbedbc2a3cef0efe63) | |
| --- | --- | --- |
| **Duración** | **15 minutos** | |
| **Resumen**  Esta prueba evaluará tu conocimiento de las habilidades y conceptos aprendidos en la Unidad 1.  **Objetivo de la prueba**  Esta prueba evaluará tu capacidad para realizar lo siguiente:   * Navegar cómodamente en el espacio en 3D y el Editor de Unity. * Añadir y manipular objetos en la Escena para colocarlos donde quieras, incluida la cámara. * Controlar el Layout del Editor de Unity para que se adapte a tus necesidades. * Crear Scripts de C# y aplicarlos a objetos. * Utilizar Visual Studio y algunas de sus funciones básicas. * Añadir componentes Rigidbody y Collider para permitir que los objetos se encuentren de forma realista. * Definir correctamente las variables e inicializarlas/asignarlas con los modificadores de acceso adecuados. | |  |
| **Preguntas**   1. ¿Qué ventana de Unity contiene una lista de todos los objetos del juego que hay en la Escena? 2. Verdadero o falso: Visual Studio no es parte de Unity. Podrías usar un editor de código diferente para editar tus Scripts de C# si lo deseas. 3. ¿Qué describe mejor la diferencia entre las siguientes imágenes, en las que el auto de la segunda imagen está más adelante de la carretera? Imagen 1 Imagen 2 4. ¿En qué orden se escriben las palabras cuando se declara una nueva variable? 5. ¿Cuál de las siguientes variables debería ser visible en el Inspector? 6. ¿Cuál es un valor posible para la variable horizontalInput? 7. ¿Qué hay de cierto en las siguientes dos líneas de código? 8. ¿Cuál de las siguientes líneas de código utiliza las convenciones de nomenclatura estándar de Unity? 9. ¿Cuál de los comentarios describe mejor el siguiente código? 10. La siguiente imagen muestra la ventana de preferencias que permite... | |  |

## Características adicionales 1: Comparte tu trabajo

| **Enlace de la lección** | [Características adicionales 1: Comparte tu trabajo](https://learn.unity.com/tutorial/bonus-features-1-share-your-work?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5caccdfbedbc2a3cef0efe63) | |
| --- | --- | --- |
| **Duración** | **1 hora** | |
| **Resumen**  En este tutorial, puedes ir más allá de lo que has aprendido en esta Unidad y compartir lo que has hecho con tus compañeros creadores.  Hay cuatro características de bonificación presentadas en este tutorial marcadas como Fácil, Medio, Difícil y Experto. Puedes intentar cualquier cosa, darle tu propio giro y compartir tu trabajo.  Este tutorial es completamente opcional, pero lo recomendamos mucho a quienes quieran llevar sus habilidades al siguiente nivel. | |  |
| **Pasos**   1. Descripción general 2. Fácil: pirámides de obstáculo 3. Medio: vehículos que se aproximan 4. Difícil: cambio de cámara 5. Experto: multijugador local 6. Orientación para la solución 7. ¿Cómo compartir tu trabajo? | | |

