

UT. 5 Desarrollo de juegos 2D y 3D

Conceptos avanzados de programación 3D.

Sistemas de coordenadas, modelos 3D, formas 3D,
transformaciones - renderización

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Aplicación de las funciones del motor gráfico. Renderización.

Aplicación de las funciones del grafo de escena. Tipos de nodos
y su utilización.

Análisis de ejecución. Optimización del código - Unity.

Conceptos avanzados de programación 3D.

Los **gráficos 3D por computadora** o **por ordenador** (en inglés *3D computer graphics*) son gráficos que utilizan una representación tridimensional de datos geométricos (a menudo cartesianos) que se almacenan en el ordenador con el propósito de realizar cálculos y representar imágenes 2D, que se pueden almacenar para verlas más tarde o mostrarlas en tiempo real. Además, el término se puede referir al proceso de creación de dichos gráficos, o al campo de estudio de técnicas y tecnología relacionadas con los gráficos 3d.

Un gráfico 3D difiere de uno bidimensional principalmente por la forma en que ha sido generado. Este tipo de gráficos se originan mediante un proceso de cálculos matemáticos sobre entidades geométricas tridimensionales producidas en un ordenador, y cuyo propósito es conseguir una proyección visual en dos dimensiones para ser mostrada en una pantalla o impresa en papel.

En general, el arte de los gráficos tridimensionales es similar a la escultura o la fotografía, mientras que el arte de los gráficos 2D es análogo a la pintura. En los programas de gráficos por computadora esta distinción es normalmente difusa: algunas aplicaciones 2D utilizan técnicas 3D para alcanzar ciertos efectos como iluminación, mientras que algunas aplicaciones 3D primarias hacen uso de técnicas 3D.

Conceptos avanzados de programación 3D.

Modelado

- La **etapa de modelado consiste en dar forma a objetos individuales** que luego serán usados en la escena creada. Existen diversos tipos de geometría para modelar con NURBS y modelado poligonal o subdivisión de superficies (en inglés subdivision surfaces). Además, existe otro tipo llamado "modelado basado en imágenes" o en inglés image based modeling (IBM), que consiste en convertir una fotografía a 3D mediante el uso de diversas técnicas, por ejemplo, la fotogrametría cuyo principal impulsor es Paul Debevec.

Hay dos tipos de técnicas de modelar que son las más representativas dentro del modelado:

- **Modelos representados por polígonos:** Uno de los sistemas utilizado por el ordenador para representar cualquier estructura son los polígonos. Un **cubo tiene 6 caras**, por lo tanto, cada una de ellas se trata de un polígono; una pirámide se compone de 4 triángulos y una base cuadrada. Sin embargo, una forma redondeada también se representa mediante polígonos, por ejemplo, un balón de fútbol se compone de 12 pentágonos y 20 hexágonos.
- **Modelos definidos por sus curvas matemáticas:** Actualmente hay otros sistemas de modelado donde el usuario trabaja con superficies curvas definidas matemáticamente. Un caso es la circunferencia, que se puede representar como un polígono de muchos lados, pero también como una **función matemática entre dos variables: X e Y**. Así mismo, el usuario trabaja con un programa vectorial para trazar curvas perfectas en un modelador no poligonal, y también dispone de diferentes tipos de herramientas para crear superficies curvas complejas.

Conceptos avanzados de programación 3D.

Composición de la escena

Esta etapa trata de distribuir los diferentes elementos (objetos, luces, cámaras...) en una escena que será utilizada para producir una imagen estática o una animación. A continuación, aparecen los diferentes aspectos que forman parte de la composición de una escena:

- **Sombra:** Definición de la forma de las sombras de los objetos. Para ello se utilizan materiales denominados shaders, algoritmos que controlan la incidencia de la luz combinando texturas con materiales.
- **Iluminación:** Creación de luces puntuales, direccionales en área o volumen, con distinto color o propiedades. Las luces tipo *omni* generan rayos de luz en todas las direcciones a diferencia de las direccionales en las cuales los rayos de luz se dirigen a una sola dirección. Además, algunos programas se ocupan de las luces tipo *domo* que iluminan a toda la escena, así también de luces que toman parámetros de laboratorio de lámparas reales. En relación con el color, se puede habilitar acorde a la escena o composición que se desee lograr y se puede configurar un ambiente con colores cálidos o fríos, los cuales se consiguen modificando los valores del RGB de cada una de las luces.

Conceptos avanzados de programación 3D.

Animación

- La animación en 3D es un proceso complejo, porque conlleva la realización previa de otros procesos como el diseño y modelado del objeto a animar. Consiste en la deformación o movimiento de los objetos de un modelo 3D a lo largo del tiempo. Para que haya animación, esta deformación o movimiento debe variar en algún aspecto respecto al tiempo: cambio de luces y formas, movimiento de objetos y cámaras, etc.

Los objetos se pueden animar a partir de:

- Transformaciones básicas en los tres ejes (XYZ), rotación, escala y traslación.

Modificaciones en formas:

- **Mediante esqueletos:** a los objetos se les puede asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. Esto ayuda al proceso de animación, en el cual el movimiento del esqueleto automáticamente afectará las partes correspondientes del modelo.
- **Mediante deformadores:** pueden ser cajas de deformación (*lattices*) o cualquier deformador que produzca, por ejemplo, una deformación sinusoidal.
- Mediante **dinámicas** para simulaciones de ropa, pelo, rígidas de objeto.

