1-.Introducción a unity

1.1 Qué es unity?

Herramienta de creación de videojuegos

Usado en multitud de proyectos

Aporta motores 3D,2D renderizado y exportación a multitud de plataformas

No es solo utilizado para creación de juegos si no también en edición de algunas películas animadas

Manual de usuario y API en constante desarrollo

Comunidad muy grande de usuarios y cantidad de información disponible

Se programa en C#

1.2 Origen y evolución de Unity

Nació en 2004 como GooBall pero fracasó, pero como habían desarrollado buenas herramientas no quisieron dejarlo morir y se acabó convirtiendo en el motor por excelencia de la app store para crear videojuegos y se fue actualizando constantemente.

Unity 2019 -> desarrollo increíble de todas las funcionalidades de Unity.

1.3 Plataformas soportadas

Unity originalmente era solo para Mac OS pero acabó siendo ejecutable también en Windows y es experimental en Linux.

Capaz de exportar juegos a más de 30 plataformas distintas entre ellas PC VR Consolas…

1.4 Licencias y creación de cuenta

id.unity.com para crear la cuenta.

La cuenta va vinculada a la asset store de Unity para garantizar que las compras en la asset store estén siempre disponibles para nosotros.

Las licencias se compran como “asientos” lo que no permite usar una sola licencia y hace necesario tener varias por ejemplo en una empresa.

2 Unity hub

2.1 Qué es Unity hub?

Unity hub es una aplicación para gestionar nuestra cuenta y licencias, nuestros proyectos y las instalaciones y actualizaciones del editor.

2.2 Descarga,instalación y configuración

Se descarga desde la página de Unity directamente.

En el menú de UnityHub podremos loguearnos en nuestra cuenta.

Configurar UnityHub (Cambiar los presets de instalación en caso de que se tenga disco duro pequeños).

Almacén en la nube y en disco duro.

Acceso a tutoriales propios de Unity, proyectos tutoriales links a documentación y presets

Detector de instalaciones de Unity tanto oficiales como la actualmente instalado y acceso a betas.

Permite añadir a posteriori componentes que no se instalaron en la instalación inicial

2.3 Creación de proyectos

Solamente tendremos que pulsar el botón New.

Poner el nombre del proyecto a crear y la localización en disco que se quiere usar, asi tambien como la versión de Unity.

Se pueden elegir templates dependiendo de lo que vayamos a desarrollar.

2.4 Apertura de proyectos

En el UnityHub se puede ver la versión y donde se localiza nuestro proyecto.

Podremos cambiar la version de Unity a utilizar, pero hay que tener cuidado y precaución ya que básicamente se podría romper todo ya que son bastante peligrosos, sobre todo al usar versiones anteriores.

3 Interfaz de usuario de Unity

3.1 Introducción al a interfaz de usuario de Unity

Bastante intimidante a primera vista, está compuesto por paneles configurables.

Conocer bien la interfaz para trabajar correctamente y de forma más cómoda.

3.2 Hierarchy View

Contiene los objetos con los que se va a trabajar.

Contiene las escenas.

Se pueden crear objetos vacíos, objetos 3D,2D, luces, Interfaces de usuario etc.

Se pueden Renombrar los objetos creados para saber lo que estamos haciendo y incluso hacer copiar y pegar.

Estructura padre hijo.

Todos los objetos son hijos de la escena y dentro de la escena se puede hacer que los objetos sean hijos de otros objetos simplemente arrastrando el objeto que se quiere que sea hijo sobre el que queremos que se quiera que sea padre.

Se puede activar o desactivar la visibilidad de los objetos en escena (como en photoshop).

3.3 Scene View: Navegación

Es el Panel central de Unity donde aparecen todos los objetos con los que se está trabajando (lo que vendría siendo la pantalla).

Se puede hacer clic con el ratón y se activa el modo navegación como si fuese un juego en primera persona, para visualizar lo que estamos haciendo, reescalar objetos etc.

Haciendo clic sobre un objeto de la jerarquía nos permitirá ir directamente a ese objeto ( en caso de alejarnos mucho y lo perdamos de vista por ejemplo).

