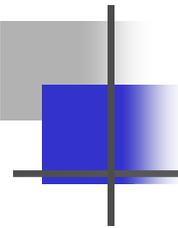


# Tema 1. Introducción

Prof. Marlon Núñez



# Contenido

---

1. Definición, dispositivos e historia
2. Géneros
3. Pros y contras de los videojuegos. Regulación.
4. Introducción a los modelos 2D, 2.5D y 3D
5. Entornos, lenguajes de programación, librerías

# Definición

- Un videojuego es un software creado para el entretenimiento en general y basado en la interacción entre una o varias personas por medio de un controlador y un aparato electrónico que ejecuta dicho videojuego.
- El aparato electrónico puede ser un computador, (ej. un PC) una máquina Arcade (ej. por monedas), una videoconsola o un dispositivo portátil (ej. teléfono móvil)

Arcade:



Consola:



Otros:



# Controladores

- Teclado
- Ratón
- Gamepad
- Joystick
- Detectores de movimiento:
  - Wiimote de Wii
  - Presión (ej. alfombras)
  - Kinect de Xbox
- Específicos:
  - pistola, volante, etc.



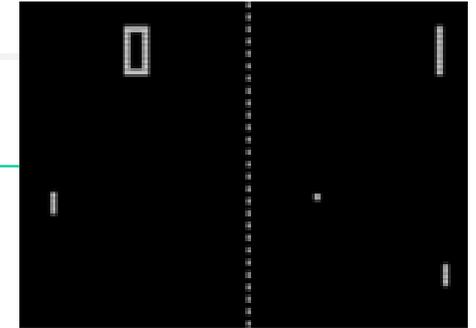
# Historia

---

- En 1958 el físico Willy Higinbotham creó el primer videojuego para explicar conceptos de física nuclear. Este evolucionará al juego *Pong*.
- Steve Russell, estudiante en la MIT (Massachusetts Institute of Technology), creó *Spacewar*, un adictivo juego de naves.
- En 1970 se creó la primera videoconsola doméstica por parte de Magnavox y de Ralph Baer y Bill Enders llamada Odyssey. Era un aparato muy primitivo: Se utilizaron 40 transistores.
- Nolan Bushnell creó una máquina que funcionaba con monedas e hizo una versión del *Spacewar* para salones recreativos llamada Computer Space.
- En 1972 Bushnell junto con Ted Dabney fundó ATARI, una compañía pensada para crear juegos de carreras.

## Historia (cont.)

- En 1972 ATARI desarrollo Pong fue el primer juego, que era una especie de tenis..
- En 1974 salió al mercado Tank para arcade y fue el primer juego que utilizó un chip ROM.
- En 1977 apareció la famosa ATARI 2600, de 8 bits, para la que se desarrollaron cientos de juegos.
- 1981 es el año histórico por excelencia a causa de la aparición del gigante NINTENDO y si primer juego: Donkey kong.
- En 1982 se creó EA (Electronic Arts), una compañía aun muy importante.
- En 1985 aparece la “revolución” de las consolas: NINTENDO saca al mercado su NES (Nintendo Entertainment System), que mostraba unos gráficos excelentes para aquella época.



# Géneros

---

Hay muchas formas de clasificar los videojuegos. La siguiente es una clasificación más.

Lucha	Plataformas
Disparos	Simulación
Aventura	Conducción
Rol	Deportivos
Estrategia	Otros

Muchos videojuegos tienen características de dos géneros o más.

# Lucha

---

- Son videojuegos basados en el combate cuerpo a cuerpo.
- Subgéneros:
  - Uno contra uno (vista lateral, vista 3D)
  - Beat'em up (1, 2... jugadores contra varios rivales)
  - Otros: avanzar y pegar, free-for-all o todos contra todos.
- Ejemplos:
  - Street Fighter, Urban Fighter, Super Smash Bros, ...

# Disparos

---

- En estos videojuegos el protagonista ha de abrirse camino o llevar a cabo una misión, a base de disparos.
- Subgéneros:
  - Disparos en primera persona o "FPS".
    - Generalmente se muestra del personaje son sus brazos y arma.
  - Disparos en tercera persona o "TPS".
    - Vista de personaje completo Objetivo: realizar misiones.
  - Disparos en primera persona multijugador masivo en línea o "MMOFPS"
  - Otros: videojuegos de pistola, *matamarcianos*
- Ejemplos:
  - Halo, Quake, Space Invaders, ...

# Aventura

---

- Son videojuegos en los que el protagonista debe avanzar en la trama interactuando con diversos personajes y objetos. Videojuegos caracterizados por la investigación, exploración, la solución de rompecabezas y un enfoque en el *relato* en vez de desafíos basados en reflejos. Mayoritariamente para un jugador.
  - Aventuras de acción
  - Aventuras gráficas
  - Aventuras conversacionales
  - Otros...
- Ejemplos:
  - Chichen Itzá o Zork, The Legend of Zelda

# Rol

---

- El protagonista interpreta un papel y debe mejorar sus habilidades mientras interactúa con el entorno y otros personajes. Atributos de los personajes (ej. vida, nivel, experiencia). Las decisiones del personaje influyen en su futuro próximo. Duración: varios días o meses.
- Subgéneros:
  - Tácticos
  - Rol multijugador masivos en línea (MMORPG) ...WoW, 2nd Life
  - Otras clasificaciones: Vista en 1ª persona, 3ª persona, Roguelike, batalla por turnos (BPT)
- Ejemplos:
  - Ragnarok Online, Final Fantasy Tactics, Star Wars Galaxy

# Estrategia

---

- El jugador deber planear y llevar a cabo una adecuada estrategia para derrotar a los enemigos, obtener recursos, construir bases, etc. Hay que manipular a un grupo de personajes y objetos.
- Subgéneros:
  - Estrategia en Tiempo real ("RTS") ... tiempo continuo
  - Estrategia por turnos ("TBS") ... analogía: ajedrez
- Otra forma de clasificarles es por su temática:
  - Gestión (ya sea esta económica o social) y bélicos.
- Ejemplos:
  - Age of Empires, Command & Conquer, Civilization, ...