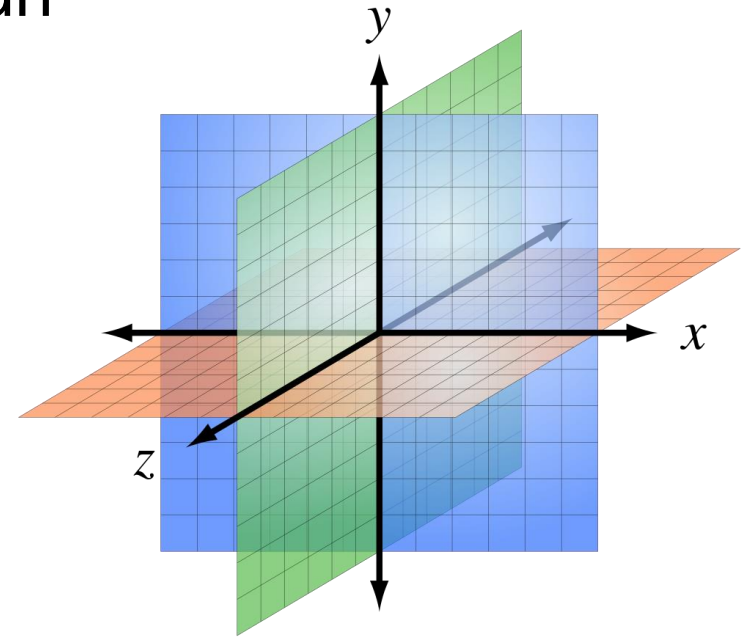
Conceptos avanzados de programación 3D.

Renderizado

- Se denomina **renderizado** (del inglés render) al proceso final de **generar la imagen 2D o animación a partir de la escena creada**. Es decir, en esta parte se procesa todo lo que es polígono, sombras, reflejos, iluminación, etc. para dar imágenes realistas, esto se puede renderizar como una única imagen o como un vídeo formado por muchas imágenes (fotograma).
- El **software de rénder** puede simular efectos cinematográficos como lens flare, profundidad de campo o motion blur (desenfoque de movimiento). Estos elementos son un producto de las imperfecciones mecánicas de la fotografía física, pero la simulación de dichos efectos aportan un elemento de realismo a la escena.
- Se han desarrollado **técnicas** con el propósito de **simular efectos atmosféricos o naturales** como lluvia, humo, fuego, niebla, etc. mediante el muestreo volumétrico o cáusticas (efecto de la luz al atravesar superficies refractantes).
- El **proceso de rénder** requiere simular una gran cantidad de procesos físicos complejos. A través de los años, la capacidad de procesamiento ha incrementado, permitiendo un nivel superior de realismo en los rénders. La mayoría de los estudios de cine utilizan Render Farm (granja de rénder) para generar imágenes de forma más rápida y eficaz.

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

- En **geometría**, un **sistema de coordenadas** es un sistema de referencia que utiliza uno o más números (*coordenadas*) para determinar unívocamente la posición de un punto u objeto geométrico.
- El orden en que se escriben las coordenadas es significativo y a veces se las identifica por su posición en una tupla ordenada; también se las puede representar con letras, como por ejemplo «la coordenada- x ». El estudio de los sistemas de coordenadas es el objeto de la **geometría analítica**, que permite formular problemas geométricos de forma "numérica".



Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

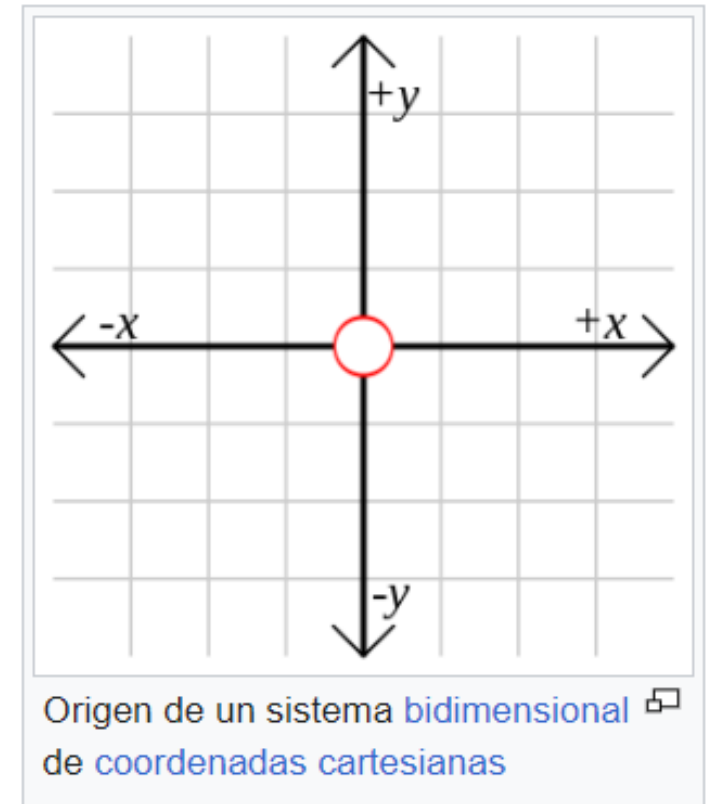
Coordenadas de objetos geométricos

- Los sistemas de coordenadas suelen utilizarse para especificar la posición de un punto, pero también pueden utilizarse para especificar la posición de figuras más complejas como rectas, planos, círculos o esferas. Por ejemplo, las coordenadas plückerianas permiten determinar la posición de una línea recta en el espacio. Cuando es necesario, el tipo de figura que se describe se utiliza para distinguir el tipo de sistema de coordenadas, por ejemplo el término coordenadas de la recta se utiliza para cualquier sistema de coordenadas que especifica la posición de una línea recta.
- Puede ocurrir que los sistemas de coordenadas para dos conjuntos diferentes de figuras geométricas sean equivalentes en términos de su análisis. Un ejemplo de ello son los sistemas de coordenadas homogéneos para puntos y rectas en el plano proyectivo.
- Los dos sistemas en un caso como este se dice que son duales. Los sistemas duales tienen la propiedad de que los resultados de un sistema pueden trasladarse al otro, ya que estos resultados son solo interpretaciones diferentes del mismo resultado analítico; esto se conoce como el principio de dualidad.

