

# Realidad Virtual aplicada a la Arquitectura:

Estudio de aplicación de tecnología para el desarrollo de videojuegos en la visualización arquitectónica

Juan López-Tarruella Maldonado

Tutores: Javier Benloch Marco  
Gustavo Salvador Herranz

Curso 2012-13

**Máster en Edificación** especialidad en Tecnología



ESCOLA TÈCNICA SUPERIOR  
ENGINYERIA  
D'EDIFICACIÓ



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

## **Agradecimientos**

A D. Javier Benloch Marco de la UPV, quien, además de ser un profesor admirable, me ha brindado todas las facilidades desde el día que le propuse ser mi tutor.

A D. Gustavo Salvador Herranz del CEU-UCH, cotutor, compañero y amigo, quien me está guiando desde mucho antes de empezar este trabajo.

A mi familia, que me ayuda en todo.

**CONTENIDO**

**1. Prólogo y objetivos .....6**

1.1 Prólogo .....6

1.2 Descripción del trabajo realizado .....7

1.3 Objetivos .....8

**2. Introducción.....9**

2.1 Breve historia de la informática gráfica .....9

El desarrollo de los aparatos de cálculo.....10

El desarrollo de los ordenadores programables..... 11

El desarrollo de la informática gráfica ..... 12

“The father of Computer Graphics” ..... 13

Las primeras investigaciones y aplicaciones industriales ..... 15

La universalización y el camino al fotorrealismo: Cine y Televisión ..... 17

Industrial Light and Magic.....18

**3 Informática aplicada a la arquitectura ..... 19**

3.1 Que es CAD.....19

CAD .....20

CAAD .....20

CAE .....20

CAM .....21

3.2 Tipos de Software utilizados en arquitectura.....22

BIM .....22

Programas de cálculo del ámbito de la construcción.....25

Modelado tridimensional .....27

Motores de render .....28

Aplicaciones 3D completas .....30

Plug-ins .....31

Postproducción.....32

Programas de visualización arquitectónica en tiempo real.....33

3.3	Proceso clásico de elaboración de una infografía arquitectónica .....	34
	Introducción .....	34
	Fases de elaboración de una infografía .....	37
	1º Delineación en 2D. ....	37
	2º Modelado 3D a partir del 2D. ....	37
	3º Aplicación de materiales al modelo. ....	39
	4º Iluminación de la escena. ....	40
	5º Renderizado .....	40
	6º Retoque digital de la imagen. ....	41
<b>4</b>	<b>Conceptos y definiciones sobre el renderizado tridimensional .....</b>	<b>42</b>
4.1	Materialidad.....	42
	Shading.....	42
	Reflexión .....	43
	Refracción.....	43
	Mapeado de textura (Texture Mapping) .....	44
	Mapeado UVW (UVW Mapping).....	45
	Mapeado de opacidad (Opacity Mapping).....	46
	Mapeado de relieve (Bump Mapping) .....	46
	Mapeado de normales (Normal Mapping) .....	47
	Mapeado de desplazamiento (Displacement Mapping) .....	48
	Mapas animados.....	48
	Mapas de procedimiento (Procedural Maps).....	48
	Material autoiluminado .....	49
	Material Cartoon .....	49
	Material SubSurface Scattering (SSS) .....	49
4.2	Iluminación .....	50
	Iluminación directa .....	50
	Iluminación indirecta .....	50
	Iluminación ambiente (ambient light).....	51
	Lightmaps .....	52
	Ambient Occlusion.....	53
	Causticas.....	53
	Image based lighting HDRI .....	54

4.3	Renderizado .....	55
	Renderizado en tiempo real.....	55
	Clipping .....	55
	Back-Side Culling .....	55
	Z Culling .....	56
	Cube Mapping.....	56
	Canales Alpha, Z-depth y otros .....	57
	Sistemas de partículas .....	57
	Físicas.....	58
	Profundidad de campo .....	58
	Desenfoque de movimiento .....	59
	Destellos de lente .....	59
	Integración de imagen generada con imagen real.....	59
<b>5</b>	<b>Realidad Virtual .....</b>	<b>60</b>
5.1	Que es la realidad virtual .....	60
5.2	Breve historia de la realidad virtual .....	61
5.3	Funcionamiento de un sistema de Realidad Virtual.....	62
5.4	Dispositivos de entrada de datos .....	63
	Dispositivos de navegación.....	63
	Dispositivos de Tracking .....	64
	Dispositivos Gestuales .....	65
5.5	Dispositivos de salida de datos .....	66
	Dispositivos de salida visual .....	66
	Representación no estereoscópica .....	67
	Representación estereoscópica .....	68
	Una clasificación de dispositivos de salida gráfica.....	70
	Localización espacial del sonido .....	71
	Dispositivos de salida auditiva .....	72
	Dispositivos Hápticos .....	74
5.6	Que es la Realidad Aumentada.....	75
5.7	Aplicaciones en ciencias de la salud .....	76
	Estudio de anatomía.....	76

Entrenamiento quirúrgico y de diagnóstico .....	76
Rehabilitación .....	76
5.8 Aplicaciones en educación y docencia .....	77
Estudios de historia del arte, arquitectura e ingeniería .....	77
Otros estudios .....	78
5.9 Aplicaciones culturales y lúdicas.....	79
5.10 Aplicaciones en arquitectura y edificación.....	80
5.11 Aplicaciones en urbanismo y paisajismo .....	81
5.12 Aplicaciones para el uso, mantenimiento y seguridad de las construcciones .....	82
5.13 Aplicaciones en Rehabilitación del Patrimonio Construido.....	84
<b>6 Game Engines.....</b>	<b>85</b>
6.1 Que es un Game Engine .....	86
Game Engines como motores de Realidad Virtual Low Cost .....	86
6.2 Algunos ejemplos de aplicación .....	88
6.3 Game Engines .....	89
CryENGINE 3.....	89
Unreal Development Kit UDK.....	90
Unity 3D .....	91
<b>7. Caso Práctico: Unity 3D. ....</b>	<b>92</b>
7.1 Comenzando a trabajar. ....	92
Interfaz de usuario .....	92
Person Controller.....	93
Cámara .....	94
7.2 Insertar contenido arquitectónico.....	94
Insertar modelos desde AutoCAD .....	95
Insertar modelos desde 3ds Max.....	98
Insertar modelos desde Sketchup.....	99
Insertar modelos desde Revit .....	100
7.3 Iluminación.....	101
7.4 Materiales .....	104

Normal shader Family .....	104
Transparent shader family .....	106
Transparent Cut Out shader family .....	106
Self-illuminated shader family.....	106
Reflective shader family .....	107
Glass refraction shader .....	107
Toon shader .....	107
Otros materiales .....	107
7.5 Entorno (terreno y vegetación).....	108
Modelado .....	108
Importación de información planimétrica .....	108
Texturizado .....	108
Creación de arbolado .....	109
Césped y elementos de detalle .....	109
7.7 Audio .....	110
7.8 Sistemas de partículas.....	111
7.9 Efectos de imagen.....	112
7.10 Interactividad – animaciones.....	113
7.11 Interactividad – físicas.....	114
7.11 Interfaz gráfica de usuario (GUI) .....	115
<b>8. Conclusiones y aspectos finales. ....</b>	<b>116</b>
8.1 Conclusiones .....	116
Sobre la informática aplicada a la arquitectura.....	116
Sobre la Realidad Virtual, Realidad Aumentada y Game Engines .....	117
Sobre la elaboración de un entorno virtual mediante un Game Engine.....	118
8.2 Interés formativo del trabajo realizado .....	120
8.3 Beneficios para la sociedad .....	120
8.4 Futuros trabajos.....	121
<b>REFERENCIAS.....</b>	<b>122</b>
<b>ÍNDICE DE IMÁGENES.....</b>	<b>126</b>

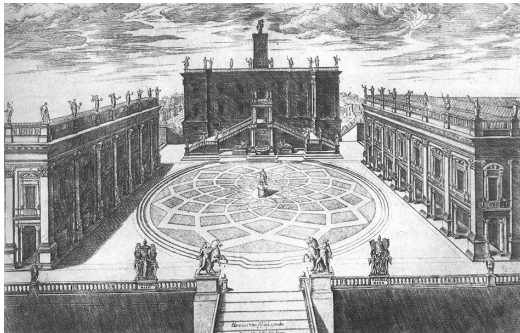


Fig 1.1- Perspectiva del diseño de Miguel Angel para la plaza del Campidoglio de Roma. 1569 Grabado de Étienne Dupérac.

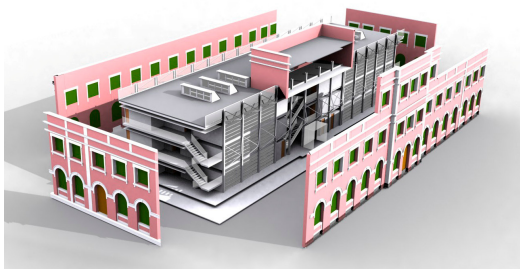


Fig 1.2- Infografía de la rehabilitación integral del Centro de Salud Valencia Malvarrosa, con fachadas proyectadas.

Imagen del autor

## 1. Prólogo y objetivos

### 1.1 Prólogo

Al igual que el escritor plasma sus pensamientos mediante palabras, el arquitecto presenta sus diseños por medio del dibujo.

Con el objetivo de transmitir las ideas de forma eficaz, además de los planos técnicos, el arquitecto se ha valido habitualmente de vistas en tres dimensiones (normalmente perspectiva cónica) para facilitar al profano la percepción de los espacios proyectados.

Con la evolución de los programas de Diseño Asistido por Ordenador (CAD) las representaciones tridimensionales arquitectónicas a mano han sido sustituidas por imágenes generadas por ordenador, debido a las ventajas que éste presenta.

Hoy en día se ha logrado un alto grado de realismo gracias a los avances de Hardware y Software. Sin embargo, hasta hace poco tiempo nuestra capacidad tecnológica solo nos permitía ofrecer imágenes estáticas o animaciones predefinidas si queríamos mantener un nivel de calidad visual.

En los últimos años, los avances en el campo de los gráficos por ordenador, debidos principalmente al desarrollo de la tecnología del videojuego, están comenzando a permitir la generación de escenarios virtuales interactivos en tiempo real.

Estas tecnologías permiten ser usadas no solo en ordenadores personales, sino en dispositivos móviles (smartphones, tablets...), videoconsolas o desde páginas web, siendo accesibles al gran público.

Además, es previsible que los avances tecnológicos afines (conectividad, acceso a la información, visión estereoscópica, sistemas de control, velocidad y capacidad de procesamiento, etc.) sufran un gran desarrollo y puedan ser aprovechados.

Siguiendo la tendencia actual, parece probable que en el futuro los entornos de Realidad Virtual sean cada vez más comunes en todos los ámbitos de la vida.





Fig 1.3- Distintos entornos 3D arquitectónicos virtuales interactivos generados mediante el motor de desarrollo de videojuegos Unity 3D.

Fuente: <http://unity3d.com/>

## 1.2 Descripción del trabajo realizado

En una Primera Parte teórica se pretende contemplar:

- Breve historia de los gráficos por ordenador.
- Descripción de los programas habitualmente usados durante el desarrollo de infografías de arquitectura.
- Descripción del proceso clásico de elaboración de representaciones arquitectónicas tridimensionales.
- Análisis de las características y prestaciones que influyen principalmente en la calidad de una representación tridimensional arquitectónica:
  - Iluminación: Raytracing, Photon Mapping, HDRI, Ambient Occlusion...
  - Materiales: mapas difusos, de opacidad, de relieve, reflexión, refracción, SSS...
  - Otros: profundidad de campo, skyboxes, mapas animados, sistemas de partículas...

En una Segunda Parte de investigación:

- Historia, desarrollo y posibilidades de la Realidad Virtual como sistema de visualización inmersiva, interactiva y en tiempo real.
- Últimos avances y expectativas en el campo de los gráficos por ordenador. Interactividad. Investigaciones actuales sobre aplicación de la Realidad Virtual en arquitectura. Evolución y tendencias del hardware y del software.

En una Tercera Parte de investigación práctica se pretenden estudiar las características y prestaciones que ofrecen actualmente los Motores 3D para desarrollo de videojuegos (game engines), como sistema de bajo coste de acceso a la Realidad Virtual.

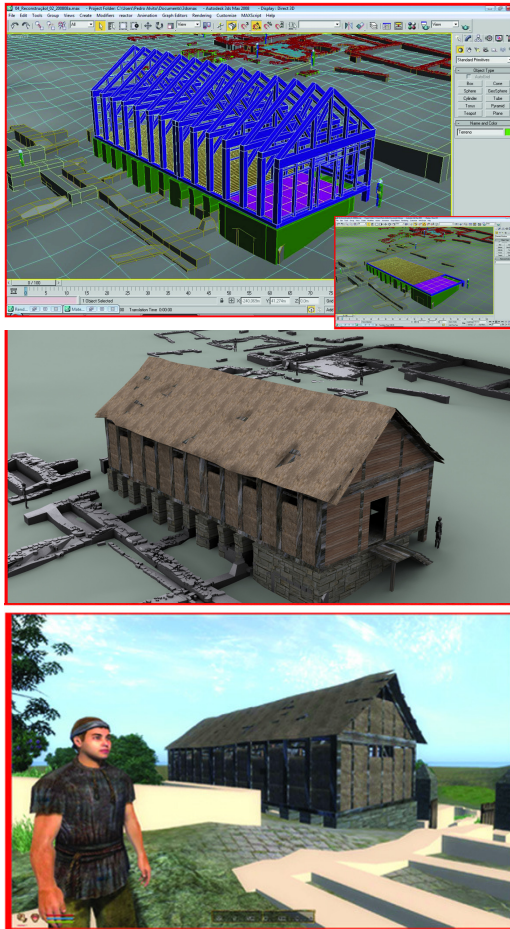


Fig 1.4- Reconstrucción arqueológica virtual de la Villa romana Casal de Freiria, usando CAD y tecnología para el desarrollo de videojuegos.

Rua, Helena y Alvito, Pedro. 2011

En esta parte experimental se probará un motor 3D concreto (*Unity 3D*) de desarrollo de Videojuegos y su interrelación con programas de diseño y modelado (*AutoCAD, Revit, Sketchup, 3dsMax...*) de uso habitual en estudios de arquitectura.

Se pretende:

- Evaluación de los procesos de inserción de modelos desde CAD y BIM.
- Evaluación de los sistemas de iluminación.
- Evaluación de los shaders de materiales.
- Evaluación de las prestaciones de audio.
- Evaluación de los sistemas de vegetación y generación de terrenos.
- Evaluación de la calidad de "inmersión" o jugabilidad: Interactividad, cinemáticas, físicas...

Posteriormente se analizará la aptitud y aprovechabilidad de las características principales de *Unity 3D* para el desarrollo de entornos arquitectónicos virtuales interactivos en tiempo real.

### 1.3 Objetivos

Se pretenden estudiar distintas maneras de integrar modelos tridimensionales realizados con programas habituales de diseño arquitectónico (CAD, BIM) en entornos virtuales mediante Motores 3D usados en el desarrollo de videojuegos y obtener conclusiones acerca de las distintas opciones, los problemas surgidos, las ventajas y las desventajas del uso de esta tecnología.

En virtud de lo analizado, concluir que campos de aplicación se pueden beneficiar del uso de este sistema de visualización arquitectónica interactiva en tiempo real, de bajo coste y gran accesibilidad.

## 2. Introducción.

En este capítulo introductorio se va a esbozar de manera resumida la evolución de la informática hasta la actualidad como aproximación a los temas de los siguientes capítulos.

### 2.1 Breve historia de la informática gráfica

Según el Diccionario de la Real Academia Española, un ordenador es una *“Máquina electrónica dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de programas registrados en ella”* (RAE, 2012).

La evolución del ordenador, o de una manera más general de la tecnología informática, ha permitido que su capacidad de cálculo pueda ser aprovechada para trabajar con elementos gráficos. Desde unos inicios en los que era un logro representar simples puntos en pantalla hemos alcanzado la capacidad de generar digitalmente mundos fotorrealistas.

Vamos a hacer un breve repaso evolutivo del desarrollo del ordenador y la informática gráfica desde varios aspectos convergentes:

- La capacidad de cálculo.
- La capacidad de programación.
- La capacidad de representación de gráficos.
- Las primeras investigaciones y aplicaciones industriales.
- La universalización y el camino hacia el fotorrealismo.

### El desarrollo de los aparatos de cálculo.

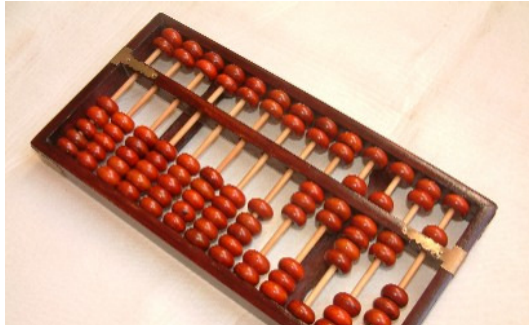


Fig 2.1- Ábaco chino

No puede señalarse una fecha exacta para el nacimiento del computador como máquina de cálculo ya que el hombre desde siempre ha necesitado resolver problemas matemáticos complejos y se ha valido de ingeniosos dispositivos desde la antigüedad. El ordenador actual no es más que la evolución de aquellos primeros aparatos.

El más antiguo antepasado del que se tiene constancia es el ábaco, comenzado a usar en Mesopotamia hacia 2700 – 2300 A.C. (Ifrac, 2001). También será usado en Egipto, Grecia y Roma.

El ábaco chino, o suanpan, que data del siglo II A.C. (Ifrac, 2001) permite ya no solo operaciones de cálculo básicas, sino también multiplicaciones, divisiones y raíces cuadradas.

Las reglas de cálculo deslizantes, desarrolladas en el siglo XVII, permitían resolver multiplicaciones y divisiones, raíces, logaritmos y operaciones trigonométricas. Han sido ampliamente usadas en ciencia e ingeniería hasta comienzos de la década de los 70 en que las calculadoras electrónicas las dejan instantáneamente obsoletas (Maor, E. 2009).

En 1642 Pascal inventa la Pascalina, la primera calculadora mecánica que permite realizar automáticamente operaciones básicas (Marguin, J 1994). Las calculadoras mecánicas se desarrollan y perfeccionan a lo largo de los siglos XVIII y XIX hasta mitad del XX.

En 1887 aparece la primera calculadora mecánica, el Comptometer, y en 1948 la calculadora Curta es el primer dispositivo calculador portátil comercializado.

La calculadora electrónica de válvulas de vacío aparece a finales de los años 1940 y es sustituida en los 50 por la tecnología de transistores, que se calienta menos (consume y se estropea menos) y tiene un tamaño menor.

En 1957 Casio comercializa la primera calculadora electrónica, el modelo 14A del tamaño de un escritorio. Una década después Texas Instruments saca al mercado la "Cal Tech", la primera calculadora personal de mano. Éste será el germen de la tecnología de microprocesadores.

Intel inventa en 1971 el primer circuito integrado comercializable, el Intel 4004, y un año después el primer microcomputador. El desarrollo de la tecnología de microprocesadores supuso una reducción de costes que disparó el desarrollo y expansión de los ordenadores personales desde finales de 1970.



Fig 2.2- Pascalina - Musée des arts et métiers

Fotografía de David Monniaux

### El desarrollo de los ordenadores programables

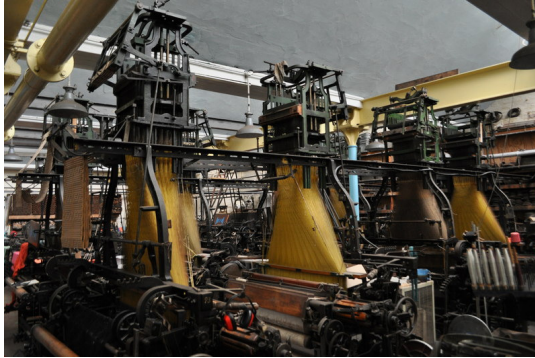


Fig 2.3- El Telar Jacquard, primera máquina « programable » mediante tarjetas perforadas.

La primera máquina que se podría definir como programable es el Telar Jacquard, aunque no sea considerado un ordenador. Inventado por Joseph Marie Jacquard en 1801, permitía preestablecer los complejos patrones de diseño que debía trazar la máquina mediante tarjetas perforadas (Hobsbawm, E 1962).

Charles Babbage, matemático e ingeniero inglés considerado “el padre de la computadora” fue quien introdujo el concepto de “computadora programable” y diseñó el primer modelo que funcionaba con tarjetas perforadas, la “Analytical Engine”. Ada Lovelace, matemática (e hija de Lord Byron) que trabajó con Babbage, desarrolló el algoritmo que permitió a la máquina calcular la secuencia de números de Bernoulli, por lo que es considerada la primera programadora de la historia.