# Plataformas

---

- Los videojuegos de plataformas se caracterizan por tener que recorrer, saltar o escalar una serie de plataformas y acantilados, con enemigos, mientras se recogen objetos para poder completar el videojuego.
- En 2D, el protagonista es conducido a través de un escenario bidimensional, efectuando un desplazamiento de izquierda-derecha y de arriba-abajo. También para 3D o su simulación (2.5D)
- Ejemplos:
  - Super Mario Bros, Sonic the Hedgehog, Donkey Kong (arcade).

# Simulación

---

- Permiten al jugador simular determinadas experiencias de la vida real. Muchas veces no tienen un objetivo concreto y por lo tanto carecen de un final propiamente dicho. Simular un construcción, controlar personajes simulados.
- Subgéneros:
  - Simuladores de construcciones (ciudades, zoologicos, etc)
  - Control de vidas (poblaciones, mascotas)
  - Otros: sociales, bailes...
- Ejemplos:
  - MS Flight Simulator, Los Sims, Petz, Dance Revolution

# Conducción de vehículos

---

- Conducción de un vehículo (coche de carreras, moto, nave espacial, lancha, avión) para llegar primero que los otros jugadores o escapar. Puede ser considerado como un subgénero de Simulación.
- Subgéneros:
  - Carreras
  - Escape
- Ejemplos:
  - Need for Speed, Gran turismo y Mario Kart (consola, arcade)

# Deportivos

---

- Recrean eventos deportivos, tanto individuales como colectivos, donde se manejan los jugadores. El objetivo es siempre ganar. Puede ser considerado como un subgénero de Simulación.
- Subgéneros:
  - Fútbol
  - Baloncesto
  - Tenis
  - Etc.
- Ejemplos:
  - Pro Evolution Soccer, Mario Tennis, NBA Live, Pong (arcade)

## Otros géneros

---

- **Party:** avanzar por turnos en un tablero virtual, superando diversas pruebas para llegar lo antes posible a la meta, o conseguir la máxima cantidad de puntos.
  - Ej. Mario Party, Rayman Raving Rabbids
- **Puzzle:** Resolver acertijos, problemas de lógica. Planear y tener la habilidad para avanzar niveles.
  - Ej: Tetris (encajar piezas), Buscaminas (identificar patrones)
- **Educativos:** Dar algún tipo de conocimiento.
  - Ej. English Training, Mi experto en francés
- **Ball games:** golpear esferas para hacerlas caer en huecos o que colisionen con otros objetos para obtener más puntos .
  - Ej. Pinball, Pong, Billar, Breakout

# Pros de los videojuegos

---

- Aportaciones sociales:
  - Fomentar la destreza motriz en ancianos y en niños con retraso mental.
  - Reducir las cefaleas en pacientes que se someten a radioterapia.
  - Facilitan las facultades de aprendizaje y de concentración de niños con este tipo de problemas.
  - Mejora algunas actividades físicas, como los reflejos, y también pueden incrementar la velocidad mental.
- Videojuegos como medios educativos: dependiendo del tipo de juego pueden enseñar diversas cosas.
- El jugador se implica, toma decisiones y ejecuta acciones. Además comprueba que su esfuerzo es recompensado.
- Potencian habilidades psicomotrices.
- Se adquieren conocimientos (lenguajes específicos, símbolos, técnicas...)

## Pros (cont.)

---

- Proporcionan un sentido del dominio, control y cumplimiento del que algunos jugadores pueden estar faltos en sus vidas, aunque no solucionan ningún problema.
  - En muchos casos aporta un aumento del autoestima y el reconocimiento social por parte de sus amigos.
  - Permiten el ejercicio de la fantasía, sin limitaciones espaciales, temporales o de gravedad.
- Favorecen el desarrollo de la coordinación oculomanual, enseñan habilidades específicas y ayudan a “aprender a aprender”.

# Contras

- La adicción a los videojuegos
  - Reflejan modelos que imperan en nuestra sociedad:
    - Violencia
    - Consumismo
    - Velocidad
    - Competición
    - Sexismo
- Pueden provocar que se les dedique mucho tiempo y dejar de lado otras tareas.
- Los jugadores que prefieren los juegos violentos pueden presentar una conducta más agresiva, impulsiva y egoísta
  - Los juegos violentos pueden predisponer la violencia con demasiada facilidad.
- Algunas personas con síntomas de fobia social utilizan estas tecnologías como refugio y defensa de su problema.
  - La familia tendrá que controlar que esto no afecte ni al desarrollo ni al estudio de los chavales.

