Preparación y exportación de modelos para la aplicación Unity

Victor Hugo Franco Serrano
UNAM, DGTIC, Departamento de Realidad Virtual
Julio 2011

1. Áreas de aplicación

Modelado, Animación, Exportación e importación de modelos

2. Índice del reporte.

3. Resumen.

El proceso de exportación de modelos tridimensionales desde 3D Studio Max a Unity se realiza a través de un formato de intercambio, el formato FBX[1], éste formato permite la transferencia de geometrías, texturas, animaciones, materiales simples y aunque posee la opción de exportar cámaras y luces estas no son reconocidas por el motor de juegos de Unity 3D.

Para realizar la exportación de los modelos estos deben tener aplicado un material estándar y pueden usar texturas de difusión, transparencia por alfas y de bump (esta última deberá ser normal map o heightmap), en el caso de mapas de especularidad, y de reflejo estos pueden ser incluidos en la exportación a FBX pero deben ser asignados dentro del engine, no es posible importar cámaras y luces desde archivos FBX estas deben crearse directamente en el engine, en el caso de las animaciones simples y esqueletales pueden estas ser exportadas a FBX e importadas a Unity sin mayor problema, aunque en el caso de la animación esqueletal son necesarios algunos requisitos[2] .

4. Introducción.

Debido a la necesidad de uso de software comercial que acelere la creación de ambientes virtuales se ha optado por el uso del Engine Unity 3D, en su versión libre, la razón preponderante en el uso de Unity es su facilidad de uso, el costo que no es tan elevado y las capacidades que el engine brinda, puede incorporar contenidos gráficos y audiovisuales de una manera eficiente ahorrando mucho tiempo de programación, además de que esta resulta fácil al incorporar los lenguaje de alto nivel C#, Java Script y Boo, este último es una variante de Phyton.

El uso de Unity abre las interrogantes sobre los formatos de modelos, texturas y demás contenido gráfico y audiovisual que el engine soporta, creando las preguntas ¿cuáles son los formatos compatibles?, ¿cómo se exportan? y ¿Cómo se importan? Y en base a que se verifica su compatibilidad, respondiendo a esto se plantea dentro del documento una serie de verificaciones rutinarias sobre los formatos más habituales que permiten la incorporación de contenido desde una aplicación o aplicaciones externas.

Basado en los resultados que arrojaran las verificaciones se establecerán los formatos compatibles y las limitaciones que estos presenten al momento de ser incorporados dentro del engine, verificando la

[1] FBX (Franco Serrano, Compatibilidad del formato FBX, 2009)
[2] Revisar requisitos en la sección discusión de resultados de este documento

transferencia de geometría, textura, materiales, iluminación, cámaras, animación y demás elementos particulares que sean de interés de los desarrolladores de aplicaciones tridimensionales para tiempo real.

5. Marco teórico.

Unity 3D es un motor de juegos que emplea las librerías gráficas de OpenGL y de DirectX que permite generar aplicaciones tridimensionales interactivas de tipo serio y también de tipo lúdico, dentro de las capacidades de este engines se encuentran el manejo de assets, que pueden ser geometrías, texturas, animaciones, materiales simples, entre otros, estos assets permiten la inclusión de contenido gráfico tridimensional, que es la materia de estudio de este documento.

Para lograr entonces la inclusión de estos assets del tipo gráfico debemos emplear el formato FBX, este formato, originalmente desarrollado por la empresa Kaydara, es en la actualidad[3] un estándar en la industria del entretenimiento digital y es usado como archivo de transferencia de contenido entre los programas más usuales de modelado tridimensional, Motion Builder, 3ds Max, Maya, Softimage, entre otros[4].

6. Parte Experimental - Método.

Las aplicaciones tridimensionales para tiempo real hacen uso por lo regular de contenido gráfico optimizado y contenido audiovisual estandarizado, además de programación de rutinas, inteligencia artificial, efectos gráficos y optimizadores de cámara, geometría y textura, para que la ejecución del sistema sea lo más fluido posible. Siendo pertinente aclarar que en el presente documento solo se analizaran y verificaran los niveles de contenido gráfico optimizado, contenido audiovisual y de optimizaciones.

Dentro de este documento se llevara a cabo la descripción del proceso de exportación de modelos a FBX, por ende el uso de la aplicación Unity 3D deberá ser aprendido por algún otro medio[5] al igual que la aplicación de exportación, sea esta 3DS max[6] o alguna otra aplicación compatible[7], es el caso de Maya, Blender, Cheetah 3D, Carrara, Lightwave, XSI, SketchUp Pro o Wings 3D, así como también algún editor de imágenes como Photoshop, Gimp, Pixlr o similares.

Revisión de las posibilidades de importación de contenido gráfico dentro de Unity 3D

Basado en las tablas de referencia de unity, que se presentan más abajo, se verificara la exportación de contenido gráfico desde la aplicación 3DS Max a través del formato de intercambio FBX, además se revisaran las opciones de exportación de materiales y texturas, animación y algunos otros elementos.

---

[3] Mayo 2011
[4] Modo,
[5] Se recomienda el tutorial Unity 3D (Luna Luz, 2011)
[6] Se recomienda el documento introducción al modelado tridimensional para tiempo real, (Franco Serrano, introducción al modelado tridimensional para tiempo real, 2011)
[7] Consultado en línea (Unity 3D, 2011)

No se realizara verificación de archivos de audio y video ni de formatos adicionales de importación.

Dentro de la documentación de la aplicación Unity 3D[8] se muestran la siguiente tabla que nos servirá para entender todas las posibilidades de importación y también a determinar las pertinentes a este documento.