En 1880 Herman Hollerith diseña un sistema de grabación de los datos producidos por las máquinas basado también en tarjetas perforadas. El manejo de datos del Censo de Estados Unidos de 1890, primer uso a gran escala del procesamiento de información mediante tarjetas perforadas, es desarrollado por la empresa de Hollerith, que posteriormente se convertiría en IBM.

En 1941 Konrad Zuse construye en Berlín la primera computadora programable electromecánica, la Z3, usada para hacer cálculos aeronáuticos por el German Aircraft Research Institute.

En 1943 Tommy Flowers construye la primera computadora programable electrónica, Colossus. Se utilizó para descifrar códigos encriptados alemanes durante la guerra.

En 1944 IBM junto con Howard Aiken de la Universidad de Harvard desarrollan Mark I, a los que seguirían Mark II, III y IV, para resolver cálculos para la Marina de Estados Unidos.

En 1946 la Universidad de Pennsylvania, con financiación de la Armada de Estados Unidos, desarrolla ENIAC. Debido a su potencia de cálculo mil veces superior a cualquier otra computadora de la época y a su tamaño físico es denominada por la prensa “Big Brain”. Como todas las supercomputadoras de la época, se dedica a realizar cálculos balísticos durante la Segunda Guerra Mundial pero, por su estructura, es considerada la primera computadora programable de uso general.

En 1957 IBM crea el lenguaje de programación de propósito general FORTRAN para su ordenador IBM704. Permitía dividir el tiempo de programación en 20 respecto al sistema empleado hasta la época (codificación a mano en lenguaje ensamblador). La inclusión de la capacidad de operar con números complejos resultaba, además, muy útil para aplicaciones de ingeniería y fue

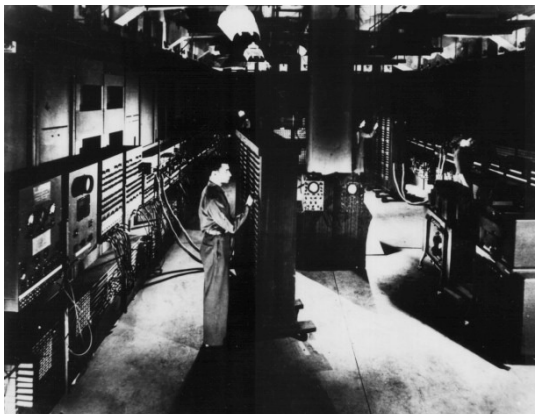


Fig 2.4- Un operario trabajando con ENIAC, “the Big Brain” en 1946.

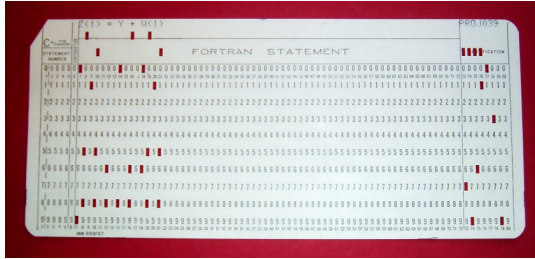


Fig 2.5- Código Fortran en una tarjeta perforada.  
Fotografía de Arnold Reinhold.

ampliamente adoptado para usos científicos. La popularidad de este lenguaje llevó a los fabricantes de ordenadores a incluir compiladores FORTRAN en sus productos. Por este motivo se considera que es el primer lenguaje de programación ampliamente extendido y soportado por distintos sistemas.

En 1960 se crea el lenguaje de programación LIST en el MIT y en 1964 hace lo propio el BASIC, creado por John George Kemeny del Dartmouth College (USA) con el objetivo de conseguir un lenguaje de programación suficientemente sencillo para ser usado por estudiantes sin conocimientos avanzados de programación.

### El desarrollo de la informática gráfica

Se considera que son gráficos por ordenador todo aquello que no es texto ni sonido. Los gráficos por ordenador nacieron para cumplir una función inicial que era la de representar la información de forma que fuera comprensible por el usuario, más allá de datos numéricos. Sin embargo el aumento de la capacidad de procesamiento de los ordenadores ha permitido abrir campos de aplicación en sectores puramente visuales, como puedan ser la industria del cine, el diseño o el videojuego.

El término “gráficos por ordenador” (en inglés “computer graphics” o “CG” es un término habitual) fue acuñado en los años 60 por William Fetter, diseñador gráfico de Boeing, para definir su trabajo (Carlson, W 2003), pero la historia comienza antes.

En 1885 ya se había inventado un tipo de válvula de vacío llamada CRT (cathodic rays tube) que permitía generar luz mediante la colisión de un electrón lanzado desde el cátodo a la superficie fosforescente del ánodo. Este tubo de CRT fue la base del primer televisor diseñado por Philo Farnsworth en 1930 y, posteriormente, la génesis del monitor.

El primer ordenador con representación gráfica de la salida de datos de la historia fue una computadora desarrollada en el MIT (Massachusetts Institute of Technology) en los años 50, la computadora *Whirlwind*. Financiada por las Fuerzas Armadas estadounidenses, el objetivo inicial era crear una computadora capaz de ejecutar un simulador de vuelo que sirviera de entrenamiento a pilotos de bombarderos.

Hasta entonces ninguna computadora había trabajado en “tiempo real”. Es decir, las computadoras existentes requerían que el investigador aportara los datos a los operarios de la máquina para que éstos los introdujeran, se procesaban y se ofrecía una salida de resultados final, muchas veces

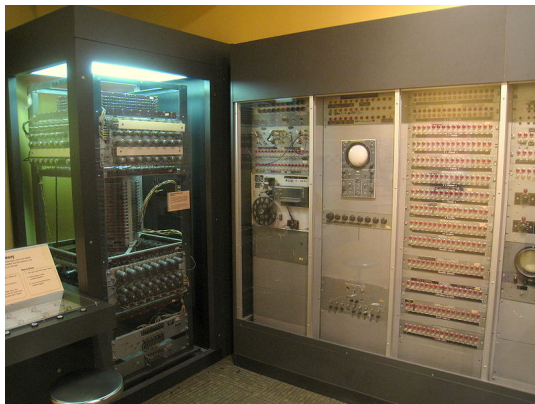


Fig 2.6- Whirlwind. La primera computadora con pantalla e introducción de datos en tiempo real mediante periférico.

Museo de las Ciencias de Boston, Massachusetts.

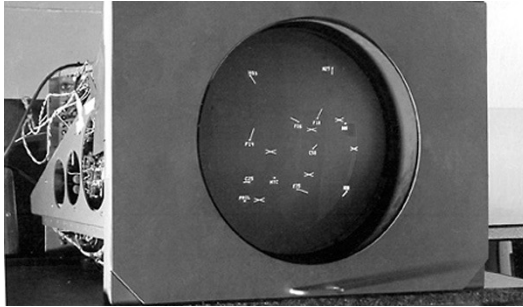


Fig 2.7- Monitor del computador Whirlwind.

después de horas (o toda una noche) de cálculos. Un simulador requería la capacidad de trabajar con datos introducidos durante la ejecución del programa y salida gráfica en tiempo real.

Whirlwind recibía datos de geoposicionamiento de una aeronave que por primera vez se pudo representar en una pantalla CRT, sobrepresionada sobre una serie de puntos que simulaban la costa de Massachusetts. Además, se diseñó un dispositivo de introducción de datos en tiempo real, un puntero, precursor de lo que ahora sería un ratón de ordenador.

Las primeras imágenes generadas por ordenador como objeto en sí mismas datan también de la década de 1950. Ben Lapovsky, pintor, matemático y artista utiliza un osciloscopio de CRT con un generador de ondas sinusoidales a las que fotografía para crear las primeras obras abstractas, que él llama "electrical compositions" (Kagan, A 1980).

### "The father of Computer Graphics"

El padre de los gráficos interactivos por ordenador es Ivan Sutherland.

Sutherland había estudiado una diplomatura y una licenciatura en ingeniería eléctrica en el Carnegie Institute of Technology y se disponía a realizar su doctorado en el MIT. Cuando buscaba temas para su tesis se topó con la computadora TX-2 del Lincoln's Laboratory del MIT.

La TX-2 no era una computadora cualquiera. Financiada (como era de esperar) por las Fuerzas Aéreas Estadounidenses, tenía 320 kilobytes de memoria (el doble que las computadoras más potentes comercializadas en la época), introducción de programas mediante tarjetas perforadas, almacenamiento de datos en cinta magnética, el primer sistema de impresión (Xerox) y un monitor de nueve pulgadas.

De forma similar a Whirlwind, disponía de un periférico de control in-situ, el *lightpen*, que permitía que el propio investigador interactuase con la máquina, en vez de necesitar un equipo de operarios que introdujesen los datos en la máquina.

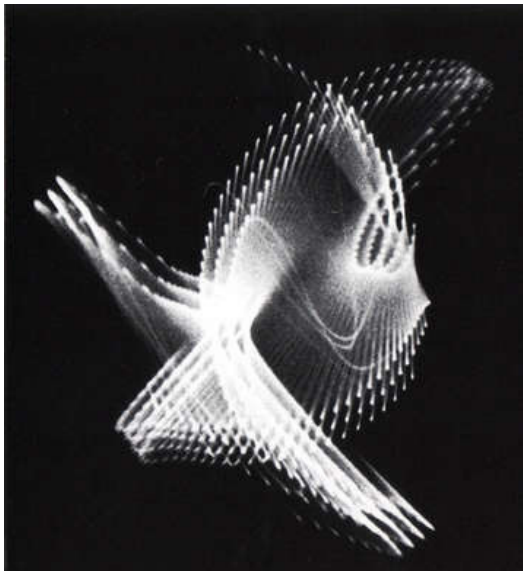


Fig 2.8- Oscillon 4, una obra de arte abstracto digital de Ben Lapovsky.

Sutherland vio con claridad las posibilidades que se abrían ante una máquina con pantalla gráfica y un lápiz óptico capaz de interactuar con ella. Trabajó sobre la TX-2 en su proyecto de tesis "*Sketchpad, A Man-Machine Graphical Communication System*" (*Sketchpad, un sistema de comunicación hombre-máquina*), que fue leída en 1963 (Sutherland, I 1963).



Fig 2.9- Sutherland trabajando en su programa Sketchpad sobre la computadora TX-2.

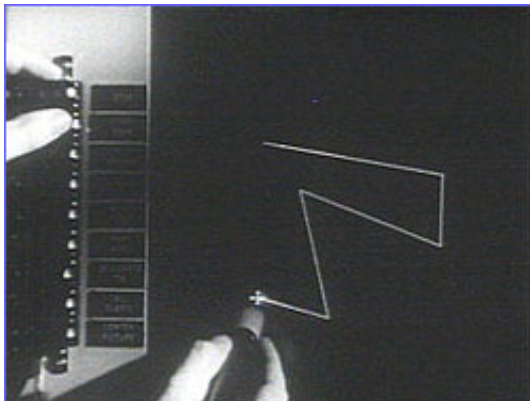


Fig 2.10- La interfaz gráfica de usuario (GUI) del Sketchpad, la primera de la historia.

Sketchpad (trad. Cuaderno de dibujo) era un programa para la TX-2 que permitía realizar dibujos técnicos perfilando directamente sobre la pantalla con el *lightpen*. Poseía algunas características muy ingeniosas que ahora son moneda común en los programas de CAD; Trabajo a escala, posibilidad de hacer zoom, rectas y encuentros perfectos, posibilidad de borrar o manipular las formas creadas, conceptos como instancia (o bloque), restricción, sistemas de coordenadas, etc....

Y la primera Interfaz Gráfica de Usuario, años antes de que existiera el término (Carlson, W 2003).

Entre 1965 y 1968 Sutherland, como profesor asociado de Ingeniería en la Universidad de Harvard inventó junto con su alumno Bob Sproull el primer dispositivo de realidad virtual (Sutherland, I 1968), una especie primitiva de gafas 3D llamadas "la espada de Damocles" probablemente de un modo irónico debido al peso de los equipos informáticos situados sobre el usuario.

Posteriormente desarrollaría el primer sistema de Realidad Aumentada, es decir, integración de imagen real con imagen digital.

Entre 1968 y 1974 fue profesor en la Universidad de Utah. Entre sus alumnos se encuentran nombres importantes de la informática gráfica, como por ejemplo Edwin Catmull, cofundador de Pixar y actual presidente de Walt Disney y Pixar Animation Studios.

En 1968 cofunda Evans and Sutherland compañía pionera dedicada a los gráficos digitales 3D.

Desde entonces ha sido profesor universitario en diversas Universidades estadounidenses y ha continuado investigando y trabajando en el mundo de la imagen digital.





Fig 2.11- DAC-1, el primer CAD comercial de la historia, desarrollado por IBM y General Motors.

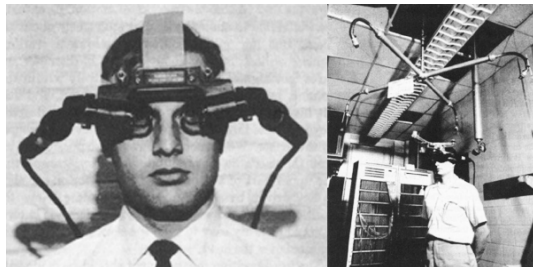


Fig 2.12- Ivan Sutherland con "La espada de Damocles", primer sistema de realidad virtual.

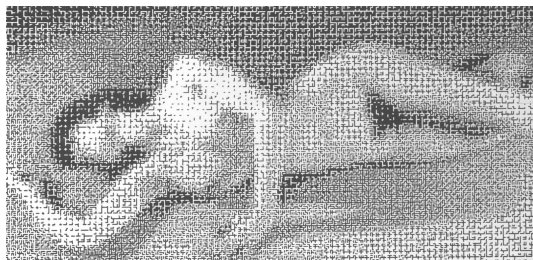


Fig 2.13- "Studies in Perception I", reconstrucción de una imagen con caracteres tipográficos. Knowlton y Harmon, 1966.

### Las primeras investigaciones y aplicaciones industriales

Se podría considerar que el primer CAD comercial fue DAC-1 (Krull, F.N. 1994).

General Motors ya era pionera en el uso de ordenadores con propósitos industriales. Desde 1952 se usaban máquinas con tecnología de tarjetas perforadas para realizar cálculos.

En 1958 se pidió al departamento de Investigación y Desarrollo que trabajasen sobre un sistema gráfico que permitiese optimizar el tiempo y minimizar errores en los diseños pues, hasta entonces, los diseños se hacían a mano por los distintos departamentos con los consiguientes problemas de coordinación cada vez que se efectuaban modificaciones en los diseños.

Lo que se pedía era un sistema que permitiera almacenar digitalmente los diseños, efectuar modificaciones, permitir rotaciones, cambios de escala y de punto de vista del modelo y, finalmente, imprimirlo.

El sistema, llamado DAC-1 (Design Augmented by Computer), fue desarrollado en colaboración con IBM en 1959. Una vez digitalizados los diseños, el programa podía convertir líneas en formas tridimensionales. Esto fue una gran innovación y generó una gran cantidad de trabajos de investigación a su alrededor.

En los años 60, con una tecnología de Hardware suficientemente desarrollada, se producen grandes avances en técnicas gráficas informáticas. Lawrence Roberts del MIT desarrolla los primeros algoritmos para ocultar líneas traseras en representaciones tridimensionales.

Ivan Sutherland propone el primer algoritmo para eliminar del cálculo las partes de un entorno tridimensional que quedan fuera del campo de visión, de manera que el procesado sea más eficiente. Como ya se ha dicho, desarrolla además sistemas de Realidad Virtual y Aumentada.

Ken Knowlton desarrolla un algoritmo que reconstruye imágenes fotográficas a partir de caracteres digitales. Nos encontramos con una aplicación artística que abre las puertas al estudio de la percepción humana.

Nelson Max desarrolla animaciones de estructuras moleculares mostrando la aptitud de la informática gráfica en la visualización científica.

En la Universidad de Utah (donde es profesor Sutherland desde el 68) se desarrollan algunos de los más importantes algoritmos y aportaciones teóricas: antialiasing (Crow), Shading (Phong y Goraud), mapeo de texturas (Catmull y Blinn), mapas de entorno y efectos atmosféricos (Blinn)...

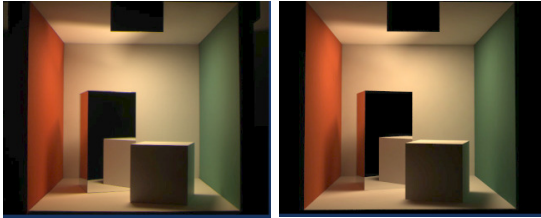


Fig 2.14- Fotografía (izquierda) e imagen de síntesis (derecha) de diversos objetos dentro de una "Cornell Box". Se realizan mediciones de luz para perfeccionar los algoritmos de los motores de render fotorrealistas.



Fig 2.15- *Texture Mapping* (superior) y *reflection mapping* (inferior) aplicados al render del T-1000 en la película de 1991 *Terminator 2*.

Copyright: Tristar Pictures.

Tras los principales avances liderados por las Universidades de Harvard, MIT y Utah, tanto el Gobierno de los Estados Unidos de América como su industria comprenden la importancia de la Investigación y Desarrollo de la informática gráfica. Se establecen fondos para impulsar los avances en este campo y se crean laboratorios e institutos de investigación en todas las Universidades norteamericanas importantes y en los años 70 se sientan las bases teóricas que se utilizan aún hoy en día.

La Universidad de Cornell (New York), que recibe su primera beca en 1973 para el Programa de Gráficos por Computador, funda un laboratorio pionero en imagen de síntesis fotorrealista. Se centran en el estudio de métodos de iluminación directa e indirecta basada en leyes físicas, dando lugar su trabajo a sistemas de renderizado (como la radiosidad) que se han aplicado posteriormente en software comercial. Uno de sus estudios más famosos es la "Cornell Box", una caja con condiciones de color e iluminación controladas que sirve de testigo para comparar distintos métodos de iluminación fotorrealista. (<http://www.cornell.edu/>).

El New York Institute of Technology recibió en 1974 generosos fondos de un inversor privado para formar su Laboratorio de Gráficos por Computador. Se compraron las máquinas más punteras de la época y se reclutó a los mejores investigadores, en gran parte antiguos estudiantes de la Universidad de Utah. El NYIT realizó grandes aportaciones para el mundo audiovisual, se desarrollaron programas de morphing (técnica de transformación de una imagen en otra, utilizada por primera vez en la película *Willow* en 1988 por Industrial Light and Magic), composición de imagen por capas (de uso habitual en películas y series para añadir fondos y elementos digitales), mapeado de texturas (utilizado en cualquier modelo tridimensional) o mapeado reflectante de entornos (usado para integrar elementos reflectantes en entornos reales, usado por primera vez en *El vuelo del navegante* en 1986).

La Universidad de Carolina del Norte se especializó en tecnología de alta velocidad de cálculo capaz de generar escenarios en tiempo real para aplicaciones científicas y médicas. Demostraron que la visualización de imágenes sintéticas puede ayudar a los científicos a comprender e interpretar datos. A partir de información suministrada por microscopio o ultrasonidos han podido generar entornos de realidad aumentada de ayuda en operaciones quirúrgicas de precisión.

El Instituto Tecnológico de California fundó su Departamento de Computer Science en 1976 reclutando a los mejores investigadores del momento, siendo Ivan Sutherland el primer seleccionado. El equipo trabajó en sistemas de render físicamente reales basados en las teorías del electromagnetismo, la cual ha desembocado en una generación de motores de render ultra realistas. También trabajaron en reflexiones anisotrópicas (la materialidad de pelo, ropa, piel...).

### La universalización y el camino al fotorrealismo: Cine y Televisión

El espectador desde siempre ha sentido la necesidad de trascender la mundanal realidad y ha buscado lo insólito a lo largo de la historia. Desde hazañas y proezas físicas y mentales hasta el mundo del misterio y el ilusionismo.

La necesidad de cautivar al espectador y atrapar su atención ha llevado a los realizadores de audiovisuales a valerse de la “pirotecnia visual”.

Quizá sea esta la razón por la que el mundo del cine y la televisión (especialmente los anuncios comerciales y videoclips musicales) siempre han estado atentos a los últimos avances en técnicas visuales y no han tardado en aplicarlos, mostrando sus posibilidades al gran público. En el caso del cine, incluso ha llegado a concebir un subgénero, el “cine de efectos especiales”.

El primer efecto especial, un efecto óptico llamado stop trick, lo descubrió George Meliés en 1896 por casualidad al detener y volver comenzar una grabación del tráfico de una calle de París. Al proyectar la película descubrió como este salto temporal “convertía” un autobús en un coche fúnebre.