## Contras (cont.)

- Los juegos agresivos no son recomendables ya que provocan ansiedad a corto plazo.
- Algunos riesgos a tener en cuenta de la adicción:
  - Algunas personas que tengan un problema psicológico previo pueden llegar a padecer trastornos de personalidad y de conducta por implicarse demasiado en el juego.
  - En algunos casos puede llegar a provocar estrés, fatiga ocular y problemas musculares, en algunos casos serios debido a la mala postura del jugador.
- Daños a la salud y daños cerebrales:
  - **Cefaleas**, que en algunos casos presentaban los mismos patrones de ondas cerebrales similares a las epilépticas.
  - **Convulsiones**, producidas por la visión de ciertas escenas:
    - Temporales, mientras dura la escena.
    - Periódicas, teniendo que administrar una droga anticonvulsiones para remitirlas.
  - **Visión borrosa**

# Código de regulación: PEGI

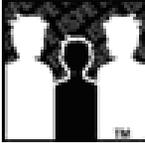
---

- PEGI es un nuevo código de autorregulación para los videojuegos.
- Se debe gracias a la ISFE (Interactive Software Federation of Europe).
- PEGI es válido en 16 países europeos:
  - Todos los países de la Unión Europea, a excepción de Alemania
  - Noruega
  - Suiza
- El objetivo es orientar a los padres del contenido que ofrecen los videojuegos y decidir si comprarlo o no.
- Es un código a escala continental, ya que han tenido en cuenta toda la variedad cultural y el “funcionamiento” de cada uno de los países participantes.

## Regulación: Código de edades

Más de 3 años	<b>3+</b>
Más de 7 años	<b>7+</b>
Más de 12 años	<b>15+</b>
Más de 16 años	<b>16+</b>
Más de 18 años	<b>18+</b>

## Descripción de contenidos

<p><b>Discriminación:</b> El juego contiene representaciones o material que puede favorecer a la discriminación.</p>	
<p><b>Drogas:</b> El juego hace referencia o muestra el uso de drogas.</p>	
<p><b>Miedo:</b> El juego puede asustar o dar miedo a niños.</p>	
<p><b>Lenguaje soez:</b> El juego contiene palabrotas.</p>	
<p><b>Sexo:</b> El juego contiene representaciones de desnudez y/o comportamientos o referencias sexuales.</p>	
<p><b>Violencia:</b> El juego contiene representaciones violentas.</p>	

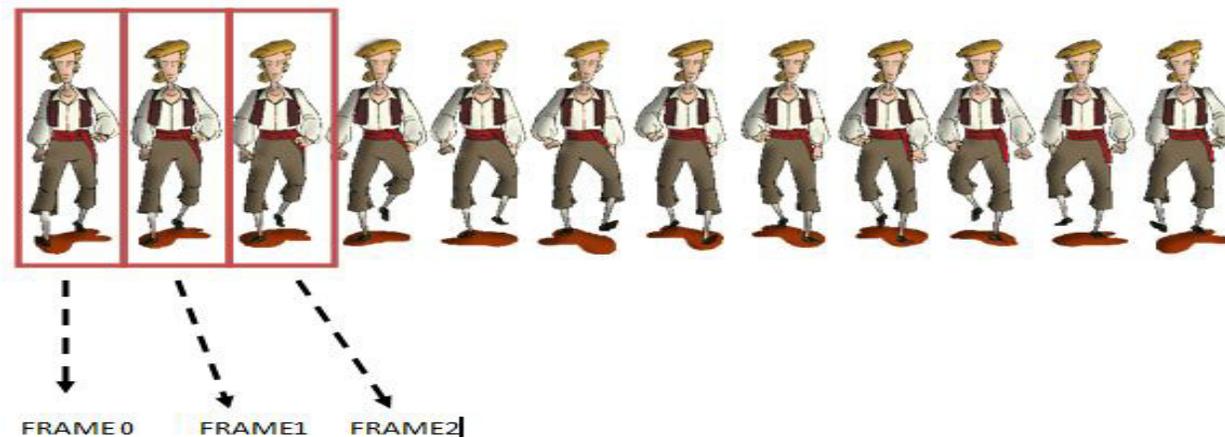
# Elementos de desarrollo

---

- Modelos
  - Modelos estáticos y animados
- Entorno de desarrollo
  - Herramientas específicas, IDE y lenguajes de programación y librerías 3d (ej. API 3D)
- Motor gráfico o de renderizado
  - Generación del objeto en su entorno (2D y 3D).
  - Los *motores de renderizado 3D* manejan perspectiva, iluminación, sombras, reflejos...
- Motor físico
  - Cálculo de la solución numérica de las ecuaciones diferenciales que gobiernan la dinámica física del entorno y sus objetos.

## Modelos y videojuegos 2D

- Los programas sitúan el fondo, los objetos del entorno. El eje x representa el ancho de pantalla y el eje y el alto.
- Los programas sitúan también a los personajes dependiendo de una lógica dada, de los controles externos y de los eventos de videojuego.
- Animación. Para representar un personaje/objeto moviéndose, se puede utilizar un dibujo.



## Modelos y videojuegos 2D (cont.)

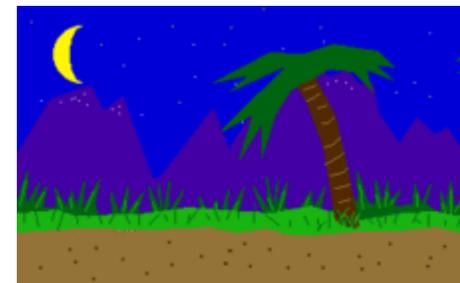
- También se puede usar un *sprite*.  
Un *sprite* es una imagen en dos dimensiones o una secuencia de imágenes que crean la ilusión de movimiento
- Los *sprites* se pueden desplazar, rotar y escalar transparentar.
- El videojuego da órdenes de cuantos *frames* por segundo dibuja y cuando lo hace y que transformadas le aplica



Hoja de *sprites*

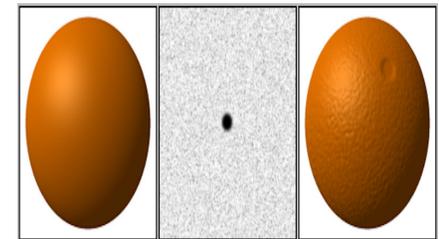
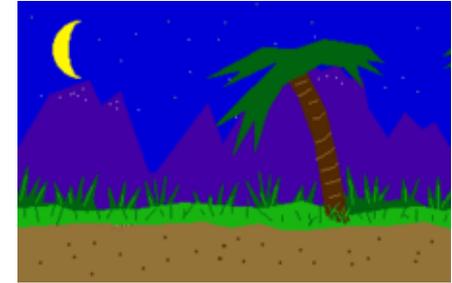
# Modelos 2D y videojuegos 2.5D

- Axonometría y proyección oblicua
  - El punto de vista se gira levemente para revelar otras facetas del medio ambiente que son visibles en una perspectiva de arriba hacia abajo o vista lateral, produciendo así un efecto tridimensional.
  
