UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS

Tecnicatura en diseño y programación de videojuegos

UNLVIRTUAL

Manipulación de Objetos en 3D

Unidad N° 2: Introducción a Unity 3D

Docentes
Jaime Fili
Nicolás Kreiff

Contenidos

¿Qué es Unity 3D?

Unity es una herramienta integrada para desarrollar videojuegos en 3D u otros contenidos interactivos, creada por la empresa Unity Technologies. Consiste en un editor para desarrollar y diseñar estos productos, así como en un motor para ejecutarlos una vez finalizados.

Figura 1. Logo de Unity.

Si bien en sus inicios era una herramienta únicamente disponible para Mac, y sólo podía publicar contenidos para Mac y Windows, actualmente el entorno de desarrollo de Unity viene en versiones para Mac OS X® , Microsoft ® y Windows ® . Además, desde la presentación de la versión para Windows en 2009, Unity continuó agregando soporte para iOS, Andriod, Wii, Xbox 360, PlayStation 3 y recientemente Adobe Flash.

Unity se ha convertido en la elección más adecuada para estudios pequeños, desarrolladores independientes y todo aquel que quiera hacer sus propios juegos. Posee una gran base de usuarios y una comunidad de desarrolladores muy activa que permite a cualquiera, desde veteranos hasta neófitos, conseguir respuestas y compartir información muy rápidamente.

Provee un excelente punto de entrada al mundo del desarrollo de videojuegos con un buen equilibrio entre características, funcionalidad y precio. La versión gratuita de Unity permite experimentar, aprender, desarrollar y vender juegos sin la necesidad de invertir. Esto ha sido uno de los factores fundamentales para convertir a Unity en el motor de más rápida adopción en los últimos años.

Adicionalmente, la variedad de plataformas de destino que soporta hacen que sea una alternativa muy atractiva en un mercado que demanda cada vez menor tiempo de producción, y en el que los desarrolladores no pueden darse el lujo de invertir en la creación de sus propias herramientas.

La versión PRO de Unity es muy accesible y provee un conjunto interesante de características. Además, existen licencias especiales que no sólo son accesibles sino también son *royalty free* -dependiendo de las plataformas de destino, que pueden no estar contempladas en la versión gratuita ni en la PRO. Esto permite a los desarrolladores crear y vender juegos sin necesidad de pagar por el derecho de uso del motor.

Características principales

Las siguientes son las principales características de Unity:

- Entorno integrado de desarrollo con jerarquías, edición visual, inspectores de propiedades detallados y previsualización del juego en vivo.

- *Deployment* en múltiples plataformas:

1. Ejecutable de Microsoft® Windows® o Mac OS® X
2. Para la web (a través del plugin Unity Web Player para Internet Explorer, Firefox, Safari, Mozilla, Netscape, Opera, Google Chrome y Camino) en Microsoft® Windows® y Mac OS® X.
3. Como widget para el dashboard de Mac OS® X.
4. Nintendo Wii.
5. iPhone / iPad.
6. Android.
7. Cliente Nativo de Google Chrome.
8. Microsoft® Xbox 360.
9. Adobe Flash Player con soporte 3D.
10. Sony® PlayStation® 3.

- Los *assets* utilizados que se utilizan en Unity son importados automáticamente al actualizarsecuando son actualizados. Unity soporta integración con 3D Studio Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop y Allegorithmic Substance.
- El motor gráfico utiliza Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iOS, Android), y APIs propietarias para Wii.
- Soporte para *bump mapping, reflection mapping, parallax mapping, screen space, ambient occlusion*, sombras dinámicas usando mapas de sombras, *render-to-texture* y efectos de posprocesado a pantalla completa.
- Posee su propio lenguaje para escribir *shaders* llamado “ShaderLab”.
- Soporte para el motor de física PhysX Engine de Nvidia.
- Soporte para *scripting* a través de Mono (*framework* .NET OpenSource) y utilizando MonoDevelop como IDE. Los programadores pueden utilizar JavaScript, C# o Boo (lenguaje inspirado en la sintaxis de Python).
- Unity Asset Server como gestor de versionado basado en PostgreSQL.
- Sistema de Audio incorporado a través de la librería FMOD con la posibilidad de reproducir archivos con compresión Ogg Vorbis.
- Reproducción de video a través del códec Theora.
- Motor de terreno y vegetación incorporado con soporte para plantado de objetos.
- Motor de Oclussion Culling de Umbra.
- Integración de *lightmapping* e iluminación global de BEAST.
- Soporte *multiplayer* de Raknet.
- A partir de la versión 3.5 se incorpora su propio soporte para *pathfinding*.