Este no sería sino el comienzo de los efectos visuales, primero ópticos o fotográficos (en los que cabe destacar la figura de Ray Harryhausen) como el *stop motion* para animar criaturas, el *matte painting* o el uso de maquetas para recrear escenarios, etc.... más tarde aparecerían los efectos mecánicos (lluvias, explosiones, marionetas y animatronics...) y los efectos especiales de maquillaje.

Por último aparecerían los efectos digitales, de la mano de la informática gráfica. Aquí encontramos desde el uso de modelos digitales tridimensionales hasta la composición digital. Aunque en un primer momento los avances técnicos desarrollados con fines científicos son aprovechados por el sector audiovisual, posteriormente las tornas se invierten. Los Mass Media son un motor económico muy importante que acabará influyendo en los caminos a desarrollar por los investigadores.

Pronto aparecerán empresas dedicadas a producir hardware y software específico para la industria, e incluso compañías creadoras de efectos digitales.

Y no solo para producir efectos “detectables”. A día de hoy generar un entorno fotorrealista perfectamente integrado en la acción puede ser económicamente mucho más rentable que construir un decorado o llevar un equipo de rodaje a algún lugar remoto.

Hasta el plano macro de chocolate líquido derritiéndose sobre una galleta en un anuncio publicitario acabará siendo completamente digital para controlar todos los aspectos de la escena.



Fig 2.16- *Viaje a la luna* (1902) de George Meliés. La primera película de ciencia-ficción que usó efectos visuales.



Fig 2.17- *Jason y los argonautas* (1963). Jason lucha contra siete esqueletos animados mediante *stop-motion* por Ray Harryhausen.

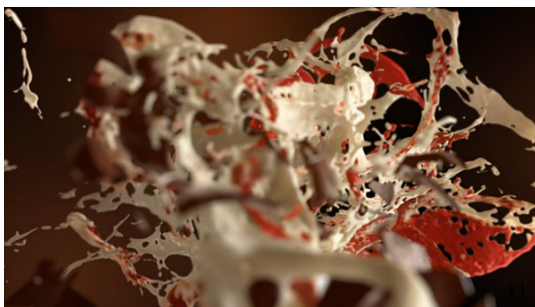


Fig 2.18- Chocolate, nata y mermelada flotan en el spot publicitario de Magnum “Temptations”, realizado por SpatialHarmonicsGroup usando el software comercial RealFlow.



Fig 2.19- Toy Story (1995), de Pixar, fué el primer largometraje animado realizado completamente por ordenador.

### Industrial Light and Magic

En 1975 George Lucas obtiene el visto bueno para comenzar la producción de la primera entrega de *StarWars*. El único problema es que no existe ninguna compañía capaz de crear los efectos especiales que requiere la historia.

Lucas decide fundar su propia empresa de efectos visuales, *Industrial Light and Magic (ILM)*, reclutando ingenieros, artistas e investigadores universitarios. El resultado mostrado en las dos primeras entregas de la saga (1977 y 1980) es algo no visto hasta la fecha, una revolución. Por ello ILM empezará a trabajar para otros directores y ganará fama de ser la principal empresa de efectos visuales del mundo (siendo, por ejemplo, la encargada de todas las películas de Steven Spielberg).

Aunque en sus inicios ILM se basaba en efectos ópticos, mecánicos y de maquillaje, en 1979 incorpora un departamento de imagen digital a cargo de Edwin Catmull, del Instituto de Tecnología de New York, que acabará convirtiéndose en Pixar, la primera compañía de animación digital, que posteriormente fue vendida a Steve Jobs y actualmente pertenece a Disney..

Industrial Light and Magic lleva ganados 16 Oscars a los mejores efectos especiales y 22 en otras categorías (<http://www.lucasfilm.com/inside/>).

Entre sus adelantos se cuentan:

- Primer personaje digital aparecido en una película, en *El joven Sherlock Holmes* (1985).
- Primer personaje digital generado en 3D, en *Abiss* (1989).
- Primeros seres vivos digitales generados en 3D fotorrealistas, en *Parque Jurásico* (1993).
- Primer largometraje animado por computador (realizado por Pixar), *Toy Story* (1995).
- Primer personaje principal completamente generado por ordenador, en *StarWars ep1* (1999).

En las últimas dos décadas ha crecido el número de grandes compañías de efectos visuales cinematográficos y de animación por ordenador.

Las principales técnicas desarrolladas en los últimos años han tenido como objetivos el máximo fotorrealismo, la perfecta inserción de imagen real y digital, la simulación de cinéticas de seres vivos y fuerzas físicas (gravedad, colisiones, abolladuras, roturas...), simulación de texturas y materiales complejos (pelo, ropa, piel, agua...), simulación de flujos de partículas y atmosféricos (polvo, nieve, lluvia, fuego...) y la interrelación actor-personaje virtual mediante *motion-capture*.



Fig 2.20- Rodaje de *Avatar* (2009 20th Century Fox) Captura del movimiento de la actriz para aplicarlo al personaje digital. Los efectos digitales están elaborados por Weta Digital.

### 3 Informática aplicada a la arquitectura

#### 3.1 Que es CAD

Nota 1:

Pese a los numerosos beneficios del CAD, éste también ha podido presentar ciertos "efectos secundarios".

Otxotorena (2011) explica como la posibilidad de generar geometrías complejas ha desembocado en una corriente arquitectónica basada en formas curvas o fractales gratuitas, edificaciones megalómanas al servicio del espectáculo, desarrolladas por lo que se ha denominado "*starchitects*" (sirvan Frank Gehry o Zaha Hadid como ejemplo).

Estas obras, de continente exuberante pero vacías de contenido, onerosas y poco funcionales han sido moda y paradigma de una sociedad "nueva rica" y así se han ubicado en ciudades que querían "estar en el mapa" (Shangay, Dubay...).

Sirva como ejemplo que Estados Unidos de America ha caído a novena posición en cuanto a "edificios más altos del mundo" frente a sus rivales asiáticos.

Nota 2:

El diseño paramétrico, basado en reglas, matemáticas y por tanto no previsible. Recuerda a los fractales de moda hace no muchos años. Invierte el proceso de lo particular a lo general. La forma final la genera el software a partir de las premisas dadas.

La informática aplicada a la arquitectura, debido a su importante componente gráfico, se ha basado principalmente en el CAD, además de los habituales programas ofimáticos de uso genérico.

Se entiende por Diseño Asistido por Ordenador o CAD (Computer Aided Design) el uso de sistemas informáticos para la creación y análisis de diseños, principalmente dibujos técnicos.

Descendientes del Sketchpad de Sutherland, inicialmente eran programas específicamente de delineación, es decir, permitían dibujar línea a línea tal como se dibujaría sobre un papel.

Al basarse en gráficos vectoriales (definidos por expresiones matemáticas) tienen intrínsecamente un alto grado de precisión, por lo que permiten geometrías perfectamente definidas, resolución de encuentros, zooms prácticamente infinitos o impresión a distintas escalas sin pérdida de calidad, por poner algunos ejemplos.

La interfaz gráfica de usuario de los programas de CAD se suele dividir en una gran área de dibujo y zonas de menús o barras de iconos de acceso a las órdenes, donde pueden distinguirse cuatro grupos:

- Creación de geometrías.
- Modificación de geometrías creadas anteriormente.
- Visualización, información e impresión.
- Gestión, configuración y ayudas al dibujo.

Las principales ventajas del uso del CAD respecto al dibujo a mano son<sup>1</sup>:

- Gran precisión de trabajo.
- Ayudas al dibujo y a la visualización.
- Ahorro de tiempo en la elaboración y modificación de los diseños.
- Mayor control y por tanto reducción de los errores.
- Facilidad de gestión, colaboración e impresión de la información.
- Creación de formas paramétricas<sup>2</sup>.

## CAD

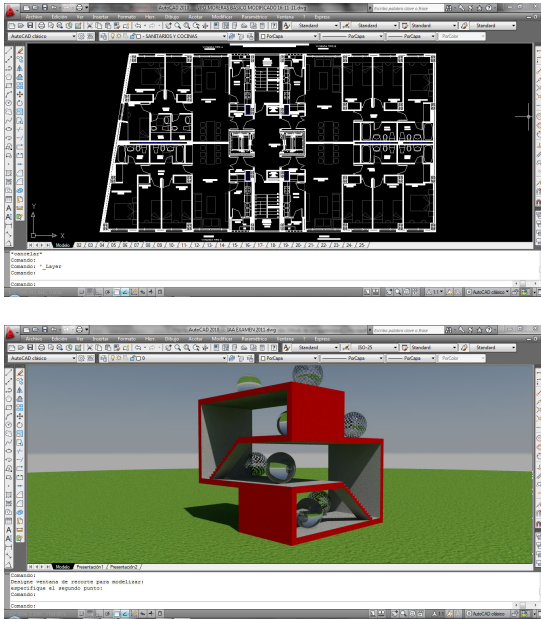


Fig 3.1- Dos capturas de pantalla del programa AutoCAD de Autodesk. En la primera se observa un plano técnico en 2D y en la segunda una representación tridimensional.

Debido al aumento de productividad y minimización de los errores, los programas de CAD se han extendido en todas las ramas del diseño industrial (automoción, aeronáutica, edificación...) que requieran planos técnicos, y han ido incorporando con el tiempo prestaciones específicas para cada ámbito (trabajo en 3D, análisis por elementos finitos, motores de render...) hasta tal punto que, actualmente, sería muy difícil delimitar donde empieza y dónde acaba el término CAD.

El término puro CAD lo podríamos entender hoy como el programa genérico definido en el apartado anterior que permita delinear prácticamente cualquier cosa.

En arquitectura estaría representado por AutoCAD, de Autodesk, que permite delinear tanto un edificio, como un mueble o un croissant (como demuestra en un artículo Enric Miralles).

Con el tiempo han aparecido otros subgéneros dentro de la expresión global CAD.

## CAAD

CAAD significa *Computer Aided Architectural Design*, traducido como *Diseño Arquitectónico Asistido por Ordenador*. Se consideran programas de CAAD aquellos que incorporan la creación explícita de objetos arquitectónicos y herramientas de ayuda al trabajo del arquitecto.

Un ejemplo sería AutoCAD Architecture (antiguo AutoCAD Architectural Desktop) de Autodesk, que permite insertar muros, puertas, ventanas y detalles constructivos, calcular directamente superficies de recintos y secciones a partir de las plantas o facilitar el renderizado de la escena.

## CAE

CAE o *Computer Aided Engineering* (Ingeniería Asistida por Ordenador) comprende generalmente los programas que basados en un entorno CAD permiten realizar análisis de los modelos, ya sean dinámicos y cinemáticos, por Elementos Finitos (tensiones y deformaciones), térmicos, de fluidos...

Ejemplos de CAE serían ANSYS, de ANSYS inc. o CATIA, de Dassault Systèmes.

Estos programas, empleados para analizar y optimizar diseños, no son de uso habitual en el sector de la construcción española, aunque en determinados casos puedan ser utilizados para estudiar estructuras singulares (tensiones, deformaciones, resistencias a impactos...).

### CAM



Fig 3.2- Imagen de un ornamento original del Banco de España en Madrid (izquierda) y de una reinterpretación (de Rafael Moneo) esculpida en piedra con tecnología de control numérico a partir de diseños CAD tridimensionales.

CAM o Computer Aided Manufacturing (Fabricación Asistida por Ordenador) comprende generalmente el uso de software destinado a dar instrucciones a máquinas de control numérico para la fabricación de objetos.

Máquinas de Control Numérico por Computador (CNC) son herramientas controladas por computador con capacidad de movimiento y giro en el espacio en dos o tres dimensiones y con distintas funciones (corte laser, hidrocorte, fresado, torno, esculpido, soldadura, pintura...).

Ejemplos podrían ser desde una sencilla cortadora laser de madera de balsa para hacer maquetas hasta un complejo brazo robot de seis grados de libertad en una línea de montaje automatizada.

La principal ventaja del CAM y la fabricación por control es poder fabricar modelos complejos generados por CAD mediante herramientas de altísima precisión.

Los programas CAM no tienen un uso directo en los estudios de arquitectura pero, en los últimos años sus aplicaciones industriales sí que han tenido cabida en ciertos elementos constructivos, casi siempre envolventes, en los que se ha aprovechado esta tecnología para incorporar diseños (paneles grabados, celosías perforadas...).

Un paso más allá lo encontramos en la ampliación del Banco de España en Madrid, realizada por Rafael Moneo. Apostando por el mimetismo con el edificio original adyacente (del siglo XIX), decide reinterpretar las esculturas y motivos figurativos de la fachada con ornamentos realizados en piedra mediante tecnología de control numérico.

### 3.2 Tipos de Software utilizados en arquitectura.

En este apartado se va a hacer un breve repaso de los tipos de software específicos que se pueden encontrar en un estudio de arquitectura o infoarquitectura además del CAD. El listado está elaborado desde el ámbito de la informática gráfica aplicada a la arquitectura por lo que no se enumerarán los programas de ofimática general ni ciertos programas específicos del sector de la construcción cuya componente gráfica es secundaria, como puedan ser los softwares de elaboración de presupuestos o de plannings temporales.

#### BIM

El concepto de BIM (o Building Information Modeling) se refiere tipo de programas orientado a la arquitectura, ingeniería y construcción que podría ser considerado el sucesor del CAD, ya que va un paso más allá en cuestión de coordinación y productividad.

En CAD, que como se dijo no es más que una herramienta de delineación, una serie de líneas independientes forman un dibujo (una planta, por ejemplo), otro conjunto de líneas independientes forman otro dibujo (un alzado), y así sucesivamente. Todos estos dibujos (planta, alzado, etc....), que también son independientes entre sí, forman la documentación gráfica del proyecto.

Es decir, una grandísima cantidad de elementos gráficos independientes acaban conformando un proyecto arquitectónico. Esto conlleva algunas desventajas con las que han lidiado los arquitectos durante años:

- Gran cantidad de tiempo no es dedicada al diseño sino a la delineación, lo cual afecta a la calidad arquitectónica y a la productividad.
- Como consecuencia de lo anterior, cualquier modificación posterior (siempre hay) implica mucho tiempo de redelineación de las distintas vistas. Igualmente, cualquier modificación implica mayor riesgo de errores y falta de coherencia entre planos.
- Al ser cada objeto gráfico independiente y sin “consciencia” de su función, no hay ninguna información automatizada que pueda ofrecer el programa. La acotación, el superficiado, elaboración de secciones, etc. serán trabajos mecánicos que deberá asumir el proyectista o delineante más o menos manualmente, otra vez con la consiguiente pérdida de productividad y riesgo de errores.

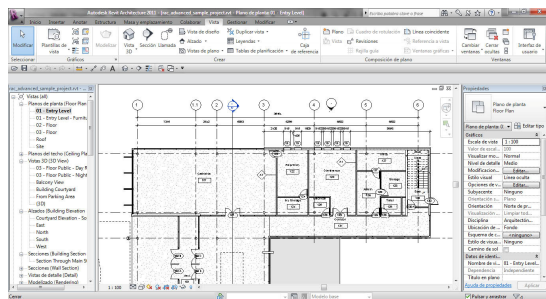


Fig 3.3- Captura de pantalla del programa BIM Revit de Autodesk mostrando un plano de planta.



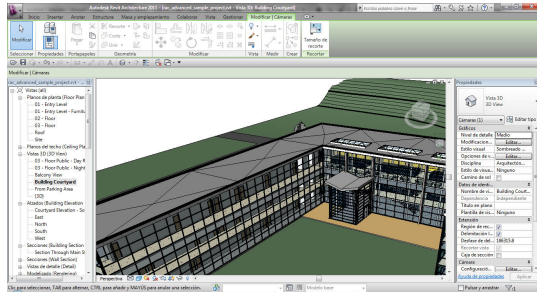


Fig 3.4- Captura de pantalla del programa BIM Revit de Autodesk mostrando una vista en perspectiva del modelo.

Con la evolución de los programas de CAD, y especialmente de CAAD, se han ido incorporando ciertas ayudas al dibujo (bloques paramétricos o con atributos, incorporación de hojas de cálculo, plug-ins de terceras compañías...) que permiten, en cierta manera, automatizar algunas tareas. Sin embargo el problema es intrínseco a la forma de trabajo del CAD, basada en la delineación.

El BIM, por el contrario, trabaja de forma totalmente opuesta. Un programa BIM es un gestor de información de elementos constructivos. Cada elemento tiene propiedades exclusivas, y todos los elementos interrelacionan de forma lógica para crear un solo modelo, un proyecto global, del que se podrá obtener la información necesaria, sea ésta en forma de vistas o de listados.

Por ejemplo, cuando se inserta un muro, el programa “sabe” que es un muro, “sabe” como tendrá que graficarse en planta, alzado o sección, “sabe” como tendrá que interrelacionar con otros elementos (frentes de forjado, otros muros o carpinterías), “sabe” la composición por capas del muro y su superficie por lo que podrá darnos mediciones exactas, y “puede saber” (si se le dice) cual es el peso, la resistencia al fuego, el precio por metro cuadrado o el color de acabado, y posteriormente, toda esta información puede ser utilizada.

La creación del modelo puede hacerse desde distintas vistas (plantas, alzados, secciones...) y, aunque parezca que se está “delineando” como en CAD, se está introduciendo información de elementos constructivos tridimensionales.

Algunas características de los BIM son:

- Se crea un modelo 3D íntegro y de él se obtienen las distintas vistas. Esto implica que se podrán obtener todas las vistas necesarias indicando simplemente el plano de corte (secciones, alzados escorizados...) y que todas las vistas serán perfectamente coherentes con el proyecto.
- Cualquier modificación en el modelo se plasma instantáneamente en todas las vistas, sin pérdida de tiempo ni fallos.
- Se puede trabajar con restricciones paramétricas. Por ejemplo, se puede indicar como han de ser los encuentros muro-forjado, o a qué distancia ha de quedar un muro bajo el forjado así que, al modificar éste último el otro adapta su forma.
- Las características de los elementos constructivos también se pueden modificar posteriormente. Un muro genérico de 25cm se puede sustituir por un muro de 35 cm de espesor, de forma que todos los encuentros e intersecciones se redibujan correctamente.

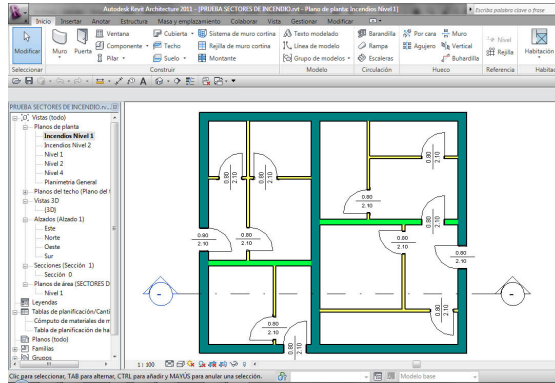


Fig 3.5- Captura de pantalla del programa Revit de Autodesk mostrando la vista en planta de una construcción sencilla en la que se ha indicado que se rellene con distintos colores los muros según su resistencia al fuego.

- El programa puede reconocer los distintos recintos, sus superficies útiles y construidas, su volumen, etc. y plasmarlo en tablas.
- Se pueden obtener todo tipo de listados a partir de la información disponible del modelo, mediciones, pesos, unidades de carpintería, etc.
- No se dibuja directamente, se dan órdenes de representación de la información del modelo. Por lo tanto, se puede indicar el tipo de grafismo, que objetos son vistos u ocultos, que líneas se han de proyectar, si se han de macizar los elementos seccionados... Se puede elegir incluso el tipo de grafismo a partir de información de los elementos (tipología, resistencia al fuego...).
- La información del modelo puede ser aprovechada por programas afines. Por ejemplo el programa BIM Revit Architecture de Autodesk interrelaciona nativamente con Revit Structure para el diseño de la estructura o con Revit MEP para el diseño de las instalaciones. Toda la información está comprendida en un único modelo arquitectónico tridimensional de manera que se pueden detectar colisiones o interferencias entre elementos estructurales, instalaciones y los acabados.
- Como efecto colateral, trabajar con modelos tridimensionales, la realización de infografías es inmediata.

Todas estas características que redundan en la productividad, la minimización de errores y el trabajo colaborativo están llevando al sector de la construcción a hacer la conversión desde el CAD.

Los grandes estudios ya utilizan el BIM, y en Estados Unidos de América y ciertos países de Europa ya es habitual su uso. A nivel de investigación también está habiendo un incremento destacable de líneas de investigación y artículos sobre el tema.

En este sentido ya se empieza a hablar de 4D-BIM al software que incorpora la dimensión temporal, 5D-BIM al que añade los factores económicos y 6D-BIM al que además coordina información de todos los agentes intervinientes durante los ciclos de vida de proyecto, obra y uso.