- *Skyboxes y skydomes*
  - Utiliza un fondo (el cielo, las montañas lejanas) creando la ilusión de alrededores distantes. es común que dicho fondo permanezca estacionaria



# Modelos 2D y videojuegos 2.5D

- *Parallax scrolling*
  - Se utilizan varias capas de sprites que se mueven en forma independiente para crear un sentido de profundidad
- Escala a lo largo del eje Z
  - Los sprites se escalan mayor o menor en función de su distancia al reproductor, produciendo ilusión de movimiento a lo largo del eje Z (hacia delante).
- Mapeado topológico, normal y por paralaje.
  - Se aplican a las texturas para simular golpes y arrugas en la superficie de un objeto sin necesidad de utilizar más polígonos. Simula profundidad de mayor realismo



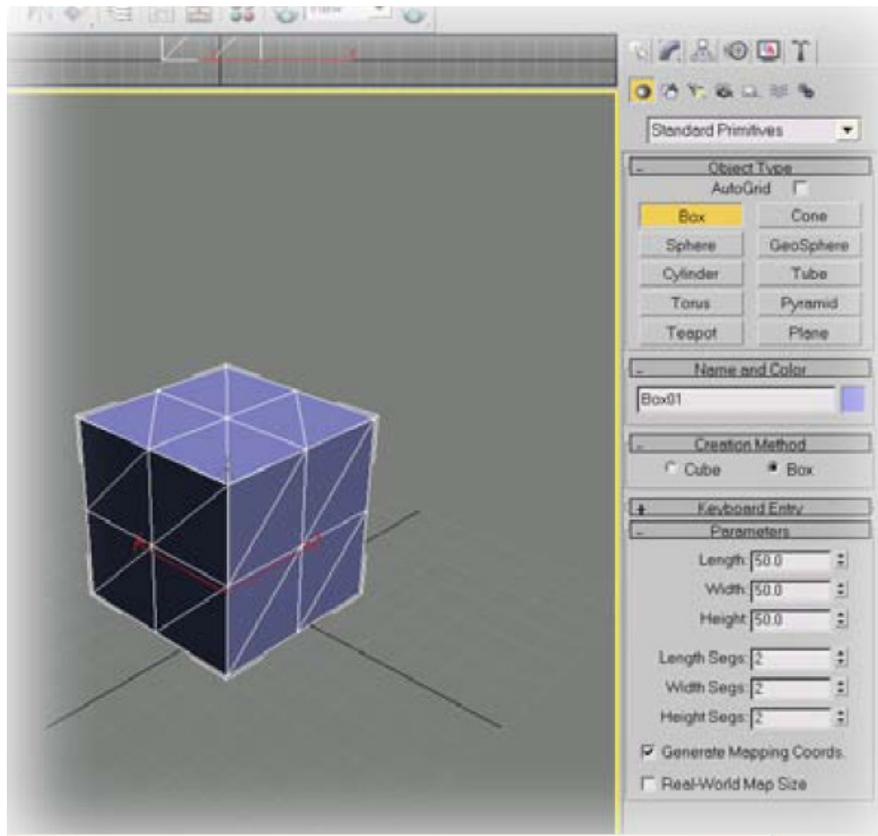
# Modelos 3D

---

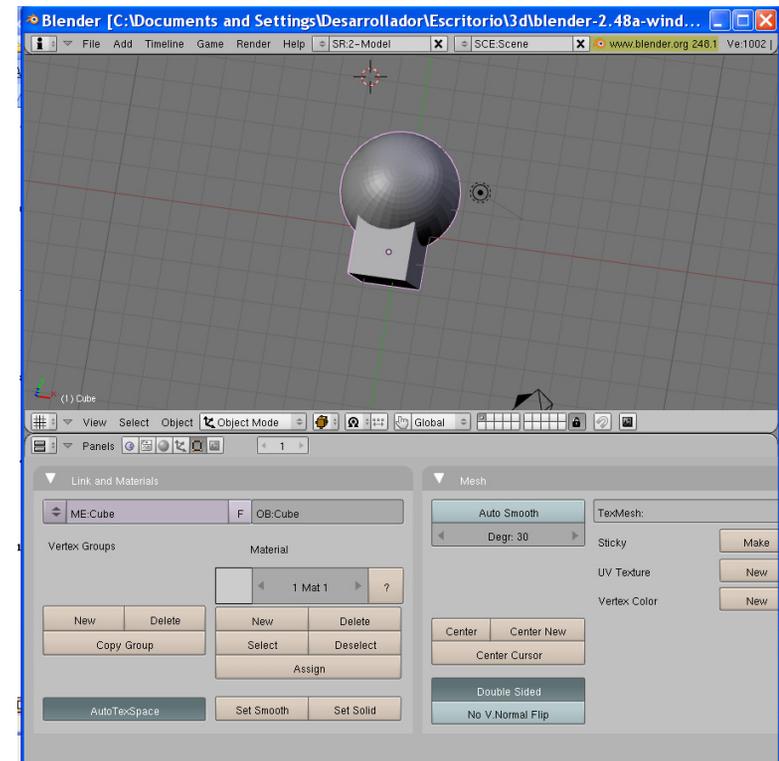
- Modelo estáticos
  - Incluyen principalmente la geometría y apariencia (ej. textura)
    - VRML, OBJ, PLY, otros
- Modelos animados
  - Incluyen principalmente geometrías, esqueletos, animaciones, texturas y sonidos
    - MD2 up to MD5, MDL, 3DS, Collada
- Herramientas más utilizadas para la creación de modelos estáticos y modelos animados:
  - Comerciales: *3D Studio* y *Maya*
  - Gratis: *Blender*