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

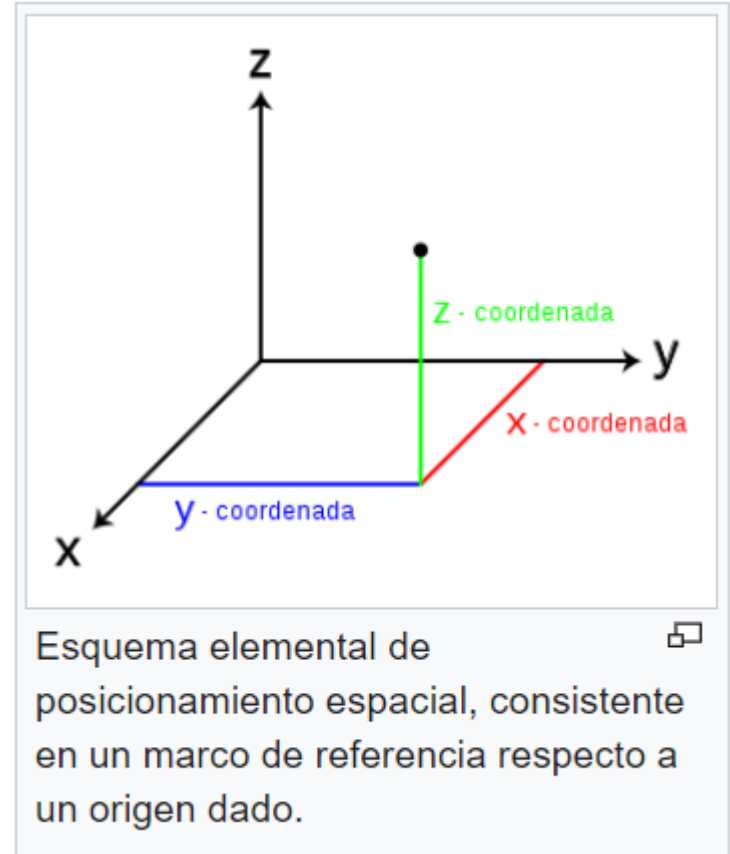
Origen de coordenadas

- El origen de coordenadas es el punto de referencia de un sistema de coordenadas. En este punto, el valor de todas las coordenadas del sistema es nulo.
- Sin embargo, en algunos sistemas de coordenadas no es necesario establecer nulas todas las coordenadas. Por ejemplo, en un sistema de coordenadas esféricas es suficiente con establecer el radio nulo siendo indiferentes los valores de latitud y longitud.
- En un sistema de coordenadas cartesianas, el origen es el punto en que los ejes del sistema se separan.



Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

- En física, geometría y análisis matemático, un objeto/modelo 3D o ente es **tridimensional** si tiene tres dimensiones. Es decir, cada uno de sus puntos puede ser localizado especificando tres números dentro de un cierto rango.
- Ejemplo: **anchura** , **altura** y **profundidad**.



Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

- En tres dimensiones, existen nueve politopos regulares: los cinco sólidos platónicos convexos y los cuatro sólidos de Kepler-Poinsot no convexos.

Politopos regulares en tres dimensiones

Class	Sólidos platónicos			Sólidos de Kepler-Poinsot					
Simetría	T_d	O_h		I_h					
Grupo de Coxeter	$A_3, [3,3]$	$B_3, [4,3]$		$H_3, [5,3]$					
Orden	24	48		120					
Poliedro regular	 $\{3,3\}$	 $\{4,3\}$	 $\{3,4\}$	 $\{5,3\}$	 $\{3,5\}$	 $\{5/2,5\}$	 $\{5,5/2\}$	 $\{5/2,3\}$	 $\{3,5/2\}$

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

En geometría son tridimensionales las siguientes figuras geométricas:

- **Poliedros de caras planas:**

- Pirámides
- Cubo
- Prismas

- **Superficies curvas:**

- Cilindro
- Conos
- Esfera o 3-esfera

Ya que todas ellas pueden ser embebidas en un espacio euclídeo de tres dimensiones. Sin embargo, hay que señalar que técnicamente la esfera, el cono o el cilindro son variedades bidimensionales (solo la cáscara) ya que los puntos interiores a ellos no son estrictamente parte de los mismos. Solo por un abuso de lenguaje o extensión del mismo informalmente se habla de esferas, cilindros o conos incluyendo el interior de los mismos.

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

Simulación 3D

- Hoy en día es posible la simulación mediante cálculos basados en la proyección de entornos tridimensionales sobre pantallas bidimensionales, como monitores o televisores. Estos cálculos requieren de una gran carga de proceso por lo que algunas computadoras y videoconsolas disponen de cierto grado de aceleración gráfica 3D gracias a dispositivos desarrollados para tal fin.
- Las computadoras disponen de las llamadas tarjetas gráficas con aceleración 3D. Estos dispositivos están formados por uno o varios procesadores (unidad de procesamiento gráfico) diseñados especialmente para acelerar los cálculos que suponen reproducir imágenes tridimensionales sobre una pantalla bidimensional y de esta forma liberar de carga de proceso a la unidad central de procesamiento de la computadora.