En España no se ha popularizado su uso práctico, quizá debido a que exceso de trabajo que derivó del Boom Inmobiliario era incompatible con el aprendizaje que requiere este tipo de programas para el usuario habitual de CAD. Sin embargo estos últimos años de crisis están siendo beneficiosos en la formación y reciclaje de los técnicos, por lo que se prevé un continuo crecimiento de usuarios.

Entre los programas comerciales más habituales cabe destacar el ya mencionado Revit de Autodesk, Allplan de Nemetschek y ArchiCAD de Graphisoft (considerado el primer BIM).

### Programas de cálculo del ámbito de la construcción

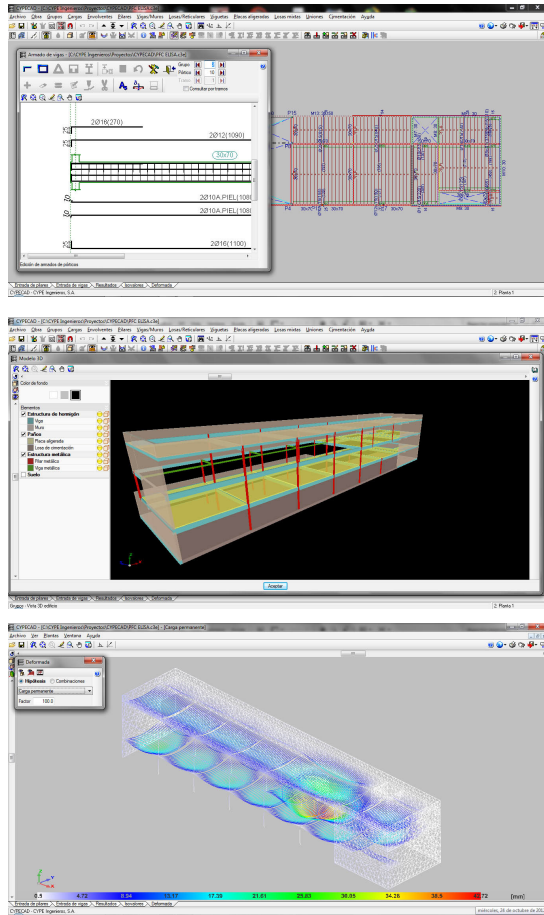


Fig 3.6- Capturas de pantalla del programa CYPECAD mostrando el área de trabajo y solución de armado de una viga (superior), visualización tridimensional de la estructura (centro), y deformaciones para una hipótesis de cálculo (inferior).

En este apartado nos referimos a los programas de cálculo de estructuras y de instalaciones que son usados habitualmente en el sector de la construcción.

Aunque en algún aspecto estos programas puedan ser considerados CAE, descrito anteriormente, se les ha diferenciado por el siguiente motivo:

Por CAE se suele entender software en el que es posible modelizar entidades para analizar sus propiedades físicas (tensiones, deformaciones, temperaturas en los distintos puntos del sólido) mediante métodos matemáticos como el de elementos finitos. Suelen ser programas con una base física y matemática genérica capaz de simular distintos aspectos de la realidad para tomar decisiones de optimización de los diseños.

Sin embargo, el software de cálculo (o diseño y cálculo) usado en el sector de la construcción, suele estar programado para un fin específico y cerrado, para presentar soluciones concretas cumpliendo una normativa en particular.

Estos programas suelen ser de mucha utilidad (están diseñados para ello) si se usan dentro del ámbito de aplicación habitual del sector pero, por el contrario, no suelen admitir modelizaciones singulares no contempladas en el programa.

Debido a que en España la *Ley de Ordenación de la Edificación* y el *Código Técnico de la Edificación* establecen que documentos componen un proyecto y como han de comprobarse los requisitos a cumplir, la mayoría de estos programas están diseñados para elaborar dichas justificaciones.

En primer lugar estaría el software de cálculo de estructuras, que generalmente permite la introducción de bases de cálculo según indica la norma, el modelado de la estructura y la introducción de cargas. Posteriormente realiza el cálculo, propone soluciones de armado (si se trabaja con hormigón) o secciones que cumplan los esfuerzos requeridos y presenta los resultados (planos y listados) en el formato concreto que se usa en el sector.

Con este software es muy rápida y cómoda la redacción de un proyecto de estructura habitual, mientras que puede ser imposible la modelización y cálculo de elementos singulares como un arco de hormigón en masa de sección triangular porque, sencillamente, no está en la base de datos del programa.

En este apartado cabría citar CYPECAD de Cype ingenieros S.A. y Tricalc de Arktec.

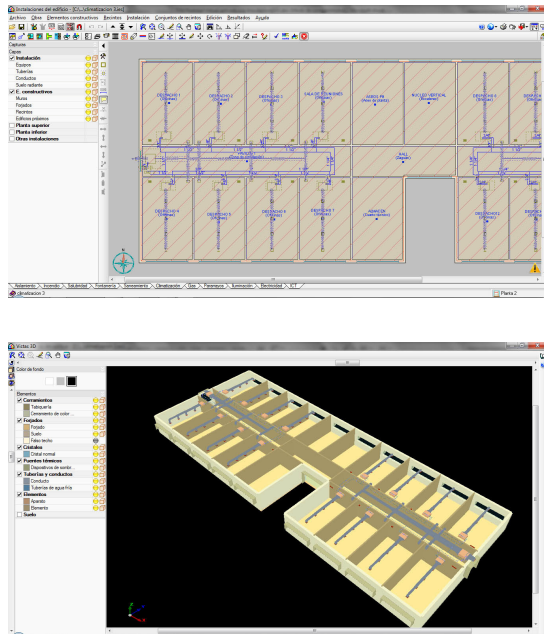


Fig 3.7- Captura de pantalla del programa CYPE Instalaciones del Edificio, mostrando el diseño en planta de una instalación de climatización (arriba), y la visualización tridimensional con dimensionado de conductos de la misma (abajo).

En el otro gran grupo se encuentra el software de diseño y cálculo de instalaciones. En este caso la oferta es mucho más amplia al cubrir todo el rango de instalaciones posibles:

- Electricidad.
- Iluminación.
- Fontanería y saneamiento, incluyendo instalaciones hidráulicas de protección contra incendio.
- Calefacción y climatización.
- Instalaciones de paneles solares para ACS y fotovoltaica.
- Telecomunicaciones.

Cabría reseñar en este grupo la suite de programas de CYPE Instalaciones del Edificio que, dentro de una misma interfaz gráfica, permite calcular las distintas instalaciones mencionadas. A modo de ejemplo, otro software sería DIALux (estudios lumínicos), VPCLima (instalaciones de climatización), Dmelect (instalaciones eléctricas)...

Habría que mencionar un tercer grupo de programas que han surgido al calor del CTE, que realizan los cálculos necesarios para justificar los apartados de los Documentos Básicos que así lo requieran, y que adoptan directamente los métodos de cálculo indicados en la normativa:

- DB-HE1: Limitación de la demanda energética. Cabría considerar en este apartado el propio LIDER, programa oficial para justificar por el método general, y toda la serie de conversores o exportadores desde otras aplicaciones, que solventan la molestia de modelar en LIDER.
- Calificación energética: CALENER (programa oficial de calificación) y exportadores desde otras aplicaciones.
- DB-HR: Protección frente al ruido: Programas de cálculo y justificación mediante el método general.
- DB-HS3: Calidad del aire interior: Software de dimensionamiento de los elementos que componen la solución de ventilación.
- DB-SI Seguridad en caso de incendio: Software de justificación de cumplimiento en los aspectos no concernientes a instalaciones hidráulicas.
- DB-SU8 Seguridad frente al rayo: Cálculo y justificación de las instalaciones de protección.

### Modelado tridimensional

Se entiende como modelado tridimensional el trabajo con sólidos y superficies que permite, a partir de formas iniciales o primitivas, y mediante ciertas operaciones (extrusiones, revoluciones, adiciones y sustracciones booleanas...), dar forma a objetos complejos.

En sus orígenes, los programas de CAD arquitectónico usados habitualmente no tenían la capacidad de trabajar en tres dimensiones y cuando se les implementó, comprendía funciones rudimentarias. Por este motivo, lo frecuente a la hora de elaborar modelos tridimensionales para su renderizado, era utilizar programas específicos de modelado tridimensional.

Con el tiempo, los programas de CAD más usuales han mejorado notablemente sus capacidades de modelado tridimensional e incluso los BIM, como se ha visto, trabajan nativamente en tres dimensiones. Sin embargo, los programas orientados a la arquitectura siguen adoleciendo de prestaciones para el trabajo con modelos complejos o de formas orgánicas. Por ello siguen teniendo cuota de mercado los programas específicos de modelado tridimensional.

Se enumeran algunos programas populares:

*Rhinceros*, de *Robert McNeel & Associates*, se especializa en modelado libre mediante curvas NURBS por lo que es idóneo para objetos con formas redondeadas, habitual en diseño industrial, automotriz, naval o ciertos tipos de arquitectura.

*Mudbox*, de *Autodesk*, o *Zbrush*, de *Pixologic*, son programas de escultura digital. Permiten esculpir y pintar a golpe de ratón una especie de "arcilla digital". Su uso habitual es el modelado de seres vivos como personajes o seres de fantasía.

*Bryce*, de *DAZ3D*, *Terragen*, de *Planetside software*, o *VUE*, de *e-on software inc.*, son modeladores de terrenos y entornos naturales basados en fractales. *VUE*, por ejemplo, ha sido utilizado por *ILM* para generar escenarios naturales para la saga *Piratas del Caribe*.

*Sketchup*, de *Google*, es un modelador de formas simples que ha adquirido muchísima popularidad debido a que es gratuito, su uso es muy sencillo e intuitivo y tiene una librería de objetos disponible (edificios, mobiliario, medios de transporte...) aportada por los usuarios prácticamente ilimitada.

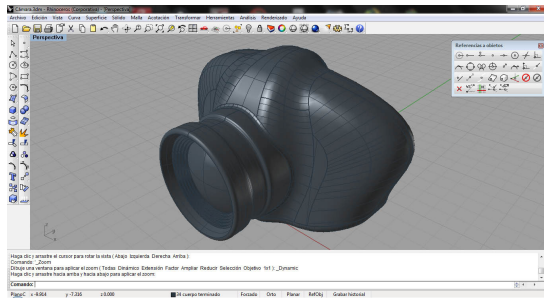


Fig 3.8- Captura de pantalla del programa Rhinoceros, especializado en el uso de curvas NURBS. Habitual en Diseño Industrial.

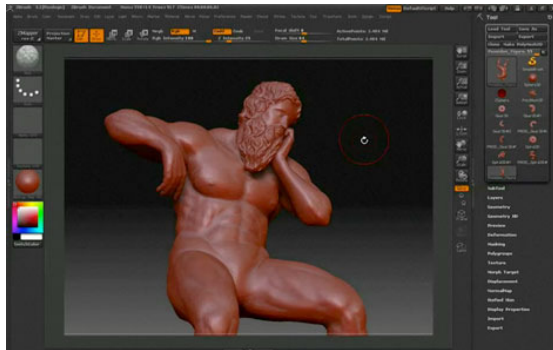


Fig 3.9- Captura de pantalla del programa Zbrush, de escultura y pintura digital. Idoneo para el modelado de personajes.

### Motores de render

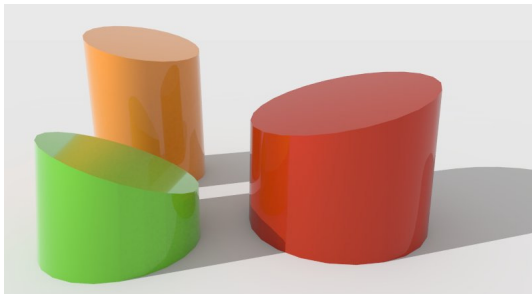
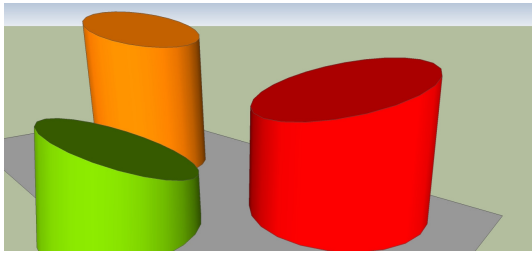


Fig 3.10- Imagen creada con el modelador Sketchup (arriba) y posteriormente renderizada con el motor Vray (abajo), incorporando iluminación directa e indirecta y reflexiones.

Renderizar significa generar una imagen virtual a partir de un modelo tridimensional.

Como se ha visto, los programas de modelado permiten crear geometría tridimensional pero, para obtener imágenes de dicha geometría hace falta renderizar el modelo. En este punto podemos encontrar varios casos:

- Algunos programas de modelado no incluyen motores de render, así que la geometría obtenida se exporta a algún programa específico de renderizado. Por ejemplo, *Sketchup* no renderiza, pero puede exportar a *Kerkithea* o *Fryrender* para generar imágenes realistas.
- Otros modeladores tienen opciones de renderizado limitadas, por lo que también se suele exportar la geometría a un renderizador potente. Por ejemplo, *AutoCAD* incorpora el motor de render *MentalRay* con características básicas, así que para imágenes muy efectivas la geometría se suele llevar a *3ds Max* que incluye características avanzadas de renderizado.
- También se puede encontrar en el mercado motores de render que funcionan como plug-ins es decir, añadidos, dentro de otro programa. Por ejemplo, el conocido motor de render *Vray* se puede instalar como plug-in en *Sketchup* para poder crear imágenes muy realistas usando éste último.
- Por último, existen programas integrados o suites, que incluyen modeladores, motores de render y muchas otras opciones, como se verá en el próximo apartado.

Por lo tanto, un motor de render es el programa, o la parte del programa, que se encarga de calcular, por distintos métodos, como se vería la geometría en el mundo real.

Los métodos y técnicas de renderizado se verán más detalladamente en el próximo capítulo, dada su importancia en el tema de estudio. Sin embargo se enumeran las funciones básicas de un motor de render genérico:

- Definición de la iluminación de la escena (situación, potencia, propiedades...).
- Definición de los materiales en escena (textura, relieve, opacidad, reflexión...).
- Definición de condiciones ambientales y fotográficas (fondo, desenfocados...)
- Cálculo (renderizado) de la escena a partir de los datos definidos.



Fig 3.11- Imagen generada con 3ds Max y renderizada con su motor MentalRay. Es una prueba de materiales con iluminación HDRI simulando el pasaje de los Uffici en Florencia.

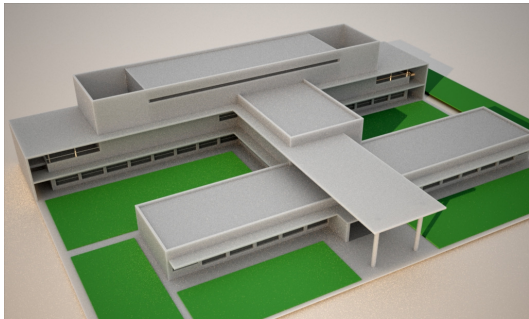


Fig 3.12- Imagen generada con AutoCAD, exportada a Sketchup y renderizada con el motor unbiased Fryrender. Es una prueba de material Sub Surface Scattering que simula la difusión de la luz de la cera, la piel o ciertos plásticos.

Algunos de los motores de render más populares son:

- Mental Ray, de Nvidia: Renderizador fotorrealista que viene incorporado de serie dentro de muchas aplicaciones, como por ejemplo AutoCAD, Revit, 3dsMax y Maya de Autodesk, Solidworks o CATIA. Trabaja con iluminación directa (por Raytracing) e indirecta (photon mapping y final gathering). Películas como Matrix Revolutions o Star Wars ep II: El ataque de los clones han usado Mental Ray para crear sus efectos visuales.

- Vray, de Chaos Group: Considerado el renderizador más usado en infoarquitectura, es una aplicación instalable en la mayoría de los programas de CAD y 3D: Rhinoceros, 3dsMax, Maya, Blender o Sketchup.

En los últimos tiempos están alcanzando notoriedad los motores de render *unbiased*. Unbiased implica cálculos puramente basados en la física de la luz, sin usar aproximaciones o artefactos que alteren los cálculos físicamente reales. En teoría, si se les deja tiempo suficiente (estos motores son más lentos) el resultado debería ser indistinguible de la realidad.

En este apartado encontraríamos:

- Indigo Renderer, de Glare Technologies: Motor físicamente real que acepta nativamente profundidad de campo, reflexiones y refracciones, causticas y efectos espectrales de descomposición de la luz.

- Fryrender, de Random Control: Otro motor unbiased que, como Vray, se puede instalar dentro de la mayoría de programas de modelado 3D.

- Maxwell Render, de Next Limit Technologies: El primer renderizador unbiased, desarrollado por un equipo español. Compatible con la mayoría de programas de modelado 3D.

Por último, se enumeran tres motores de render gratuitos:

- Kerkithea: Motor biased – unbiased gratuito compatible con el modelador tridimensional Sketchup (también gratuito).

- Luxrender: Motor unbiased gratuito y de código abierto con prestaciones muy avanzadas.

- Yafaray: motor gratuito y de código abierto compatible con Blender, programa de modelado y animación tridimensional gratuito.

### Aplicaciones 3D completas

Se entiende por aplicaciones completas, soluciones completas o Suites 3D al software que incorpora distintas herramientas para lograr un producto final prácticamente acabado. Pueden incluir:

- Modelador 3D, y opciones de importación desde otras aplicaciones.
- Sistemas de iluminación y materiales.
- Herramientas de animación.
- Herramientas para la animación de personajes, como rigging, skinning y cinemáticas inversas.
- Aplicaciones específicas como:
  - Sistemas de partículas.
  - Sistemas de fluidos.
  - Sistemas físicos (gravedad, colisiones...)
- Motor de render, y posibilidad de incorporar nuevos motores de render.
- Composición o edición de video.

Estos programas suelen ser complejos de dominar (o de dominar todo su potencial) y suelen expresar sus prestaciones en el mundo de la animación profesional. Sin embargo, algunos de ellos han sido usados en el mundo de la infografía arquitectónica cuando se ha requerido niveles de calidad que un programa de CAD habitual no es capaz de ofrecer. Destacan:

- *3ds Max*, de *Autodesk*: Programa de modelado, animación y renderización de completísimas características. Históricamente se ha convertido en el software de visualización arquitectónica por excelencia, en un principio desarrollando todas las fases de la infografía y, cuando los programas de CAD adquirieron la suficiente capacidad de modelado tridimensional, encargándose del texturizado, iluminación y renderizado. No obstante, sus posibilidades van mucho más allá de lo que se aprovecha en la visualización arquitectónica.
- *Blender*, de *Blender Foundation*: Software que está ganando gran popularidad al ser gratuito (mientras que sus competidores directos cuestan miles de euros) y de código abierto, lo que reporta mejoras y actualizaciones constantes ya que cualquier persona puede colaborar en el perfeccionamiento del programa.



Fig 3.13- Fotogramas del cortometraje *The Third and the Seventh*, del español Alex Roman.

Homenaje a la fotografía y la arquitectura realizado con 3ds Max y renderizado con Vray.



### Plug-ins

El término plug-in es un neologismo tomado del inglés que significa “enchufable” o “conectable”.

En informática el término hace referencia a programas, utilidades o herramientas diseñadas para ser utilizadas dentro de otra aplicación, pudiendo ser desarrolladas por el creador de la aplicación o por empresas independientes. En los apartados anteriores se han ido viendo algunos ejemplos dispersos y, aunque los plug-ins tienen infinidad de aplicaciones, aquí se reseñarán los más comunes dentro del campo de la informática para la arquitectura:

- Ayudas a la elaboración de documentación del proyecto de arquitectura: En este apartado se encuentran plugins que funcionan dentro de CADs genéricos incorporando ordenes de ayuda a la elaboración del proyecto arquitectónico.

*Proyecto Da Vinci*, de *Urbicad*, es un plugin para AutoCAD basado en la inserción de bloques paramétricos que automatiza la delineación de cimentaciones y estructuras, muros, carpinterías, equipamiento y mobiliario, señalización, urbanismo, detalles constructivos y 350 patrones de sombreado.

*AsuniCAD* ha desarrollado también diversos productos para AutoCAD: *Lands* facilita la elaboración de proyectos de paisajismo y jardinería. *LT Express 2* incorpora 100 ordenes nuevas de ayuda a la gestión de capas, dibujo, textos, superficies y cotas.

- Exportadores: Permiten exportar información entre distintos formatos de archivo. Por ejemplo, *vpLIDERCAD*, del Departamento de Termodinámica Aplicada de la UPV, permite transformar geometría generada con AutoCAD en información legible desde LIDER.