# Modelados 3D estáticos

## 3D Studio Max



## Blender 3D



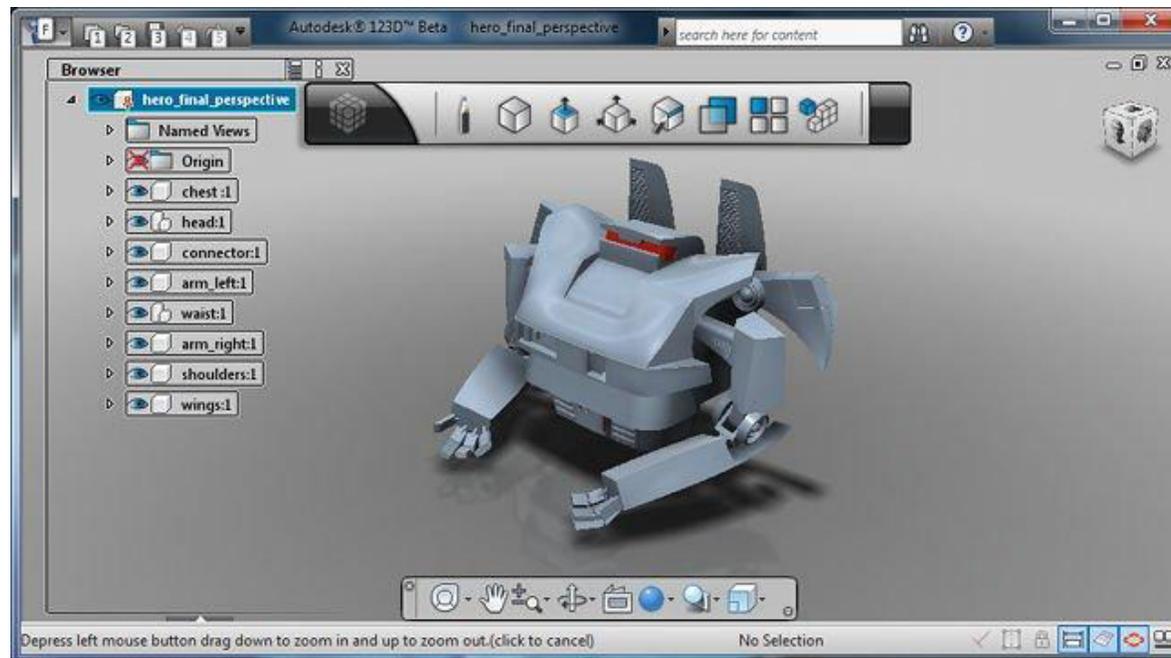
# Modelo 3D estáticos: herramientas gratuitas

## Autodesk 123D

Edición y visualización de modelos estáticos.

Formatos exportados:

\*.OBJ, \*.SAT, \*.STL, \*.STEP,...



# Modelos animados 3D

- Son ficheros y que contienen la información necesaria para representar un personaje:
  - Esqueleto
  - Animaciones
  - Textura
  - Otros aspectos: audio, física
- Hay varios formatos (MD2, Blender, MDL). Un formato muy utilizado en videojuegos para PC es Tal vez el formato más conocido en el mundo PC, es el MDL. Hay ficheros MDL de animales, soldados, Golems, etc. de libre distribución por Internet

# Modelos animados 3D: editores

Herramientas comerciales: Maya y 3D Studio

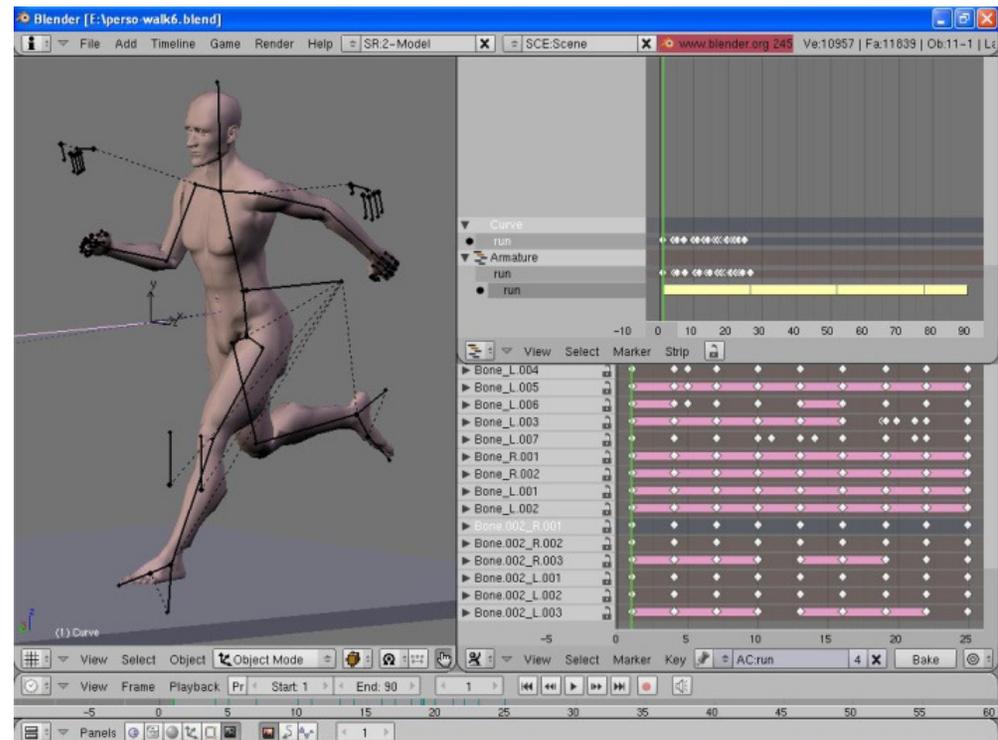
Herramientas gratuitas:

Blender

Edición y visualización de  
modelos estáticos y  
animados.

