

«Talento Tech»

Desarrollo de Videojuegos

Unity 3D

Clase 01



Clase N° 1 | Introducción al 3D

Temario:

- Diferencias entre 2D y 3D en Unity.
- Navegación en el espacio 3D: vistas, movimiento de cámara, herramientas.
- Transformaciones en 3D: posición, rotación, escala.
- Materiales básicos.
- Iluminación inicial.

Objetivos de la clase

En esta primera clase vas a comenzar a formar parte del equipo de TalentoLab. Como desarrollador en este proyecto innovador, tu misión será construir los cimientos del mundo virtual de Nexus. Para eso, vas a dominar herramientas clave del motor Unity, entendiendo cómo funciona un entorno 3D, cómo se manipulan objetos, y cómo la luz y los materiales pueden transformar una escena técnica en un universo inmersivo. Cada conocimiento técnico tendrá una aplicación concreta dentro del mundo que estás ayudando a crear.

Diferencias entre 2D y 3D en Unity.

Antes de sumergirnos en las diferencias técnicas, es importante entender **por qué esta distinción es uno de los primeros grandes cruces de caminos en el desarrollo de un videojuego**.

En el mundo del desarrollo con Unity, crear un proyecto en **2D o 3D** no es solo una cuestión estética: **define la arquitectura base del juego**, los componentes disponibles, la lógica de física, el tipo de assets que usarás, y el modo en que el jugador experimenta tu mundo.

Elegir entre 2D o 3D es como decidir si vas a contar una historia con dibujos planos o construir un mundo tridimensional donde el jugador pueda caminar, mirar alrededor y explorar.

Unity y su flexibilidad

Unity es un motor gráfico potente y versátil que permite desarrollar tanto juegos 2D como 3D, y también combinarlos. Al iniciar un proyecto, uno de los primeros pasos es justamente seleccionar entre estas dos configuraciones:

- En **2D**, Unity optimiza la interfaz y los componentes para trabajar en un plano bidimensional.
- En **3D**, se activa una estructura de escena con profundidad, luces reales y una cámara con perspectiva.

Esta elección **no es irreversible**, pero sí condiciona el flujo de trabajo, el tipo de interacción y las herramientas disponibles desde el inicio.

¿Por qué es clave para vos como desarrollador?

Porque:

- Te prepara para tomar decisiones **conscientes y justificadas** en la preproducción.
- Te ayuda a **evitar errores de arquitectura** (por ejemplo, usar componentes 3D en un juego 2D puede traer problemas de rendimiento o colisiones incorrectas).
- Amplía tu visión como diseñador y desarrollador, permitiéndote adaptar tu enfoque según el tipo de juego o experiencia que querés crear.

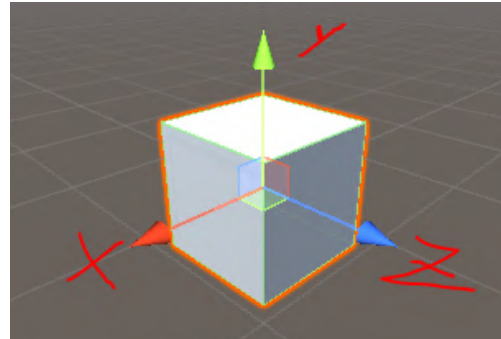
Ejemplo: "Imaginá que una empresa te pide que armes un prototipo rápido de un minijuego. ¿Lo hacés en 2D porque necesitás velocidad? ¿O lo hacés en 3D para mostrar un entorno más inmersivo? Saber las diferencias entre ambos te permitirá tomar esa decisión en minutos, y no perder horas reestructurando más adelante."

¿Qué significa trabajar en 2D o 3D?

Espacio dimensional

Elegir 2D o 3D define la “física” del mundo que vas a construir.

- **2D:** Se trabaja en un plano X-Y; útil para juegos de plataformas, rompecabezas o side-scrollers.
- **3D:** Añade un tercer eje, Z, que aporta profundidad. Ideal para mundos tridimensionales, simuladores o juegos inmersivos.