## Introducción a la gestión de proyectos y al trabajo en equipo

| **Enlace de la lección** | [Introducción a la gestión de proyectos y al trabajo en equipo](https://learn.unity.com/tutorial/introduction-to-project-management-and-teamwork?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0) | |
| --- | --- | --- |
| **Duración** | **20 minutos** | |
| **Resumen**  La mayoría de los trabajos en el mundo del diseño y el desarrollo requieren el trabajo en equipo y el uso de herramientas de gestión y planificación de proyectos para garantizar el éxito de los mismos. En este tutorial, aprenderás los conceptos clave y las mejores prácticas utilizadas para la planificación de proyectos, las fases de producción, la gestión de proyectos y el trabajo en equipo.  **Habilidades**  Gestión de proyectos básica   * Planificación de proyectos en el ciclo de desarrollo en tiempo real * Gestión de proyectos en el ciclo de desarrollo en tiempo real   **Materiales**   * [Ejemplo de una carta de proyecto](https://connect-prd-cdn.unity.com/20201117/bb422f61-9b42-4521-9828-910aae22e39c/Example%20of%20Project%20Charter.pdf?_gl=1*kofcjx*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgyMjA5OS43MS4xLjE2Mjk4MjQwNjEuMzk.&_ga=2.76522390.242656276.1629722835-1660102106.1600431951) (.pdf) * [Documento de diseño del proyecto](https://connect-prd-cdn.unity.com/20201117/99fd6c6b-0d44-42f5-9373-eb8a1887cacf/Project%20Design%20Doc%20TEMPLATE%20-%20%5BGDOC%5D.pdf?_gl=1*kofcjx*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgyMjA5OS43MS4xLjE2Mjk4MjQwNjEuMzk.&_ga=2.76522390.242656276.1629722835-1660102106.1600431951) (.pdf) | |  |
| **Pasos**   1. Descripción general 2. Resumen de las fases de producción 3. Resumen de la planificación de proyectos 4. Documentos de diseño y planes de proyecto 5. Gestión de proyectos y seguimiento del progreso 6. Ejercicio: identifica las herramientas para apoyar la gestión y el seguimiento del proyecto 7. ¿Cómo preparar la publicación? 8. Publicación del proyecto 9. Actividades operativas y retrospectivas 10. Resumen | | |

## Unidad 2: Jugabilidad básica

| **Enlace de la lección** | [Jugabilidad básica](https://learn.unity.com/project/unit-2-basic-gameplay) | |
| --- | --- | --- |
| **Duración** | **6 horas 20 minutos** | |
| **Resumen**  En esta unidad, programarás un juego de vista aérea con el objetivo de lanzar comida a los animales hambrientos (que vienen en estampida hacia ti) antes de que te esquiven. Para ello, te familiarizarás con algunos de los conceptos más importantes de la programación y de Unity, como las declaraciones condicionales, la generación de valores aleatorios, los arreglos, la detección de colisiones, los Prefabs y las posiciones de una instancia. Al completar esta unidad, aprenderás a programar un juego básico que permita lanzar proyectiles y mover al jugador para mantener el juego vivo.  **Esto es lo que aprenderás**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE   Creación de *scripts* de aplicación básica   * ¿Cómo utilizar estructuras lógicas comunes para controlar la ejecución del código? * ¿Cómo implementar los tipos de datos adecuados? * ¿Cómo escribir un código que se integre en un sistema existente? * ¿Cómo implementar un estilo de código que sea eficiente y fácil de interpretar? * Nuevos conceptos de prototipos:   Depuración básica   * Diagnostica y corrige código que compila, pero no logra rendir según lo esperado. | |  |

### 

## Unidad 2: Introducción

| **Enlace de la lección** | [Unidad 2: Introducción](https://learn.unity.com/tutorial/unit-2-introduction?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6) | |
| --- | --- | --- |
| **Duración** | **15 minutos** | |
| **Resumen**  Un video introductorio a la Unidad 2, donde aprenderás a implementar la jugabilidad básica.  Ver el video de introducción [aquí](https://learn.unity.com/tutorial/unit-2-introduction?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6#5d1ba822edbc2a002175788d). | |  |
| **Pasos**   1. Unidad 2: Introducción | |  |