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

El **modelado 3D** es el proceso de desarrollo de una representación matemática de cualquier objeto tridimensional (ya sea inanimado o vivo) a través de un software especializado. Al producto se le llama modelo 3D. Se puede visualizar como una imagen bidimensional mediante un proceso llamado renderizado 3D o utilizar en una simulación por computadora de fenómenos físicos. El modelo también se puede crear físicamente usando dispositivos de impresión 3D.

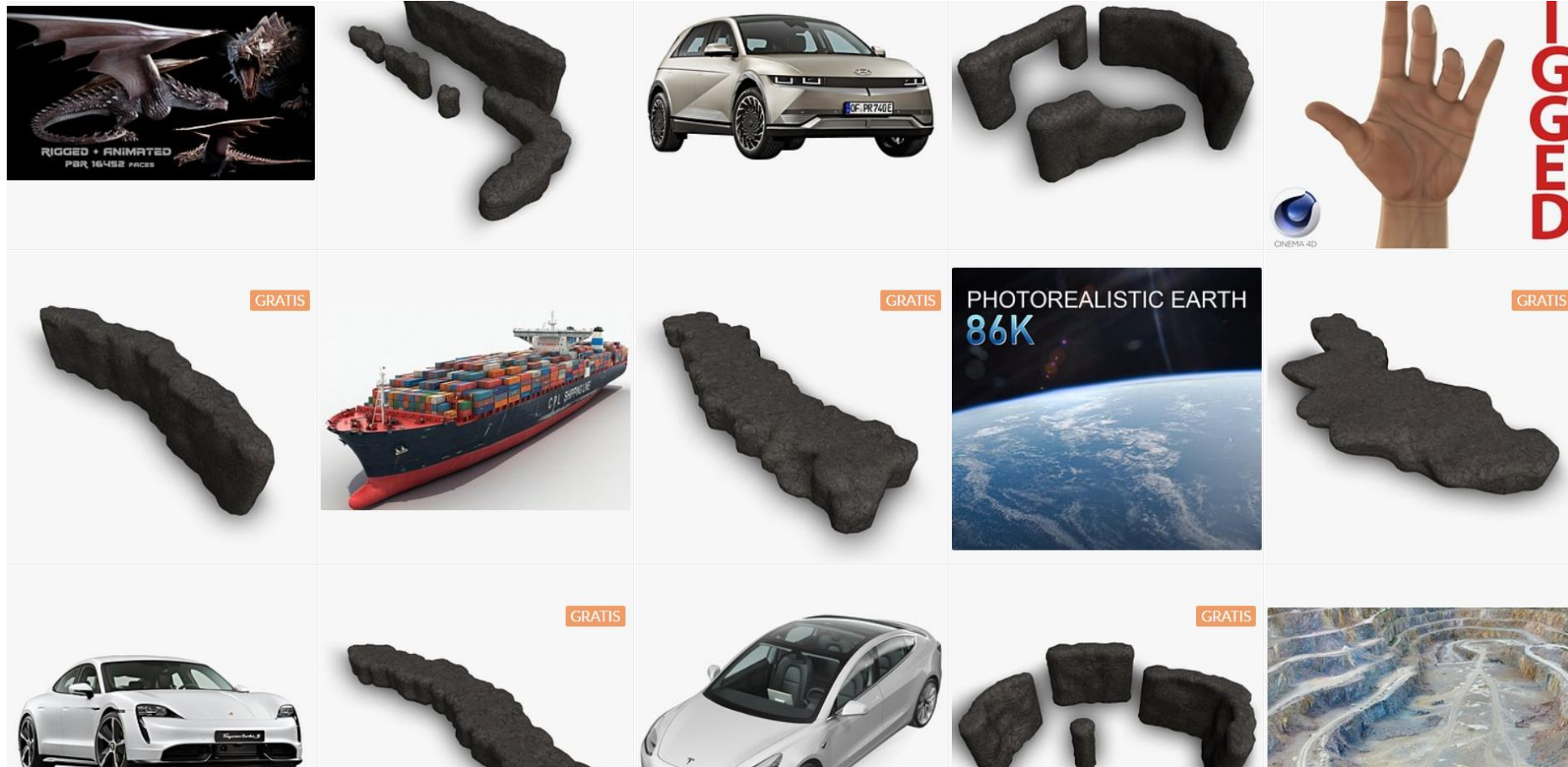
Los **modelos pueden ser creados automática o manualmente**. El proceso manual de preparar la información geométrica para los gráficos 3D es similar al de las artes plásticas y la escultura.

El software de modelado 3D es un tipo de software de gráficos 3D utilizado para producir modelos tridimensionales. Los programas individuales de este tipo son llamados «Aplicaciones de modelado» o «modeladores».

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

- Los **modelos 3D** representan un objeto tridimensional usando una colección de puntos en el espacio dentro de un espacio 3D, conectados por varias entidades geométricas tales como triángulos, líneas, superficies curvas, etc. Siendo una colección de datos (puntos y otro tipo de información), los modelos 3D pueden ser hechos a mano, a través de algoritmos o bien escaneados.
- **Los modelos 3D son ampliamente usados en gráficos 3D.** De hecho, su uso pre-data se extiende al uso de gráficos 3D en ordenadores. Algunos videojuegos usan imágenes pre-renderizadas de modelos 3D como sprites antes de que los ordenadores pudieran renderizarlas en tiempo real.