Cámara

Implica cómo el jugador “ve” el mundo: plano y claro vs. profundo e inmersivo.

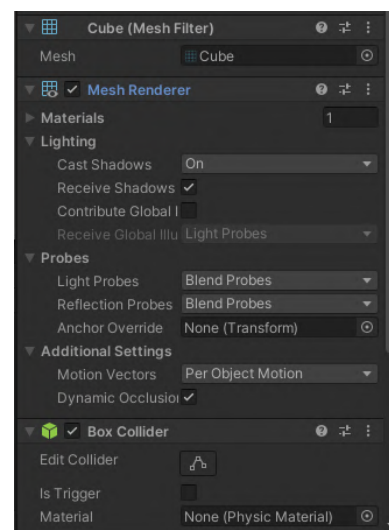
- **2D:** La cámara es ortográfica por defecto (sin perspectiva ni profundidad).
- **3D:** La cámara utiliza perspectiva por defecto, mostrando profundidad y distancia entre objetos.



Componentes

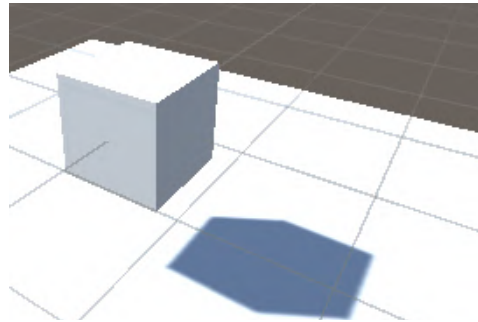
Esto determina qué herramientas usarás y cómo se comportan los objetos.

- **2D:** Sprites, Colliders 2D, Rigidbody 2D.
- **3D:** Meshes (geometrías tridimensionales), Colliders 3D, Rigidbody 3D.



Luces y sombras

- **2D:** Las luces suelen ser decorativas, sin generar sombras reales.
- **3D:** Las luces afectan la apariencia del entorno, generando sombras dinámicas y efectos más realistas.



¿Cómo elegir?

Elegí **2D** si:

- Buscás un desarrollo más rápido.
- El enfoque está en la mecánica o narrativa simple.
- Tenés un equipo con experiencia en arte 2D.

Elegí **3D** si:

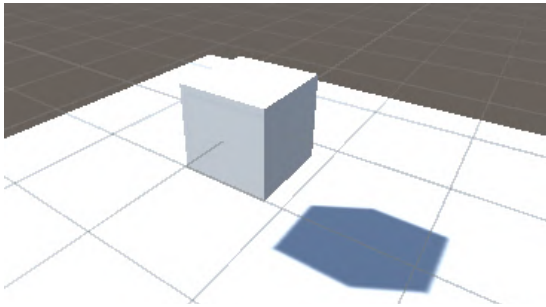
- Querés una experiencia inmersiva.
- Necesitás explorar entornos desde distintos ángulos.
- Tu juego depende de la profundidad espacial (por ejemplo, plataformas en diferentes alturas).

Navegación en el espacio 3D

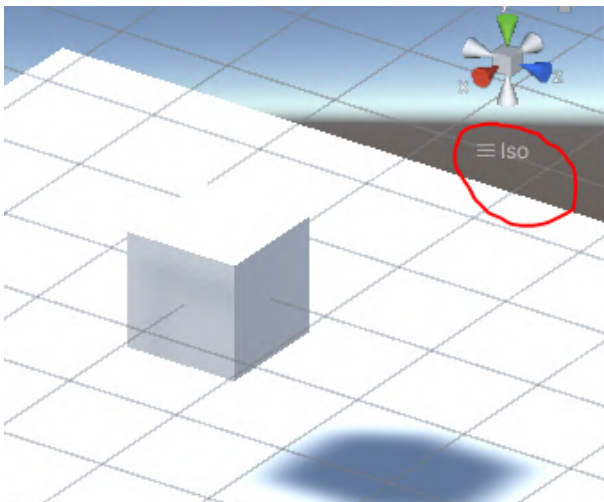
Vistas en Unity

Unity permite alternar entre diferentes formas de “ver” tu escena. Aprender a manejarlas es clave para crear y ubicar objetos con precisión.