### 

## Lección 2.1: Posición del jugador

| **Enlace de la lección** | [Lección 2.1: Posición del jugador](https://learn.unity.com/tutorial/lesson-2-1-control-player-bounds-with-if-then-statements/?contentId=5f7229b2edbc2a001f834db7&missionId=5f71fe63edbc2a00200e9de0&pathwayId=5f7e17e1edbc2a5ec21a20af&projectId=5cdcc312edbc2a24a41671e6&tab=overview&uv=2020.3) | |
| --- | --- | --- |
| **Duración** | **1 hora** | |
| **Resumen**  Al inicio de esta unidad, crearás un nuevo proyecto para tu segundo prototipo y harás funcionar el movimiento básico del jugador. Primero elegirás qué personaje quieres, con qué tipos de animales te gustaría interactuar y qué comida te gustaría darles. Le darás al jugador un movimiento básico de lado a lado como hiciste en el prototipo 1, pero luego usarás enunciados condicionales if-then para mantener al jugador dentro de los límites.  **Resultado del proyecto**  El jugador podrá moverse de izquierda a derecha en la pantalla según las pulsaciones de las teclas izquierda y derecha del usuario, pero no podrá salir de la zona de juego por ningún lado.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE   Creación de *scripts* de aplicación básica   * ¿Cómo utilizar estructuras lógicas comunes para controlar la ejecución del código?   **Materiales**  [Prototipo 2: Archivos de inicio](https://connect-prd-cdn.unity.com/20210507/bfd26de3-a68a-4a16-8cf6-8eacf2bb7f75/Prototype%202%20-%20Starter%20Files.zip?_gl=1*10cj57e*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgyOTM4NS43Mi4xLjE2Mjk4Mjk5MDAuNTk.&_ga=2.54428587.242656276.1629722835-1660102106.1600431951) (.zip) | |  |
| **Pasos**   1. ¿Cómo crear un proyecto nuevo para el prototipo 2? 2. ¿Cómo añadir al jugador, los animales y la comida? 3. ¿Cómo obtener la entrada horizontal del usuario? 4. ¿Cómo mover al jugador de izquierda a derecha? 5. ¿Cómo mantener al jugador dentro de los límites? 6. ¿Cómo limpiar el código y las variables? 7. Resumen de la lección | | |

### 

## Lección 2.2: Comida voladora

| **Enlace de la lección** | [Lección 2.2: Comida voladora](https://learn.unity.com/tutorial/lesson-2-2-launch-projectiles-by-instantiating-prefabs?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6) | |
| --- | --- | --- |
| **Duración** | **1 hora 10 minutos** | |
| **Resumen**  En esta lección, harás que el jugador pueda lanzar un proyectil a través de la Escena. Primero escribirás un nuevo *script* para lanzar el proyectil hacia delante. A continuación, guardarás el proyectil junto con todos sus *scripts* y propiedades mediante un nuevo e importante concepto en Unity llamado Prefabs. El jugador podrá lanzar un proyectil Prefab con un toque de la barra espaciadora. Por último, hay que añadir límites a la Escena, y eliminar cualquier objeto que salga de la pantalla.  **Resultado del proyecto**  El jugador podrá presionar la barra espaciadora y lanzar un proyectil Prefab a la Escena, el cual se destruirá cuando salga de los límites del juego. Los animales también se eliminarán de la Escena cuando abandonen los límites del juego.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE   Creación de *scripts* de aplicación básica   * ¿Cómo utilizar estructuras lógicas comunes para controlar la ejecución del código? | |  |
| **Pasos**   1. ¿Cómo hacer que el proyectil vuele hacia delante? 2. ¿Cómo convertir el proyectil en un Prefab? 3. ¿Cómo probar la pulsación de la barra espaciadora? 4. ¿Cómo lanzar un proyectil cuando se pulsa la barra espaciadora? 5. ¿Cómo convertir los animales en Prefabs? 6. ¿Cómo destruir los proyectiles fuera de la pantalla? 7. ¿Cómo destruir los animales fuera de la pantalla? 8. Resumen de la lección | | |
| **Nueva funcionalidad**   * El jugador puede desplazarse hacia la izquierda y la derecha con base en la pulsación de las teclas izquierda y derecha del usuario. * El jugador no podrá abandonar el área de juego por ninguno de los lados.   **Nuevos conceptos y habilidades**   * Ajustar el tamaño del objeto * Enunciados condicionales if * Operadores mayor que y menor que | | |