3.4 Scene View: Gizmo

Básicamente el Gizmo(una brújula) indica la referencia entre nuestra cámara y el punto sobre el que estamos rotando la vista, para que nos podamos orientar en el eje de coordenadas y en nuestro mundo.

Son los ejes XYZ , haciendo clic sobre una de las perspectivas se alinea directamente con los ejes, como en autocad.

3.5 Scene View: Draw Mode

Es a efectos prácticos el modo de visualización de los objetos en pantalla mientras trabajamos ( para visualizar aristas, lineas de trazado, contrastes de color).

Se trabaja normalmente en modo wireframe (Visualizar el objeto tal cual se verá en el juego).

3.6 Scene View: Opciones de vista (Parte I)

Se puede cambiar entre visualización 2D y 3D.

Se puede encender y apagar la luz(crear iluminación para ver como quedaría, desde la cámara).

Se puede activar y desactivar también el sonido.

Activar y desactivar efectos como la Skybox.

3.7 Scene View: Opciones de vista (Parte II)

Herramientas personalizadas: Muestra las herramientas personalizadas.

Cámara: permite editar la velocidad de movimiento de la cámara, el campo de visión, el foco..), solo es la cámara que se usa para visualizar, no tiene nada que ver con la del juego, se puede decir a Unity distancias de renderizado.

Gizmos: Se puede editar el tamaño relativo de los iconos que indican el campo de visión la luz, el objeto seleccionado…

Panel de búsqueda: Nos permite buscar objetos específicos, al escribir el nombre los objetos de la escena cuyo nombre coincida conservarán su color, el resto lo perderán.

3.8 Herramientas: Transformaciones

Herramienta de mano: Sirve para moverse.

Herramienta mover: Permite mover los objetos por la escena dependiendo del plano que nosotros seleccionemos.

Herramienta rotación: Permite rotar objetos en la escena, otra vez, dependiendo del plano que seleccionemos rotara de una manera o otra, o rotarlos de forma libre.

Herramienta escala: Permite escalar los objetos cambiandolos el tamaño, se pueden escalar también dependiendo del eje seleccionado(solo crecerá en una dirección) o se puede hacer crecer o encoger a todo el objeto entero de forma proporcionada.

Herramienta Rect Tool: Esta pensada sobre todo para trabajar en 2D, permite rotar y cambiar el tamaño de los objetos , se puede usar en 3D, pero solo trabajara sobre un plano.

Herramienta Universal: Es una herramienta que combina todas las anteriores en una sola, muy útil cuando hay que hacer muchos ajustes de forma continuada.

Herramientas personalizadas: Muestra las herramientas personalizadas (programadas por ti mismo).

Todas las herramientas tienen atajos de teclado.

3.9 Herramientas: Pivot/Center,Local/Global

Pivot: Permite utilizar el pivote de un objeto(punto 0,0) para desplazar uno o más objetos en una selección múltiple siempre se muestra el pivote del último objeto seleccionado

Center: Es como Pivot, pero Center se posiciona sobre un punto intermedio entre los dos o más objetos seleccionados.

Dependiendo de la herramienta seleccionada, cada objeto girará sobre su propio pivote, o sobre el punto intermedio entre ellos.

Global: Es un sistema de coordenadas global para todos los objetos.

Local: Es el sistema de coordenadas del propio objeto.

Por ejemplo en una habitación si te desplazas sobre le sistema global de la habitación tú te moverás hacia el norte por ejemplo, pero si te giras hacia el oeste, si te sigues desplazando sobre el sistema global, seguirías desplazandote hacia el norte, pero si utilizas el Local te desplazarías al Este

3.10 Herramientas: Play,Pause,Step

Play: Ejecutar el juego de forma continua.

Pause: Pausar el juego.

Step: Ejecutar el juego paso a paso (como cuando estamos haciendo debug en Java o C#) se puede usar cuando el juego está pausado y avanza frame a frame.

El juego se puede iniciar en modo pausa para poder ejecutar el paso a paso y ver mas fácil que es lo que está pasando.

3.11 Herramientas: Collab,Cuenta,Layer y Layouts

Collab: Es un sistema de control versiónes, permite trabajar en un proyecto con varias personas.