Adicionalmente, Unity posee una tienda de componentes llamada “Unity Asset Store” que ha permitido crear una comunidad de desarrolladores muy sólida, aportando soluciones (tanto pagas como gratuitas) para diversas necesidades asociadas al desarrollo de los videojuegos.

Tanto desde el punto de vista académico como profesional, esto le incorpora un valor agregado muy importante a la plataforma.

Instalando Unity

La instalación de Unity es un proceso sencillo y directo. En este apartado también se explicará cómo preparar el entorno de trabajo para compilación y *deployment* en plataformas mobile.

Requisitos de Sistema

Requisitos de Sistema para desarrollar en Unity

- Windows: XP SP2 o superior; Mac OS X: CPU Intel y “Leopard” 10.5 o superior.
- Tarjeta de Video con 64 MB de VRAM, soporte para *pixel shader* o unidades de 4 texturas.
- Para usar Occlusion Culling se requiere una GPU con soporte para Occlusion Query.

Requisitos para desarrollar en Unity para iOS

- Una Mac basada en chipset Intel.
- Mac OS X “Snow Leopard” 10.6 o superior.

Requisitos para desarrollar en Unity para Android

- En adición de los requisitos generales desarrollar en Unity.
- Windows: XP SP2 o superior; Mac OS X: CPU Intel y “Leopard” 10.5.8 o superior.
- SDK de Android y Java Development Kit (JDK).

Instalación Básica

El primer paso es descargar el instalador desde la siguiente URL: http://unity3d.com/unity/download/

Unity está disponible para instalar en entornos **Microsoft® Windows®** y **Mac OS® X** únicamente. Por el momento no existe versión para Linux aunque se estima que en el corto plazo podría haber una solución para esto.

Una vez descargado el instalador, se debe ejecutar y seguir las instrucciones.

En Windows, la instalación por defecto crea el grupo de programas “Unity”, y en Mac OS crea una carpeta en *launchpad* también llamada Unity.

Se recomienda elegir la prueba de gratis durante 30 días de la versión PRO que incluye iOS y Android PRO. Finalizado el tiempo de prueba se puede volver a activar ya sea con una licencia paga o con la licencia gratuita.

Nota
Al finalizar la instalación, Unity se activa mediante una conexión a internet. Abrirá una ventana del navegador por defecto y pedirá que se elija el tipo de licencia.

En la imagen anterior se pueden apreciar los elementos que quedan luego de instalar el producto. Los más importantes son el ícono de "Unity" y el de "MonoDevelop"; este último es el que se utilizará para escribir los scripts en C#. La aplicación "Unitron" es el editor de *scripting* original de Unity y todavía se provee como alternativa para aquellos que no quieran usar MonoDevelop.

Entorno de Desarrollo para iOS

Además de los requisitos de sistema que ya se mencionaron, para compilar e instalar aplicaciones en dispositivos iOS es necesario tener instalado **Xcode IDE de Apple.**

Se puede descargar gratuitamente desde la **AppStore de Apple**, accesible desde cualquier computadora Mac con Mac OS X 10.6 o superior. Se debe tener en cuenta que para poder instalar Xcode se debe tener actualizado Mac OS X a la última versión disponible.

Sólo se podrá hacer uso del módulo iOS de Unity en sistemas operativos Mac OS X.

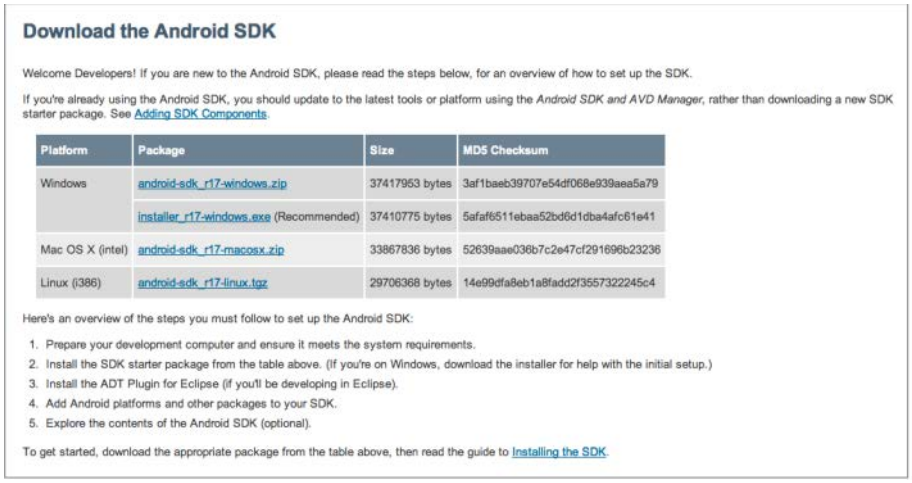
Por más de que la licencia lo permita este módulo no será accesible desde un sistema basado en Windows.

Sin embargo, el desarrollo se puede realizar sin inconvenientes en un entorno Windows y luego cambiar de plataforma en un entorno Mac OS X.

Entorno de Desarrollo para Android

A diferencia de iOS, para poder compilar e instalar en sistemas Android se pueden utilizar tanto sistemas basados en Windows como en Mac OS.

Para descargar el SDK de Android ir a la siguiente URL y seguir las instrucciones: http://developer.android.com/sdk/index.html



Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather than downloading a new SDK starter package. See Adding SDK Components.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r17-windows.zip	37417963 bytes	3af1baeb39707e54df068e939aea5a79
	installer_r17-windows.exe (Recommended)	37410775 bytes	5afaf6511ebaa52bd6d1dba4afc61e41
Mac OS X (intel)	android-sdk_r17-macosx.zip	33867836 bytes	52639aae036b7c2e47cf291696b23236
Linux (i386)	android-sdk_r17-linux.tgz	29706368 bytes	14e99dfa8eb1a8fadd2f3557322245c4

Here's an overview of the steps you must follow to set up the Android SDK:

1. Prepare your development computer and ensure it meets the system requirements.
2. Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)
3. Install the ADT Plugin for Eclipse (if you'll be developing in Eclipse).
4. Add Android platforms and other packages to your SDK.
5. Explore the contents of the Android SDK (optional).

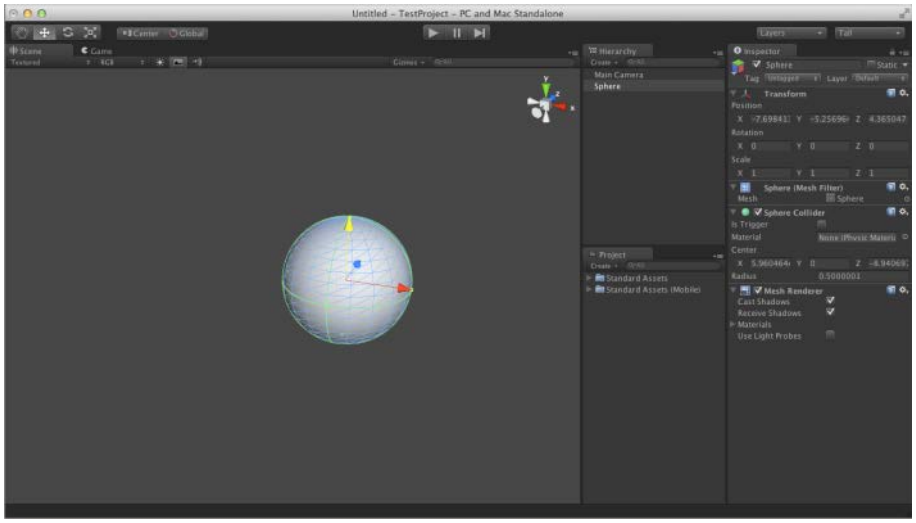
To get started, download the appropriate package from the table above, then read the guide to Installing the SDK.

Finalizada la instalación de los prerrequisitos, el usuario se encuentra en condiciones de compilar e instalar tanto para dispositivos iOS como Android.

Los detalles de la generación de binarios para las mencionadas plataformas se verán más adelante.

Interfaz de Usuario

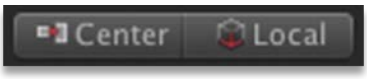
En la imagen a continuación se puede apreciar la apariencia general de la interfaz de usuario de Unity.



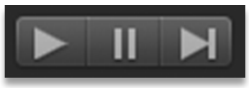
La disposición de la misma se puede configurar de diferentes maneras, según la preferencia del usuario.

Barra de Herramientas

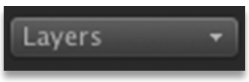
En la esquina superior derecha encontramos las herramientas de edición para panear la escena, mover, rotar y escalar los objetos.



Estas herramientas afectan la forma en la que se muestra la escena de edición.



Los botones Play / Pause / Step son utilizados en la pestaña de previsualización del juego.



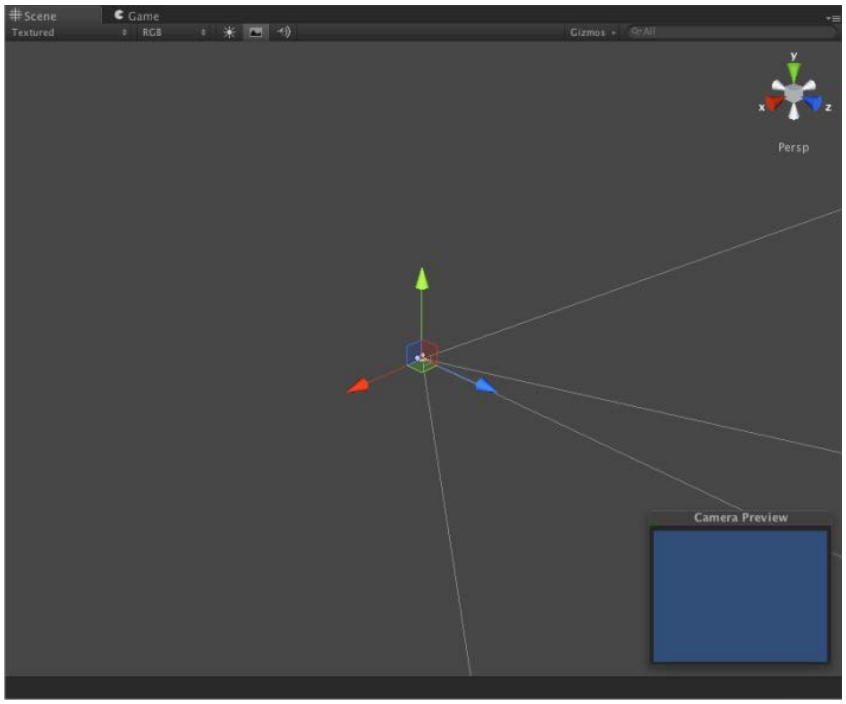
El *dropdown* de capas permite seleccionar que objetos son mostrados en la vista de escena.

El *dropdown* de distribuciones permite seleccionar de qué manera desea el usuario que se muestre la interfaz de usuario de Unity.

Vista de Escena

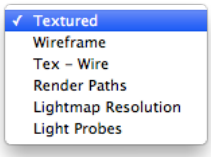
La porción central de la pantalla está destinada a las pestañas de edición de escena y a la de previsualización del juego. En la imagen de ejemplo a continuación hay una escena vacía, solamente con la Main Camera. Al seleccionarla, aparece un cuadro de previsualización en la esquina inferior derecha que muestra qué está viendo la cámara en ese momento.

En la misma imagen, sobre la esquina superior derecha se puede observar la disposición actual de los ejes coordenados. Haciendo click sobre cada uno de ellos, o del cubo central se puede cambiar la perspectiva de visualización.



La pestaña de edición de escena posee varias opciones que se pueden ver debajo del título de la misma.

De izquierda a derecha son:



Modo de Dibujado: permite elegir como se desea ver los objetos en la escena. Por defecto se muestran los objetos texturados.

- **Textured:** muestra las superficies con sus texturas visibles.

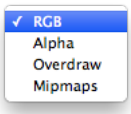
- **Wireframe:** dibuja las mallas en forma de malla de alambres.

- **Tex-Wire:** muestra las mallas texturadas y con una capa superpuesta de malla de alambres.

- **Render Paths:** muestra los *rendering paths* para cada objeto utilizando un código de colores: VERDE indica iluminación diferida, AMARILLO indica *forward rendering* y ROJO indica *vertex lit*.

- **Lightmap Resolution:** superpone una grilla en la escena que muestra la resolución de los *lightmaps*.

Modo de Renderizado: permite elegir qué método de renderizado se utilizará para mostrar la escena:



- **RGB:** renderiza la escena con todos los objetos coloreados normalmente.

- **Alpha:** renderiza los colores con alpha.

- **Overdraw:** renderiza los objetos como siluetas transparentes. Los colores transparentes se acumulan haciendo fácil identificar los lugares en donde un objeto se dibuja por encima de otro.

- **Midmaps:** muestran el tamaño ideal de la textura usando un código de color. ROJO indica que la textura es más grande de lo necesaria (a la distancia y resolución actual); AZUL indica que la textura debería ser mayor. Naturalmente, el tamaño ideal de las texturas dependen de la resolución a la que el juego correrá y cuán cerca puede llegar a estar la cámara de superficies en particular.

- **Scene Lighting, Game Overlay y Audition Mode**: a la derecha del modo de renderizado se encuentran tres botones que controlan otros aspectos de la representación de la escena.

El primero permite seleccionar si la vista será iluminada con el esquema de iluminación por defecto o si utilizarán luces agregadas a la escena. El esquema por defecto es utilizado automáticamente hasta que se sume una nueva luz a la escena.

El segundo botón controla si los *skyboxes* y elementos de la GUI serán renderizados en la vista de escena. También muestra y oculta la grilla de ubicación.

El tercer botón enciende y apaga las fuentes de audio de la escena.

- **Gizmos**: son indicadores visuales que se superponen a los objetos en la escena para indicar diversas cosas. En la imagen a continuación se puede visualizar los distintos gizmos que se pueden habilitar (o deshabilitar) en la vista de edición de escena.

Unity posee un API especial para que el desarrollador cree sus propios gizmos en caso de que lo necesite.

Vista de Juego

La vista de juego se renderiza de la cámara (o cámaras) en el juego. Es representativo del producto final.

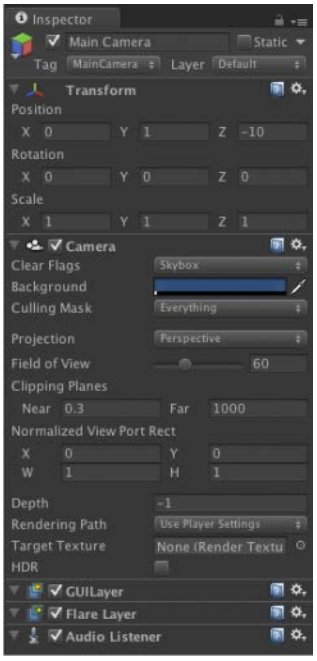
Al igual que la vista de escena, posee opciones para controlar su funcionamiento.

- **Aspect Dropdown:** desde aquí se puede forzar el *aspect ratio* de la vista de juego con diferentes valores. Es posible utilizarlo para testear cómo el juego se verá en diferentes resoluciones y *aspect ratios*.

- **Maximize on Play:** mientras esté habilitado, la vista del juego se maximizará al 100% de la ventana del editor.

- **Stats:** muestra la ventana con las estadísticas del renderer.

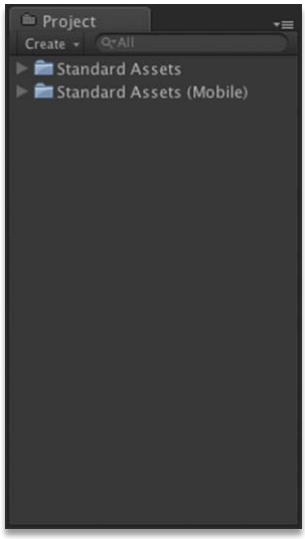
- **Gizmos:** mientras está habilitado, se muestran los mismos gizmos que aparecen en la vista de escena. Esto incluye los dibujados con cualquiera de las funciones en la clase Gizmos.



Inspector: Los juegos en Unity están compuestos de múltiples *GameObjects* que contienen mallas, scripts, sonidos u otros elementos gráficos como las luces. El "inspector" muestra información detallada acerca el GameObject seleccionado en ese momento, incluyendo los componentes adjuntados y sus propiedades.

Con el inspector se puede modificar la funcionalidad de un GameObject en la escena , así como cualquier propiedad que aparezca en él; incluso las variables de los scripts pueden cambiar sin modificar el script. Se puede utilizar el inspector para cambiar variables en tiempo de ejecución para experimentar y encontrar el equilibrio correcto de *gameplay* en un juego. En un script, si se define una variable pública de tipo objeto (como GameObject o Transform) se puede arrastrar y soltar GameObjects o Prefabs en el inspector para hacer la asignación de manera visual.

Project View



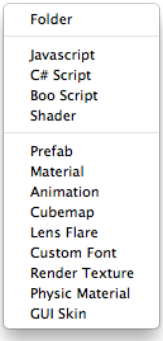
Todo proyecto en Unity contiene una carpeta llamada "Assets". Los contenidos de dicha carpeta se muestran en la vista de proyecto (Project View). Aquí es donde se almacenan todos los *assets* que componen el juego, como escenas, scripts, modelos 3D, texturas y Prefabs.

Si se hace click derecho sobre cualquier asset en la vista de proyecto, se puede elegir la opción "Reveal in Explorer" ("Reveal in Finder" en Mac) para ver cómo está almacenado el asset en el sistema de archivos.

Para agregar assets a un proyecto, se pueden arrastrar desde cualquier lugar del sistema operativo a la vista de proyecto. Alternativamente, se puede hacer la opción "Assets > Import New Asset" del menú de opciones. Una vez hecho esto, el asset estará listo para ser utilizado en el juego.

Nota
Es recomendable nunca mover assets del proyecto usando el sistema operativo ya que puede romper cualquier metadata asociada con el asset. Es mejor siempre usar la vista de proyecto para organizarlos.

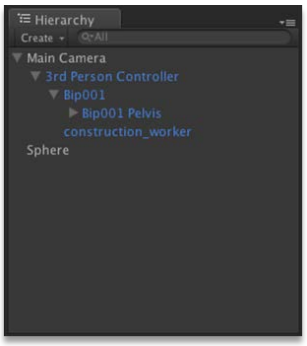
Las **escenas** se pueden pensar como niveles individuales del juego y también son almacenadas en la vista de proyecto. Para crear una nueva escena se utiliza el comando CTRL+N (Command+N en Mac) y para guardar una escena actual CTRL+S (Command+S en Mac).



Algunos assets deben ser creados desde Unity. Para esto se debe usar el *drop-down* denominado "Create", o bien click derecho en la vista de proyecto y elegir la opción "Create".

Esto permitirá agregar scripts, prefabs o carpetas para mantener el proyecto organizado. Se puede renombrar cualquier asset o carpeta presionando "**F2**" en Windows, **Enter** en Mac, o con dos clicks espaciados sobre el nombre del asset. Si se mantiene presionada la tecla **Alt** mientras se expande o contrae un directorio, todos los subdirectorios serán expandidos o contraídos.

Hierarchy



La vista de jerarquía contiene todos los GameObjects en la escena actual. Algunos de estos son instancias directas de assets, como pueden ser modelos 3D, y otros son instancias de Prefabs (si bien se verán más adelante, estos son objetos personalizados que serán el elemento más común en los desarrollos en Unity). Se puede seleccionar y "parentizar" objetos en la jerarquía de Unity. A medida que se agregan o remueven objetos de la escena, estos aparecerán y desaparecerán de la jerarquía también.

Parenting: Unity utiliza el concepto de parenting (o "parentizar", es decir, conventir en padre de). Para hacer que cualquier objeto sea hijo de otro, simplemente se debe arrastrar el que se desea convertir en hijo sobre el que se desea que sea padre en la vista de jerarquía. Un hijo va a heredar el movimiento y la rotación del padre. Se puede expandir y contraer el padre para ver sus hijos en la jerarquía sin afectar el comportamiento del juego.

Referencias

Documentación OnLine de Unity: http://unity3d.com/support/documentation/

Unity (game engine): http://en.wikipedia.org/wiki/Unity_Technologies

Beginning 3D Game Development with Unity – Apress 2011