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

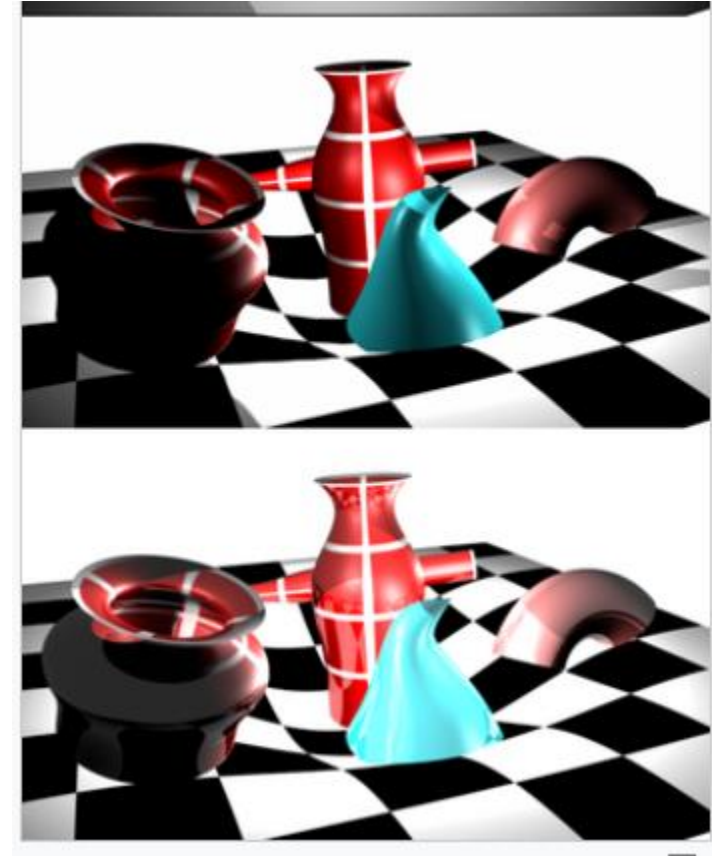
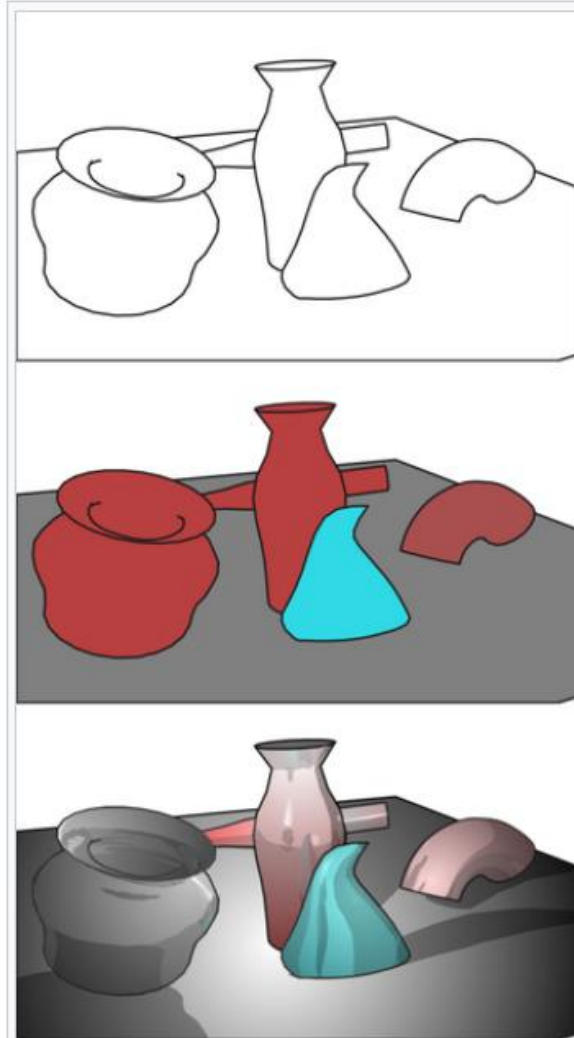


Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

- El término **renderización** (del inglés rendering) es un anglicismo para representación gráfica, usado en la jerga informática para referirse al proceso de generar imagen fotorrealista, o no, a partir de un modelo 2D o 3D (o en lo que colectivamente podría llamarse un archivo de escena) por medio de programas informáticos. Además, los resultados de mostrar dicho modelo pueden llamarse render.
- Cuando se completa la imagen previa (generalmente un boceto de *wireframe*), se utiliza la representación, que agrega texturas de mapa de bits o texturas por procedimientos, luces, mapeado topológico y posición relativa a otros objetos. El resultado es una imagen completa que ve el consumidor o el espectador previsto.
- Para las animaciones de películas, se deben representar varias imágenes (cuadros) y unirlos en un programa capaz de hacer una animación de este tipo. La mayoría de los programas de edición de imágenes en 3D pueden hacer esto.

Sistemas de coordenadas, modelos 3D, formas 3D, transformaciones - renderización

Ejemplo renderizado
sobre una misma
escena:



Propiedades de los objetos: luz, texturas, reflejos, sombras.

- **La luz** (del latín lux, lucis) es la parte de la radiación electromagnética que puede ser percibida por el ojo humano. En física, el término luz se considera como parte del campo de las radiaciones conocido como espectro electromagnético, mientras que la expresión luz visible señala específicamente la radiación en el espectro visible.
- **La luz**, como todas las radiaciones electromagnéticas, está formada por partículas elementales desprovistas de masa denominadas fotones, cuyas propiedades de acuerdo con la dualidad onda-partícula explican las características de su comportamiento físico. Se trata de una onda esférica.
- **La óptica** es la rama de la física que estudia el comportamiento de la luz, sus características y sus diferentes manifestaciones.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Una **textura** es una imagen de mapa de bits en el espacio (u,v) o (s,t) que se aplica a una figura, **ya sea tridimensional o bidimensional**, para darle realismo, mediante un programa de gráficos especializado. Algunos tipos comunes de texturas incluyen:

- **Texturas de color:** Se utilizan para aplicar patrones de color y detalles visuales a la superficie de un objeto.
- **Texturas de normal map:** Permiten simular detalles de relieve en una superficie, lo que mejora la apariencia de objetos tridimensionales al interactuar con la luz.
- **Texturas de desplazamiento:** Se utilizan para alterar la geometría de una superficie virtual, creando efectos de relieve y deformación.
- **Texturas Specular:** Controlan la forma en que un objeto refleja la luz, permitiendo simular superficies brillantes o rugosas.
- **Texturas de emisión:** Hacen que una superficie emita luz propia, lo que puede ser útil para representar materiales autoluminosos o efectos especiales.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Las texturas son ampliamente utilizadas en diversas aplicaciones de gráficos por computadora, como:

- **Videojuegos:** Las texturas se utilizan para representar personajes, entornos, objetos y efectos visuales en juegos, mejorando la apariencia y la inmersión.
- **Animación y efectos visuales:** Las texturas son fundamentales para crear imágenes realistas y detalladas en animaciones y efectos visuales en películas, comerciales y otros medios audiovisuales.
- **Diseño arquitectónico:** Las texturas se aplican a modelos arquitectónicos virtuales para simular materiales y detalles realistas, permitiendo una visualización precisa de los diseños.
- **Simulaciones:** En campos como la medicina, la ingeniería y la física, las texturas se utilizan para simular materiales y fenómenos, lo que permite una representación precisa y realista.
- **Realidad Extendida:** Las texturas son esenciales para crear entornos virtuales inmersivos y realistas en aplicaciones de realidad virtual.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

El **mapeado de texturas** establece cómo se sitúa la textura sobre el objeto al momento de proyectarse. Algunas texturas son tridimensionales y por lo general no necesitan un mapeado detallado ya que no se verán exactamente deformaciones o imperfectos. Cuando hay texturas de imagen bidimensionales existen una serie de inconvenientes. Cuando la textura es plana, el objeto no lo es. Se debe colocar de alguna manera esa figura “plana” sobre una superficie. Las técnicas de mapeado de texturas pueden ser de dos tipos:

- **Coordenadas Ortogonales:** éstas utilizan las propias coordenadas 3D de los objetos proyectados de forma automática. Aparentemente esta es la más sencilla ya que no necesita ninguna configuración adicional.
- **Mapeado UV:** se asigna una coordenada para cada vértice del objeto que más adelante se va a interponer.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

La **iluminación en los videojuegos**, al igual que la iluminación en el cine, son un conjunto de técnicas y conceptos en experiencias/ambientes reales transmitidas a un juego para dar en cada caso una sensación diferente. A través de la iluminación podemos transmitir emociones, crear ambientes y llamar la atención de los jugadores hacia los lugares u objetos que el creador quiere que preste más atención.

El desarrollo técnico de los últimos años ha dado lugar a un gran realismo gráfico y a mayores oportunidades de interacción a través de la iluminación en los entornos de los videojuegos.

La iluminación aplicada a los videojuegos se inspira en la tradición pictórica como referencias y adaptarlas a los diferentes escenarios dependiendo de cada videojuego.

Uno de estos aspectos es el “**Clarooscuro o Tenebrismo**” (técnica pictórica desarrollada en el Barroco), en donde, la luz siempre estará dirigida hacia algún objeto o personaje para poder resaltar las sombras y a su vez, la luz. Esto realizará atmósferas y paisajes que **llaman aún más la atención del jugador.**

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Entre los múltiples tipos de iluminación posibles en el desarrollo de videojuegos, hay que tener en cuenta cuál escoger a la hora de desarrollar nuestro juego, teniendo en cuenta el uso que va a hacer dicha iluminación.

Una **mala gestión de su uso** puede conllevar a un **mal rendimiento** de nuestro juego, pese a que la iluminación hace que nuestro juego cobre vida, también sacrifica una considerable cantidad de rendimiento.

El **proceso de iluminación final** se puede llevar a cabo con **Unity o Unreal Engine (u otro motor de videojuegos)**, los cuales son unos software de desarrollo de videojuegos potentes y eficientes.

La diferencia entre estos dos es que Unity tiene un uso más fácil que Unreal Engine.

Algunos juegos conocidos como *Genshin Impact*, *Rust*, *Resident Evil*, *Assassin 's Creed*, *Hearthstone*, *Outlast*, *Life Is Strange 2*, *Fortnite*, *Dragon Ball, FighterZ*... Han usado técnicas de iluminación con los software de desarrollo de videojuegos mencionados anteriormente.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

La **dramaturgia de la Luz en los videojuegos** se puede escenificar de diversas maneras, el objetivo es conseguir transmitir una sensación o un sentimiento al jugador para que la experiencia sea lo más inmersiva posible, acompañando las escenas y la historia.

Algunos de los recursos más usados son:

- **Iluminación con tonos fríos** para transmitir tristeza, soledad, etc
- **Iluminación con tonos cálidos** para transmitir felicidad, calidez, etc
- **Luz tenue/intensa con una oscuridad profunda** para guiar al jugador.
- **Luz natural** para estimar la hora o la meteorología en la que transcurre el juego.
- **Luz secundaria con una intención concreta** (peligro, seguridad, ansiedad, etc...)

Independientemente de cual sea el escenario en el que se encuentre el jugador, la iluminación puede afectar de maneras diversas a los sentimientos que pueden llegar a transmitir esa escena.

Dependiendo del **tipo de iluminación y del tipo de fuente de luz** un mismo escenario puede transmitir diferentes sensaciones siempre acordes a la temática del juego y/o el entorno en el que se encuentra el jugador.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

La iluminación, cuando se utiliza para guiar las acciones del jugador, **se clasifica en dos tipos: diegética y extradiegética**. La iluminación diegética es aquella que se encuentra en entornos oscuros e interiores y la iluminación extradiegética es aquella que es causada por la luz del día o que se encuentra en exteriores.