Cuenta/nube: Abrir las organizaciones que colaboran con Unity (abrir el juego en la nube, compras dentro del juego, anuncios).

Layers:Son las capas del juego, se puede ocultar o mostrar capas de objetos enteras, es como las capas de photoshop.

Layouts: Permite modificar la vista del programa de Unity a nuestro gusto.

3.12 Game View: Display y Aspect Ratio

Game View: Es lo que muestra la cámara, lo que ve el Jugador, se pueden usar varias pantallas, Unity soporta hasta 8 pantallas y cada una sería una cámara diferente, por ejemplo en un simulador de vuelo en una pantalla se vería el mundo y en otro las cartas de navegación comunicación con la torre… etc, solo para Windows, Linux y Mac.

Aspect Ratio: No es recomendable usar free aspect, es el tamaño de pantalla que se va a usar, con free aspect se puede dar el tamaño de pantalla que se quiera, pero nos arriesgamos a que algo quede fuera de la pantalla del juego.

Se pueden crear aspect ratios personalizados que en este caso se utilizarían por ejemplo para juegos de móvil, VGA (estilo clásico).

3.13 Game View: Panel derecho

Maximize On Play: Redimensiona la pantalla de juego automáticamente al tamaño de nuestra pantalla al iniciarlo.

Mute Audio: Sirve para quitar el sonido, útil para cuando has iniciado el juego miles de veces y estás cansado de la música de inicio.

Vsync: Opción de la última versión de Unity, sirve para utilizar el Vsync en el propio menú de desarrollo, el Vsync lo que hace es hacer que no se mande la orden de actualización hasta que se ha terminado de renderizar todo, para evitar el tearing.

Stats: Muestra la carga del procesador, los decibelios(ruido) del juego, lo que tarda la CPU en procesar, cantidad de vértices,triángulos..

3.14 Project View

Contiene todos los assets que se podrían utilizar en el juego, modelos, audios,fondos…

Para copiar un proyecto completo solamente es necesario copiar la carpeta Assets Packages y ProjectSettings.

Cada archivo de assets contiene archivos meta que Unity utiliza para saber qué recursos están relacionados dentro del proyecto, peligroso borrar desde el buscador, hacerlo desde dentro de Unity ya que se podría perder el proyecto entero.

Ser organizado con las carpetas es primordial para no perderse y que sea todo un lío.

3.15 Inspector

Es una vista dinámica, básicamente el inspector nos da información de cada asset,sonido,prop,luz opciones de modelo, importación y renderizado.

De los objetos que estan en nuestra escena podemos cambiar el color de los objetos, hacer que assets que no tienen visualización dentro del juego en la programación nos salga visualmente.

Permite modificar la posición de los objetos y hacer que sean estáticos.

Desde el Inspector se puede hacer depuración (Debug) individualmente cada objeto.

3.16 Consola

Muestra los mensajes de registro de Unity cuando el programa está funcionando

Clear on Play: limpia la consola cada vez que se pulse el botón de jugar.

Error Pause: Pausa automáticamente si iba a ocurrir un error para poder localizarlo fácilmente.

Se pueden hacer búsquedas concretas en la consola de Unity.

3.17 Personalización del layout

Shift+Espacio sirve para maximizar la ventana de la que tenga el foco el ratón.

Los paneles de Unity se pueden mover con total libertad simplemente arrastrandolos y se pueden acoplar con otros paneles, lo que se suele hacer es coger el panel del juego dejarlo flotante y ponerlo en otro monitor por ejemplo y trabajar libremente.

La configuración de layouts que tengamos se puede guardar y en caso de cargarnos los layouts de Unity y liarla mucho podemos volver a poner los layouts de base de Unity con solo pulsar un botón

3.18 Menús

La mayoría de opciones del menú son duplicados de las opciones que están en los propios layouts y, en general las principales opciones son las de cualquier IDE como abrir guardar proyectos copiar, pegar abrir y cerrar paneles que hayamos cerrado sin querer etc.

3.19 Preferencias

Editar->Preferencias, son las preferencias básicas del editor, es decir opciones del programa.