- Utilidades específicas de 3D: Aplicaciones completas 3D como 3ds Max facilitan la expansión de sus funcionalidades mediante plugins específicos como *Realflow*, de Next Limit Technologies, para trabajar con fluidos, *FumeFX*, de Afterworks, para simular humo o VUE, de e-on technologies, para generar vegetación.

- Motores de render: Como ya se explicó, muchos motores de render funcionan como plugin incorporándose a los programas de modelado 3D habituales.

### Postproducción

Normalmente, las imágenes o animaciones digitales no se consideran terminadas tal como son generadas por el motor de render, necesitan un acabado final. Estas últimas correcciones pueden comprender:

- La corrección de valores de luminosidad, contraste o color (etalonaje).
- La inserción o eliminación de elementos en la imagen generada (personajes, fondos...).
- La adición de efectos atmosféricos, lumínicos o desenfoques.
- La aplicación de filtros de efectos artísticos (simulación de acuarelas, lápiz, cartoon...).
- La sobreimpresión de rótulos o textos.

Para ello se suele utilizar software de edición digital de imágenes y retoque fotográfico en el caso de imágenes estáticas, o de composición digital de video en el caso de animaciones y, hay que hacer notar que en ambos casos el programa suele trabajar en dos dimensiones.

El trabajo de postproducción no se debe ser considerado despreciable, por ser la última fase de creación de las infografías. Las diferencias entre imágenes sin y con postproceso pueden ser tal como la de una fotografía (o grabación) amateur y otra profesional.

Ejemplos de software de edición digital de imágenes serían el archiconocido *Photoshop*, de *Adobe*, programa de referencia en su ámbito o *Gimp*, de *Gimp.org*, alternativa gratuita que está alcanzando gran difusión.

Ejemplos de composición digital de video serían *After Effects*, de *Adobe*, o *Combustion*, de *Autodesk*.



Fig 3.14- Imagen original (arriba) y procesada (abajo) con Adobe Photoshop.

Se han corregido valores cromáticos y se ha insertado un elemento externo.



Fig 3.15- Imagen procesada con Adobe Photoshop.

Se ha efectuado etalonaje digital para realzar los colores de la Toscana y se ha modificado un edificio (al fondo) para aportar mayor dramatismo a la escena.

Nota 1:

La Persistencia de la visión (POV) es un fenómeno fisiológico descubierto por Joseph Plateau que explica que, por efecto de la impresión de la imagen en la retina, el ojo humano ve a unos 10 “fotogramas” por segundo (dependiendo de las condiciones).

Por lo tanto, sucesiones de imágenes similares a mayor velocidad no se interpretarán como distintas, sino como una escena en movimiento.

Por ello el cine se emite a 24 frames por segundo y la televisión (europea) a 25 fps obteniendo la sensación de movimientos fluidos.

Actualmente, los avances en neurofisiología han demostrado que la persistencia retiniana no funciona tal como se ha descrito, ya que es un fenómeno cerebral y no óptico.

Sin embargo, se sigue aceptando el término y explicación iniciales, en ámbitos no científicos, ya que es más intuitivo y comprensible.



Fig 3.16- Interior generado en tiempo real por TwinMotion 2.

### Programas de visualización arquitectónica en tiempo real

En los últimos años, la velocidad de cálculo de las representaciones tridimensionales por ordenador se ha multiplicado exponencialmente debido a la evolución de las tarjetas gráficas (y en parte esta evolución ha sido por el impulso económico que supone el mercado del videojuego).

En un principio, la tarjeta gráfica era un elemento que se encargaba de traducir los datos gráficos recibidos por la CPU (unidad central de procesamiento) en información comprensible por el dispositivo de salida. La revolución llega cuando se empieza a incorporar su propia memoria RAM y su propia GPU (unidad de procesamiento de gráficos) a la misma.

La GPU, al igual que la CPU, es un procesador de datos. La diferencia es que, mientras la CPU del ordenador debe tratar todo tipo de información, la GPU está diseñada y optimizada para tratar un solo tipo de datos, los relativos a la representación tridimensional. Es por ello que es mucho más eficiente y libera recursos al resto del sistema.

Cuando el aumento en la velocidad de tratamiento de la información tridimensional llega a ser tal, que permite el cálculo y representación de más de 10 escenas por segundo<sup>1</sup>, podemos empezar a hablar de visualización en tiempo real.

El compromiso es el siguiente, a más velocidad de refresco de la escena (o fotogramas por segundo), menos cálculos se pueden realizar y por tanto menos calidad de la escena representada.

Si se limitan los frames por cumplimiento del fenómeno POV, el resultado directo es que el aumento de potencia de las GPUs ha ido permitiendo hacer más cálculos y por lo tanto introducir mayor número de objetos (realmente de polígonos) en escena con condiciones de iluminación y texturizado más complejas.

A día de hoy existen ya algunas aplicaciones dedicadas a la visualización arquitectónica que aprovechan este potencial para ofrecer visualización interactiva con alto grado de realismo:

- *Twinmotion 2* importa archivos de CAD, BIM o programas 3D y permite generar entornos arquitectónicos de gran realismo en alta resolución.
- *Lumion* permite también realizar recorridos en tiempo real, con unas prestaciones gráficas un poco inferiores al anterior.
- *LumenRT*: A partir de modelos de Sketchup y tras un preproceso de cálculo de iluminación indirecta (texture baking), permite la visualización en tiempo real de los modelos.

### 3.3 Proceso clásico de elaboración de una infografía arquitectónica

#### Introducción

A la hora de abordar la realización de infografías es habitual definir algunos objetivos previos:

1º Escala de infografía: urbanismo, exterior, interiorismo, detalle...

2º Nivel de detalle del modelado: desde volumetría hasta máximo detalle.

3º Nivel de detalle de los materiales e iluminación: desde conceptual (tipo maqueta) a fotorrealista.

4º Calidad y resolución de salida.

El objetivo es que el esfuerzo, tiempo y dinero empleados sean correspondientes al resultado perseguido. Porque es un hecho que el incremento de nivel en cualquiera de los puntos implica un aumento exponencial de recursos.

Por ejemplo, si se requieren imágenes del exterior de un edificio no es necesario modelar su interior o, como máximo, se modelan los recintos visibles a través de las ventanas. De igual manera, para obtener imágenes de interiorismo no haría falta reconstruir todo el edificio sino un escenario con el recinto en cuestión, como un plató de rodaje.

Una vez definido esto, los pasos habituales para la generación de una imagen tridimensional realista arquitectónica han sido habitualmente:

1º Delineación en 2D.

2º Modelado 3D a partir del 2D.

3º Aplicación de materiales al modelo.

4º Iluminación de la escena.

5º Renderizado.

6º Retoque digital de la imagen.

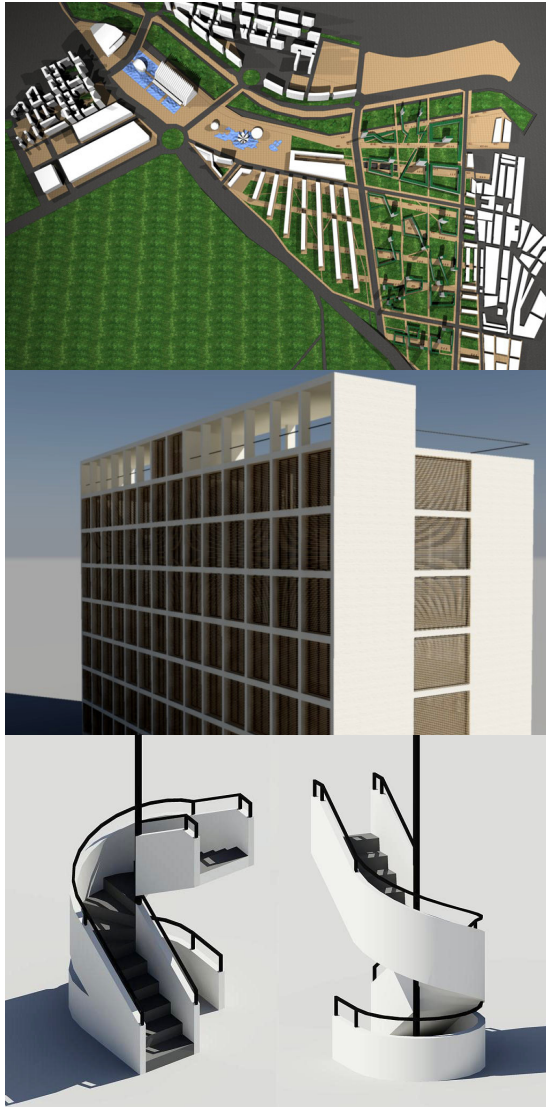


Fig 3.17- Distintas escalas implican distintos niveles de detalle. Urbanismo (arriba) Exterior (centro) y detalle (abajo).

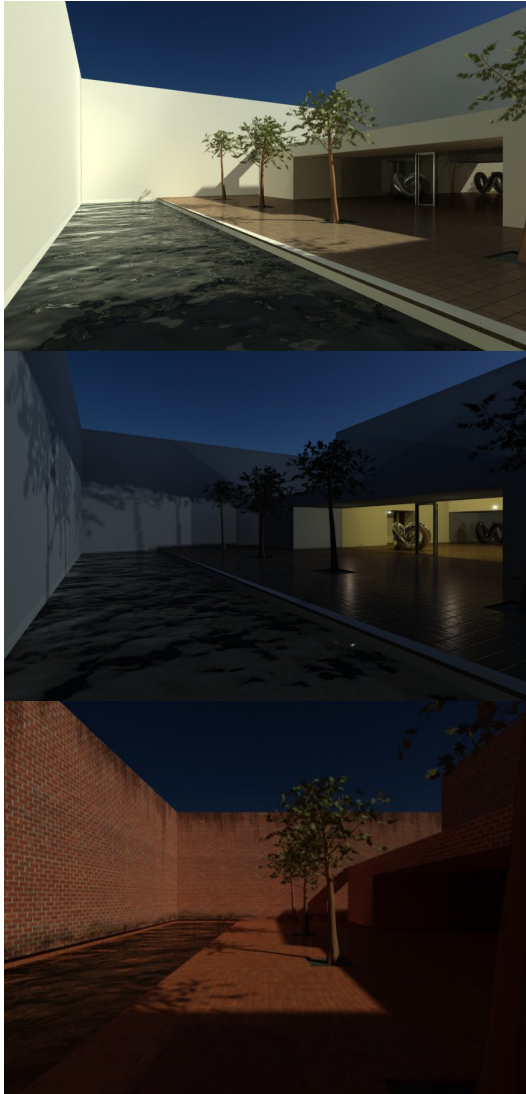


Fig 3.18- La infografía fotorrealista permite visualizar los proyectos en distintas situaciones. Escena diurna (arriba), nocturna (centro) y con cambio de material (abajo) de la Casa Guerrero de Alberto Campo Baeza.

En épocas anteriores se delineaba en 2D en un programa de CAD (p.e. AutoCAD) aprovechando su gran precisión, y posteriormente se exportaba a algún otro programa (p.e. 3ds Max) para modelar en 3D, aplicar materiales y luces y renderizar la imagen.

Posteriormente, los programas de CAD adquirieron prestaciones de trabajo en 3D suficientes para que el modelado 3D se hiciera en el programa de CAD. Los siguientes pasos se harían en el programa de renderizado.

Tiempo después los programas CAD van adquiriendo capacidad de renderizar imágenes con iluminación y texturizado de una cierta calidad. Por ejemplo, AutoCAD a partir de su versión 2008 incorpora el motor de render Mental Ray (propio de 3ds Max) con capacidad de aceptar iluminación fotométrica (físicamente real) e indirecta (reflejada por las superficies y con fuga de color).

En este momento ya se puede generar desde cero y con un único programa imágenes realistas, lo cual redundaría en la comodidad de usar (y previamente haber aprendido) un solo programa.

En la actualidad, los programas BIM (p.e. Revit) tienen la ventaja de que trabajan nativamente en tres dimensiones. Por lo tanto desaparece la primera etapa. Además, también suelen incluir la posibilidad de aplicar materiales y luces y renderizar escenas en el mismo programa.

Sin embargo, para conseguir un fotorrealismo extremo siguen siendo necesarios los programas específicos (el mismo 3ds Max) que permiten muchas otras opciones como la generación de cáusticas (reflexiones y refracciones reales a través de fluidos y materiales transparentes) sombreado con ambient occlusion, profundidad de campo (desenfoque de cámara) y desenfoques de movimiento (motion Blur), animaciones complejas incluyendo cinemáticas inversas (animación de personajes), creación y animación de telas, pelo (o césped), partículas y fluidos (lluvia, viento...), efectos especiales, integración con imagen y video real, etc.

Todas estas técnicas avanzadas de representación fotorrealista requieren un enorme esfuerzo en formación y práctica, y gran cantidad de recursos (de tiempo y económicos) y suelen quedar fuera del ámbito habitual del arquitecto.

En el caso de externalizar la elaboración de las infografías a profesionales, se suele aportar el archivo 2D, que los infógrafos “limpiarán” y utilizarán para generar el 3D. Éstos, si no son arquitectos, es habitual que desarrollen todo su trabajo en 3ds Max, al que suelen estar acostumbrados. Sin embargo, tal y como se ha comentado, es poco probable que expriman todas las posibilidades que ofrece el programa, pues dispararía el presupuesto, y lo habitual es que se negocie un nivel de calidad estándar.



Fig 3.19- Infografía fotorrealista.

Modelo 3D insertado en una fotografía del solar (arriba) y edificio terminado (abajo).

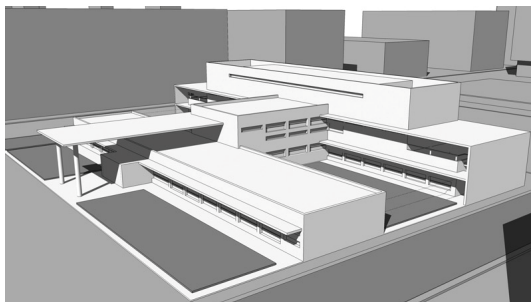


Fig 3.20- Infografía no realista.

Con presupuestos y plazos de tiempo limitados también se pueden obtener buenos resultados.

Igualmente ocurre con la elaboración de animaciones. Procesar un fotograma con calidad fotorrealista y efectos incluidos, a una resolución 720 × 576 píxeles (calidad DVD) puede tardar fácilmente más de 10 minutos en un ordenador común. Un segundo de video, a 24 fotogramas por segundo, 4 horas. Y un minuto 10 días.

Es por ello que para grandes animaciones (casi siempre para la industria del cine) el trabajo se divide en equipos especializados (guionistas, modeladores 3D de personajes, decorados y paisajes, iluminadores, animadores, postprocesadores, montadores, técnicos de sonido, etc...) y el renderizado se realiza en lo que se suele denominar "granjas", naves con un número elevado de potentes ordenadores conectados entre sí que se reparten los fotogramas a generar y vuelcan el resultado a un ordenador central, consiguiendo agilizar así los largos tiempos de cálculo.

Sin embargo el trabajo habitual en arquitectura y construcción es bien distinto. Las animaciones rara vez se realizan debido a su elevado coste y bajo retorno. Se solicita para grandes proyectos (concursos internacionales, grandes infraestructuras...) o a petición de un promotor inmobiliario para mostrar su producto (grandes urbanizaciones o edificios emblemáticos).

Generalmente el arquitecto crea imágenes. Y generalmente no es necesario (ni práctico) que éstas lleguen a un nivel fotorrealista extremo. La infografía actual cumple las funciones clásicas de la maqueta, mostrar una representación de lo que aún no existe, estudiar la escala y la inserción en el entorno, las proporciones de los espacios, la iluminación... tanto para estudio personal por parte del proyectista como para facilitar la visualización de lo diseñado al cliente (o al jurado de un concurso).

Para este tipo de trabajo, la metodología expuesta al comienzo del apartado cumple perfectamente las expectativas. Incluso, las infografías no fotorrealistas están ganando gran aceptación últimamente, llegando casi a ser una moda.

### Fases de elaboración de una infografía

Se desarrollan a continuación las fases enumeradas en el apartado anterior:

#### 1º Delineación en 2D.

La delineación como paso previo al modelado 3D no puede ser igual que la trazada para un plano técnico, lo que importa son las líneas principales de los elementos que se vayan a modelar. Es una delineación más “limpia”.

Por eso cuando se trabaja a partir de planos existentes, que es lo habitual, todas las cotas, textos, sombreados y demás elementos auxiliares son eliminados, así como el mobiliario y equipamiento que ya se añadirá posteriormente en 3D.

Es habitual crear capas con los nombres de los materiales o colores a usar, y ordenar los elementos en ellas. En un modelo infográfico lo importante de los objetos es su apariencia, no su función.

#### 2º Modelado 3D a partir del 2D.

Se ha de tener en cuenta que modelando en 3D aparecen multitud de líneas que se cruzan en el espacio creando confusión cuando el modelo adquiere cierta complejidad. Trabajando en 3D es prioritario el orden, y éste lo aportan las capas.

Por ello, es práctica común desdoblar capas para un mismo material ordenando, por ejemplo, por planta (3D-SUELOS-P0, 3D-SUELOS-P1, 3D-SUELOS-P2...) para poder desactivar elementos terminados y que no interfieran en el modelado.

Las órdenes más habituales en los programas de modelado 3D son:

- Creación de primitivas: Elementos 3D nativos como prismas, esferas, conos, toroides...
- Extrusión de formas 2D: Estirado hacia arriba; una línea creará un plano y un círculo un cilindro.

La mayoría de las partes de un modelo se generan por extrusión (estirado hacia arriba) a partir de elementos 2D. Por ejemplo, un forjado se crea extruyendo 30 cm un contorno de forjado en planta o una tabiquería extruyendo 2,5 m un contorno de tabiquerías en planta.

- Barrido de secciones 2D a lo largo de trayectorias: Por ejemplo para generar pasamanos.
- Revolución de formas 2D alrededor de un eje: Creación de columnas, lámparas...

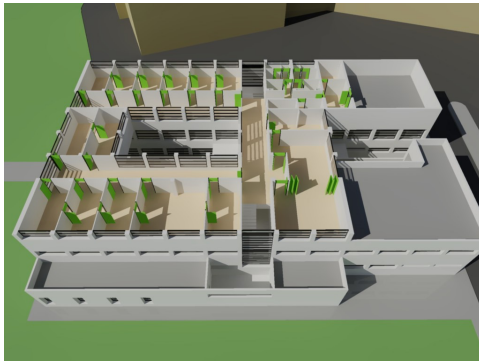
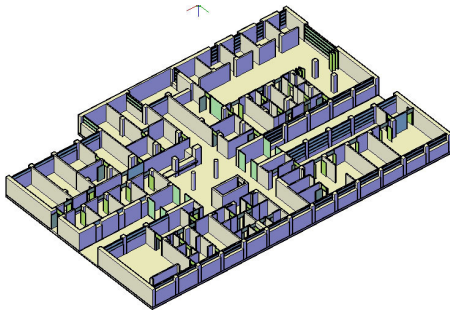
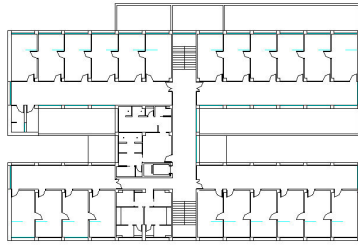


Fig 3.21- Fases del modelado. Delineado base en 2D (arriba), modelado por extrusión (centro) y modelado terminado.

Los elementos de cada planta están en capas diferentes para poder apagarlas a voluntad y trabajar con comodidad.

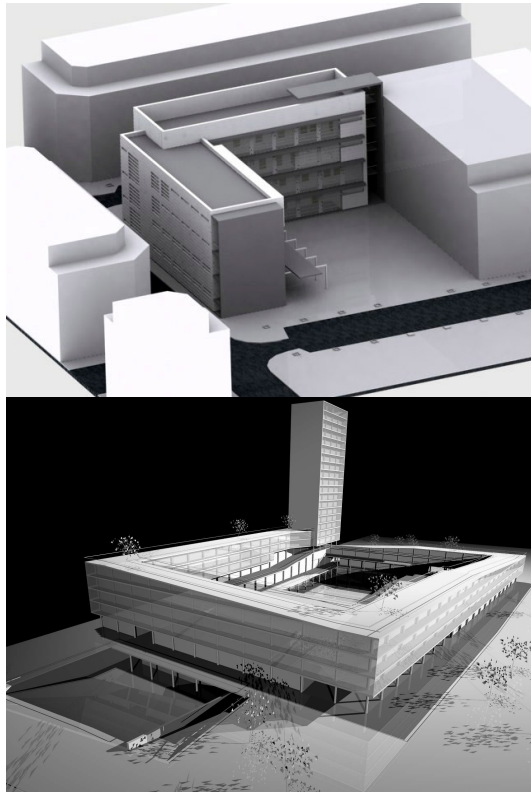


Fig 3.22- Los modelos con excesivo detalle o gran número de elementos son más difíciles de manejar.