3D Package Support

	Meshes	Textures	Anims	Bones
Maya .mb & .ma[1]	*	*	*	*
3D Studio Max .max[1]	*	*	*	*
Cheetah 3D .jas[1]	*	*	*	*
Cinema 4D .c4d[1 3]	*	*	*	*
Blender .blend[1]	*	*	*	*
modo .lxo[2]	*	*	*	*
Autodesk FBX	*	*	*	*
COLLADA	*	*	*	*
Carrara[1]	*	*	*	*
Lightwave[1]	*	*	*	*
XSI 5.x[1]	*	*	*	*
SketchUp Pro[1]	*	*		
Wings 3D[1]	*	*		
3D Studio .3ds	*			
Wavefront .obj	*			
Drawing Interchange Files .dxf	*			

Fuente: Unity 3d

[1] Import uses the application's FBX exporter. Unity then reads the FBX file.
[2] Import uses the application's COLLADA exporter. Unity then reads the COLLADA file.
[3] Cinema4D 10 has a buggy FBX exporter.

De igual manera en la documentación de Unity 3D[7] nos muestra la información de compatibilidad con los diversos medios audiovisuales

Image Formats

Photoshop .psd and .tiff are imported with layers automatically flattened.
JPEG, PNG, GIF, BMP, TGA, IFF, PICT and many other image formats are supported.

Supported Audio & Video Formats

MP3 and Ogg Vorbis .ogg audio files are natively supported, depending on platform. On mobile platforms, audio is converted to MP3 to take full use of hardware decompression.
AIFF, WAV and most other audio format are supported, ideally suited for sound effects. Optional compression can be configured in the Unity Editor.
MOD, IT, S3M, XM tracker files are fully supported.
Ogg Theora video is natively supported.
Video MOV, AVI, ASF, MPG, MPEG, MP4VIDEO files are transcoded by Unity with a configurable bitrate.

Fuente: Unity 3d

Other File Formats

XML and text files with .xml and .txt extensions can be referenced at runtime.
Any other file types, such as RTF and DOC, can be used for project notes and to-do lists.

[8] Documentación Unity 3D (Unity 3D, 2011)

Análisis de las opciones de exportación al formato FBX

Siendo la materia de estudio el contenido gráfico no podemos dejar de lado el proceso en el que este se prepara para su exportación[9] y para ello 3ds Max será la aplicación que será empleada dentro de la parte experimental del documento para realizar la exportación del contenido gráfico al formato FBX para, posteriormente, importarlo dentro del engine de juegos Unity 3D.

En la siguiente imagen se muestran las opciones de exportación del formato FBX[10] y del lado derecho se describen brevemente.

FBX

A. Dentro de la sección de Geometry podremos activar o desactivar la exportación de grupos de suavizado, las tangentes y binormales, al subdivisor turbosmooth, manejo de instancias, selection sets entre otras es recomendable no exportar turbosmooth para tiempo real.

B. Dentro de la sección de Animation podemos activar o desactivar la exportación de la secuencia animada, además podremos hornear la animación (esta operación es recomendable) que realiza una conversión entre las curvas de animación y las integra al archivo FBX.

Aquí se especifica si queremos exportar skin y morphs para personajes animados a través de esqueletos y algunas opciones de optimización y conversión de curvas de animación.

[9] Exportación (Franco Serrano, Preparación y exportación de modelos y escenarios para tiempo real, 2011)
[10] **Autodesk 3ds Max FBX Plug-in Guide** (Autodesk, 2011)

C. Dentro de la sección Cameras activaremos o inactivaremos la exportación de las cámaras que estén definidas en la escena.

D. Dentro de la sección Lights activaremos o inactivaremos la exportación de las luces que estén definidas en la escena, recordemos que FBX solo exporta luces simples todas las demás serán convertidas al formato de las luces simples.

E. Dentro de la sección Embebed Media activaremos o inactivaremos la integración de las texturas que estén definidas en los objetos, si esta opción se deja deseleccionada las texturas estarán referenciadas por lo que deberán acompañar en carpeta al archivo FBX.

F. Dentro de la sección Units se seleccionara la conversión al sistema de unidades del sistema inglés o del sistema decimal, si se han trabajado los modelos con una conversión y equivalencia adecuada es recomendable dejar la opción Automatic activada.

G. Dentro de la sección Axis Conversion se seleccionara el eje que apunta hacia arriba (UP axis) pues cada programa define como es el manejo de sus ejes coordenados, por ejemplo 3ds max maneja Z como UP, DirectX maneja Y como up igual que maya y Unity maneja Y como eje superior, se recomienda dejar el eje Y como eje superior al exportar a FBX.

H. Dentro de la sección de FBX File Format se puede seleccionar el tipo de codificación del archivo si es binaria o ASCII, siendo binario un formato más veloz en su lectura, las versiones de FBX permiten manejar rangos de compatibilidad conversiones anteriores de **3DS MAX** o maya pues si ambos programas tiene instalados los exportadores.

Verificaciones estándar de compatibilidad

Las verificaciones estándar de compatibilidad[11] son una serie de pruebas que serán corridas a los diferentes contenidos que son usados dentro de las aplicaciones tridimensionales para tiempo real y permiten la verificación de elementos esenciales usados dentro del pipeline de producción de estas y permite entender que problemáticas encontrará el usuario al hacer uso de los formatos de intercambio, la exportación de los mismo y su importación dentro de los engines o librerías graficas usadas.