Por ejemplo podemos hacer que Unity Recompile después de parar el juego o que pare el juego directamente y lo recompile.

Unity gratis NO TIENE TEMA OSCURO.

3.20 Project Settings

La mayor diferencia entre project setting y preferencias es que Project Settings son configuraciones específicas del proyecto de Unity que estamos llevando a cabo y no del programa(menú editar, Project Settings).

3.21 Package Manager

Window->Package manager

Nos permite añadir nuevos paquetes ( o quitar ) a Unity y nos permite actualizar los mismos directamente desde aquí.

Es posible que los paquetes que instalemos tengan dependencias por otros paquetes, lo que quiere decir que tendremos que descargarnos todos los paquetes que sean necesarios.

Built-in Packages: Son paquetes propios de Unity y nos permiten desactivar sistemas enteros, por ejemplo si se va a realizar un juego 2D se puede desactivar el 3D para aligerar el propio juego.

3.22 Shortcuts

Nos permite editar los atajos de teclado para ponerlos a nuestro gusto como en cualquier ID, se pueden crear varios perfiles por si se trabaja con otras personas y cada uno tiene sus preferencias (Edit->Shortcuts).

4 Conceptos esenciales

4.1 GameObjects

Todo objeto que tenemos en Unity y está en el clasificador de jerarquías, se podría decir que un GameObject es un contenedor de componentes.

4.2 Componentes

Son piezas funcionales de código, que mediante el mismo indican cómo va a funcionar un determinado objeto

Los componentes se pueden ver en el Inspector y cada componente guarda diversas variables.

Todo gameobject tiene como minimo un componente:

-Transform ya que es necesario para posicionar un objeto en la escena.

Los componentes pueden tener dependencias entre ellos, es decir un componente puede necesitar de otro para funcionar.

Se le pueden quitar dependencias a un componente, cambiar su orden, pero NO se le puede quitar el Transform, para ello hay que eliminar el componente.

4.3 Creación de componentes

Los componentes son lineas de código de C# que añaden funcionalidades a los GameObjects

4.4 Tags

Los tags sirven para marcar y/o clasificar objetos

Se puede utilizar para diferenciar objetos por codigo

Se crean desde Edit-> Project Settings-> Tags & Layers

Cuando se crean duplicados, los duplicados no se llaman igual, ya que tienen “un apellido”, entonces al intentar buscarlos no los encontramos, añadiendo tags esto se remedia ya que el tag no cambia.

No se pueden renombrar, habría que borrar y crear de nuevo.

4.5 Tags: Ejemplo de uso

Demostración de funcionalidad mediante el siguiente código:

private void OnCollisionEnter(Collision other){ if(other.gameObject.CompareTag("Enemigo")){ Debug.Log("Destruyendo enemigo: " + other.gameObject.name); Destroy(other.gameObject); } else if(other.gameObject.CompareTag("Amigo")){ Debug.Log("Hola amigo!"); }

Hace que los objetos al caer y tocar un suelo al que se le ha añadido el componente a los marcados con el tag enemigo los destruye y a los marcados con el tag amigo los saluda.

4.6 Layer:Creación, asignación y uso

Las capas nos permiten agrupar objetos por categorías lo que nos permite definir a que objetos afectan algunos sistemas como las físicas, iluminaciones etc

Al contrario que a los tags a las capas si que se les puede cambiar el nombre sin necesidad de borrarlos.

Si se quiere borrar una capa simplemente con dejar el nombre vacío sirve para borrarla.

Se puede desactivar la interacción entre capas, para por ejemplo que un objeto atraviese a otro.

4.7Escenas:Concepto y uso

Contienen el entorno y objetos del juego

Se podrían tener escenas por ejemplo para un menú, opciones del juego, una pantalla de carga etc.

Ctrl+S shortcut para guardar.

Al guardar la escena se guarda todo, incluyendo los settings.

Las escenas están en la categoría de Assets

Las escenas se pueden cargar o haciendo clic en ellas (a mano en preview) o mediante código(cuando el juego está en funcionamiento).

4.8 Escenas: Edición multi-escena

En muchas ocasiones es útil poder trabajar con varias escenas a la vez.