En este caso, ambos modelos son macizos porque no se necesitaban renders del interior.

- Solevado: Generación de volúmenes y superficies a partir de una serie de secciones transversales.
- Giros y desplazamientos 3D: Para ubicar elementos en su posición correcta.

Por ejemplo, existen elementos que por su geometría no se pueden extruir directamente sobre el eje z a partir de la planta (ciertas escaleras, rampas o cubiertas inclinadas). En estos casos suele ser habitual extruir a partir de una sección apoyada en el plano XY y posteriormente girar o alinear espacialmente el objeto hasta su ubicación definitiva.

- Booleanas de unión, diferencia e intersección: Permiten sumar y restar sólidos para crear otros más complejos.

La mayoría de huecos (huevo de forjado para escalera o huevo de puertas y ventanas en muros) se realizan restando (usando *booleanas de diferencia*) al elemento original un prisma con la forma del huevo.

- Órdenes de manipulación de caras, aristas o vértices.
- Ordenes de corte por un plano, curvado de aristas, achaflanado, etc...

En el ámbito de la arquitectura de líneas rectas la mayor parte del trabajo se realiza mediante extrusiones, uniones y diferencias y cambios de posición en el espacio. Arquitecturas singulares, como el museo Guggenheim de Bilbao, pueden requerir mecanismos de creación más complejos que se salen del ámbito habitual del arquitecto.

Los modelos complejos (p.e. un edificio) se elaboran como unión de partes más sencillas (fachadas, forjados, carpinterías). El orden de modelado suele ser, en general, de lo más general al detalle, (que suele ser de lo más grande a lo más pequeño) o en su defecto de lo más fácil a lo más difícil.

Los elementos repetibles (puertas, carpinterías...) cuando se modelan se suelen convertir en bloques o grupos (según nomenclatura de cada programa) para facilitar su inserción. A veces, viviendas o plantas enteras se agrupan a su vez para generar edificios, y edificios se agrupan para generar urbanizaciones.

Siempre se tiene en cuenta que cuanto más detalle tiene el modelado y más número de elementos, más "pesa" el modelo y más cuesta de renderizar después.



### 3º Aplicación de materiales al modelo.

Los materiales se aplican a los objetos del dibujo para proporcionar un efecto realista. Los materiales constan de propiedades como color, brillo, transparencia, reflexión, etc. y tienen la posibilidad de usar mapas de texturas.

Los mapas de texturas son patrones que se pueden aplicar a los materiales para afectar a su apariencia, opacidad y relieve.

Los mapas recubren los objetos como un empapelado de pared. Se pueden ajustar sus dimensiones, su posición y su modo de proyección en los objetos (plano, cuadrado, cilíndrico y esférico).

El proceso de aplicación de materiales consiste en asignar éstos a los diferentes objetos de la escena, corrigiendo los valores hasta conseguir la apariencia deseada.

Normalmente todos los programas tienen bases de datos de materiales y además permiten la creación de otros nuevos definiendo sus propiedades.

Hay que tener en cuenta un par de puntos:

- El nivel de realismo de todos los materiales de la imagen suele ser similar, pues en caso contrario el resultado global presenta estridencias (y una cadena es tan débil como su eslabón más débil).
- Incrementar el nivel de realismo de los materiales aumenta exponencialmente el esfuerzo empleado.



Fig 3.23- El mapeado se basa en imágenes frontales con los bordes corregidos para que no se marquen los solapamientos.

Un mapeado puede aplicarse para simular un material (arriba) o todo un paño de fachada (abajo).

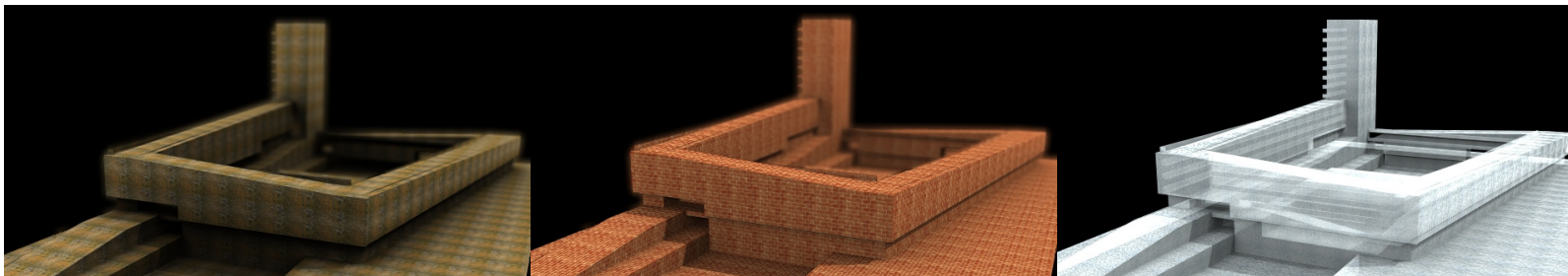


Fig 3.24- Distintas texturas alicadas a un mismo modelo.



Fig 3.25 - La tetera de Utah iluminada mediante sol y cielo (arriba) y mediante un mapa esférico HDRI (abajo).

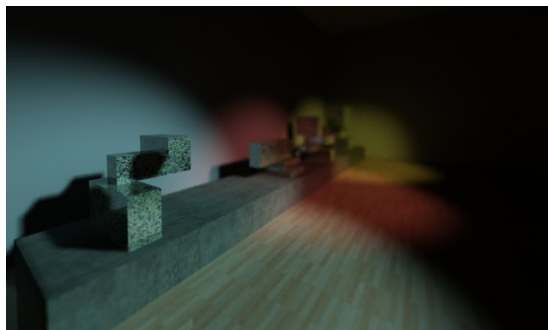


Fig 3.26- Iluminación interior mediante proyectores de colores. La escena está simulando el desenfocado en profundidad de una cámara física.

#### 4º Iluminación de la escena.

En general siempre es más sencilla la iluminación de exteriores; un sol proporciona iluminación directa con sombras proyectadas y una iluminación indirecta envuelve el conjunto.

La iluminación de interiores se complica. Si los recintos se iluminan con luz natural, ésta ha de rebotar en las distintas superficies hasta bañar toda la escena de un modo natural (lo cual suele requerir una serie de pruebas hasta conseguir el resultado buscado) y en ocasiones es necesario apoyar con luces artificiales.

En cuanto a la iluminación indirecta (rebotada por las superficies) a mayor número de fotones (muestras) y mayor número de rebotes, mayor realismo. Pero también mayor tiempo de cálculo.

En cuanto a los tipos de luz, el software siempre suele permitir usar un sol con posición físicamente real según ubicación, orientación, fecha y hora, y luces omnidireccionales (bombillas), o direccionales (focos y proyectores).

A partir de ahí casi todas las propiedades son configurables (intensidad, color...), incluso se pueden configurar efectos irreales como luces que no proyecten sombras o no se atenúen con la distancia.

Un tipo de iluminación más avanzada y realista es la basada en light probes HDRI. Esto consiste en fotografiar un entorno real con unas condiciones particulares (imagen de 360º y alto rango dinámico) y utilizar esta información para generar una bóveda que envolverá e iluminará la escena.

El resultado conseguido es una mayor complejidad lumínica (en vez de un foco de luz tenemos miles de grados de intensidad lumínica a lo largo de una semiesfera), mayor realismo en las reflexiones de los materiales metálicos, y una concordancia lumínica exacta a la hora de insertar el modelo tridimensional en el entorno en el que se tomó la fotografía.

#### 5º Renderizado.

El renderizado es el proceso de ajuste de las variables del motor de render y las opciones de salida de la infografía. Estas dependen en gran medida del motor concreto seleccionado pero, en general, aumentar la calidad (en cualquier aspecto) o la resolución de salida implica aumentar el número de cálculos y el tiempo de procesado. Es en este paso en el que se suelen ajustar efectos como nieblas o desenfoques de cámara (depth of field).

### 6º Retoque digital de la imagen.

Como se describió, este último paso se suele efectuar en un editor de imágenes y es el que aporta la apariencia definitiva. En general, la inserción de personajes o arbolado, efectos lumínicos y desenfocos de ciertos puntos de la escena son mucho más cómodos de realizar en modo “collage” que mediante opciones avanzadas de la aplicación 3D.

En el mundo real los objetos se ensucian, en el virtual no. Existen dos maneras de aportar suciedad e irregularidades a la escena (y por tanto realismo) mediante los editores de imágenes.

La primera manera es editar los propios patrones de texturas que usarán los materiales, generar pequeñas diferencias.

Imaginemos un conjunto de edificios de ladrillo. En el mundo real no tienen todas las zonas de todos los edificios exactamente la misma tonalidad, si elegimos un mismo patrón de textura en nuestro modelo, esto si ocurrirá. Sin embargo, si clonamos la textura, hacemos pequeñas variaciones de tono y las aplicamos a los distintos edificios, el efecto será más real al renderizar.

La segunda manera es “ensuciar” la imagen terminada de igual forma que se ensucia la realidad. Con las herramientas adecuadas se puede oscurecer zonas expuestas al desgaste, marcar las huellas de la lluvia, restos de grafiti, entornar persianas de forma aleatoria... como ocurriría en la realidad.



Fig 3.27- Imagen acabada con inclusión de arbolado y personajes. Más fácil y cómodo que trabajar con elementos 3D.



Fig 3.28- Textura de ladrillo modificada para crear manchas de suciedad (arriba). Posteriormente se utiliza esta textura para el acabado superior de los muros (abajo).

## 4 Conceptos y definiciones sobre el renderizado tridimensional

En este apartado se clarifican algunos conceptos relacionados con el renderizado tridimensional que pueden ser usados en capítulos posteriores.

### 4.1 Materialidad

En el apartado de materialidad se incluyen los conceptos relacionados con la apariencia de los objetos representados.

#### Shading

El concepto de shading hace referencia a la manera en la que la luz interactúa con un punto de la superficie y como varía esta interacción a lo largo de las superficies.

La luz, al tocar una superficie, puede reflejarse o dispersarse dependiendo de la micro estructura superficial del objeto.

Reflexión implica que la luz golpea la superficie y todos los rayos (o fotones) rebotan en la misma dirección, con el mismo ángulo de salida que el de incidencia según las leyes de la física. Los materiales con microestructura superficial plana (p.e. metales o vidrios) reflejan gran cantidad de la luz recibida y provocan destellos o deslumbramientos (highlights) cuando el plano de reflexión forma el mismo ángulo con la fuente de luz que con el punto de vista.

Dispersión implica que, a nivel microscópico, la superficie no es plana. Por ello los fotones al golpear la superficie rebotan en múltiples direcciones. Los materiales rugosos (p.e. telas o yesos) tienen apariencias sin brillos ni reflejos.

Se conoce como shader al algoritmo que describe cómo ha de actuar la luz al contacto con una superficie. Algunos tipos son:

- Flat shading: Algoritmo que ilumina cada polígono de un objeto con un tono según la posición del vector normal del plano del polígono y la ubicación e intensidad de la fuente de luz. Genera formas facetadas.

- Gouraud shading: Algoritmo que genera formas con tonos suavemente difuminados, disimulando los vértices de los polígonos.

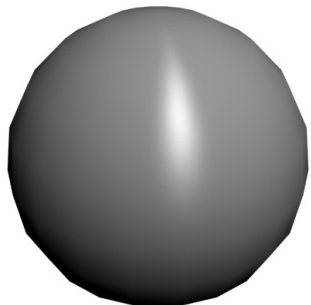
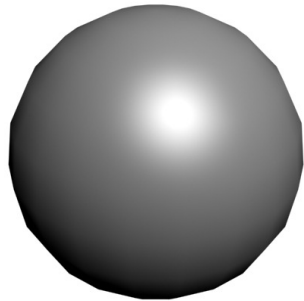
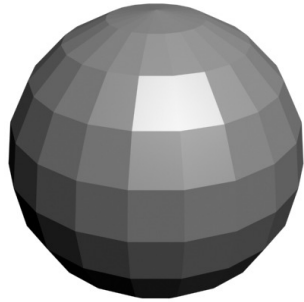


Fig 4.1- Distintos tipos de shaders sobre un mismo objeto. Flat (arriba) , Phong (centro) y anisotrópico (abajo).



Fig 4.2- Propiedades de reflexión (arriba) y refracción (abajo) aplicados a un mismo objeto.

- Phong shading: Algoritmo que genera gradientes de tono suavizados, como Gouraud, pero enfatizando destellos de reflexión de la fuente de luz.
- Anisotropic shading: Algoritmo que permite definir distintos niveles de reflexión según el plano de incidencia de la luz. Comportamiento anisotrópico tienen los materiales con acabado superficial no isotrópico, por ejemplo metales que han sufrido tratamientos industriales (cortes, pulidos...).

Normalmente todos los algoritmos de shading permiten controlar una serie de propiedades visuales del material, como son el color, los niveles de brillo y la opacidad.

### Reflexión

Las propiedades de reflexión permiten que una superficie refleje su entorno. Un espejo ideal tiene un valor de reflexión del 100%. Además, en el mundo real prácticamente todos los objetos lisos, satinados o pulidos presentan reflexión en mayor o menor medida (maderas lacadas o barnizadas, terrazos y mármoles pulidos, alicatados, metales, vidrios...) y su correcta simulación requiere del uso sutil de esta propiedad.

Los motores de render habitualmente calculan la reflexión por trazado de rayos desde el punto de vista hasta el objeto reflectante, rebotando según las reglas de la geometría hasta encontrar un objeto no reflectante. El cálculo de las reflexiones consume muchos recursos ya que multiplica el espacio y es común que se pueda limitar el número de rebotes máximos que puede efectuar un rayo.

### Refracción

La refracción es una propiedad física por la cual un rayo modifica su trayectoria al atravesar medios materiales con distinto índice de refracción, dependiendo éste de la velocidad de propagación de la onda en el medio. Según la Ley de Snell, la distorsión del rayo depende del ángulo de incidencia en la interfaz entre medios y de sus respectivos índices de refracción.

En nuestro caso los rayos son luminosos y los medios al menos parcialmente transparentes.

Este fenómeno nos permite apreciar que un vaso contiene agua, o distinguir entre una pompa de jabón y una bola de cristal.

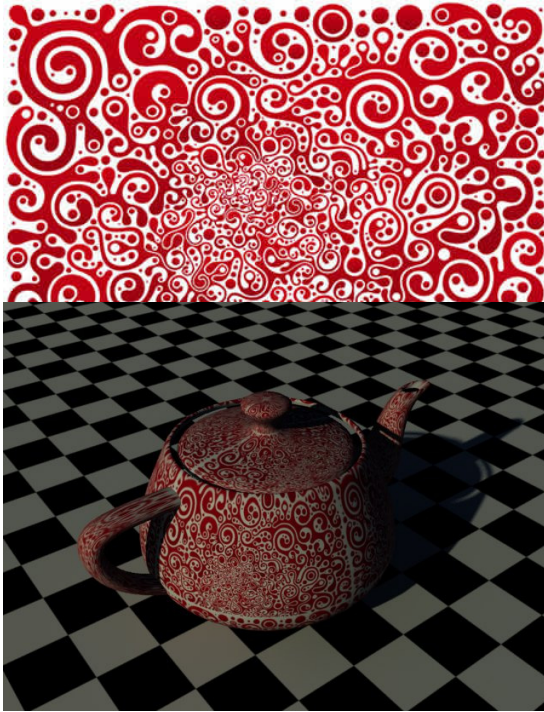


Fig 4.3- Imagen bipmap (arriba) y aplicación como mapa de textura sobre un objeto (abajo).

Los motores de render permiten simular materiales transparentes definiendo su índice de refracción (IOR) a partir de valores tabulados habituales. Igualmente permiten definir el número máximo de rebotes para limitar los tiempos de cálculo excesivos.

La refracción es un fenómeno usado en infografía de diseño de producto (junto con las caústicas) o en interiorismo hiperrealista. Sin embargo, en arquitectura no tiene prácticamente sentido su uso ya que el material transparente predominante es el vidrio plano cuya refracción es despreciable.

### **Mapeado de textura (Texture Mapping)**

El mapeado de textura consiste en aplicar una imagen bitmap en una superficie, como si se pegara una lámina o se empapelara un muro.

Cuando la apariencia superficial de un elemento no es continua (p.e. ladrillo caravista, piedra, etc.) el acabado no se puede definir con un solo color. La solución visualmente perfecta consistiría en modelar geoméricamente cada sillar de piedra o ladrillo de manera individual, y las juntas con sus rugosidades. Sin embargo en la práctica esto sería un trabajo inabarcable. La solución habitual en estos casos es utilizar una imagen frontal del material a imitar, con los bordes corregidos para poder repetirse a lo largo y a lo ancho sin acusar las uniones.

La imagen puede envolver las superficies repitiéndose infinitamente o puede ser aplicada una sola vez como si se tratara de una lámina sobre el objeto.

### Mapeado UVW (UVW Mapping)

Cuando se aplica un mapa a una superficie es necesario definir sus coordenadas de aplicación:

- La posición de inicio del mapeado: Igual que se define la posición de arranque de un empapelado de pared o de un pavimento con juntas.
- La orientación o inclinación de la imagen en el plano de aplicación: Igual que se puede alicatar con las piezas horizontales, verticales o diagonales.
- El tamaño del mapa: Es imprescindible ajustar el tamaño de la imagen respecto al objeto donde se inserta. Por ejemplo, el tamaño de un caravista para que el resultado parezca real y no una casa de juguete.

Los ejes de las coordenadas de textura se definen habitualmente como UV (o UVW) para distinguirlas de las coordenadas XY (o XYZ) utilizadas habitualmente por los objetos.

Igualmente es necesario definir el modo de proyección de la imagen sobre el objeto. Aunque se ha explicado que el texturizado equivale a un empapelado, realmente la imagen es proyectada sobre el objeto. Es necesario definir si la proyección es plana, cúbica, cilíndrica o esférica (y definir los ángulos de proyección) para adaptarse a todo tipo de superficies.

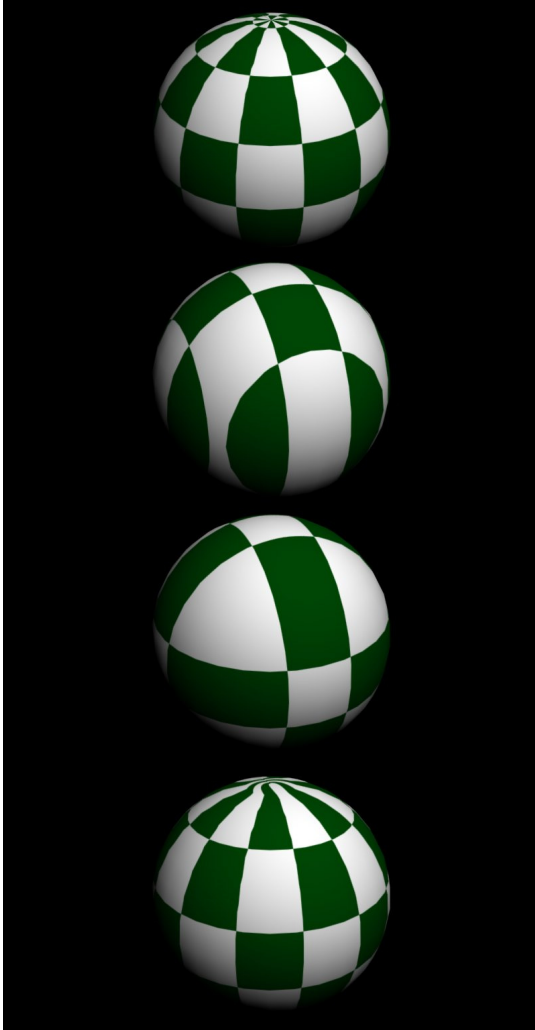


Fig 4.4- Distintos tipos de proyección de un mismo mapeado. De arriba a abajo, esférico, plano, cúbico y cilíndrico.