Nivel geométrico

Procedimiento aplicado a cada tipo de geometría:

1. Creación del tipo de geometría
    a. Primitivas
    b. Splines
    c. También llamadas líneas o paths
    d. Polígonos
    e. NURBS
    f. Patch
    g. Deformadores de geometría
    h. Geometría compuesta
    i. Meta-objetos
2. Creación de modificadores adicionales en caso de que sea requerido
3. Preparación para la exportación
4. Exportación al formato FBX
5. Revisión de problemáticas de exportación
6. Importación dentro del engine Unity 3D
7. Revisión de las geometrías dentro del engine y en la organización interna de las carpetas
8. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
9. Análisis de datos y planteamiento de conclusiones

Nivel textura

Procedimiento aplicado a cada textura:

1. Revisión de estándares de uso[12] de textura en aplicaciones tridimensionales para tiempo real

[11] Verificaciones estándar basadas en el documento Delimitación de estándares de uso (Franco Serrano, Delimitación de estándares de uso de contenidos gráficos y multimediales en aplicaciones tridimensionales para tiempo real, 2011)
[12] Delimitación de estándares de uso (Franco Serrano, Delimitación de estándares de uso de contenidos gráficos y multimediales en aplicaciones tridimensionales para tiempo real, 2011)

   a. Potencia de 2
   b. Formatos
      i. Para transparencia
      ii. Para reflejo
      iii. Para relieve
      iv. Para especularidad
   c. Tipos de archivo
   d. Manejo de transparencia por textura
   e. Texturas tile o clamp
   f. Manejo de coordenadas de textura
   g. Multicanales, multi-coordenadas y multi-texturas
2. Asignación de texturas a un material simple
3. Asignación de material simple a una primitiva básica (esfera)
4. Creación de geometría con ID´s de multimaterial
5. Creación de material Multi/subobjeto y Multi-texturas
6. Asignación de multi-materiales, multi-coordenadas y multi-texturas a un objeto 3d
7. Preparación para la exportación
8. Exportación al formato FBX
9. Revisión de problemáticas de exportación
10. Importación dentro del engine Unity 3D
11. Revisión de las texturas dentro del engine y en la organización interna de las carpetas
12. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
13. Análisis de datos y planteamiento de conclusiones

Nivel material

Procedimiento aplicado a cada tipo de material:

1. Revisión de estándares de uso[12] de materiales en aplicaciones tridimensionales para tiempo real
   a. Blending
   b. Composite
   c. Cubemapping
   d. Environment mapping
   e. Baking
   f. Procedural
   g. Vertex color
   h. Multi/Sub-objeto
   i. Transparencias
      i. Por alfas
      ii. Por material

   j. Shaders
2. Asignación de diferentes tipos de materiales a una primitiva básica (esfera)
3. Creación de coordenadas de textura
4. Implementación de texturas adicionales a un material simple
5. Preparación para la exportación
6. Exportación al formato FBX
7. Revisión de problemáticas de exportación
8. Importación dentro del engine Unity 3D
9. Revisión de las texturas y materiales dentro del engine
10. Revisión de la organización interna de las carpetas de materiales
11. Preparación y corrección de materiales dentro del engine
12. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
13. Análisis de datos y planteamiento de conclusiones

Nivel luces

1. Revisión de estándares de uso[12] de luces en aplicaciones tridimensionales para tiempo real
   a. Luz omnidireccional
   b. Luz direccional
   c. Luz dirigida tipo spot
   d. Luces de otro tipos
2. Iluminación de varios objetos primitivos y un piso
3. Preparación para la exportación
4. Exportación al formato FBX
5. Revisión de problemáticas de exportación
6. Importación dentro del engine Unity 3D
7. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
8. Análisis de datos y planteamiento de conclusiones

Nivel cámaras

1. Revisión de estándares de uso[12] de cámaras en aplicaciones tridimensionales para tiempo real
   a. Cámara con objetivo
   b. Cámara libre
2. Creación de las cámaras
3. Exportación al formato FBX
4. Revisión de problemáticas de exportación
5. Importación dentro del engine Unity 3D
6. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
7. Análisis de datos y planteamiento de conclusiones

Nivel animación simple

1. Creación de animación simple con primitivas básicas
    a. Uso del motor de física Reactor
    b. Conversión a animación simple
2. Exportación a FBX
3. Revisión de problemáticas de exportación
4. Importación dentro del engine Unity 3D
5. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
6. Análisis de datos y planteamiento de conclusiones

Nivel animación esqueletal

1. Uso de personaje base human con animación esqueletal
2. Revisión de pesos asignados al skinning de la geometría
3. Reparación de problemas
4. Exportación a FBX
5. Revisión de problemáticas de exportación
6. Importación dentro del engine Unity 3D
7. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
8. Análisis de datos y planteamiento de conclusiones

Nivel Partículas

1. Revisión de estándares de uso[12] de partículas en aplicaciones tridimensionales para tiempo real

    a. Particle Flow
    b. Snow
    c. Particle Array
    d. Super Spray
    e. Spray
    f. Blizzard
    g. Particle Cloud

2. Creación de sistema de partículas
3. Exportación a FBX
4. Revisión de problemáticas de exportación
5. Importación dentro del engine Unity 3D
6. Recopilación de datos útiles para el uso de archivos FBX en Unity3D

7. Análisis de datos y planteamiento de conclusiones

Optimizadores y su uso en Unity

1. Revisión de estándares de optimización de escenas[12] para tiempo real en Unity
    a. Billboard
    b. Visibility Culling[13]

    Cortar lo que no se ve, existen cuatro casos:

        i. Occlussion Culling - Caras ocultas detras de otros objetos
        ii. Viewfustrum culling - Aquello que esta fuera del cono de visión
        iii. Backface Culling - Caras que no ve el observador
        iv. Contribution Culling - Aquellos que son tan pequeños que no contribuyen a la imagen

    c. Places y Visibility Groups
    d. Level Of Detail LOD
    e. Cajas de colisión y sistema de física
    f. Horneado de textura y de sombras
    g. Shaders
    h. Espacialización de geometrías
        i. Arreglo ineficiente de nodos
        ii. Arreglo eficiente de nodos
2. Creación o implementación de los optimizadores en Unity o 3DS Max
3. Revisión de posibilidades de exportación a FBX de optimizadores de escena
4. Exportación de estas en FBX
5. Revisión de problemáticas de exportación
6. Importación dentro del engine Unity 3D
7. Recopilación de datos útiles para el uso de archivos FBX en Unity3D
8. Análisis de datos y planteamiento de conclusiones

7. Resultados y discusión de Resultados.

Nivel Geométrico

En el caso particular de las geometrías se encontró que todo objeto geométrico que es importado a Unity es creado y convertido a polígonos[14] y solo por eso, no es posible exportar primitivas, ni splines, ni

[13] Manejo de escenas (Ramos Nava, 2006)

NURBS, ni geometría Patch o geometría compuesta, meta-objetos o deformadores activos, todos estos tipos de geometría son convertidos a polígonos cuando son exportados a FBX.