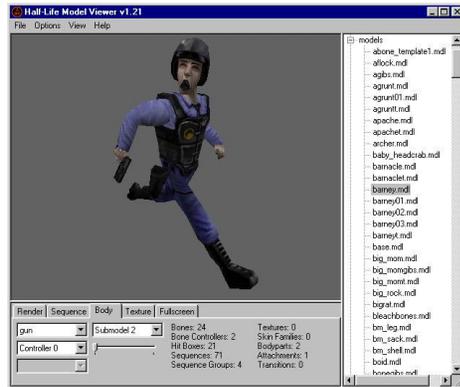
Formatos exportados:

\*.Blend, \*.MDL, etc



# Modelos animados 3D: *visualizadores* de ficheros MDL (aparte de Blender)

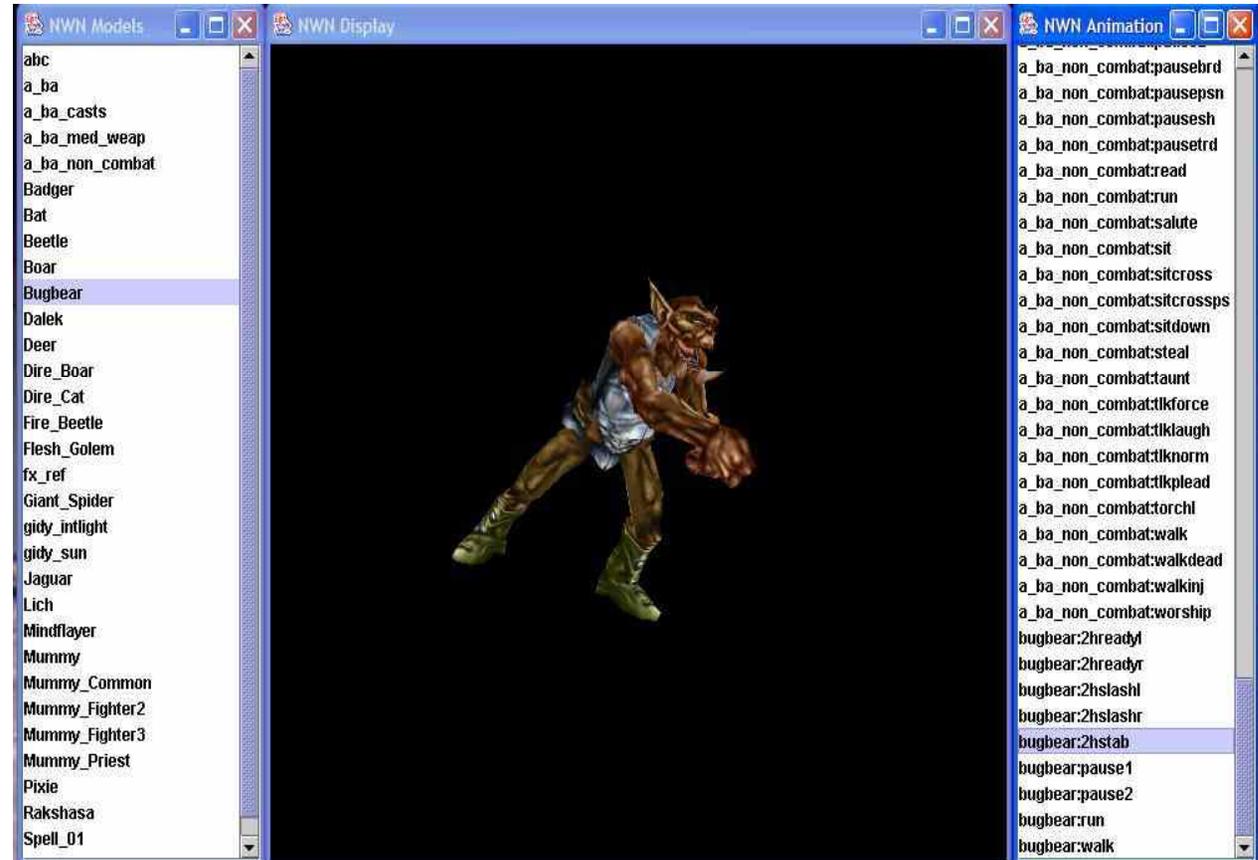
Cada fichero MDL tiene el personaje y sus animaciones:



Half-Life model viewer

personaje

sus animaciones



NWN Java 3D Model Viewer

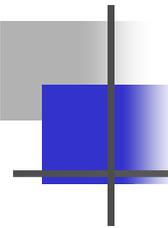
Hay también editores de ficheros MDL (ej. Valve Hammer editor)