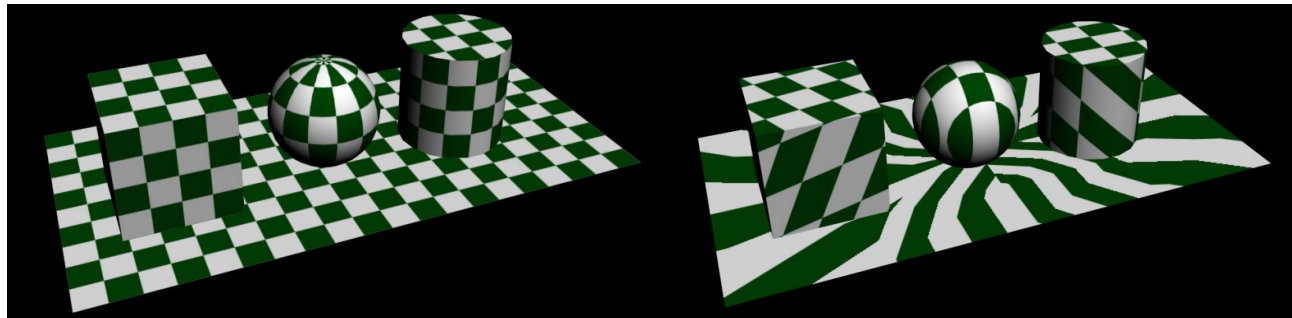


Fig 4.5- La importancia de un correcto mapeado. A la izquierda y a la derecha distintas proyecciones de un mismo mapeado sobre los objetos.

A la izquierda el plano presenta un mapeado plano orientado en su plano (valga la redundancia), el cubo un mapeado cúbico, la esfera un mapeado esférico y el cilindro un mapeado cilíndrico.

A la derecha el plano presenta mapeado esférico, el cubo mapeado cúbico con los ángulos de proyección no coincidentes con los planos del objeto, y la esfera y el cilindro sendos mapeados planos con distintos ángulos de aplicación.

### Mapeado de opacidad (Opacity Mapping)

Al igual que una fotografía o dibujo puede proyectarse en un objeto para modificar su imagen superficial, también puede hacerlo para definir que partes son visibles. Normalmente se utilizan imágenes en blanco y negro en las que el blanco define zonas opacas, el negro invisibles (o viceversa) y el espectro de grises todo el gradiente de opacidad (normalmente 256 niveles).

Usos: Por ejemplo, si se quiere simular un enrejado barroco existen dos opciones:

- Modelar todo el enrejado geoméricamente, con el consiguiente esfuerzo, pérdida de tiempo y cantidad de polígonos que eternizarán el tiempo de cálculo.
- Modelar un simple muro y aplicarle un mapa de opacidad con una imagen frontal en blanco y negro definiendo el motivo barroco.

El segundo método es mucho más rápido de realizar y calcular y su apariencia es correcta en vistas generales. El primer método solo se debería usar para primeros planos de escenas fotorrealistas.

### Mapeado de relieve (Bump Mapping)

Imaginemos un muro de ladrillo caravista. Se modela un prisma y se le aplica una textura de ladrillo en la superficie. El resultado no quedará real, parecerá una fotografía de unos ladrillos pegada en el muro. Esto es así porque falta el relieve del muro, los rehundidos de las llagas y la textura del ladrillo.

Para solucionar este problema se puede modelar cada junta del caravista, pero será este un trabajo tedioso. Igualmente, se puede simular relieve en una superficie por medio de imágenes, siendo éste un método mucho más sencillo que modelar geoméricamente los relieves.

El Bump mapping se utiliza cuando el material a representar presenta relieves superficiales, siendo el más sencillo de los mapeados de relieve.

Funciona mediante imágenes en blanco y negro donde el blanco define movimiento hacia el exterior, el negro hacia el interior y el espectro de grises todo el gradiente de desplazamientos. El motor de render calcula la tonalidad de cada punto de la superficie a partir de la información planimétrica definida por el mapa.

En este caso, el mapa no modifica la geometría física, solo la manera de representarse el elemento (equivaldría a un maquillaje para marcar facciones). Por ello, cuando la superficie es vista desde un

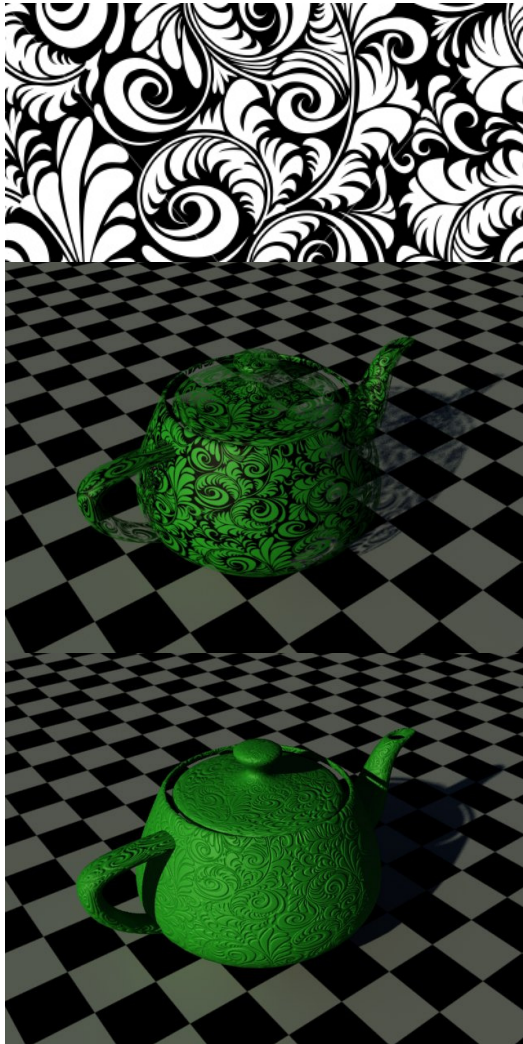


Fig 4.6- Una imagen (arriba) utilizada para definir la opacidad (centro) y el relieve (abajo) de un objeto.



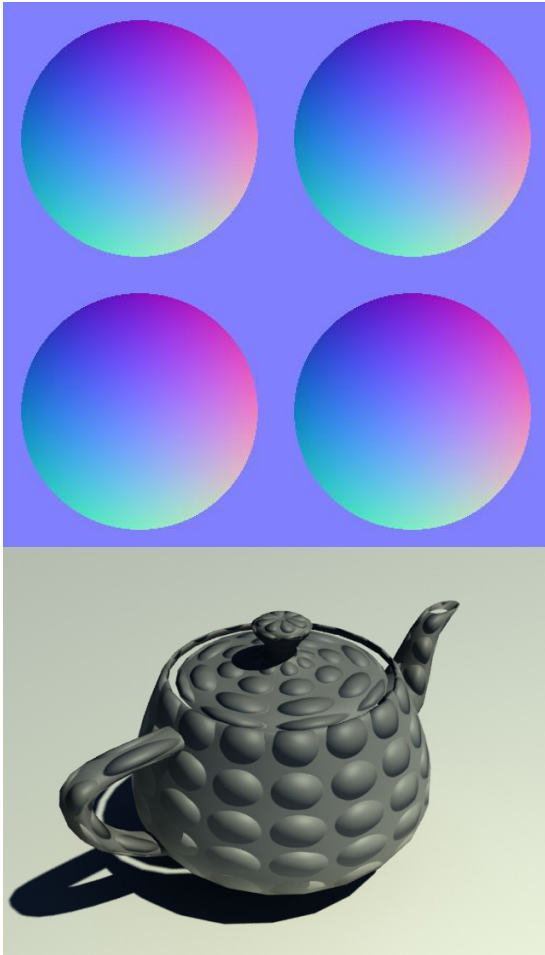


Fig 4.7- A partir de una imagen RGB en color (arriba) se pueden simular desplazamientos en los tres ejes.

La imagen (abajo) es más realista que con un Bump Map, pero no deja de ser un efecto óptico.

En la silueta y en la sombra se aprecia que no existen relieves.

lateral se pierde el efecto tridimensional e, igualmente, las sombras proyectadas por el elemento no presentan relieves.

Sin embargo, siendo el mapeado de relieve más básico y rápido de calcular, el resultado es correcto en vistas generales.

### Mapeado de normales (Normal Mapping)

El mapeado Bump solo aporta información de la "altura" de cada punto de la superficie, al basarse en imágenes en blanco y negro. Una evolución de los mapas Bump son los mapas de normales, basados en imágenes en color.

El color es un espacio tridimensional, lo que quiere decir que cualquier color está generado a partir de distintas combinaciones de tres primarios. En el caso del modelo sustractivo, utilizado en impresión, esos colores son cian, magenta y amarillo. En el caso del modelo aditivo, empleado en fuentes lumínicas (en monitores) los colores son rojo, verde y azul. Es el llamado modelo de color RGB (red, green, blue).

Por lo tanto, una imagen en color RGB puede aportar información tridimensional vinculando los valores de R, G y B a X, Y y Z para cada pixel de la imagen, modificando tridimensionalmente la superficie a tratar.

Normalmente los mapas de normales se utilizan para generar imágenes de gran realismo con pocos polígonos, lo cual es necesario para agilizar tiempos de cálculo. El proceso es el siguiente:

- Se genera un modelo 3D de gran detalle (número de polígonos), por ejemplo un personaje.
- Se crea un mapa de normales del modelo 3D, que queda almacenado como una imagen en color.
- Se crea una versión esquemática del modelo 3D con un número de polígonos mucho menor.
- Se aplica el mapa normal al modelo esquemático para obtener la calidad del detallado.

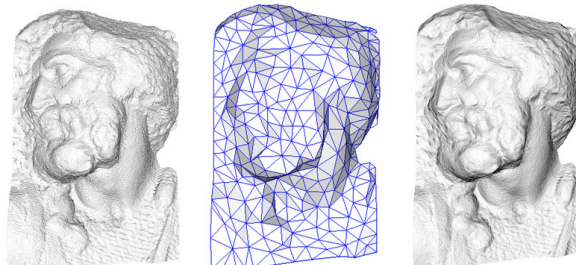


Fig 4.8- Calidad con pocos polígonos:

La imagen de la izquierda tiene 4.000.000 polígonos.

La imagen del centro tiene 500 polígonos.

La imagen de la derecha tiene 500 polígonos y se ha aplicado un normal map extraído de la primera.

### Mapeado de desplazamiento (Displacement Mapping)

Los mapas Bump y Normales no modifican la geometría del objeto, simplemente falsifican la información que recibe el motor de render a la hora de colorear la superficie. Por eso fallan en las siluetas y en la proyección de sombras, donde se aprecia el acabado liso.

Existe otro tipo de mapa, el mapa de desplazamiento, que si que permite modificar la geometría a partir del gradiente de color de una imagen en blanco y negro. No obstante, su uso no es habitual ya que altera las superficies generando cantidades enormes de polígonos que ralentizan enormemente los cálculos.

### Mapas animados

En la mayoría de los programas, los mapeados pueden hacerse usando, además de imágenes, animaciones y videos. De esta manera se puede simular una televisión dentro de un escenario virtual o, mediante un mapa bump animado, generar ondulaciones en superficies de agua.

### Mapas de procedimiento (Procedural Maps)

Además de utilizar imágenes importadas (fotografías o dibujos), un mapeado puede hacerse a partir de dibujos generados matemáticamente. Es lo que se conoce como mapas de procedimiento.

Las ventajas principales son que no marcan solapes (al no ser una imagen repetida), ocupan poca memoria (al ser líneas de programación) y que pueden ofrecer múltiples modificaciones paramétricas. Los mapas generados pueden utilizarse para definir texturas, opacidades, relieves, o cualquier otro tipo de propiedad.

Por ejemplo, las texturas en damero (checkers) se pueden simular con mapas de procedimiento que permiten elegir los colores y dimensiones de las teselas. Existen otros muchos tipos de mapas que simulan nubes, fractales, rugosidades, ondulaciones, etc. y se pueden animar fácilmente.



Fig 4.9- A diferencia de un Bump Map (centro), un Displacement Map (abajo) si que modifica su geometría física a partir de una imagen (arriba).

El número de polígonos y los tiempos de cálculo se disparan.

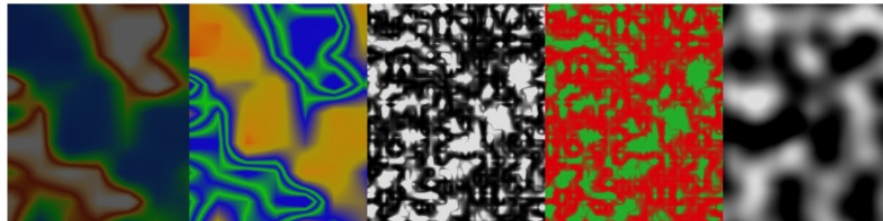


Fig 4.10- Distintos mapas de procedimiento.



Fig 4.11- Material autoiluminado.

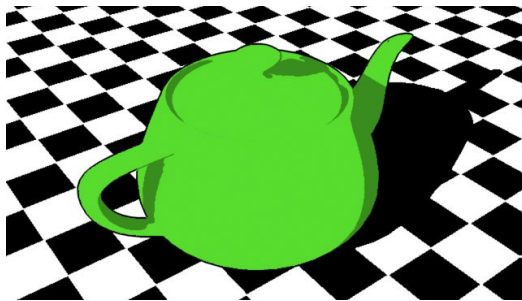


Fig 4.12- Material no fotorrealista o "Cartoon".

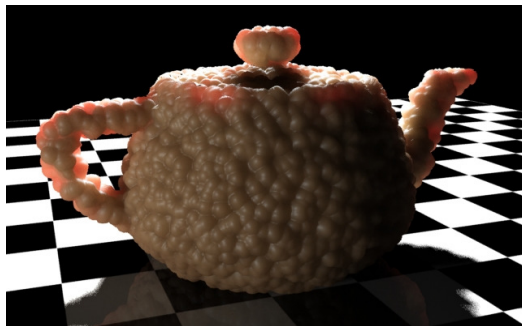


Fig 4.13- Material SSS.

### **Material autoiluminado**

Algunos motores de render permiten el uso de materiales autoiluminados. En el caso más básico esto implica que se pueden ver sin necesidad de que les afecte luz externa, en el más avanzado, que el objeto puede iluminar el entorno.

Esto puede tener aplicaciones en el caso de escenas arquitectónicas nocturnas (carteles publicitarios, tubos de neón...).

### **Material Cartoon**

Igualmente, existen materiales que pueden ser renderizados de forma no fotorrealista, consiguiendo distintos acabados tipo "dibujos animados". Esto se puede aplicar a objetos sueltos (a nivel de material) o a toda la escena (a nivel de render).

### **Material SubSurface Scatering (SSS)**

En los materiales con comportamiento SubSurface Scattering (dispersión bajo la superficie) la luz atraviesa en mayor o menor medida la capa superficial, se dispersa y retroilumina en un punto cercano. Es una especie de translucencia que se puede apreciar en la cera, la piel humana y en ciertos materiales de construcción como el alabastro.

Aunque los motores de render pueden simularlos, en el ámbito de la arquitectura no tiene sentido aplicar materiales SSS a menos que se busque un efecto concreto, debido al aumento de tiempo de cálculo y en la dificultad para conseguir el aspecto adecuado.

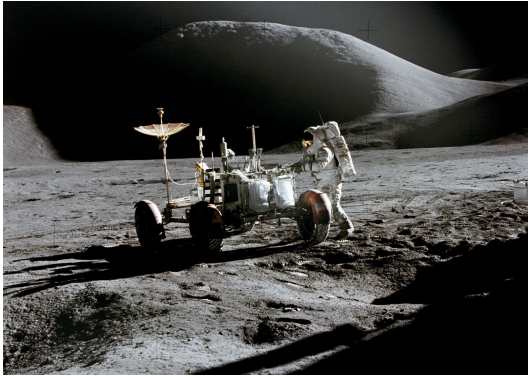


Fig 4.14- En la luna, debido a la ausencia de atmósfera solo existe iluminación directa y pequeñas reflexiones del suelo.

Las sombras son muy acusadas y los contrastes muy altos.

Fotografía cortesía de la NASA.



Fig 4.15- En la tierra, sin embargo, la atmósfera y los objetos del entorno producen una iluminación indirecta que aporta matices a los colores.

## 4.2 Iluminación

### Iluminación directa

El método más sencillo de cálculo de iluminación. El rayo parte de la fuente de luz y golpea las superficies. La intensidad luminínica depende de la intensidad del emisor y del ángulo de incidencia en el objeto. Es un cálculo geométrico sencillo.

Cuando los ordenadores no tenían potencia suficiente para calcular la iluminación indirecta, era el tipo de iluminación más habitual en la generación de imágenes virtuales.

El problema es que la apariencia de las imágenes iluminadas solo con luz directa no se parece a como se iluminan los objetos en la tierra, como se verá a continuación.

### Iluminación indirecta

Imaginemos la siguiente escena:

*Una persona bajo una sombrilla en la playa, en una mañana de verano bajo un sol abrasador.*

Si se generara la escena únicamente con iluminación directa ocurriría un fenómeno insólito, una sombra completamente negra cubriría a la persona bajo la sombrilla impidiendo verla. De hecho, cualquier objeto a la sombra o a contraluz desaparecería en la oscuridad más absoluta. Esto es así porque la iluminación directa solo ilumina los objetos golpeados directamente por el rayo de luz.

Esto es irreal.

En la tierra los objetos están iluminados de forma directa e indirecta, pudiéndose distinguir, a grandes rasgos, dos tipos de iluminación indirecta:

1º Iluminación atmosférica: Además de la fuente de luz principal (el sol) que proyecta rayos luminosos de gran intensidad desde una dirección concreta, la atmósfera actúa como un difusor hemisférico emitiendo luz, más débil y más o menos homogénea, desde todos los puntos del espacio.

2º Iluminación reflejada: La luz, cuando golpea los objetos, es reflejada por éstos dependiendo de su color y acabado superficial. Los fotones rebotan en las distintas superficies hasta que pierden intensidad. Además se produce el fenómeno llamado sangrado de color; la luz reflejada en una superficie roja saldrá teñida de este color.

Si consideramos la escena de la playa con iluminación indirecta ocurre lo siguiente:

1º Las superficies expuestas directamente al sol son iluminadas intensamente con un tono dorado y proyectan sombras.

2º Las superficies no bañadas por el sol, pero a la intemperie son iluminadas desde todas las direcciones por la bóveda celeste, con menor intensidad y sin generar sombras definidas ya que la iluminación envolvente las anula.

3º Incluso los elementos ocultos del entorno, como la persona bajo la sombrilla, la parte inferior de las mesas de playa o el interior de los apartamentos, son iluminados por la luz reflejada en el suelo y resto de elementos del entorno.

Los motores de render actuales permiten calcular iluminaciones indirectas por distintos métodos, algunos más cercanos a la realidad física y otros con aproximaciones matemáticas para agilizar el proceso.

La iluminación indirecta atmosférica, que es la principal en exteriores abiertos, puede ser calculada a partir de imágenes de entorno HDR, como se verá más adelante.

La iluminación indirecta basada en interreflexiones difusas, que es la principal en interiores, puede ser calculada por distintos algoritmos como radiosity, ray tracing, path tracing o photon mapping, que en general tratan de simular cuanta luz rebotada por los objetos llega a cada punto de la escena.

La mayor parte de los métodos de iluminación indirecta que aportan realismo a la escena no pueden ser aplicados, a día de hoy, en la generación de entornos en tiempo real debido a la cantidad de cálculos que se requieren.

Por ello se han desarrollado ciertas estrategias o artificios para emular, de forma más o menos real, la iluminación indirecta.

### **Iluminación ambiente (ambient light)**

El más antiguo y básico es la iluminación ambiente. Consiste en indicar un valor mínimo de iluminación de la escena. De esta manera, las zonas no iluminadas no quedan negras. Sin embargo los colores obtenidos son planos y se aprecia falta de interrelación entre los objetos de la escena.

Los programas de renderizado antiguamente hacían extensivo uso de este recurso.

Las imágenes no fotorrealistas generadas en Sketchup también.



Fig 4.16- Interior renderizado solo con iluminación directa (arriba) y con iluminación directa más indirecta (abajo).

La luz que entra por la ventana rebota cinco veces bañando toda la estancia.

La primera imagen tardó once segundos en renderizar, la segunda setenta y ocho, usando los algoritmos photon mapping y final gathering en AutoCAD.

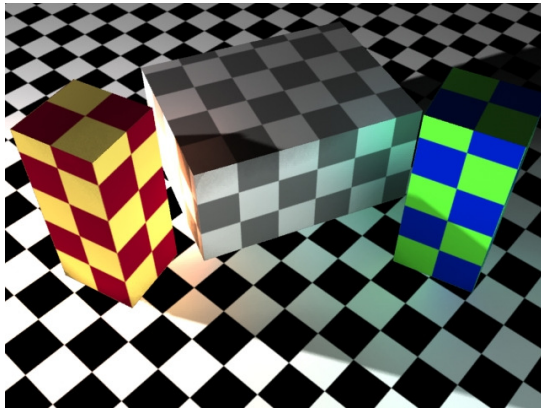


Fig 4.17- El cubo gris recibe sombras, iluminación indirecta y sangrado de color de los objetos cercanos.