Tabla de exportación de geometría

Tipo de geometría	Limitaciones
Primitivas	FBX no soporta primitivas, al exportarse deben convertirse a mesh por ende Unity nunca importara una primitiva desde un paquete externo, se debe tomar en cuenta que al convertir las primitivas se pierde toda asignación de Skin y de Morphs. Si se requiere de una primitiva básica se recomienda crearla directamente en Unity
Splines	No soporta la exportación de splines estos se convierten en unity a objetos vacíos, como nota adicional existe una extensión comercial llamada Rage Spline[15] que permite el manejo de gráficos bidimensionales en Unity.
Polígonos	La exportación de polígonos, al exportarse convierte los quads en triángulos y puede llegar a tener problemas con la dirección de los ejes volteando algunas normales
NURBS	FBX no soporta curvas NURBS deben convertirse a mesh por ende Unity nunca importara una NURBS desde un paquete externo, se debe tomar en cuenta que al convertir la curva NURBS se pierde toda asignación de Skin y de Morphs.
Patch	FBX no soporta geometría PATCH esta debe convertirse a Triangle mesh por ende Unity nunca importara una geometría PATCH desde un paquete externo, se debe tomar en cuenta que al convertir el patch se pierde toda asignación de Skin y de Morphs.
Deformadores de	FBX no soporta deformadores de geometría esta debe convertirse a mesh por ende Unity nunca importara una deformador de geometría desde un paquete externo, se debe tomar en cuenta que al convertir el objeto deformado se pierde

[14] Se sobreentiende que un modelo está constituido por polígonos, ejes y vértices
[15] Rage Spline de Juha Kiili http://juhakiili.com/blog/

geometría	toda asignación de Skin y de Morphs. Como nota adicional existe una aplicación comercial llamada Mega Fiers[16] que permite usar los deformadores de geometría de forma similar a la usada en 3DS Max Bend, Twist, FFD, Displace, Taper entre otras.
Geometría compuesta	En el caso de la geometría compuesta que por definición emplea 2 o más objetos geométricos de clases similares, sean primitivas, mesh o poly, el resultado de exportación ser la conversión de la geometría compuesta en una geometría única, por ejemplo en el caso de los booleanos de sustracción[17] se conservara el resultado final de la operación perdiendo los objetos individuales que lo componían, de igual forma en una operación compuesta de loft[18] se pierde la generatriz y directriz pero se exporta el resultado de esta operación, sería similar a convertir en 3DS Max el objeto compuesto a geometría tipo mesh o tipo poly, se debe tomar en cuenta que al convertir el objeto deformado se pierde toda asignación de Skin y de Morphs.
Meta-objetos	FBX no soporta meta objetos, durante la exportación a FBX deben convertirse a mesh, por ende Unity nunca importara un meta-objeto desde un paquete externo, se debe tomar en cuenta que al convertir el objeto deformado se pierde toda asignación de Skin y de Morphs.

Nivel textura

Unity conserva todos los tipos usuales de textura asignada a materiales estándar, pero no los asigna de forma automática, la única textura asignada de manera automática es la de difusión, esta contiene a su vez el canal alfa, los demás se deben asignar manualmente, tal es el caso de normal maps, environment maps, heightmasp entre otros.

Toda textura usada en objetos que serán usadas en aplicaciones para tiempo real deben ser en potencias de 2 sin importar el engine usado, OSG, Unity o Virtools , esta medida refiere al ancho y alto en pixeles y se mide usualmente en valores que van de 32, 64, 128, 256, 512, 1024 y 2048, texturas menores y mayores pueden existir pero no se recomienda solo en casos particulares, también es posible combinar potencias de 2 por ejemplo que mida 512 de alto y 256 de ancho pero siempre conservando la proporción de la potencia de 2.

[16] Mega-Fiers de Chris West http://u3d.as/content/chris-west/mega-fiers/1Qa
[17] Booleanos de sustracción (Franco Serrano, introducción al modelado tridimensional para tiempo real, 2011)
[18] LOFT (Franco Serrano, introducción al modelado tridimensional para tiempo real, 2011)

Los formatos soportados por Unity son Photoshop[tm] (psd) sin layers, tiff sin layers, JPEG, PNG, GIF, BMP, TGA, IFF, PICT entre otros.

Las escalas recomendables para texturas son 256, 512 y 1024, es posible usar escalas mayores y menores desde 32 hasta 4096, el uso de mipmaps es automatico al importar las texturas, esta opción puede deshabilitarse en cualquier momento, la optimización de texturas en unity se maneja con compresiones de DirectX existen los tipo DXT5 y DXT1, también es posible manejar texturtas sin compresión de direct x y se pueden manejar de 16 y de 24 bits y con alfa de 32 bits.

En el caso de la Transparencia por alfas se logra integrando el canal alfa en la textura de difusión y se debe tomar en cuenta que la especularidad es exportada a través del canal alfa de la textura de difusión, por ende no pueden convivir una textura con especularidad con una con transparencia por alfas se debe elegir una.