### 

## Lección 2.3: Estampida de animales aleatorios

| **Enlace de la lección** | [Lección 2.3: Estampida de animales aleatorios](https://learn.unity.com/tutorial/lesson-2-3-random-animal-stampede?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6) | |
| --- | --- | --- |
| **Duración** | **1 hora** | |
| **Resumen**  Nuestros animales Prefabs caminan por la pantalla y se destruyen al salir de los límites, pero no aparecen en el juego a menos que los arrastremos. En esta lección haremos que los animales aparezcan por sí solos, en un lugar aleatorio de la parte superior de la pantalla. Para ello, crearemos un nuevo objeto y un nuevo *script* para gestionar todo el proceso de generación.  **Resultado del proyecto**  Cuando el usuario pulse la tecla S, un animal elegido al azar aparecerá en una posición aleatoria en la parte superior de la pantalla, caminando hacia el jugador.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejorar el código simple al utilizar las características de un IDE.   Creación de *scripts* de aplicación básica   * ¿Cómo implementar los tipos de datos adecuados?   **Materiales**  [Plan de la lección 2.3: Estampida de animales aleatorios](https://connect-prd-cdn.unity.com/20190516/ebd9ef79-cece-4b54-be5d-8afa9f7fb147_Lesson_Plan_2.3___Random_Animal_Stampede.pdf?_gl=1*slu2us*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTgyOTM4NS43Mi4xLjE2Mjk4MzI4NjIuNTk.&_ga=2.109421830.242656276.1629722835-1660102106.1600431951)(.pdf) | |  |
| **Pasos**   1. ¿Cómo crear un administrador de creación? 2. ¿Cómo generar un animal cuando se pulsa la tecla S? 3. ¿Cómo generar animales aleatorios desde el arreglo? 4. ¿Cómo establecer al azar la ubicación de la generación? 5. ¿Cómo cambiar la perspectiva de la cámara? 6. Resumen de la lección | | |
| **Nueva funcionalidad**   * El jugador puede presionar la S para generar un animal. * La selección de animales y la ubicación de generación son aleatorias. * Seleccionar el modo de proyección de la cámara (perspectivo u ortográfico)   **Nuevos conceptos y habilidades**   * Administrador de creación * Arreglos * Códigos clave * Generación aleatoria * Variables locales frente a globales * Proyecciones perspectivas frente a isométricas | | |

### 

## Lección 2.4: Decisiones de colisiones

| **Enlace de la lección** | [Lección 2.4: Decisiones de colisiones](https://learn.unity.com/tutorial/lesson-2-4-collision-decisions/?contentId=5f7229b2edbc2a001f834db7&missionId=5f71fe63edbc2a00200e9de0&pathwayId=5f7e17e1edbc2a5ec21a20af&projectId=5cdcc312edbc2a24a41671e6&tab=overview&uv=2020.3) | |
| --- | --- | --- |
| **Duración** | **50 minutos** | |
| **Resumen**  Nuestro juego está avanzando muy bien, pero hay algunas cosas importantes que debemos añadir antes de que esté terminado. En primer lugar, en vez de presionar S para generar los animales, los generaremos con un temporizador para que aparezcan cada pocos segundos. A continuación, añadiremos colisionadores a todos nuestros Prefabs y haremos que lanzar un proyectil contra un animal lo destruya. Por último, mostraremos un mensaje de «Game Over» cada vez que algún animal consiga esquivar al jugador.  **Resultado del proyecto**  Los animales aparecerán en un intervalo de tiempo y caminarán por la pantalla, desencadenando un mensaje de «Game Over» si consiguen esquivar al jugador. Se destruirán si el jugador los golpea con un proyectil para alimentarlos.  **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE   Creación de *scripts* de aplicación básica   * ¿Cómo escribir un código que se integre con un sistema existente? * ¿Cómo implementar un estilo de código que sea eficiente y fácil de interpretar? | |  |
| **Pasos**   1. ¿Cómo hacer un nuevo método para generar animales? 2. ¿Cómo generar animales en intervalos de tiempo? 3. ¿Cómo agregar un componente Collider y Trigger? 4. ¿Cómo destruir objetos al colisionar? 5. ¿Cómo desencadenar un mensaje de «Game Over»? 6. Resumen de la lección | | |
| **Nueva funcionalidad**   * Los animales se generan en intervalos de tiempo y caminan por la pantalla * Cuando los animales consiguen rebasar al jugador, se muestra un mensaje de «Game Over». * Si un proyectil colisiona con un animal, ambos objetos se eliminan.   **Nuevos conceptos y habilidades**   * Crear métodos y funciones personalizados * Utilizar invokeRepeating() para repetir un código * Colliders y Triggers * Funciones de desactivación * Registrar mensajes de depuración en la consola | | |