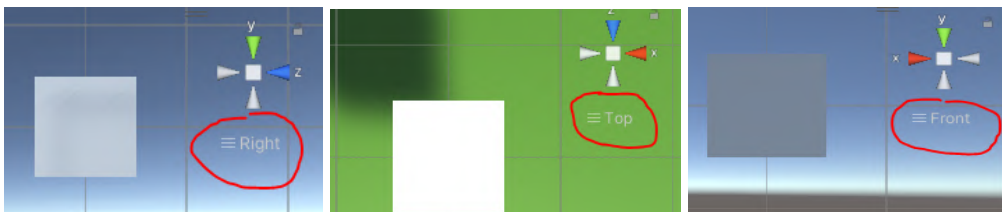
- **Perspectiva:** Vista tridimensional tradicional.



- **Ortográfica o isométrica:** Útil para diseño técnico o visualización plana.



- **Vistas planas:** Top, Front, Right para enfoques precisos.



💡 **Consejo:** Se puede alternar entre vistas haciendo clic en el gizmo de ejes en la esquina superior derecha de la ventana de la escena.

Herramientas de navegación

Movimiento de cámara

La cámara es tu “ojo” dentro del mundo 3D. Aprender a moverla con soltura te permitirá trabajar con rapidez y precisión.

Acción	Control
Rotar la vista	Clic derecho + mover el mouse
Desplazar lateralmente (pan)	Clic central (scroll) + mover
Zoom	Rueda del mouse
Enfocar objeto seleccionado	Seleccioná el objeto y presioná F

💡 **Consejo:** mantené siempre una mano en el mouse y otra en el teclado para combinar navegación y edición sin interrupciones.

Herramientas de manipulación de escena

Una vez que dominás la cámara, necesitás usar las herramientas básicas de Unity para transformar objetos en el espacio 3D.

Herramienta	Atajo	Función
Move Tool	W	Mover el objeto (ejes X, Y, Z)
Rotate Tool	E	Rotar en cualquier eje
Scale Tool	R	Cambiar tamaño
Rect Tool	T	Solo útil para objetos UI o planos
Transform Tool	Y	Combina las anteriores

💡 Estas herramientas aparecen en la barra superior y pueden aplicarse a cualquier objeto seleccionado.

Transformaciones en 3D

En Unity, **toda entidad en la escena** es un “GameObject” que puede transformarse a través de 3 ejes: **X, Y, Z**.

Transformación	¿Qué hace?	Ejemplo visual
Posición	Mueve el objeto en el espacio	Colocar una puerta en la pared
Rotación	Cambia la orientación (ángulo)	Inclinar una lámpara hacia abajo
Escala	Aumenta o reduce el tamaño	Agrandar una montaña o achicar una caja

Estas transformaciones se aplican desde el componente **Transform** de cada objeto (visible en el panel **Inspector**).

Materiales básicos.

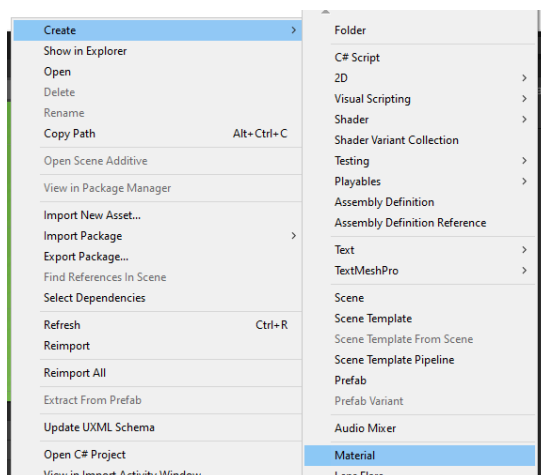
Hasta ahora colocaste objetos en la escena, los moviste, los rotaste y escalaste. Pero todavía todo se ve... genérico. Cubos grises, esferas sin textura, formas que no comunican nada. Para que el mundo de Nexus empiece a tener personalidad, necesitás aprender a aplicar materiales y texturas que definan su estilo visual.