El relieve tipo bump se logra a través de un mapa de normales y puede mejorarse con un heightmap y un shader de parallax mapping, todos estos mapas deben definirse desde 3ds max, las texturas puedes ser de tipo tile o clamp al igual que las coordenadas de textura.

En el caso de multi-materiales, multi-coordenadas y multi-texturas al exportarse se pierden los multicanales y las múltiples coordenadas de textura, cabe mencionar que para manejar canales adicionales para cálculo de lightmapping y ambient occlussion estos deberán hacerse dentro del engine de unity.

Nivel material

Se encontró que Unity solo soporta la exportación de materiales simples, propiamente esto es porque el formato de intercambio, FBX, convierte todos los materiales en materiales simples, es posible exportar un objeto geométrico con zonas definidas por ID´s que hagan uso de multi-materiales, aunque todos estos serán estándar, los materiales que hagan uso de reflejo, bump o transparencia pueden mantener las texturas que los definen tomando en cuenta que deben realizarse algunas modificaciones pertinentes dentro del engine de Unity.

Tipo de material	Limitaciones
Blendig	No soporta blending directo de la aplicación, este debe ser puesto ya en el engine
Composite	No soporta al material composite lo que sucede es que este material y todos sus sub-materiales se pierden y son reemplazados por un material simple

Environment mapping	Es soportada la exportación del environment map pero este debe ser modificado de forma manual en unity 3D para asignarle la característica de environment map
Cubemapping	Para la opción de cube mapping existen dos soluciones, la primer es usar un environment mapping y convertirlo a Cube Map a través de unity, (consultar la sección anexos para más información), para posteriormente usarlo como mapa de reflejo, el segundo método es a través de caras separadas de un cubemap estas se importan de forma manual en Unity y se debe generar un mapa tipo cubemap y posteriormente asignarles a cada cara la textura correspondiente, Unity o soporta archivos dds con cubemaps por tira o por cruz por esto solo se puede usar un environment o caras separadas previamente.
Baking	Es posible realizar la exportación de geometría texturizada usando mapas pre-calculados como si fueran texturas simples manejando solo un canal y una sola coordenada de textura por modelo.
Procedural	Los materiales procedurales de 3DS Max no pueden pasar directamente al formato FBX y por ende deben ser pre-calculados como mapas simples o construidos a través de shaders dentro del engine.
Vertex color	No soporta la exportación de colores por vértice, estas no pueden ser convertidas al formato FBX y no pueden ser importados en Unity
Multi/Sub-objeto	Soporta multi-materiales con todos los canales disponibles
Transparencias	Es posible tener transparencia por alfas estas deben estar integradas en la textura de difusión o pueden ser generadas dentro de Unity lo cual no es muy recomendable. No es posible tener transparencias por material de forma directa al exportar, esta debe ser programada en un shader particular dentro de Unity
Shaders	No es posible exportar shaders a través del formato FBX, tampoco es posible exportar e importar shaders generados a través de 3DS Max.[19]

[19] Shaders usando Shader FX (Franco Serrano, Aplicación Shader FX, 2008)

Nivel luces y nivel cámaras

Luces	Todas las luces son convertidas en objetos vacíos (empty objects) cuando son importados a Unity.
Cámaras	Todas las cámaras son convertidas en objetos vacíos en Unity

Como dato adicional en FBX las luces de tipo diferente a las simples son convertidas en luces puntuales, tal es el caso de la luz tipo sky y la luz de mental ray tipo área omni, también son convertidas a luces tipo spot simple como en el caso de la luz de mental ray tipo spot.

Nivel animación simple

La animación simple es exportada de manera eficiente usando el formato FBX y es importada de igual manera en Unity, debe recordarse que la animación creada a través de modificadores no puede exportarse usando FBX, pues esta última no es soportada.

Nivel animación esqueletal

Debemos manejar siempre animaciones con un frame-rate de 30 cuadros por segundo, (en maya el default es 24 por ende deberá ser cambiado antes de animar), usar el modificador Skin (solo para 3ds max) es esencial pues con el modificador Physique puede haber muchos problemas en la conversión, además de considerar que los modificadores usuales aplicables a las articulaciones en el modificador skin no pasaran a FBX por lo que es recomendable manejar a conciencia la distribución y asignación de pesos en los vértices.

Es necesario agregar al inicio de las animaciones cinco cuadros con nuestro personaje en posición T pues este puede ser ocupado por el ragdoll para simular muertes o física aplicada a un cuerpo inerte modificado por la gravedad y obstáculos sólidos.

Para realizar la exportación de un modelo tridimensional que haga uso de animación esqueletal debemos tomar en cuenta el horneado o Baking de la animación, para que las curvas de animación se conviertan en cuadros de animación, por esto mismo es muy importante que al exportar desde 3ds max seleccionemos la opción bake animation, pero en maya deberemos realizar el horneado de animaciones desde el mismo maya, además de estar seguros de que se eliminó la historia antes de exportar.

No debe existir ningún modificador arriba del skin pues este se perderá y podría modificar la distribución de pesos en el skinning.

Nivel Partículas

No es posible exportar sistemas de partículas en FBX estos son convertidos en objetos vacios Empty Objects, los sistemas de partículas deberán ser creados dentro del engine de unity.

Optimizadores en Unity

Billboard

No es posible exportar billboards directamente desde 3ds max usando FBX, pero es posible exportar los planos que serán billboards para que posteriormente en código se puedan usar estos como billboards, por desgracia el código debe asignarse por objeto y además deberá seleccionarse la cámara, digamos si tienes 1000 billboard a cada uno deberas asignarle el script y la cámara.

Es posible crear billboards en Unity usando el siguiente código[20] de C#

```
using UnityEngine;
 using System.Collections;

 public class LookAtCameraYonly : MonoBehaviour
 {
   public Camera cameraToLookAt;

   void Update()
   {
      Vector3 v = cameraToLookAt.transform.position - transform.position;
      v.x = v.z = 0.0f;
      transform.LookAt(cameraToLookAt.transform.position - v);
   }
 }
```

Visibility Culling[21]

Delimitando el concepto de que Visibility culling corta lo que no se ve para evitar el overdrawing o sobredibujado de objetos de manera excesiva y por ende poco optima y existen cuatro casos posibles:

- Occlussion Culling - Caras ocultas detrás de otros objetos

[20] Look at camera Y only code (Carter, 2009)
[21] Manejo de escenas (Ramos Nava, 2006)

- View-fustrum culling - Aquello que esta fuera del cono de visión
- Backface Culling - Caras que no ve el observador
- Contribution Culling - Aquellos que son tan pequeños que no contribuyen a la imagen

Basado en la experimentación de uso de FBX a Unity se encontró que no es posible exportar ningún tipo de Visibility culling por lo que este deberá realizarse dentro de Unity, cabe mencionar que al exportar a FBX podemos apoyar la creación de los optimizadores, por ejemplo en el View Fustrum Culling los objetos deben estar separados si existe mucha distancia entre ellos pues si son un solo objeto este no es bien calculado por el view frustrum culling.

Se realizaron las pruebas de cada una en Unity encontrando que:

Occlussion Culling

Es posible realizar la optimización de Occlussion Culling dentro de Unity[22], se debe definir la geometría como estática y luego en la ventana de Occlussion Culling se deben modificar algunos valores para posteriormente crear las zonas de oclusión y luego pre-calcular estas y guardarlas en un archivo independiente.

View-fustrum culling

El view frustrum culling viene incorporado en todas las cámaras de unity y sus valores pueden ser modificados manualmente por el usuario definiendo en donde inicia el cono de proyección y en donde es cortado por el plano de corte.

Para realizar eficientemente el fustrum culling en unity los objetos deberán estar separados, pueden estos estar jerarquizados pero deben ser reconocidos por unity como objetos independientes, por ende es recomendable separar las geometrías que ocupen mucho espacio en unidades, por ejemplo dos edificios que se encuentren lejanos, sucede lo mismo con el cálculo de luz con deferred lighting, este cálculo es menos optimo si los objetos están muy separados es mejor dividirlos en dos objetos.

Backface Culling

Todos los shaders de Unity, exceptuando los que se usan para la vegetación, tienen activa la prueba de Backface culling por lo que el cálculo de la cara inversa de la normal de los polígonos no se representara, para cambiar esta optimización deberá ser editado o creado un shader que contenga una prueba nula de culling por caras, pues si se cambia el backface culling por front face se obtendrá el resultado inverso sin calcularse alguna de las caras.

Contribution Culling

No existe una implementación directa de este algoritmo dentro de unity es necesario programarlo.

[22] Occlussion Culling (Unity Technologies, 2010)

Per-layer cull

Existe en unity un sistema por capas de culling este es llamado per-layer cull y permite separar por capas a los objetos que constituyen una escena esto con el objetivo de que la distancia de corte de estos objetos en relación con la cámara varie al del plano de corte de esta última, haciendo que objetos de un mismo layer que desaparecerían a 1000 unidades puedan desaparecer a una distancia menor y preservando a los objetos que no estén en este layer.

Places y Visibility Groups

El equivalente en unity a estos algoritmos de optimizacion es la implementacion de occlussion culling en las escenas de unity.

Level Of Detail LOD

No es posible exportar niveles de detalle desde 3ds max a unity haciendo uso de FBX, por lo que se recomienda el uso de UNILOD[23] dentro de unity esta extensión de unity permite la gestión de niveles de detalle dinámicos que en la versión pro de unity permiten la generación automática de estos, en la versión indie de unity no es posible la simplificación automática de las mallas.

Cajas de colisión y sistema de física

Es posible definir geometría invisible como cajas de colisión dentro de unity, estas cajas pueden exportarse desde programas de modelado, aunque también es posible asignarles geometría de colisión directamente en unity sin crearla en algún software de modelado, para ello se deberá seleccionar el objeto y luego definir el tipo de colisión que se requiera, si es poco compleja se elegirá entonces la de caja aunque si se necesita más precisión se elegirá la de esfera y así sucesivamente hasta llegar a un nivel alto de detalle de la geometría de colisión, la recomendación es manejar siempre geometría de colisión de poca complejidad para que los cálculos de colisión no sea exagerados.

La física en tiempo real será implementada directamente en Unity, pues no es posible exportar modelos con física asignada desde 3DS Max.

Horneado de textura y de sombras

Es posible exportar el modelo FBX con mapas horneados pero estos deben estar en completemap o en su defecto tener el lightmap separado para que este se use en unity 3d, además de esto es posible generar mapas de luces dentro de unity dándole un plus en la versión pro pues esta versión puede pre-calcular iluminación global y generar mapas de manera automática definiendo su calidad y la

[23] UNILOD http://forum.unity3d.com/threads/42524-UniLOD-BETA-Level-of-detail-and-streaming-support-UPDATE

complejidad y detalle de la proyección geométrica automática por lo que no es necesario precalcular desde el programa de modelado.

8. Conclusiones y Recomendaciones.

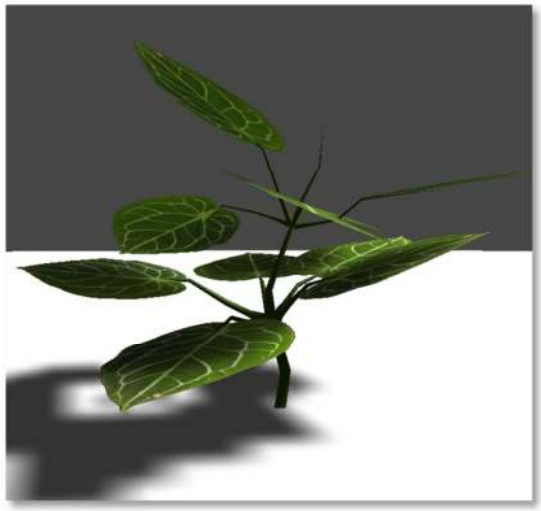
Escala

La escala es fundamental pues Unity maneja la escala en equivalencia a centímetros por ejemplo 1 unidad equivaldría a 0.01metros dando como escala que una unidad equivale a 1 centímetro, es fundamental que los modelos sean exportados de esta manera al formato FBX por lo que debemos seleccionar siempre la escala de forma manual y elegir centímetros en todo momento, caso diferente es el de la aplicación 3d en la que realicemos los modelos, en las aplicaciones 3d podemos manejar la equivalencia usual que es 1 unidad equivale a 1 metro pero siempre recordar exportar en centímetros.

Materiales de doble lado

Para manejar materiales de doble lado se recomienda que un programador modifique los shaders pertinentes, esto con el fin de calcular ambas normales de la superficie de un objeto y que este se represente por ambos lados, por ejemplo en una especie vegetal en donde las hojas deberán ser representadas por ambos lados, este paso es necesario si se quiere conservar la optimización de los escenarios y modelos a través de la sustitución de geometría con textura.

Geometría con cálculo de doble normal



Geometría con cálculo de una sola normal

9. Bibliografía.

Autodesk. (2006). *Autodesk FBX* . Recuperado el 2009, de http://www.autodesk.com/fbx

Autodesk. (Noviembre de 2007). *Interoperability and FBX Technology.* Recuperado el 2009, de http://images.autodesk.com/adsk/files/fbx_whitepaper_nov2007.pdf

Autodesk. (November de 2007). *Maya FBX Plug-in Guide.* Recuperado el 2009, de White Paper: http://www.kxcad.net/autodesk/maya/Maya_Documentation/Maya_Plug-in_200611.pdf

Autodesk. (2011). *Export*. Recuperado el 09 de 06 de 2011, de Autodesk 3ds Max FBX Plug-in Guide: http://download.autodesk.com/us/fbx/2012/3dsmax/index.html

Autodesk, Inc. (2006). *3ds Max FBX Plug-in Guide.* Recuperado el 2006, de http://www.kxcad.net/autodesk/3ds_max/3ds_Max_FBX_Plug-in_Guide.pdf

Carter, N. (19 de 10 de 2009). *LookAtCameraYonly.* Recuperado el 14 de 06 de 2011, de unify community: http://www.unifycommunity.com/wiki/index.php?title=LookAtCameraYonly

Franco Serrano, V. H. (2008). *Aplicación Shader FX.* Recuperado el 13 de 06 de 2011, de Departamento de Realidad Virtual: http://sistemas.tic.unam.mx/images/stories/realidadvirtual/doc_acad/manuales/shaderfx.pdf

Franco Serrano, V. H. (Octubre de 2009). *Compatibilidad del formato FBX.* Recuperado el 26 de Abril de 2011, de Sistemas y servicios institucionales DGTIC UNAM: http://sistemas.tic.unam.mx/images/stories/realidadvirtual/doc_acad/reportes/fbx.pdf

Franco Serrano, V. H. (2009). *Método de produccion de contenido gráfico para tiempo real.* Recuperado el 2009, de http://www.ixtli.unam.mx/tutoriales/produccion_contenido_grafico/produccion_contenido_grafico.pdf

Franco Serrano, V. H. (06 de 2011). *Delimitación de estándares de uso de contenidos gráficos y multimediales en aplicaciones tridimensionales para tiempo real.* Recuperado el 06 de 2011, de Realidad Virtual: http://sistemas.tic.unam.mx/es/realidadvirtual/documentos-academicos/

Franco Serrano, V. H. (01 de 02 de 2011). *introducción al modelado tridimensional para tiempo real.* Recuperado el 09 de 06 de 2011, de Departamento de Realidad Virtual: http://sistemas.tic.unam.mx/images/stories/realidadvirtual/doc_acad/cursos/intro3d.pdf

Franco Serrano, V. H. (01 de 02 de 2011). *Preparación y exportación de modelos y escenarios para tiempo real.* Recuperado el 09 de 06 de 2011, de Cursos: http://sistemas.tic.unam.mx/es/realidadvirtual/documentos-academicos/402-cursos

Luna Luz, F. (07 de 06 de 2011). *Intoducción a Unity 3D.* Recuperado el 07 de 06 de 2011, de Departamento de realidad virtual: http://sistemas.tic.unam.mx/images/stories/realidadvirtual/doc_acad/tutoriales/unity.pdf

Ramos Nava, M. d. (15 de 05 de 2006). *Manejo de escenas.* Recuperado el 14 de 06 de 2011, de Realidad virtual: http://sistemas.tic.unam.mx/images/stories/realidadvirtual/doc_acad/cursos/manejo_escenas.pdf

Unity 3D. (2011). *Asset Importing*. Recuperado el 9 de Junio de 2011, de Unity 3D Editor: http://unity3d.com/unity/editor/importing

Unity Technologies. (24 de 09 de 2010). *Occlusion Culling.* Recuperado el 14 de 06 de 2011, de Unity Manual: http://unity3d.com/support/documentation/Manual/Occlusion%20Culling.html

Wikimedia Foundation. (2009). *FBX*. Recuperado el 2009, de http://en.wikipedia.org/wiki/FBX

10. Apéndices.

- Donde conseguir las versiones del exportador para 3D Studio Max, Maya y Blender

En la dirección electrónica de la compañía Autodesk:
http://usa.autodesk.com/adsk/servlet/pc/item?siteID=123112&id=10775855
Es posible encontrar los plugins de exportación, conversión y visualización del formato FBX para 3D Studio Max y Maya en Windows y además para Maya en MAC y LINUX aunque las aplicaciones poseen ya alguna versión preinstalada del exportado de FBX, al igual que Blender que ya trae un exportador integrado.

Instalación de plug-in de exportación FBX

Los exportadores del formato FBX para 3DS MAX y Maya se encuentran en la dirección electrónica:
http://usa.autodesk.com/adsk/servlet/pc/index?id=6837478&siteID=123112

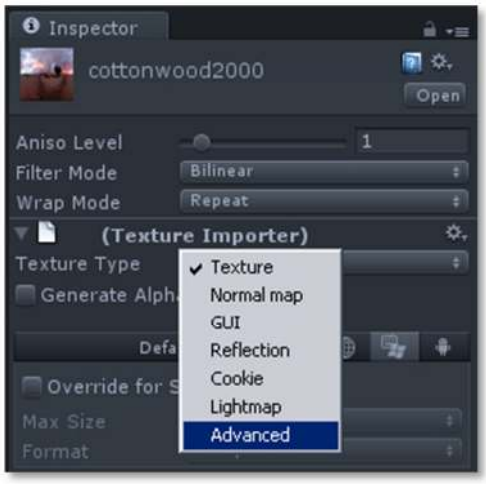
Existen varias versiones del formato FBX y estas también dependen de que versión de 3DS MAX o maya estemos trabajando, por lo que para elegir una versión debemos estar seguros de que esta existe en ambas aplicaciones o que es compatible entre ellas, por ejemplo intercambio de modelos entre maya y max o entre max y unity.

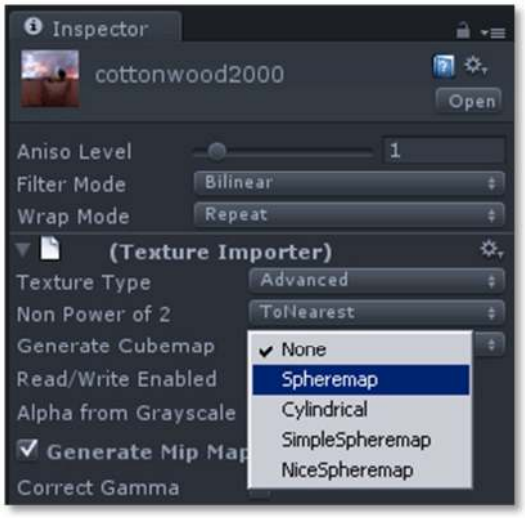
Conversión de environment map a Cube Map a través de unity

El proceso de conversión de environment a cubemap en unity comienza importando la textura de environment en unity, esto se puede hacer de forma manual o puede venir agregada[24] en un modelo FBX.

[24] Traducción contextual de la palabra embebed

Ya en la textura debes seleccionar en las opciones de tipo de textura como Advanced, esta opción desplegara nuevas funciones de la textura entre ellas la que nos interesa que es Generate Cubemap

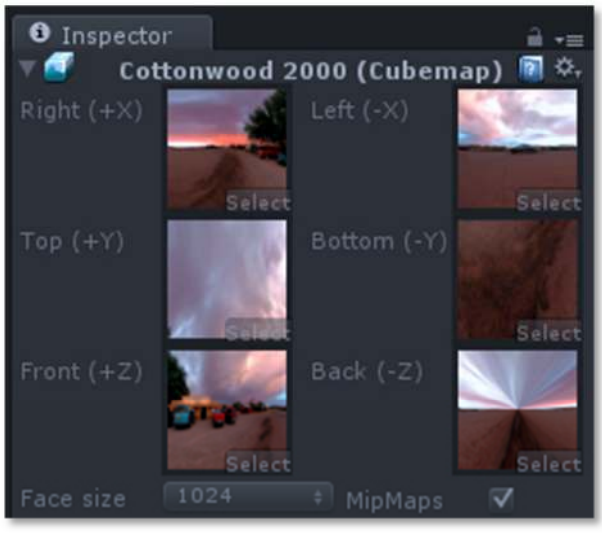


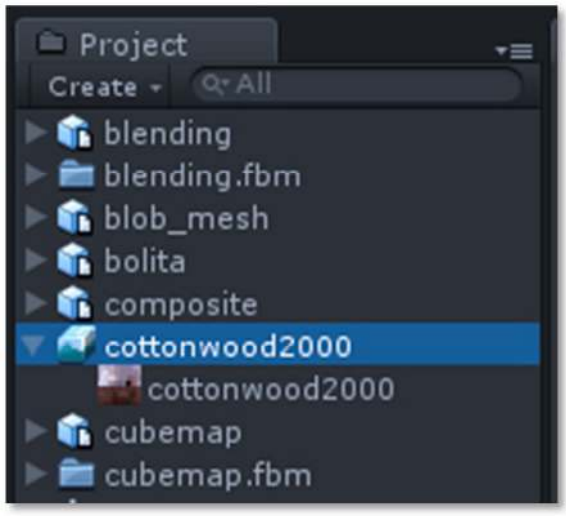


Dentro de las opciones de Generate Cubemap deberemos seleccionar la opción Spheremap para que realice la conversión entre tipos de panorama de latitud longitud a single files cubemap y podrá ser utilizado para simular un reflejo.

Una vez que ha terminado la operación se mostrarán las imágenes separadas que constituyen al cubemap.

En la imagen podemos apreciar a las seis imágenes que definen cada lado del cubo de reflejo.





Dentro de la ventana Project podemos observar el cubemap que se representa por un cubito azul turquesa que dentro tiene la imagen original de environment map tipo latitud longitud.

Aplicado al shader de reflejo simulado podemos observar como este se visualiza de manera similar al metal reflejante y este reflejo es dependiente de las coordenadas del mundo por ende si la primitiva gira el reflejo mantendrá las coordenadas de mundo.