La iluminación es útil a la hora de guiar al jugador. Al señalar la ruta que éste debe tomar, se evitan tiempos muertos en los que se busque la salida sin éxito alguno. La clave para guiar al jugador mediante la iluminación es la intensidad, el contraste y el movimiento. Lo que hace el juego más accesible y permite al jugador moverse fluidamente por el entorno.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Cómo se **diseña la iluminación** es algo que no sólo depende del motor gráfico sino de la intencionalidad de lo que se desea transmitir. Lo principal es diseñar las funciones narrativas de la iluminación y luego ver cuáles son las necesidades técnicas y la capacidad de los motores de juego utilizados para trabajar en la iluminación de un videojuego

En la aplicación existen las **luces estáticas**, las cuales no cambian de posición al salir de luces fijas; es una técnica muy utilizada para ahorrar recursos en los videojuegos y optimizar el rendimiento de estos ya que solo necesitan ser procesadas a la hora de crear el juego.

Por otro lado están las **luces dinámicas**, que son luces interactivas con la escena 3D creada y que van variando dependiendo del escenario o lugar de la emisión. Por ejemplo, si en un videojuego amanece y anochece, la posición de las sombras varía de un momento a otro.

Podemos distinguir entre **aspectos formales**, **aspectos compositivos** y **aspectos técnicos**.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Aspectos formales

Los aspectos formales son aquellos que pueden hacer variar la intensidad, dirección, etc.. de la luz.

- **Difusión:** Deformación de la luz después de rebotar o reflejarse en una superficie.
- **Color:** Según la paleta de colores que se use en un videojuego se puede transmitir un sentimiento u otro.
- **Direccionalidad:** Se refiere a la dirección que tiene la luz, por ejemplo en los videojuegos la direccionalidad de la luz suele ser utilizada para decir al jugador que camino puede o debe tomar.

Aspectos compositivos

Los aspectos compositivos son aquellos que definen la forma en la que es compuesta una escena, es decir dónde y cómo se sitúan las fuentes de luz.

- **Pueden ser obvios** - Es decir, aparecen en la pantalla del jugador de forma que puede ver directamente la o las fuentes de luz que iluminan la escena.
- **Pueden ser no tan obvios** - Es decir, el jugador puede deducir de dónde procede la iluminación pero no lo puede ver con certeza, es decir la fuente de luz no es visible para este.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

Aspectos técnicos

Los aspectos técnicos son aquellos que influyen en la iluminación dependiendo del software que sea utilizado para crear la iluminación.

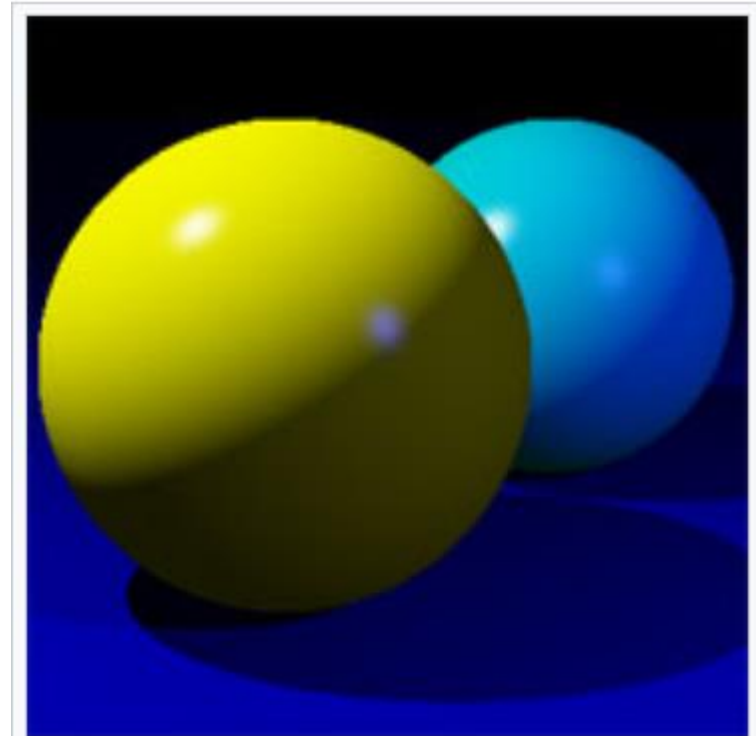
Iluminación juegos 2D y 3D

Existen diferentes formas de trabajar la iluminación en los videojuegos. Podemos diferenciar dos grandes ámbitos según sean juegos trabajados en dos o tres dimensiones:

- **Juegos diseñados en 2D:** Las luces en juegos 2D son, en realidad, sprites con una opacidad y un contraste determinados, de esta manera se mezcla con el entorno y da un efecto de iluminación.
- **Juegos diseñados en 3D:** Las luces en juegos 3D son fuentes de luz a las cuales se les puede cambiar la intensidad y el contraste, entre otros, y se puede modificar para que, al chocar con un objeto, el rastro de luz se vea interrumpido y de un efecto de sombra.

Propiedades de los objetos: luz, texturas, reflejos, sombras.

- Un **resaltado especular** o **reflejo especular** es el punto brillante de luz que aparece en objetos brillantes cuando está iluminado (por ejemplo, ver imagen a la derecha). Los resaltados especulares son importantes en los gráficos de computadora 3D, ya que proporcionan una señal visual fuerte para la forma de un objeto y su ubicación con respecto a las fuentes de luz en la escena.



Reflejos especulares sobre un par de esferas.

Aplicación de las funciones del motor gráfico. Renderización.

Render

El renderizado es el proceso de generar una imagen (realista o no) a partir de una referencia 2D o un modelo 3D a través de un programa especializado. Para poder hacer un renderizado, se requiere de varias cosas: texturas, geometría, luces, sombras y una cámara. Al resultado de este proceso se le llama **render**.

Cada render tiene un tipo de iluminación específica, hay estos tipos:

Realtime

- La iluminación “**Realtime**” consta de ir generando, en tiempo real, todas las luces y sombras de la escena, cada vez que cambies una luz en tu escena, esta se actualizará en tiempo real dándote un feedback “instantáneo”.

Baked

- La iluminación “**Baked**” consiste en hacer un cálculo de las luces que se encuentran en la escena, renderizarlos y plasmarlos en un mapa de luces, el cual se sobrepondrá sobre las texturas para generar una falsa ilusión de sombras, es decir si se calcula la luz y después se mueve un objeto, la sombra que arrojaba ese objeto se mantiene y el objeto actual no arrojará sombra hasta que se recalcule la luz nuevamente.

Ray Tracing

- El trazado de rayos o “**Ray Tracing**” es un algoritmo que se utiliza para el renderizado de imágenes que consiste en el cálculo del camino de la luz como píxeles en el plano de una imagen y simula el efecto de la luz sobre las superficies con las que interactúa. Las imágenes que son producidas gracias a esta técnica tienen un **grado muy alto de realismo**, aunque necesitas una mayor capacidad en tu ordenador. Por ejemplo, un videojuego con el ray tracing activado tendrá un rendimiento menor que un videojuego sin el ray tracing activado.
- Hay otras tecnologías a tener en cuenta en el aspecto del color dentro del renderizado en los videojuegos.

Aplicación de las funciones del motor gráfico. Renderización.

HDR

- El **HDR (High Dynamic Range)** es una tecnología que está implantada en las pantallas y en las cámaras fotográficas. Aunque se puede configurar y desactivar si se desea, lo que hace es conseguir información de captura y reproducir una escala de grises más amplia.
- Por lo tanto hay una **gran diferencia entre luces y sombras que hace que se vea más natural** y se acerque más a lo que ven los humanos. **El HDR en los videojuegos mejora la experiencia** ya que aplica un mejor rendimiento de color múltiple, una saturación más profunda, un contraste más diverso, reflejos más brillantes y sombras más oscuras.
- También hay diferentes tipos de HDR: HDR10, Dolby Vision, HDR10+, y HLG. El HDR10 es el más usado ya que no hay que pagar ningún tipo de licencia. El HDR10 asigna unos valores predeterminados durante todo el tiempo, a diferencia de HDR10+ y Dolby Vision que cambian los valores dependiendo del momento.

Espacio de color

- Un **Espacio de Color es una organización concreta de los colores**. En la actualidad dependiendo del dispositivo que los esté reproduciendo, los colores pueden verse de manera distinta. La forma en la que los dispositivos organizan los colores antes de mostrarlos se llama Espacio de Color. Hay muchos tipos, entre los más comunes se encuentran el **RGB y el CMYK**.

Análisis de ejecución. Optimización del código

- Unity.

Optimizando Scripts

- Esta sección demuestra cómo a usted le gustaría ir acerca de optimizar los scripts y métodos actuales que su juego utiliza, y también va en detalle acerca de las razones por las cuales las optimizaciones funcionan, y por qué aplicarlas le van a beneficiar en ciertas situaciones.

El Profiler es el Rey

- No hay tal cosa como un checkbox para marcar que le van a asegurar que su proyecto corra sin problemas. Para optimizar un proyecto lento, usted tiene que profile (perfilar) para encontrar los infractores específicos que toma una cantidad de tiempo desproporcionado. Intentar optimizar sin profiling (perfilar) o sin entender los resultados que el profiler (perfilador) le da es como intentar optimizar con una venda puesta

Profiler móvil Interno

- Usted puede utilizar el internal profiler para averiguar qué tipo de procesos está poniendo lento su juego, sea física, scripts, o el renderizado, pero usted no puede profundizar a los scripts y métodos específicos para encontrar los problemas en realidad.

Optimizado por Diseño

- Intentar desarrollar algo que es rápido del comienzo es arriesgado, por que hay un trade-off entre gastar tiempo por perder el tiempo haciendo cosas que sería igual de rápido si no se optimizarán y hacer cosas que tienen que ser cortadas o reemplazadas más tarde por ser demasiado lento.
- Esto toma intuición y conocimiento del hardware para hacer unas buenas decisiones en este aspecto, especialmente porque cada juego es diferente y lo que podría ser una optimización crucial para un juego puede ser un fracaso en otro.

Información de Unity - Más info:

<https://docs.unity3d.com/es/2019.4/Manual/MobileOptimizationPracticalScriptingOptimizations.html>

Bibliografía

- https://es.wikipedia.org/wiki/Gr%C3%A1ficos_3D_por_computadora
- https://es.wikipedia.org/wiki/Sistema_de_coordenadas
- <https://es.wikipedia.org/wiki/Renderizaci%C3%B3n>
- https://es.wikipedia.org/wiki/Resaltado_especular
- <https://es.wikipedia.org/wiki/Luz>
- [https://es.wikipedia.org/wiki/Textura_\(gr%C3%A1ficos_por_computadora\)](https://es.wikipedia.org/wiki/Textura_(gr%C3%A1ficos_por_computadora))
- https://es.wikipedia.org/wiki/Mapeo_de_texturas
- https://es.wikipedia.org/wiki/Iluminaci%C3%B3n_en_los_videojuegos
- <https://docs.unity3d.com/es/2019.4/Manual/MobileOptimizationPracticalScriptingOptimizations.html>