¿Qué es un material en Unity?

Un **material** define **cómo se ve** la superficie de un objeto. Determina el color, el brillo, si refleja la luz o si tiene una textura como madera, piedra o metal.

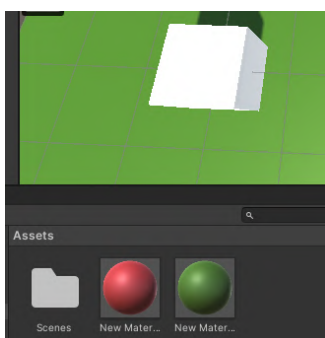
Creación de materiales

- Ruta: Assets > Create > Material.

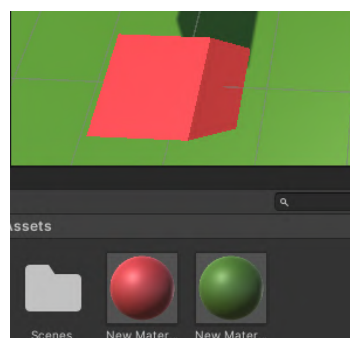


- El material aparecerá en el panel **Project**.
- En el **Inspector**, verás sus propiedades principales.
- Asignar el material a un objeto arrastrándolo al modelo.

Antes:

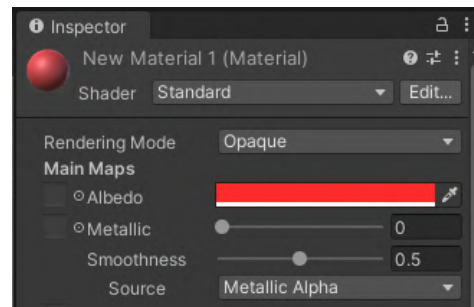



Después:



Propiedades de un material

- **Albedo:** Define el color base o la textura del material.
- **Metallic:** Controla cuánto parece **metal** el material.
- **Smoothness:** Define el nivel de **suavidad/reflexión**..



 **Tip:** Estas propiedades interactúan con la luz. Cambiar la “smoothness” en una escena bien iluminada produce efectos muy notables.


¿Cómo poner una textura a un material en Unity?

Asignar una textura a un material:

- Selecciona el material creado.
- En el Inspector, localiza la propiedad Albedo.
- Haz clic en el pequeño cuadro al lado del color en Albedo.
- Selecciona la textura deseada de tus assets o arrastrarla directamente al campo de Albedo.

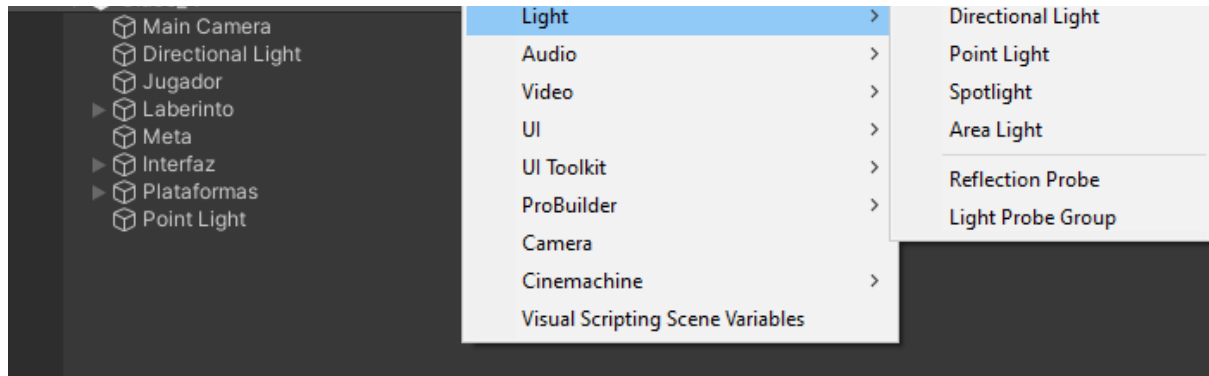
Aplicar el material al objeto:

- Arrastra el material desde el panel Project al objeto 3D en la escena o en la Jerarquía.
- Alternativamente, selecciona el objeto 3D, ve al Inspector, y en su componente Mesh Renderer, asigna el material en la sección de Materials.

