GAME OBJECTS

Andrés Felipe López Velásquez, Cristian Arturo Paz Álvarez, Noe Santiago Castellanos Salazar

¿QUE ES?

Los GameObjects son objetos fundamentales en Unity que representan personajes y el escenario. Estos no logran nada por sí mismos, pero funcionan como contenedoras para Componentes, que implementan la verdadera funcionalidad. Por ejemplo, un objeto Light es creado al adjuntar un componente Light a un GameObject.

Un GameObject siempre tiene el componente Transform adjunto (para representar la posición y orientación) y no es posible quitar esto. Los otros componentes que le dan al objeto su funcionalidad pueden ser agregados del menú Component del editor o desde un script. También hay muchos objetos útiles pre-construidos (figuras primitivas, cámaras, etc) disponibles en el menú GameObject.

Ya que los GameObjects son una parte importante de Unity, hay una sección de GameObjects en el manual con detalles extensivos acerca de ellos. Usted puede encontrar más acerca de controlar los Gameobjects desde scripts en la página de referencia de scripting del GameObject.

OTROS COMPONENTS

Mirando la Cámara Principal del GameObject, se puede ver que contiene una colección diferente de Components. Especialmente, un Camera Component, a GUILayer, un Flare Layer, y un Audio Listener. Todos estos Components proporcionan funciones adicionales al GameObject. ¡Sin estas, no habría nada representando los gráficos del juego para la persona que está jugando! Rigidbodies, Colliders, Particles, y Audio son todos diferentes Components (o combinaciones de Components) que pueden ser agregados a cualquier GameObject.

Los Components son las tuercas y tornillos de los objetos y comportamientos de un juego. Son las piezas funcionales de cada GameObject. Si todavía no entiende la relación entre Components y GameObjects, lea la página de GameObjects antes de seguir adelante.

Un GameObject es un contenedor para muchos Components distintos. Por defecto, todos los GameObjects automáticamente tienen un Transform Component. Esto es

porqué el Transform dicta dónde el GameObject está ubicado, y cómo es girado y escalado. Sin un Transform Component, el GameObject no tendría lugar en el mundo. Intente crear un GameObject vacío.

Recuerde que siempre puede utilizar el Inspector para ver qué Components son adjuntados al GameObject seleccionado. A medida que los Components son agregados y quitados, el Inspector va a siempre mostrar cuáles están actualmente adjuntos. Usted va a utilizar el Inspector para cambiar todas las propiedades de cualquier Component (incluyendo scripts).

AGREGANDO COMPONENTS

Usted puede agregar Components al GameObject seleccionado a través del menú de Components. Nosotros vamos a intentar esto añadiendo un Rigidbody a un GameObject vacío que creamos. Selecciónelo y escoja Component->Physics->Rigidbody del menú. Cuando lo haga, va a ver las propiedades del Rigidbody aparecer en el Inspector. Si oprime Play mientras que el GameObject vacío esté seleccionado, podrá tener una pequeña sorpresa. Inténtelo y note cómo el Rigidbody le ha agregado funcionalidad al GameObject que estaba vacío. (La posición Y del GameObject comienza a disminuir. Esto es porqué el motor de física de Unity está causando que el GameObject se caiga gracias a la gravedad.)

Puede adjuntar cualquier número o combinación de Components a un solo GameObject. Algunos Components trabajan mejor en combinación con otros. Por ejemplo, el Rigidbody trabaja con cualquier Collider. El Rigidbody controla el Transform a través del motor de física NVIDIA PhysX, y el Collider permite que el Rigidbody se choque e interactúe con otros Colliders.

EL TRANSFORM COMPONENT

Es imposible crear un GameObject en Unity sin un Transform Component. El Transform Component es uno de los más importantes Components, ya que todas las propiedades Transform del GameObject están habilitadas por sus usos en este Component. Define la posición, rotación, y escala del GameObject en el mundo/Scene View del juego. Si un GameObject no tuviera un Transform Component, sería nada más que alguna información en la memoria del computador. Efectivamente no existiría en el mundo.

El Transform Component también habilita un concepto llamado Parenting, que es utilizado a través del Unity Editor y es una parte critica de trabajar con GameObjects. Para aprender más sobre el Transform Component y Parenting, lea Transform Component Reference page.