La escena en la que se cargan los managers y separar el escenario en distintas escenas para facilitar la carga de la escena y el trabajo de varias personas a la vez.

Es muy fácil trabajar con escenas múltiples a la vez, deben estar en la misma jerarquía para poder cargarlas y descargarlas, esto significa que, al cargarlas los objetos que pertenezcan a otra escena se verán en la escena principal, evitando así una carga innecesaria o un cambio abrupto entre distintas “fases” del juego.

5 Asset Store

Es la tienda de Assets de Unity:

-Código.

-Texturas.

-Materiales.

-Modelos.

-Animaciones.

-Utilidades.

-De pago y gratuitos

Unity ofrece distintos packs de ejemplo como parte de sus tutoriales

6 Prefabs

6.1 ¿Qué es un prefab?

Un prefab es un GameObject que nos permite almacenar componentes y reutilizarlo como un asset, sería algo similar a una plantilla de objetos, como por ejemplo entornos(el interior de una habitación si hay que hacer muchas casas por ejemplo).

Los cambios realizados sobre el prefab se realizan automáticamente a todas sus instancias creadas.

6.2 Creación de un prefab

Simplemente para crear un prefab hay que arrastrar el GameObject a la carpeta que elijamos y a esto se le llamaría instanciar un prefab, se puede instanciar sobre el editor o sobre el código del juego.

Para añadir un prefab a nuestro juego simplemente lo tendremos que arrastrar a la ventana.

6.3 Prefab Mode

Los prefabs se pueden modificar para que afecten a todas las instancias

Para entrar en el modo de edición con hacer doble click sobre el prefab ya se podría editar

Esto abre una escena nueva en la que estará nuestro prefab (plantilla) y todo lo que editemos en esta escena afectará al prefab.

6.4 Instance Override

Permite hacer pequeños cambios en las instancias del prefab sin romper su vínculo con el prefab.

-Añadir componentes.

-Cambiando valores de propiedades.

-Eliminando componentes.

-Añadiendo hijos.

(Efectivamente el del curso dice que es programador y no artista vaya bicho mas feo)

Tiene limitaciones:

-No se pueden eliminar GameObjects hijos del prefab.

-No se pueden cambiar los padres de un prefab.

6.5 Aplicar valores desde la instancia

Si seleccionamos el objeto que hemos editado podemos editar todos los Overrides que le hemos hecho en la opción de overrides, y aquí se puede editar los cambios y aplicarlo directamente al prefab, haciendo que todos los objetos cambien así mismo estos cambios también se pueden deshacer los cambios.

6.6 Prefabs anidados

Como el propio nombre indica nos permite tener prefabs dentro de otros prefabs

Anidar un prefab es como añadir cualquier objeto.

Cada prefab mantiene su independencia.

6.7 Variantes

Son variaciones de otros prefabs, son similares a los overrides de instancias pero usando prefabs

Para crearlo solamente hay que hacer:

Click derecho sobre un prefab Create/Prefab Variant o arrastrar el prefab otra vez a la carpeta proyecto y Unity ya pregunta si se quiere crear un variant o un prefab nuevo.

6.8 Desempaquetado de Prefabs

Es la operación inversa de un Prefab, para ello solamente hay que hacer click derecho en el objeto y unpack prefab y volvería a ser un GameObject normal.

7 Cache Server

7.1 Qué es el Cache Server?

Es una herramienta que incluye Unity para ahorrar tiempo cuando se cambia entre distintas plataformas o cuando se quiere importar Assets que ya han sido usados por otra persona que trabaja en el mismo proyecto que nosotros.

Unity no utiliza el formato original de los assets, los convierte a un formato para la plataforma sobre la que está trabajando.

Básicamente guarda una copia de los recursos ya convertidos para que cuando sea necesario volver a usarlos no tengan que volver a codificarse por completo.

7.2 Servidor Local vs Remoto

Se puede trabajar con él en modo Local, Remoto o simplemente desactivado

-Deshabilitado: no es recomendable, los recursos tendrán que importarse cada vez que se cambie de plataforma.

-Local: Si no es necesario compartir la caché porque solo trabajamos nosotros.

-Remoto: En un servidor externo, para trabajar en un equipo.