### 

## Desafío 2: Jugar a traer la pelota

| **Enlace de la lección** | [Desafío 2: Jugar a traer la pelota](https://learn.unity.com/tutorial/challenge-2-play-fetch-with-random-values-and-arrays/?contentId=5f7229b2edbc2a001f834db7&missionId=5f71fe63edbc2a00200e9de0&pathwayId=5f7e17e1edbc2a5ec21a20af&projectId=5cdcc312edbc2a24a41671e6&tab=skills&uv=2020.3) | |
| --- | --- | --- |
| **Duración** | **1 hora** | |
| **Resumen del desafío**  Utiliza tus habilidades para generar números aleatorios y arreglos para programar este desafío en el que caen pelotas del cielo aleatoriamente y tienes que enviar a tu perro a atraparlas antes de que lleguen al suelo. Para completar este reto, tendrás que asegurarte de que tus variables se han asignado correctamente, tus enunciados condicionales if se han programado correctamente, tus colisiones se detectan perfectamente, y los objetos se generan de forma aleatoria.  **Resultado del desafío**   * Se genera una bola aleatoria (de 3) en una posición X aleatoria sobre la pantalla. * Cuando el usuario presiona la barra espaciadora, se genera un perro que corre para atrapar la pelota. * Si el perro colisiona con la pelota, esta se destruye. * Si la pelota toca el suelo, se muestra un mensaje de depuración «Game Over». * Los perros y las pelotas se eliminan de la Escena cuando salen de la pantalla.   **Habilidades**  Comprensión del código básico   * ¿Cómo interpretar código simple? * Mejora el código simple al utilizar las características de un IDE   Creación de *scripts* de aplicación básica   * ¿Cómo utilizar estructuras lógicas comunes para controlar la ejecución del código? * Depuración básica * Diagnostica y corrige código que compila, pero no logra rendir según lo esperado.   **Materiales**  [Desafío 2: Archivos de inicio](https://connect-prd-cdn.unity.com/20210506/b167f1c0-bbb1-4537-9b9b-c9e599ed16a0/Challenge%202%20-%20Starter%20Files.zip?_gl=1*6vrtvr*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYyOTg3MzY5OC43My4xLjE2Mjk4NzQ0OTYuNTA.&_ga=2.84911386.242656276.1629722835-1660102106.1600431951) (.zip) | |  |
| **Pasos**   1. Descripción general 2. Advertencia 3. Se generan perros en la parte superior de la pantalla 4. El jugador genera pelotas verdes en lugar de perros 5. Las pelotas se destruyen en cualquier lugar cerca del perro 6. No se está eliminando nada afuera de la pantalla 7. Solo se genera un tipo de pelota 8. Bono: el intervalo de generación siempre es el mismo 9. Bono: el jugador puede presionar varias veces la barra espaciadora | | |

### 

## Laboratorio 2: Nuevo proyecto con elementos primitivos

| **Enlace de la lección** | [Laboratorio 2: Nuevo proyecto con elementos primitivos](https://learn.unity.com/tutorial/lab-2-new-project-with-primitives?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6#) | |
| --- | --- | --- |
| **Duración** | **1 hora** | |
| **Resumen del desafío**  Crearás y configurarás un proyecto que pronto se transformará en tu propio proyecto personal. Por ahora, utilizarás formas primitivas (como esferas, cubos y planos) como marcadores de posición para tus objetos a fin de que puedas añadir funcionalidades de la manera más eficiente posible, sin que te quedes atascado con los gráficos. Para que quede claro qué objeto es cada uno, también le darás a cada uno un material de color único.  **Resultado del proyecto**   * Todos los objetos clave aparecerán en la Escena como objetos primitivos con la cámara posicionada adecuadamente para el tipo de proyecto. | |  |
| **Habilidades**  Creación de *scripts* de aplicación básica   * Nuevos conceptos de prototipos: | | |
| **Pasos**   1. ¿Cómo crear un nuevo proyecto y cambiar el nombre de la Escena? 2. ¿Cómo crear un plano de fondo? 3. ¿Cómo crear un jugador primitivo con un nuevo material? 4. ¿Cómo ubicar la cámara en función del tipo de proyecto? 5. Enemigos, obstáculos, proyectiles y materiales 6. ¿Cómo exportar un archivo de respaldo de Unity Package? 7. Resumen de la lección | | |