 **Consejo visual:** Combiná textura con un color base para efectos más interesantes.

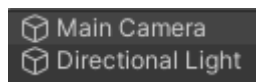
Iluminación

Las luces en Unity 3D son elementos que iluminan escenas y objetos en un entorno 3D. Para crear una luz, vamos a ir a la *Jerarquía*, clic derecho -> Light -> Y nos va a aparecer una lista de luces:

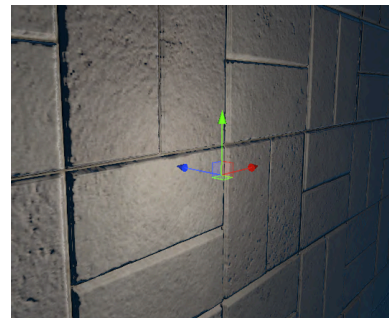


Vamos a explicar las más importantes, que son las siguientes:

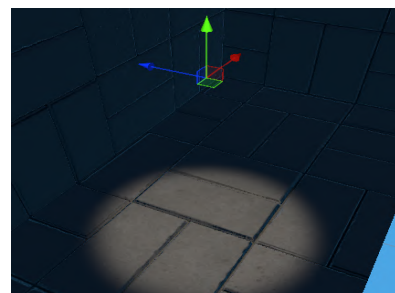
Directional Light: Emite luz en una dirección, funciona como el sol de la escena. Cuando creamos una escena nueva siempre vamos a ver una *Directional Light* y una *Main Camera*.



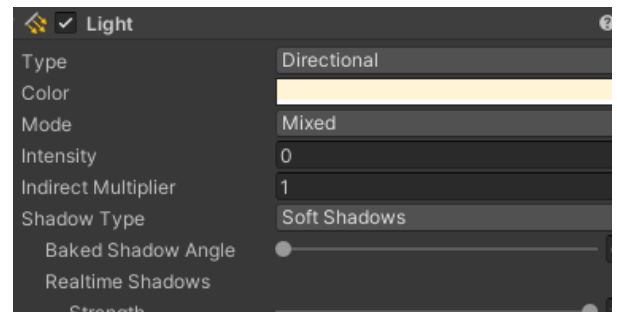
Point Light: Emite luz desde un punto en todas las direcciones. Tiene atenuación con la distancia y produce sombras. No es una luz que emite luz desde el centro, sino que ilumina alrededor desde el centro.



Spotlight: Es un tipo de luz de foco en forma de cono. Esta hace que la luz parta de un punto hacia un lugar en concreto.



Existen otros tipos de luces, pero con estas tres podemos hacer casi todo lo que deseemos en nuestros proyectos. En general, más allá de ciertas diferencias que hay entre los distintos tipos de luces que repasamos, estos comparten una serie de variables que vamos a modificar para conseguir el efecto deseado:



- **Type:** Podemos elegir el tipo de luz antes mencionadas (*Directional*, *Point*, *Spotlight*, etc).
- **Color:** Cambia el color de la luz emitida.
- **Intensity:** Probablemente la variable más importante de las luces, podemos elegir que tan potente es la luz emitida.

¡Talento Lab te está esperando!



¡Les damos la bienvenida a este camino increíble! Imaginá que recibiste una invitación para participar en el proceso de selección de Talento Lab, una startup ubicada en Buenos Aires. ¿Cuál sería el reto? Completar una pasantía de aprendizaje que pondrá a prueba todas tus

habilidades y aprendizajes. A partir de este momento un equipo de expertos te guiarán en este emocionante viaje.

Acerca de TalentoLab.