EDITANDO COMPONENTS

Uno de los grandes aspectos de los Components son la flexibilidad. Cuando se adjunta un Component a un GameObject, hay diferentes valores o Properties en el Component que pueden ser ajustadas en el editor mientras se construye un juego, o por scripts cuando esté corriendo el juego. Hay dos tipos de propiedades ( Properties): Values y References.

Este componente contiene una sola propiedad de Reference (referencia), y siete propiedades de valor. Un Audio Clip es la propiedad de referencia. Cuando esta Audio Source empieza a reproducirse, intentará reproducir el archivo audio que está referenciado en la propiedad Audio Clip. Si ninguna referencia se hace, un error va a ocurrir ya que no hay audio para reproducir. Usted debe referenciar el archivo dentro del Inspector. Esto es fácil como arrastrar un archivo de audio del Project View a la propiedad Reference o utilizando el Object Selector (selector de objetos).

Los componentes pueden incluir referencias a cualquier otro tipo de Componente, GameObject, o Assets. Usted puede leer más acerca de cómo asignar referencias en la página acerca de editar propiedades.

Las propiedades restantes en el Audio Clip son todas propiedades con valor. Estas pueden ser ajustadas directamente desde el Inspector. Las propiedades de valor del Audio Clip son todas palancas, valores numéricos, campos desplegables, pero las propiedades con valor también pueden ser strings de texto, colores, curvas, y otros tipos. Puede leer más acerca de estos y sobre editar propiedades de valor en la página acerca de editing value properties](EditingValueProperties.html).

ELIMINAR (REMOVE)

Un comando Remove Component está disponible para casos en dónde ya no se necesite el component adjunto al GameObject. Observe que hay algunas combinaciones de components que dependen de cada uno (eg, Hinge Joint solo funciona cuando un Rigidbody también esté adjunto); verá un mensaje de advertencia si trata de eliminar components que otros dependan.

PROBANDO LAS PROPIEDADES

Mientras su juego esté en Play Mode, usted es libre de cambiar las propiedades en Inspector de cualquier GameObject. Por ejemplo, puede querer experimentar con diferentes alturas de salto. Si crea una propiedad Jump Height en el script, puede entrar en Play Mode, cambiar el valor, y oprimir el botón de salto para ver qué pasa. Luego, sin salirse de Play Mode puede cambiarlas nuevamente y ver los resultados en segundos. Cuando se salga de Play Mode, sus propiedades se van a revertir a sus valores pre-Play Mode, para que no se pierda nada de trabajo. Este flujo de trabajo le da un poder increíble con el cual experimentar, ajustar, y refinar su juego sin la necesidad de invertir mucho tiempo en ciclos de iteración. Inténtelo con cualquier propiedad en Play Mode. Creemos que va a estar impresionado.

TRANSFORM

El componente Transform determina la Position, Rotation, y Scale de cada objeto en la escena. Cada GameObject tiene un Transform.

CREANDO COMPONENTES

Cuando usted cree un script y lo adjunte a un GameObject, el script aparece en el inspector del GameObject al igual que un Componente. Esto se debe a que los scripts se vuelve Componentes cuando son guardados - un script es simplemente un tipo especifico de Componente. En términos técnicos, un script compila como un tipo de Componente, y es tratado como cualquier otro componente por el motor de Unity. Entonces básicamente, un script es un componente que usted está creando usted mismo. Usted definirá sus miembros que serán expuestos en el Inspector, y se ejecutará cualquier funcionalidad que usted ha escrito.

DESACTIVANDO GAMEOBJECTS

Un GameObject puede ser temporalmente eliminado de una escena marcándolo como inactivo. Esto puede ser hecho usando la propiedad activeSelf del script o con la casilla de activación en el Inspector

APPLICANDO UNA ETIQUETA(TAG)

El Inspector va a mostrar la etiqueta(Tag) y Layer->Layers drop-down menus justo abajo del nombre de cualquier GameObject. Para aplicar una etiqueta a un GameObject, simplemente abra el despegable de etiquetas y escoja la etiqueta que requiera

CREANDO NUEVAS ETIQUETAS

Para crear una nueva Etiqueta, haga click en la opción “Add tag…” al final del menú desplegable. Esto abrirá el Tag Manager en el Inspector. El Tag Manager es descrito aquí.

Las capas(Layers) aparecen similares a las etiquetas(Tags), pero son usadas para definir cómo Unity debe representar los GameObjects en la escena. Mire la página de Layers para más información.

REFERENCIAS BIBLIOGRAFICAS

<https://docs.unity3d.com/es/530/Manual/class-GameObject.html>