# API 3D de bajo nivel: OpenGL

---

Un API 3D es una librería de clases para gestionar objetos 3D

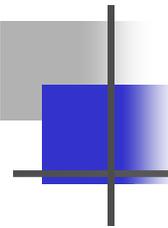
- OpenGL (Open Graphics Library), diseñada en 1992, es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.
- La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos.
- Existen extensiones de terceros para poder realizar algunos cálculos especiales, por ejemplo, para ciertas tarjetas de video 3D se cuentan con funciones especiales.
- Alternativa al OpenGL en el mundo Windows: DirectX



## API 3D de bajo nivel: OpenGL (cont.)

---

- OpenGL es el API multiplataforma mas ampliamente utilizado (Windows, MacOS X, Linux and Unix systems)
  - OpenGL hace posible la presentación en tiempo real de 3D.
  - OpenGL es estable.
  - Aunque es posible programar en OpenGL usando C/C++ o Java (JOGL), se considera como un aAPI algo complejo para trabajar directamente. Por está razón es utilizado a traves de APIs de más alto nivel como Ogre3D, Java3D, Xith3D, entre otros.
- Alternativa : DirectX / Direct3D



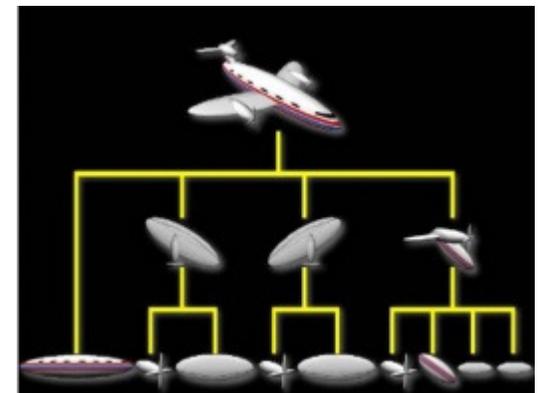
## API de bajo nivel en Windows: DirectX

---

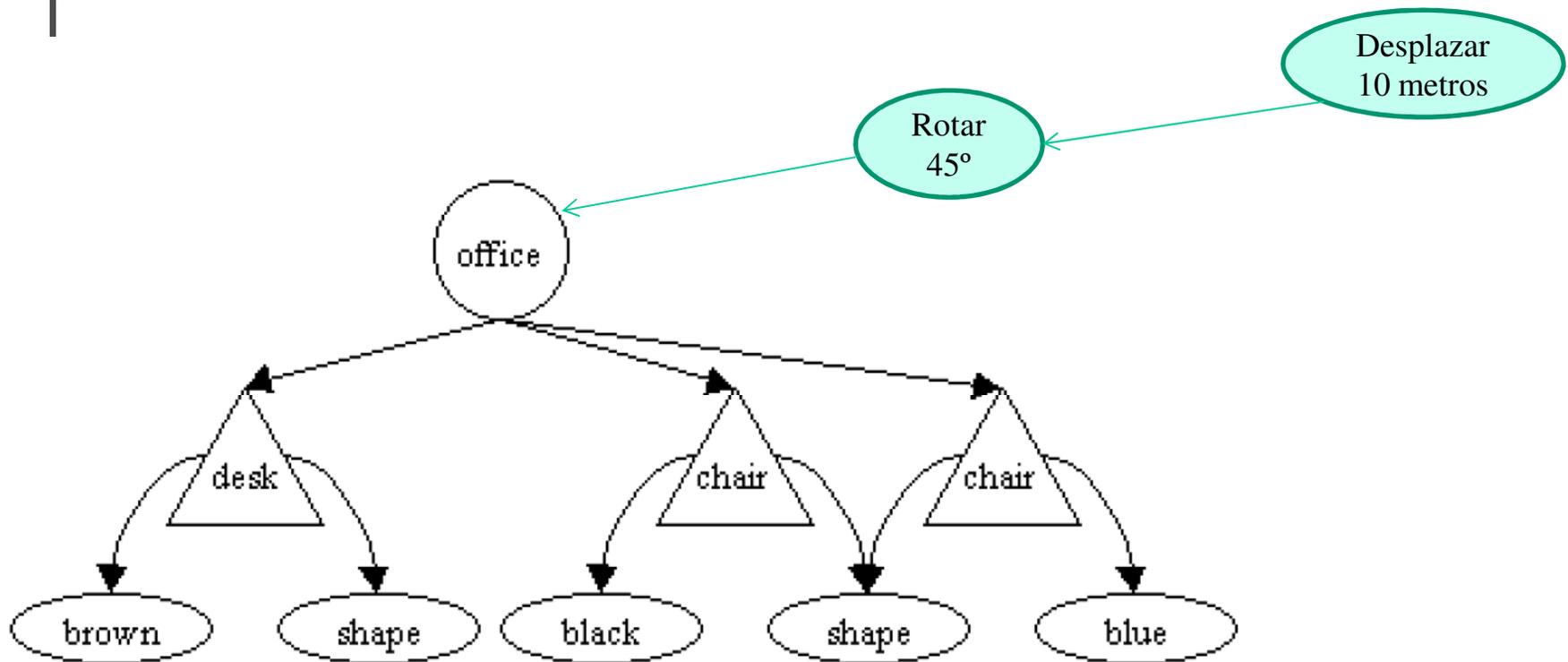
- DirectX es la base para la mayoría de los juegos desarrollados en Windows. Por este motivo, Microsoft ha diseñado un framework denominado XNA para la programación de videojuegos. Está disponible para sus consolas Xbox y Xbox 360.
- Direct3D es parte de DirectX específica para la programación de 3D.
- Se tiene acceso directo al conjunto de instrucciones del microprocesador.