En un futuro cercano, una innovadora empresa de tecnología llamada TalentoLab ha lanzado un proyecto secreto para revolucionar el mundo de los videojuegos: "Nexus". El objetivo es crear un videojuego 3D completamente inmersivo que combine historia, jugabilidad y un entorno visual espectacular. Los mejores talentos del mundo digital han sido reclutados para formar parte de este ambicioso proyecto. En este curso, tú eres parte de un equipo que debe trabajar en colaboración para desarrollar un juego exitoso y llevar a cabo todas las etapas del proceso creativo y técnico.

Equipo TechLab:



Elizabeth

Product Owner y Lead
GameDesigner



Luigi

Diseñador UX UI y
Lead Artist



Giuseppe

Desarrollador



Roberta

Desarrolladora

Situación inicial en TechLab.



Gracias al crecimiento emergente de nuestra empresa, hemos recibido pedidos de incontables clientes que desean que construyamos las bases de sus proyectos. Te hemos contratado con el fin de engrosar nuestras filas y ampliar nuestros talentos.

Para empezar este nuevo proyecto, deberás manipular de la mejor manera cada elemento en 3D que posee el motor gráfico que usamos: **Unity**. Así que partamos diferenciando lo nuevo (3D) de lo viejo (2D) y creando los primeros objetos de nuestro juego.

Ejercicios prácticos:



Elizabeth, la Product Owner desea que te familiarices con la idea del proyecto. Así que el primer reto como nuevo miembro, es demostrar tu capacidad para trabajar con las herramientas básicas del motor gráfico Unity y empezar a construir los cimientos del mundo de este proyecto.

Actividades iniciales:

1. Preparando el terreno:

Como parte de tu pasantía en TalentoLab, uno de los objetivos es familiarizarte con las herramientas 3D de Unity. Para comenzar:

- Crea un nuevo proyecto en Unity y configura un entorno 3D básico.
- Explora las vistas y herramientas para navegar en el espacio tridimensional.
- Experimenta con transformaciones en 3D: mueve, rota y escala objetos básicos (cubos, esferas, etc.) para comprender cómo manipularlos en el espacio.

2. Construcción inicial del mundo:

Ahora, TalentoLab te pide que empieces a diseñar el concepto de un nivel para Nexus. Siguiendo la filosofía de la empresa, este nivel será un punto de partida que evolucionará con cada clase. Para esta actividad:

- Descargá algunos assets gratuitos de la [Unity Asset Store](#) o creá tus propios modelos simples en Unity.
- Usa esos elementos para diseñar un prototipo básico de un entorno que represente una posible sección del juego. Podría ser un paisaje, un escenario urbano o un entorno abstracto.
- Pensá en el estilo visual y las sensaciones que querés transmitir: ¿futurista, natural, misterioso?

3. Documentación del concepto:

TalentoLab valora tanto la creatividad como la organización. Por eso, además de construir tu nivel, deberás presentar una pequeña descripción de tu idea:

- ¿Cómo describirías el mundo de tu juego?
- ¿Qué tipo de experiencia querés ofrecer al jugador?
- ¿Qué desafíos visuales o jugables creés que podrían encajar en ese entorno?

4. Iteración continua:

Recordá que esta idea inicial no es definitiva. TalentoLab te anima a ir ajustando y ampliando tu mundo con cada clase, añadiendo nuevas mecánicas, detalles visuales y desafíos.

Materiales y recursos adicionales.

Luces:

<https://docs.unity3d.com/Manual/Lighting.html>

Materiales:

<https://docs.unity3d.com/es/2018.4/Manual/class-Material.html>

Preguntas para reflexionar.

1. ¿Por qué elegimos un juego 2D o 3D? ¿Qué diferencias o aportes nos puede dar cada estilo?
2. ¿Qué importancia tienen los materiales, texturas y luces en mi juego?
3. ¿Por qué es importante conocer tanto el 2D como el 3D a la hora de diseñar un juego?

Próximos pasos.

En la próxima clase veremos cómo generar movimientos básicos en nuestro personaje dentro del entorno 3D y empezaremos a crear mecánicas básicas de movimiento como el DoubleJump y el Dash.



Buenos Aires
aprende
Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad