Apartado 1: descripción 🡪 historia del motor y descripción general

Descripción general

El motor gráfico utiliza Direct3D (en Windows), OpenGL (en Mac y Linux), OpenGL ES (en Android y iOS), e interfaces propietarias (Wii). Tiene soporte para mapeado de relieve, reflexión de mapeado, mapeado por paralaje, pantalla de espacio oclusión ambiental (SSAO), sombras dinámicas utilizando mapas de sombras, render a textura y efectos de post-procesamiento de pantalla completa.

El soporte integrado para Nvidia (antes Ageia), el motor de física PhysX, (a partir de Unity 3.0) con soporte en tiempo real para mallas arbitrarias y sin piel, ray casts gruesos, y las capas de colisión.

El scripting viene a través de Mono. El script se basa en Mono, la implementación de código abierto de .NET Framework. Los programadores pueden utilizar UnityScript (un lenguaje personalizado inspirado en la sintaxis ECMAScript), C# o Boo (que tiene una sintaxis inspirada en Python). A partir de la versión 3.0 añade una versión personalizada de MonoDevelop para la depuración de scripts.

Unity también incluye Unity Asset Server - una solución de control de versiones para todos los assets de juego y scripts, utilizando PostgreSQL como backend, un sistema de audio construido con la biblioteca FMOD, con capacidad para reproducir audio comprimido Ogg Vorbis, reproducción de vídeo con códec Theora, un motor de terreno y vegetación , con árboles con soporte de billboarding, determinación de cara oculta con Umbra, una función de iluminación lightmapping y global con Beast, redes multijugador RakNet y una función de búsqueda de caminos en mallas de navegación.

DAFO

**Fortalezas:**

* Independiente de plataforma: no estamos anclados a un solo sistema operativo.
* Ofrece un editor gráfico de fácil uso y muy potente, el cual permite importar objetos y modelos 3D hacia nuestros proyectos de forma fácil y rápida.
* La versión gratuita de la herramienta es muy completa y puede satisfacer de buena manera las necesidades e intereses de los desarrolladores de videojuegos.
* Permite incorporación de nuevas características mediante la incorporación de plugins específicos.
* Cuenta con un Asset Store en el cual podemos encontrar recursos que podemos utilizar en nuestros proyectos, dichos recursos pueden ser tanto gratuitos como de pago, pero en caso de que nos sea difícil trabajar con el aspecto de diseño gráfico del juego este Asset Store nos puede ser de gran utilidad.
* Amplia documentación en los foros de la comunidad.

**Debilidades:**

* No permite empezar desde unas bases, o de unas plantillas, por decirlo de alguna manera, e ir luego implementando detalles. Si no que debes empezar de cero con cada juego.
* La existencia de licencias más caras puede suponer una limitación para desarrolladores freelance o grupos de desarrollo pequeños. Las licencias más caras aportan principalmente mejoras gráficas y de rendimiento, pero solo vale la pena hacerte con ellas si tienes un equipo mediano o grande, y un proyecto ambicioso.

**Amenazas:**

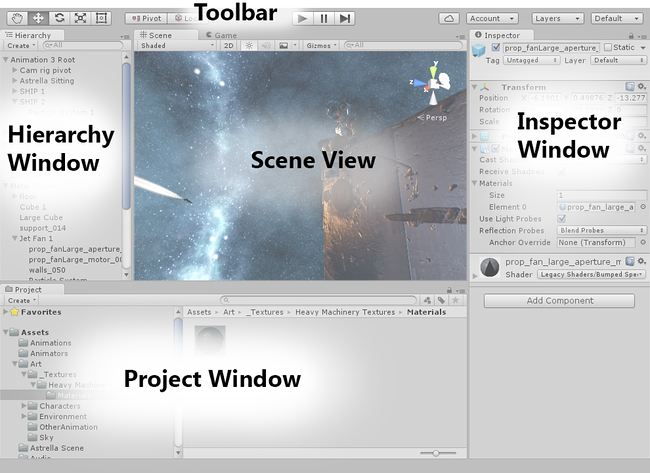
* Los competidores actuales están en constante desarrollo.
* Surgen nuevos competidores.

**Oportunidades:**

1. DISEÑO
   1. Interfaz

<http://docs.unity3d.com/es/current/Manual/LearningtheInterface.html>

La interfaz de Unity es bastante sencilla, está compuesta por los siguientes elementos:



La **barra de herramientas** proporciona acceso a las principales funciones de Unity. A la izquierda contiene las herramientas más básicas para la manipulación de la escena y los. En el centro están los controles de reproducción. A la derecha se encuentran los accesos a la nube y a la cuenta de usuario, seguidos de un menú de visibilidad de las capas, y finalmente el menú Editor de diseño (que proporciona algunos diseños alternativos y permite guardar los diseños propios).

La **ventana de proyecto** muestra su biblioteca de activos o recursos disponibles para utilizar en el proyecto, ordenados por categorías. Todo lo que se guarde en la carpeta predefinida del proyecto se importará automáticamente.

La **vista de escena** es donde pondremos todos los elementos de nuestro juego. Permite editar la escena visual, ya sea desde una perspectiva 3D o 2D, dependiendo del tipo de proyecto que se está trabajando. Tanto los objetos como los escenarios se pueden arrastrar a la posición deseada. Contiene 2 pestañas: scene y game, cambiando a la pestaña game podremos ver cómo se comportaría el juego con los elementos y las acciones programadas en la vista de escena.

La **ventana de jerarquía** La jerarquía muestra la forma en que todos los elementos de nuestro juego están relacionados entre sí.

La **ventana de inspector** permite ver y editar todas las propiedades de objeto que se haya seleccionado. El contenido de esta ventana variará en dependencia del tipo de objeto seleccionado, ya que los diferentes tipos tienen diferentes propiedades.

* 1. Recursos predefinidos

<https://www.assetstore.unity3d.com/en/>

<http://docs.unity3d.com/es/current/Manual/HOWTO-InstallStandardAssets.html>

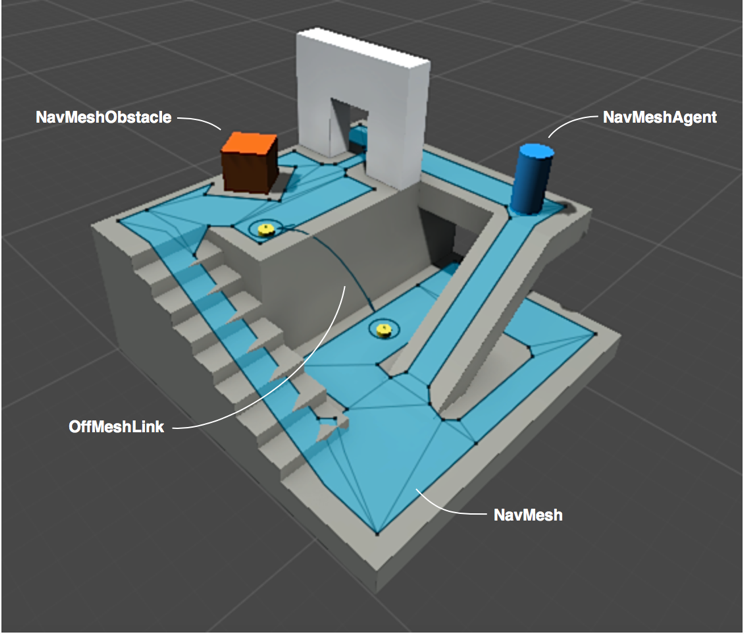
<http://docs.unity3d.com/es/current/Manual/UISystem.html>

<http://docs.unity3d.com/es/current/Manual/UIReference.html>

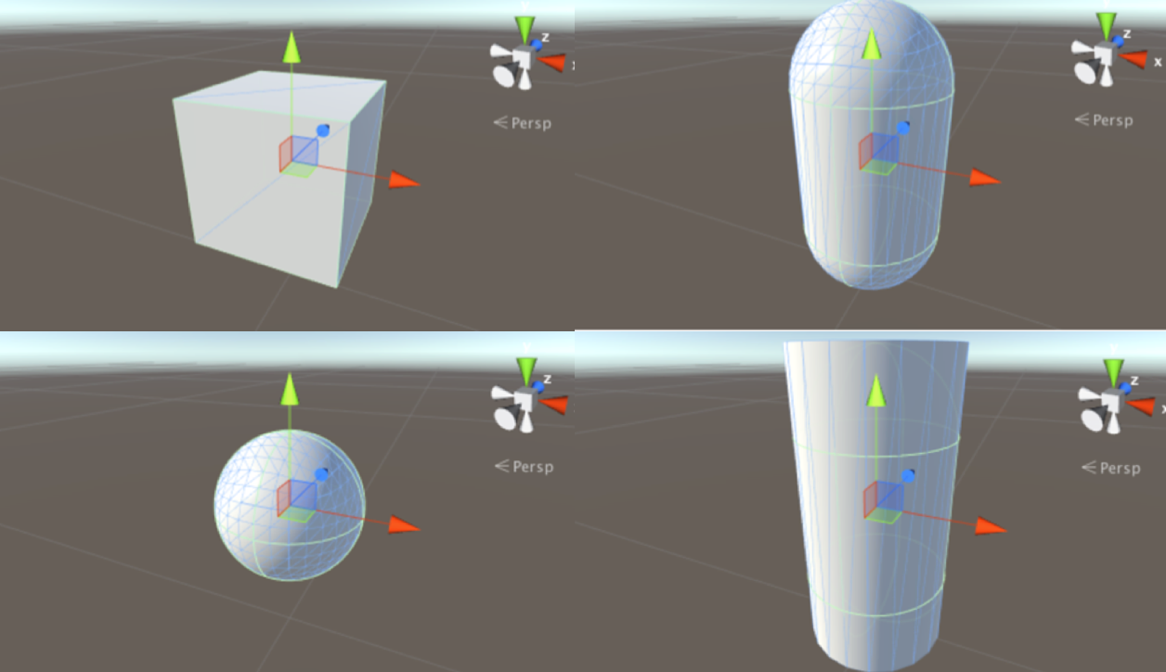
Con la descarga del programa, descargamos también una serie de assets predefinidos (Standard Assets). Estos son principalmente 2D, cámaras, personajes, efectos, entorno etc. Se pueden transferir entre proyectos usando paquetes Unity (Unity packages), que se pueden exportar e importar.

Unity cuenta con una serie de componentes de interacción como son:

* Botones con eventos UnityEvents OnClick.
* Toggle: casilla de verificación **Is On** que determina si el toggle está actualmente apagado. Este valor es cambiado cuando el usuario haga clic . Éste también tiene un UnityEvent **OnValueCHanged** para definir lo que hará cuando el valor sea cambiado.
* Sliders (Deslizadores) para seleccionar valores mínimos y máximos.
* Barras de desplazamiento.
* Campos de texto.

El sistema de navegación le permite crear personajes que se pueden mover de forma inteligente en el mundo del juego. El sistema de navegación utiliza mallas de navegación para razonar sobre el medio ambiente. Las mallas de navegación se crean automáticamente de su geometría de la escena. obstáculos dinámicos permiten alterar la navegación de los personajes en tiempo de ejecución, y fuera de la malla de vínculos le permiten construir acciones específicas, tales como la apertura de puertas, o saltando desde una cornisa. En esta sección se describe la navegación Unidad y la búsqueda de caminos en detalle.

Unity puede trabajar con modelos 3D de cualquier forma, se pueden crear con el software de modelado. Sin embargo, también hay una serie de tipos de objetos primitivos que pueden ser creados directamente dentro de Unity, por ejemplo cubos, esferas, cápsulas o cilindros.



1. CARACTERÍSTICAS GENERALES
   1. Multiplataforma

<https://unity3d.com/es/unity>

<https://unity3d.com/es/unity/multiplatform>

El número de plataformas sobre las que puede correr un videojuego desarrollado en Unity está creciendo constantemente.

Para dispositivos móviles:

* Despliegue en un clic para Android, iOS, Windows Phone y Tizen.
* Gran cantidad de optimizaciones gracias a características como la agrupación de assets.
* Herramientas y flujos de trabajo 3D y 2D dedicados y fáciles de usar.

Para realidad virtual y realidad aumentada:

* Pipeline de renderizado altamente optimizado.
* Capacidades de iteración rápida del editor de Unity.
* Native Oculos Rift, Gear VR and Playstation VR están disponibles para desarrolladores Unity. Procimamente lo estarán Microsoft HoloLens y Sream VR/Vive.

Juegos de escritorio:

* Posibilidad de crear juegos de escritorio de alta calidad gracias al shader físico de Unity.
* Realtime Global Illumination de Enlightnen, una simulación de cómo se transporta la luz físicamente. Significa que es una manera de simular cómo la luz se pasa entre las superficies en la escena 3D, lo que mejora el realismo visual en el juego. No sólo eso, sino que también puede ayudar en la ambientación y gameplay de tu juego. Los algoritmos de GI toman en cuenta la luz que viene directamente de una fuente de luz (iluminación directa) y también la luz que se refleja por las superficies en la escena con diferentes materiales (iluminación indirecta).
* Suministra soporte para despliegues con un clic en plataformas PC, Mac y Linux.

Juegos para consola:

* Más fácil que nunca para desarrolladores independientes publicar para plataformas de consolas.
* PS4, Xbox One, PlayStation Mobile, PlayStation Vita y Wii U de forma gratuita.
* Los procesos de aprobación varían según la plataforma, la comunicación hay que establecerla directamente con los propietarios para solicitar más información y para acceder a las compilaciones de despliegue en consolas de Unity.

Para web:

* Plug-in web gratuito.
* WebGL de Unity altamente optimizado que puede ofrecer un rendimiento a velocidades nativas. Ya ha sido ampliamente utilizado para crear numerosos títulos comerciales de éxito.

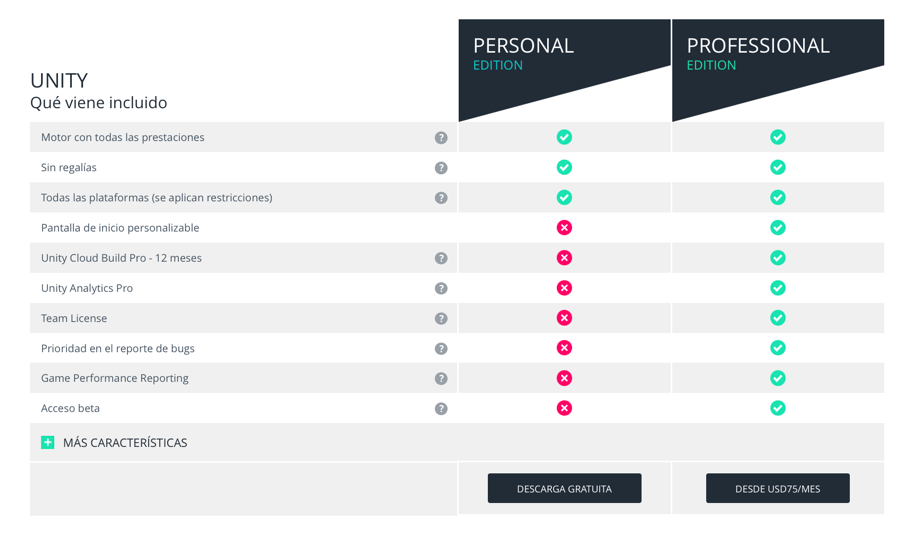
Para Smart Tvs

* Desarrollo para tvOS , Android Tv y Samsung Smart Tv.
  1. Licencia

Unity ofrece diferentes tipos de licencia, con diferentes características:

Para **desarrolladores**:

* Licencia gratuita con una serie de características.
* Licencia de pago mensual con características más específicas.



Para **estudiantes:**

Acceso sencillo a la plataforma Unity a escuelas, estudiantes y docentes, todo ello con el conjunto completo de prestaciones profesionales del editor comercial de Unity. Además de la tecnología líder de la industria que es accesible a una amplia gama de usuarios, Unity suministra una amplia gama de tutoriales, ejemplos de proyectos de juego y recursos comunitarios.

* Descuentos especiales en cursos, certificados, contenidos y servicios para instituciones educativas
* Una biblioteca de aprendizaje de videos tutoriales y ejemplos de proyectos.
* Entradas con descuento para estudiantes y profesores a conferencias de desarrolladores de Unity.
* Una amplia comunidad en línea de usuarios comprometidos.
* Programa de licencias gratuitas a las instituciones educativas de primaria y secundaria y programas de aprendizaje.
* Kit de herramientas Unity.

Para **empresas**:

Las Soluciones Empresariales a la medida garantizan que Unity respalde plenamente las metas creativas y de negocios de la organización.

* Incluyen actualizaciones.
* Descuentos por volumen y de otro tipo.
* Licencias de códigos fuente.
* Soporte a medida bajo pedido.

1. RENDIMIENTO
   1. Optimización

<http://docs.unity3d.com/es/current/Manual/OptimizingGraphicsPerformance.html>

(herramientas del entorno para optimizar)

**Optimización del CPU**

Con el fin de que el CPU haga menos trabajo, es bueno reducir la cuenta de los objetos visibles. Unity cuenta con **Draw call batching**. Para dibujar un objeto en la pantalla, el motor tiene que emitir un draw call (llamado de dibujo) al API de gráficas (e.g. OpenGL o Direct3D). Los Draw Calls a menudo son costosos, con el API de gráficas haciendo un trabajo significante para cada draw call, causando una sobrecarga en el rendimiento por el lado del CPU. Esto en su mayoría es causado por los cambios de estado hechos entre los draw calls (e.g. cambiando a un material diferente), que causa pasos de validación costosos y de translación en el driver gráfico. Unity utiliza varias técnicas para hacer frente a esto: Static Batching (combinar objetos estáticos en grandes mallas, y hacerlas de una manera más rápida) y Dynamic Batching (para pequeñas mallas, transforma sus vértices en la CPU, agrupa varios y dibuja una sola vez).

**Rendimiento de Iluminación**

La iluminación que no es computarizada para nada siempre es la más rápida. **Ligthmapping** permite computar una iluminación estática solo una vez, en lugar de computarla cada frame. El proceso para generar un ambiente lightmapped toma solo un poco más que simplemente colocar una luz en la escena en Unity, pero va a correr mucho más rápido (2–3 veces por 2 luces por-píxel) y se verá mucho mejor ya que el lightmapper puede suavizar los resultados de una iluminación global.

Optimizar una iluminación por-píxel salva ambos CPU y GPU: el CPU tiene menos draw calls por hacer, y el GPU tiene menos vértices para procesar y píxeles por rasterizar para todos estos renderizadores de objetos adicionales. Unity cuenta con las Quality Settings que sirven para modificar cuántas luces terminan como luces de píxel y cuántas como luces de vértice. Cada luz calcula su importancia basada en qué tan lejos está desde el mesh y qué tan intensa su iluminación es. Adicionalmente, algunas luces son más importantes que otras puramente del contexto del juego. Por esta razón, cada luz tiene un ajuste Render Mode que puede ser establecido a Important o Not Important; las luces marcadas como Not Important van a típicamente tener una carga menor de renderización.

**GPU: Compresión de Textura y Mipmap**

Utilizar **Compressed Textures** va a disminuir el tamaño de las texturas (resultando en unos tiempos de carga más rápidos y huellas de memoria más pequeñas) y también puede aumentar el rendimiento de renderización. Las texturas comprimidas utiliza solo una fracción del ancho de banda de memoria necesitado para texturas. De la misma forma, habilitar Generate Mip Maps para texturas utilizadas en escenas 3D puede ayudar limitar la cantidad de datos de textura transferida cuándo el GPU está renderizando, una textura mip mapped va a habilitar el GPU para utilizar una textura de baja-resolución para triángulos pequeños.

**LOD y Distancias de Eliminación Por-Capa (Per-layer Cull Distances)**

**Level of Detail** permite que elementos pequeños puedan hacerse invisibles en largas distancias, mientras que los objetos más grandes todavía serían visibles. De esta forma se reduce la carga de CPU y GPU.

La eliminación selectiva de oclusión (**Occlusion Culling**) es una característica que desactiva la renderización de objetos cuando estos no están vistos por la cámara ya que son oscurecidos por otros objetos.

1. UTILIDAD
   1. Compatibilidad con programas externos

<http://unityspain.com/topic/10633-exportar-modelo-de-unity-a-programa-3d-externo/>

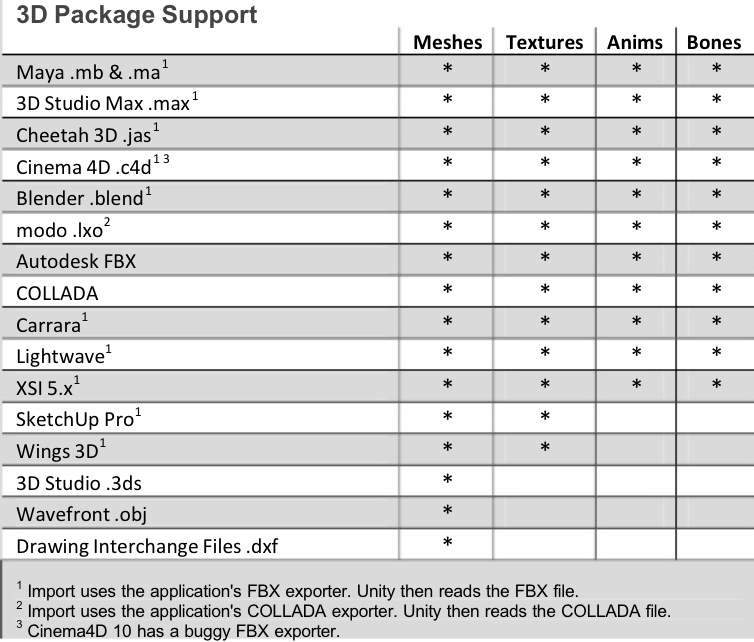
<http://www.pablomarcos.net/siempre-es-mejor-utilizar-fbx-antes-que-obj-en-unity3d/>

<http://docs.unity3d.com/es/current/Manual/HOWTO-importObject.html>

<http://docs.unity3d.com/es/current/Manual/AudioFiles.html>

Unity admite diferentes formatos de mallas, OBJ, FBX, Blend, MA, MB, MAX… Y por supuesto cada modelador trabajará con una aplicación 3D distinta, ya sea Maya, Blender, Strata, Lightwave, 3DS Max, Cinema4D y muchos más.

Para importar una malla Propietaria en Unity (como MA, MB, MAX o Blend) se necesita tener instalado el software 3D apropiado a la hora de importarla. Para archivos MA y MB se necesita tener Maya instalado, para archivos MAX se necesita 3DS MAX instalado, para archivos Blend, Blender, y lo mismo con el resto de formatos de archivo Propietarios. Esto es porque el importador de mallas de Unity lo que realmente hace es exportar el archivo Propietario a un archivo FBX sin que nos enteremos. Durante la importación, Unity carga el software de modelado 3D asociado a ese archivo y utiliza su exportador de FBX interno para obtener una versión de ese archivo en FBX. Por lo tanto, para Unity las mallas Propietarias crean una dependencia con el software que han sido modeladas ya que Unity necesita ese software para importar la malla. Si el software no está presente durante la importación, entonces la importación fallará. Una vez importada la malla, el software con el que ha sido modelada ya no es necesario.

A continuación se muestran los formatos soportados:

**Imágenes:** Photoshop .psd y .tiff se importan con capas acopladas de forma automática. También soporta JPEG, PNG, GIF, BMP, TGA, IFF, PICT y otros muchos formatos.

**Audio y vídeo:** MP3 y archivos de audio Ogg Vorbis .Ogg son compatibles de forma nativa, dependiendo de la plataforma. En las plataformas móviles, el audio se convierte en MP3 para tener un uso completo de la descompresión de hardware. AIFF, WAV y otros formatos de audio también son compatibles, ideal para los efectos de sonido. En Unity Editos se puede configurar una compresión . Son totalmente compatibles los formatos MOD, IT, archivos de seguimiento S3M, XM. Theora .ogg es soportado de manera nativa. Los formatos de vídeo MOV, AVI, ASF, MPG, MPEG, archivos MP4video se recodifican.

**Otros formatos soportados:** archivos XML y .txt pueden ser referenciados en tiempo de ejecución. Cualquier otro tipo de archivo, como RTF y DOC, se pueden utilizar para las notas de proyectos y listas de tareas pendientes.

* 1. Servicios

<http://docs.unity3d.com/Manual/UnityServices.html>

Unity trae una creciente gama de servicios gratuitos para ayudar a los desarrolladores hacer juegos y participar, retener y rentabilizar el público. Algunos de estos servicios son:

* **Unity Ads** es una herramienta de monetización para los juego. Es una red de publicidad móvil que ofrece beneficios líderes en el mercado. La red sólo muestra los anuncios de videojuegos a gamers, lo que los hace mucho más relevantes. Eso es muy bueno para los anunciantes que buscan impulsar las descargas de su juego y para los editores que buscan generar ingresos con la venta de publicidad.
* **Unity Analytics** es una sencilla pero potente plataforma de datos que proporciona análisis para los juego de Unity, averiguar quiénes son los jugadores y su comportamiento en el juego. Plataformas compatibles: iOS, Android, Windows Phone 8.1, Windows Store 8.1 (escritorio), Windows Store 10.0 (Escritorio), Mac, PC, Linux, Web Player (Sólo disponible con el SDK), WebGL.
* **Unity IAP** hace que sea fácil de implementar compras en su aplicación a través de las más populares tiendas de aplicaciones. Las plataformas de apoyo incluyen: App Store de iOS, Mac App Store, Google Play y Windows Store.

1. PROGRAMACIÓN
   1. Lenguaje

<http://academiaandroid.com/scripts-lenguajes-programacion-unity/><http://developeando.net/unity3d/>

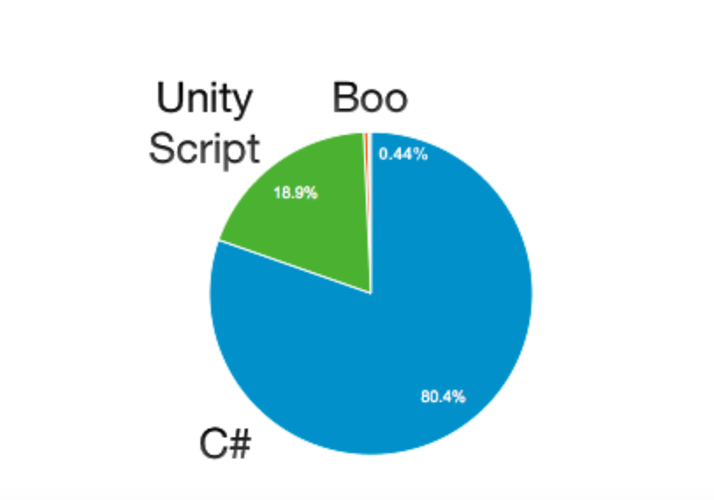
Podemos controlar el comportamiento de los GameObjects en Unity asignándoles componentes predefinidos y configurando sus propiedades a través de una interfaz gráfica. A pesar de su versatilidad, es posible que estos componentes sean insuficientes si queremos crear una mecánica de juego más personalizada o compleja.

Para conseguir ese objetivo, Unity nos ofrece la posibilidad de implementar *scripts*. Un *script* es un pequeño programa que podemos asociar a un GameObject como si se tratara de un componente más, proporcionándole una funcionalidad concreta que definimos nosotros. Podemos escribirlos en alguno de los lenguajes que Unity soporta de forma nativa: C#, UnityScript y Boo.

**C#**: Es el lenguaje que prima el propio motor de juegos Unity, ya que le da un soporte especial y basa toda su documentación en él.

* Es el más utilizado, con mucha diferencia, por la comunidad de usuarios de Unity, lo que facilita luego el aprovechamiento de scripts creados por terceros en nuestros propios videojuegos.
* Su similitud con Java, lo hacen muy fácil de utilizar por aquellos que trabajan en el entorno Android.

**Boo**: Lenguaje de programación orientado a objetos, de tipos estáticos para la Common Language Infrastructure, con una sintaxis inspirada en Python y un énfasis en la extensibilidad del lenguaje y su compilador. Ha sido descontinuado a efectos prácticos en las nuevas versiones de Unity, ya que no pueden crearse scripts en este lenguaje directamente en el editor. Esta falta de soporte, debida según Unity al pequeño porcentaje de usuarios que lo utilizan, hace que sea mejor descartarlo.

**Unity Scripting** es una ‘versión’ particular de Javascript. Aunque más utilizado que Boo, está lejos de las cifras de C# y la falta de documentación oficial es una señal a tener en cuenta.

* 1. API (librerías, scripts)

Unity cuenta con una serie de clases predefinidas, además de las que podamos crear. Estas clases aportan funcionalidades que facilitan la creación de videojuegos. Algunos ejemplos de estas clases son:

* **MonoBehaviour**: Clase base para todos los nuevos scripts. Proporciona una lista de todas las funciones y eventos que están disponibles para scripts estándar adjuntos a Game Objects. Importante para cualquier interacción o control sobre objetos individuales en el juego.
* **Transform**: Cada Game Object tiene una posición, rotación y escala en espacio (ya sea 3D o 2D), y esto es representando por el componente Transform. Además de proporcionar esta información, tiene muchas funciones útiles que pueden ser utilizadas para mover, escalar, girar, re-apadrinar y manipular objetos, al igual que convertir coordenadas desde un espacio a otro.
* **Rigidbody/ Rigidbody2D:** Para la mayoría de elementos del gameplay, el motor proporciona un conjunto de herramientas más fácil para mover objetos, detectando triggers y colisiones, y aplicando fuerzas. La clase Rigidbody (o su equivalente en 2D, Rigidbody2D) proporciona todas las propiedades y funciones necesarias para jugar con la velocidad, masa, fricción, fuerza, colisión, etc.

Además de estas clases Unity ofrece una serie de APIs que facilitan otras funciones del desarrollo de un videojuego, como son la integración con redes sociales, la jugabilidad o las capacidades multijugador.

**API Social**

Es el punto de acceso de Unity para características sociales, tal como:

* Perfiles de usuario
* Lista de amigos
* Logros
* Estadísticas / Clasificaciones

Proporciona una interfaz unificada a diferentes back-ends sociales, tal como XBox Live o el GameCenter, y es utilizado principalmente por programadores en el proyecto del juego.

**Playable API**

Esta nueva API proporciona una forma de crear herramientas, efectos u otros mecanismos de juego mediante la organización y la evaluación de las fuentes de datos en una estructura de árbol.

**API de Alto Nivel**

(HLAPI) es un sistema para construir capacidades multijugador para juegos de Unity. Maneja las tareas comunes que son requeridas para juegos multijugador.

1. GRAFICOS
   1. Renderizado

<http://docs.unity3d.com/Manual/30_search.html?q=rendering>

Modos de renderizado:

Unity tiene diferentes **modos** que permiten elegir si el objeto utiliza la transparencia, y si es así, qué tipo de mezcla va a utilizar.

* Opaco: Es el valor predeterminado, y es adecuado para objetos sólidos normales sin zonas transparentes.
* Recorte: Le permite crear un efecto de transparencia que tiene bordes duros entre las áreas opacas y transparentes. En este modo, no hay áreas semitransparentes, la textura es o bien 100% opaco, o invisible. Esto es útil cuando se utiliza la transparencia para crear la forma de materiales tales como hojas, o una tela con agujeros y los suelos.
* Transparente: Adecuado para la representación realista de materiales transparentes tales como plástico transparente o de vidrio. En este modo, el propio material se enfrentarán a los valores de transparencia (basado en el canal alfa de la textura y de la alfa del color de tinte), sin embargo reflexiones y una excelente iluminación permanecerá visible a plena claridad como es el caso de materiales transparentes reales.
* Fade: Permite que los valores de transparencia a desaparecer por completo un objeto, incluso de cualquier reflejos especulares o reflexiones que pueda tener. Este modo es útil si se desea animar un objeto aparición o desaparición paulatina. Se ISS no adecuado para la representación realista de materiales transparentes, tales como vidrio o plástico transparente debido a las reflexiones y pone de relieve también se desvanecieron.