# APIs de alto nivel: manejadores de grafos de escenas 3D

- Un grafo de escena es un colección de nodos en un grafo o estructura en árbol.
- Cada nodo puede tener varios hijos. Una operación sobre un grupo automáticamente se aplica a todos sus miembros.
- Concatenar varias operaciones espaciales (desplazamiento, rotación, escalado) es sencillo para realizar complejas operaciones.
  - Un grupo de objetos puede ser movido uno solo
- Un nodo puede estar relacionado de alguna con cualquier nodo del árbol
- Los lenguajes y entornos de programación serán más cómodos (fáciles de programar) si están orientados a escenas (scene-oriented)



## APIs de alto nivel: Grafo de escena 3D (cont.)

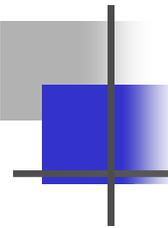


- Cada nodo puede tener varios hijos.
- Una operación sobre un grupo automáticamente se aplica a todos sus miembros.
- Una concatenación de nodos padres permiten desplazar, rotar, escalar un grupo.
- Un nodo puede estar relacionado de alguna con cualquier nodo del árbol.

## API de alto nivel Java3D

---

- Es una API para procesamiento digital de imágenes en 3D en lenguaje java, la cual basa su funcionamiento en OpenGL o DirectX.
- Es un API 3D de propósito general
- Se deriva del paquete javax.media.j3d.\* También se utilizan las clases java.awt.\*, y javax.vecmath.\*
- Orientado a grafo de escenas. El grafo de escenas permite organizar objetos por un criterio que no necesariamente es su situación espacial.



## API de alto nivel Xith3D

---

- API inspirado en Java3D, pero especializado en la creación de juegos.
- Al contrario que Java3D, Xith3D permite hacer uso de las funciones opengl, a través de JOGL.
- Requiere la importación librerías, incluidas algunas de Java3D, JavaOpen GL
- Aún no funciona sobre DirectX.
- Orientado a escenas.

## API de alto nivel Ogre3D

---

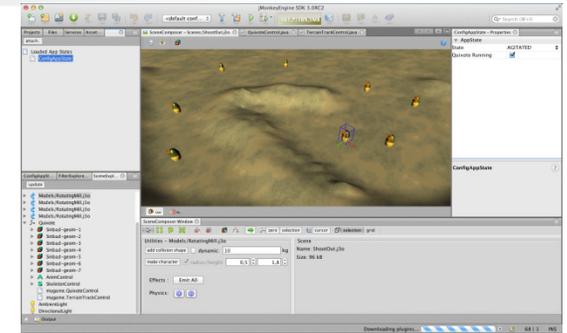
- Ogre (Object Oriented Graphics Engine) es un motor de gráficos orientado a escenas y multiplataforma.
- Es un proyecto open source
- Entre estas características está el soporte para PlayStation y exportadores para las herramientas de modelación tridimensional más usadas en el mercado.
- Ogre no es un engine diseñado solo con los juegos en mente, es un engine de graficos 3D general.
- Ogre no trae soporte nativo para sonido ni física,
- Orientado a escenas

# Motores (*engines*)

Los motores ofrecen:

- un entorno de desarrollo amigable
- herramientas para la creación de escenas
- herramientas para la creación de modelos
- los programas principales y disparadores de eventos
- librerías gráficas API 3D para gestionar objetos 3D
- librerías para permitir la física
- librerías para permitir la IA
- IDE: entorno propio o integrado en otro (ej Netbeans, Eclipse)

Ejemplos: jMonkey, Unity3D, Unreal



jMonkey

# Lenguajes de programación

---

- Para programar un videojuego nuevo es importante tener en cuenta:
  - Las capacidades del lenguaje de programación para el videojuego específico a desarrollar y su entorno de ejecución
  - La velocidad de la programación (comodidad de uso)
  - La velocidad de la ejecución
- Cualquiera de los lenguajes mencionados (LUA, UScript, C++, C# o Java) se pueden utilizar para programar videojuegos profesionales. C++/C# generan código con prestaciones algo mejores que Java.

# Lenguajes, APIs y motores (*engines*)

- Mundo MODs para PC
  - Desarrollo de videojuegos a partir de otros
  - Juego Unreal (1998) → Unreal Engine → Unreal Tournament
  - Motores: Unreal engine
- Mundo C++/C#
  - *APIs orientado a escenas*: Ogre, librerías Unity3D, librerías XNA (C#),
  - *APIs no orientado a escenas*: OpenGL para C++/C#
  - Motores: Unity3D engine, XNA engine
- Mundo Java
  - *APIs orientado a escenas*: Java 3D, Xith3D, Ogre4J, librerías jMonkey
  - *APIs no orientado a escenas*: JOGL , LWJGL para Java
  - Motores: jMonkey engine

# Mundo MOD

- Un MOD es una modificación de un videojuego. Para su desarrollo se pueden requerir herramientas específicas de la empresa que desarrolló el video juego, o conocer su lenguaje de programación.
- En el caso de videojuegos 3D para PC, algunos son “modables” es decir se pueden hacer modificaciones, y pueden requerir mayor/menor conocimiento de lenguajes de programación.
- Pueden haber herramientas MOD para modificar un videojuego:
  - Parcialmente en texturas, personajes (ej. Far Cry 4, SimCity 4)
  - Totalmente: otras MOD (ej. Valve Hammer Editor ) se pueden usar para desarrollar nuevo videojuegos a partir de otro (ej. el videojuego *Counter-Strike* es una modificación del juego *Half-Life*).
- Este paradigma ha permitido crear Engines (ej. Unreal engine)

# APIs y Motores en el mundo Java

- El bucle básico del videojuego se puede desarrollar utilizando un API 3D de propósito general (ej. Java3D u Ogre3D)
- Si se usa un Motor (Engine), el bucle básico viene ya desarrollado. El programador deberá añadir funcionalidad.