## Prueba 2

| **Enlace de la lección** | [Prueba 2](https://learn.unity.com/quiz/quiz-2-z8?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6) | |
| --- | --- | --- |
| **Duración** | **15 minutos** | |
| **Descripción general de la prueba**  Esta prueba evaluará tu conocimiento de las habilidades y conceptos aprendidos en la Unidad 2.  **Objetivo de la prueba**  Esta prueba evaluará tu capacidad para realizar lo siguiente:   * Crear un enunciado condicional if-then para aplicar la lógica básica en el proyecto, incluido el uso de los operadores mayor que (>) y menor que (<). * Transformar un GameObject en un Prefab que se pueda colocar en la Escena. * Trabajar con Prefabs para añadir eficiencia a los flujos de trabajo. * Obtener la entrada de datos del usuario con GetKey y KeyCode para comprobar las presiones específicas del teclado. * Utilizar arreglos para crear una lista accesible de objetos o valores, y seleccionar aleatoriamente un objeto de ese arreglo. * Generar valores al azar para asignar aleatoriamente las posiciones de generación. * Cambiar la perspectiva de la cámara para que se adapte mejor al juego. * Repetir funciones en un intervalo de tiempo con InvokeRepeating. * Escribir métodos personalizados para que el código sea más legible. * Detectar las colisiones y destruir los objetos que se encuentran entre sí. * Mostrar mensajes en la consola con Debug Log. | |  |

## Características adicionales 2: Comparte tu trabajo

| **Enlace de la lección** | [Características adicionales 2: Comparte tu trabajo](https://learn.unity.com/tutorial/bonus-features-2-share-your-work?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f71fe63edbc2a00200e9de0&projectId=5cdcc312edbc2a24a41671e6#) | |
| --- | --- | --- |
| **Duración** | **1 hora** | |
| **Resumen**  En este tutorial, puedes ir más allá de lo que has aprendido en esta Unidad y compartir lo que has hecho con tus compañeros creadores.  Hay cuatro características de bonificación presentadas en este tutorial marcadas como Fácil, Medio, Difícil y Experto. Puedes intentar cualquier cosa, darle tu propio giro y compartir tu trabajo.  Este tutorial es completamente opcional, pero lo recomendamos mucho a quienes quieran llevar sus habilidades al siguiente nivel. | |  |
| **Pasos**   1. Descripción general 2. Fácil: movimiento vertical del jugador 3. Medio: animales agresivos 4. Difícil: interfaz de usuario del juego 5. Experto: barra de hambre de los animales 6. Orientación para la solución 7. ¿Cómo compartir tu trabajo? | | |

## Evaluación de la misión 1

| **Prueba:** [¿Cómo administrar el flujo y los datos de la Escena?](https://learn.unity.com/quiz/quiz-manage-scene-flow-and-data?labelRequired=true&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | | |
| --- | --- | --- |
| **Entrega:** [continuación de datos en un nuevo archivo](https://learn.unity.com/tutorial/submission-data-persistence-in-a-new-repo?labelRequired=true&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | | |
| **Una entrega exitosa incluirá lo siguiente:**   * Un enlace al archivo de GitHub del proyecto que muestre varias entregas con mensajes. * Un poco de continuación de datos entre escenas * Un poco de continuación de datos entre sesiones | | |
| **Pasos**   1. Descripción general 2. Fácil: movimiento vertical del jugador 3. Medio: animales agresivos 4. Difícil: interfaz de usuario del juego 5. Experto: barra de hambre de los animales 6. Orientación para la solución 7. ¿Cómo compartir tu trabajo? | | |