Renderizar esta escena consume tiempo de cálculo.

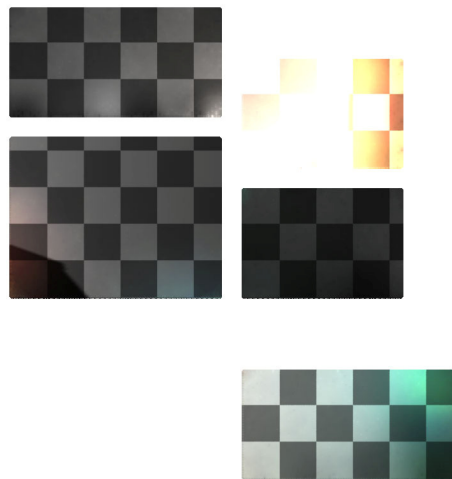


Fig 4.18- Se precálculan los efectos del entorno sobre cada cara del objeto y se guardan como mapas de textura.

### Lightmaps

Una manera de simular iluminación fotorrealista en tiempo real es el uso de lightmaps (mapas de iluminación), lo que en España se conoce como “tostar texturas”.

El método consiste en lo siguiente:

- 1º Precalcular la iluminación indirecta con un método realista dedicando todo el tiempo que sea necesario.
- 2º Tomar el resultado de la iluminación realista de cada superficie y guardarlo como un mapa de textura.
- 3º Re-aplicar la textura con iluminación incluida a las superficies.
- 4º Ahora es posible renderizar solo con iluminación directa porque la indirecta está “pintada” sobre cada superficie.

Como analogía, equivaldría a como mediante maquillaje facial o un trampantojo modifica la realidad “pintando” sombras y reflejos.

La principal limitación de esta técnica es que los objetos a los que se aplican las texturas “tostadas” han de permanecer en la misma posición que cuando se calculó la iluminación, pues si se movieran se “llevarían” las sombras y brillos con ellos.

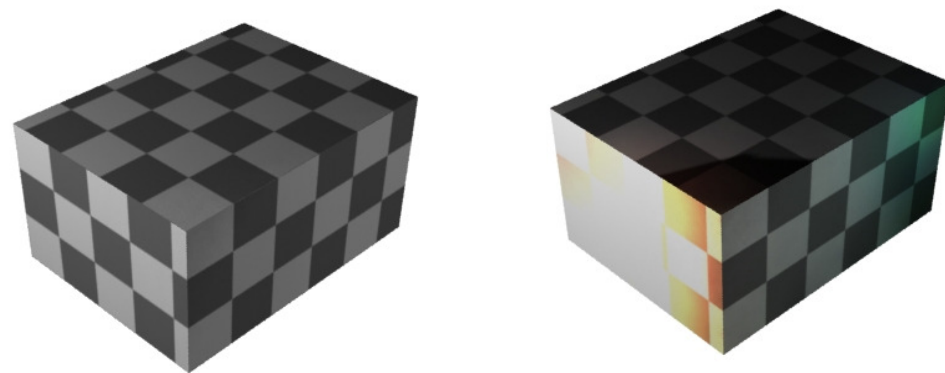


Fig 4.19- Render del prisma aislado, solamente con iluminación directa.

La imagen de la izquierda muestra la textura original, la imagen de la derecha la textura “tostada” con las sombras y reflexiones adheridas. Las dos imágenes tardan el mismo tiempo de generarse aunque la segunda sea más realista si se devuelve a su entorno.

### Ambient Occlusion

Imaginemos el interior de un cubo blanco con una fuente de luz. La luz golpea las superficies del cubo y los fotones comienzan a rebotar repetidas veces hasta que pierden intensidad. Los fotones golpean todas las superficies iluminándolas. Sin embargo, las superficies más expuestas serán más veces alcanzadas (y más iluminadas) al contrario que las menos expuestas (o más ocluidas).

La oclusión ambiental es un ingenioso algoritmo que oscurece las zonas recónditas de una escena, como esquinas, zonas entre objetos próximos u oquedades, simulando el efecto de oscurecimiento que se produciría por la oclusión de dichas zonas ante los rebotes de fotones.

La oclusión ambiental no es un sistema de iluminación indirecta (lo cual suele requerir mucho tiempo de cálculo) pero el efecto resultante es muy similar y mucho más rápido de calcular.

Además, los oscurecimientos producidos por unos objetos sobre otros tienden a dar unidad a la escena.

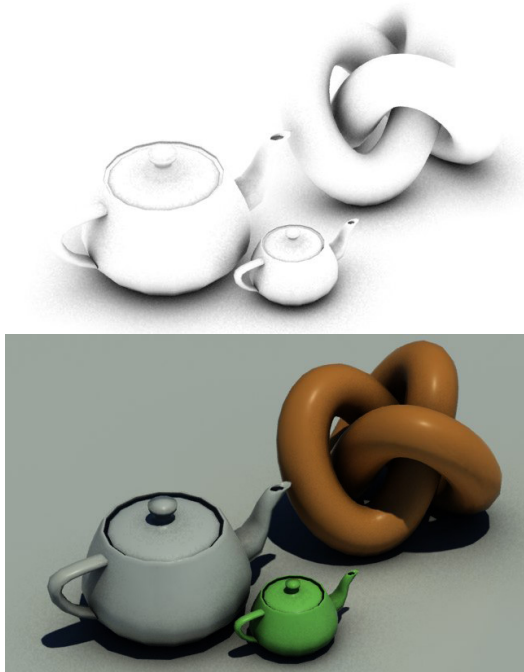


Fig 4.20- Canal de Ambient Occlusion (arriba) e imagen (abajo) con el canal superpuesto para simular iluminación indirecta.

### Causticas

Las causticas son destellos producidos por la reflexión y refracción de la luz a través de objetos transparentes. Se pueden observar en una copa de vino al sol o en los reflejos ondulantes en una piscina.

Aunque los motores de render pueden calcularlas, su uso es prácticamente nulo en arquitectura ya que consumen recursos y no aportan un aumento de calidad apreciable.

Puede tener sentido su uso en infografías en primer plano de diseño de productos de vidrio o plástico transparente, o incluso en escenas de interiorismo hiperrealista (bodegones digitales).

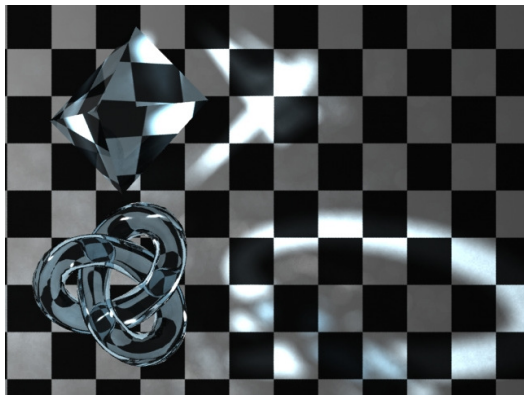


Fig 4.21- Causticas generadas por un prisma y un nudo toroide transparente.

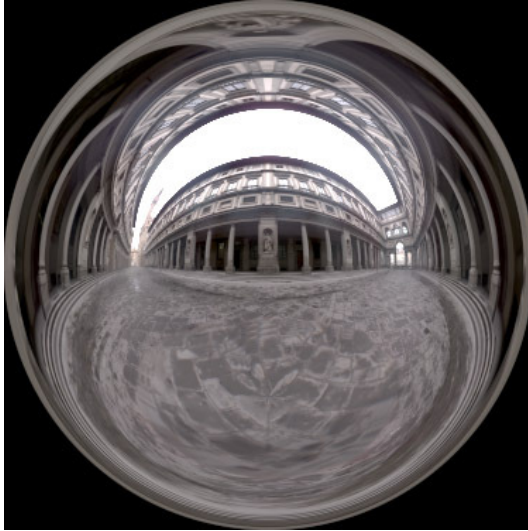


Fig 4.22- Fotografía esféricada (light probe) de alto rango dinámico de la Galeria de Uffici.

Fotografía cortesía de Paul Debevec.

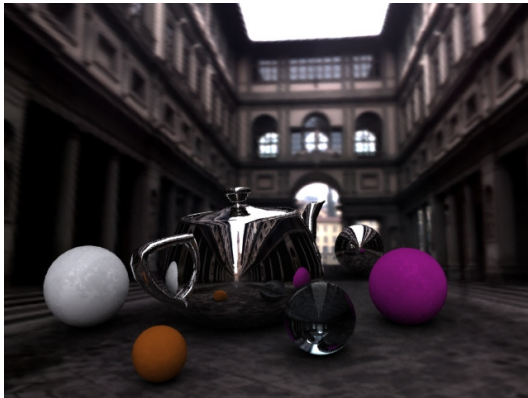


Fig 4.23- Iluminación de los objetos a partir de la imagen anterior. Se observa que, además de iluminar, se ha utilizado la imagen como mapa de reflexión en los objetos cromados.

### Image based lighting HDRI

La iluminación tradicional de una infografía ha consistido, habitualmente, en ubicar una serie de luces virtuales en la escena. En un exterior sería un único sol y una bóveda celeste y en un interior las luminarias pertinentes.

Sin embargo, la realidad es mucho más compleja. Por ejemplo, en un exterior el cielo puede estar parcialmente nublado y no iluminar homogéneamente e, igualmente, pueden existir edificios u otros elementos que obstaculicen la iluminación ambiental. Se puede considerar que los objetos son iluminados por la infinidad de puntos más o menos luminosos, y de una infinidad de tonalidades, que conforman el entorno.

Esta discrepancia entre condiciones de iluminación real y virtual queda aún más patente cuando se integra imagen real y virtual. El elemento virtual no “encaja” en la imagen ya que su iluminación no es coherente con el resto de elementos.

La solución para simular correctamente la iluminación de un entorno consiste en fotografiar dicho entorno y utilizar la imagen para iluminar la escena virtual (Debevec, P. 2002).

1º La imagen a tomar del entorno ha de ser envolvente. Para ello hay dos opciones, o tomar una sola fotografía con un objetivo especial (o fotografiando una esfera especular), o tomar seis fotografías en la dirección de las normales de un cubo.

2º Es muy recomendable que la fotografía sea de alto rango dinámico (HDR), esto es que contenga información de un gradiente de luminancias elevado. Por ejemplo, guardando la imagen en formato RAW o TIFF de más de 8 bits.

3º Se “monta” el entorno virtual a partir de la fotografía envolviendo de forma esférica los objetos virtuales, o montando un “cube map” con las seis imágenes.

4º Se renderiza utilizando la imagen como proyector de luz.



### **4.3 Renderizado**

#### **Renderizado en tiempo real**

El renderizado en tiempo real se utiliza en aplicaciones interactivas, como juegos, e implica que los cálculos de generación de las imágenes se han de realizar a una velocidad suficiente para crear sensación de movimiento continuo en el ojo humano. Esto es, al menos, veinticuatro imágenes por segundo.

El objetivo principal, en estos casos, es conseguir el máximo fotorrealismo en el tiempo disponible. Para ello normalmente se han de desechar muchas técnicas o algoritmos físicamente aproximados y hacer uso de simplificaciones o artificios que puedan simular la realidad de manera aceptable. Algunos de estos artificios pueden ser los lightmaps (tostado de texturas pre-renderizadas), el uso de modelos de bajo número de polígonos junto con normal maps o efectos como profundidad de campo, desenfoque de movimiento y destellos de lente para simular cámaras físicas.

#### **Clipping**

Clipping es cualquier algoritmo destinado a descubrir que objetos (realmente que polígonos) se encuentran dentro del campo visual. Esto tiene una gran importancia, sobre todo en el ámbito de las renderizaciones en tiempo real, (por ejemplo videojuegos) ya que limita el área y el número de polígonos que se han de calcular. Cuantos menos objetos (polígonos) contiene una escena, más recursos puede dedicar a mejorar la calidad de ésta.

Por ejemplo, en una vista urbana a nivel de calle todos los objetos que quedan fuera de la visión de la cámara no existen a efectos de cálculo.

#### **Back-Side Culling**

Las formas tridimensionales están compuestas por polígonos bidimensionales, y los polígonos (como una hoja) tienen dos caras, una anterior definida por un vector normal al plano y otra posterior.

Back-Side Culling es el algoritmo que evita que se calcule la cara posterior de cada polígono para reducir tiempos de renderizado.

Por ejemplo, en una vista urbana a nivel de calle las fachadas de los edificios pueden estar definidas por polígonos rectangulares. No tiene sentido calcular la representación de las caras que apuntan al

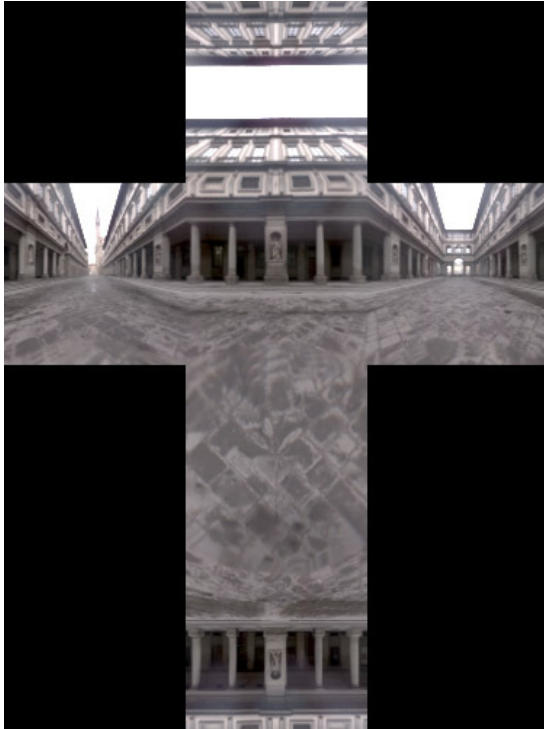


Fig 4.24- Cube Map de la Galería de los Uffici. Cortesía de Paul Debevec.

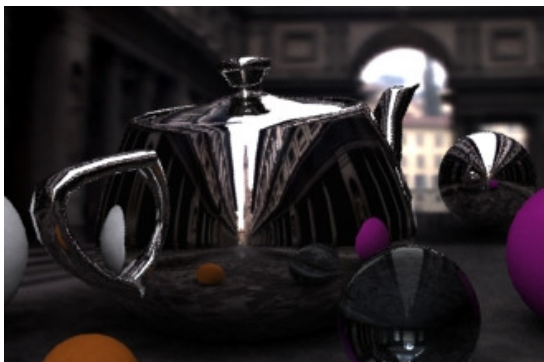


Fig 4.25- El Cube Map aplicado como fondo y mapa de reflexión de la tetera.

interior de los edificios pues no se verán en escena. Para el caso de que un polígono tenga su normal invertida (por un mal modelado) y no se renderice existen órdenes de volteo de normales.

### Z Culling

Z Culling u Oclusion Culling es el algoritmo que elimina del renderizado los objetos que quedan ocultos detrás de otros, usado también para optimizar los tiempos de cálculo.

Por ejemplo, en una vista urbana a nivel de calle, todos los elementos de las calles adyacentes quedan eliminados del cálculo al estar obstruidos por los edificios en primer término.

### Cube Mapping

Cube mapping es una manera de generar entornos lejanos (cielos, paisajes, etc.) de calidad y que no consuman recursos. Consiste en encerrar el entorno de simulación con seis imágenes que forman un cubo (cuatro lados, suelo y techo). El punto de vista se sitúa centrado en el interior del cubo y, por efecto de la perspectiva, las imágenes se funden y generan un fondo continuo.

Las imágenes pueden ser fotografías obtenidas de la realidad o renders de alto realismo pregenerados. Igualmente pueden ser imágenes fijas o animadas.

La ventaja principal (y por la cual se usa a menudo en videojuegos o simulaciones en tiempo real) es que el entorno fotorrealista está prerenderizado y no consume cálculos durante la simulación.

La desventaja principal es que el entorno se sitúa en el infinito y no puede ser alcanzado nunca.

Su uso habitual es para generar Skyboxes o fondos aunque se puede utilizar para generar iluminación exterior realista, como se vio en el apartado específico.

### Canales Alpha, Z-depth y otros

En la generación de un render, además de obtener la imagen realista, se pueden obtener otra serie de imágenes (normalmente llamados canales) que aportan información sobre la escena:

- Alpha: Representa en escala de grises la transparencia de la escena.
- Reflexión: Representa solo las reflexiones de la escena.
- Refracción: Representa solo las refracciones de la escena.
- Iluminación: Representa en escala de grises la iluminación de la escena.
- Z-depth: Representa en escala de grises la profundidad de la escena.
- Material id: Representa en color plano cada material identificado en la escena.
- Matte: permite seleccionar objetos para hacer una máscara.

Los canales Alpha, Material ID y Matte tienen importancia puesto que facilitan, posteriormente, la selección de partes de la escena y su montaje o manipulación.

El canal Z-Depth es especialmente importante para uso en visión estereoscópica, ya que presenta en escala de grises la distancia de los objetos de la escena (simularía la "visión" de un radar o de un murciélago), que puede ser aprovechada posteriormente para generar tridimensionalidad.

### Sistemas de partículas

Los sistemas de partículas permiten generar lluvia, nieve, humo o fluidos, con un comportamiento dinámico propio e interactivo con otras fuerzas de la escena (viento, por ejemplo). Su uso en el mundo del videojuego es extensivo ya que permiten simular explosiones y efectos atmosféricos que incrementan el realismo y la sensación de presencia.

Igualmente pueden facilitar la comprensión del espacio virtual al usuario al aportar información de profundidad, movimiento y paralaje que permiten al cerebro interpretar las distancias.



Fig 4.26- Render de una serie de nudos toroides (arriba), canal Z (centro) y canal Material ID (abajo).



Fig 4.27- Simulación de físicas:

Dos fotogramas de una animación en la cual cae (gravedad) una tela reflectante (cuerpo blando) sobre un cuerpo humano (cuerpo rígido), lo envuelve (detección de colisiones) y posteriormente se deforma y estira cuando el cuerpo se mueve (cinemáticas inversas).

La escena se ilumina con una imagen HDR para aportar realismo.



Fig 4.28- Escena con el fondo desenfocado al usar una baja profundidad de campo y simulación de un destello de lente.

### Físicas

Generalmente, los programas de animación y renderizado y los motores de creación de videojuegos permiten simular propiedades físicas de los objetos (gravedad, masas, fuerzas, aceleraciones, inercias, rozamientos...) para recrear de forma realista acciones que no podrían ser animadas intuitivamente de forma manual.

Los dos elementos principales son el motor de físicas, que es la parte del programa que simula la física newtoniana, y el detector de colisiones, que es la parte que detecta cuando interrelacionan los objetos.

Los tipos de físicas más comunes se podrían clasificar en:

- Física de cuerpos rígidos: Propiedades físicas de cuerpos indeformables.
- Física de cuerpos blandos: Propiedades físicas de cuerpos deformables.
- Física de Ragdolls (peleles) y cinemáticas inversas: Comportamiento cinemático de cuerpos con articulaciones (cualquier personaje humano, animal o robótico simulado).
- Sistemas de partículas y dinámica de fluidos: Explicados en el punto anterior.

### Profundidad de campo

En óptica y fotografía se denomina profundidad de campo a la zona de la imagen que queda enfocada. Biológicamente es un fenómeno natural que se produce en el mecanismo de enfoque del ojo humano. En fotografía es un recurso artístico que permite resaltar el elemento principal de la composición del fondo.

En ambos casos el efecto depende de la distancia focal de las lentes, de la distancia del objeto enfocado y de la apertura del diafragma.

Una imagen generada por computador, por su naturaleza, no sigue las leyes de la óptica, por eso, en principio, no existe el desenfoco. Con el objetivo de representar la realidad tal como la ve el ojo humano se han implementado algoritmos que simulan la profundidad de campo permitiendo incluso definir, en la mayoría de los programas, el tipo de cámara, objetivo, sensibilidad, distancia focal, y apertura de diafragma a imitar.

### Desenfoque de movimiento

Cuando se toma una fotografía a un objeto en movimiento, éste aparecerá desenfocado, en mayor o menor medida, dependiendo de su velocidad y del tiempo de exposición fotográfica. El desenfoque de movimiento es el causante de que no se vean nítidamente las llantas de un vehículo en movimiento o las aspas de un helicóptero.

Por el mismo motivo que el expuesto en el punto anterior, las imágenes sintéticas no presentan desenfoque de movimiento (motion blur) de manera natural. Sin embargo se puede forzar su efecto en la mayoría de los programas con el fin de reproducir la realidad.

### Destellos de lente

Otro fenómeno óptico que puede ocurrir en la realidad y se puede simular en las imágenes de síntesis son los destellos de lente producidos por la reflexión y refracción de los rayos lumínicos en el propio objetivo de la cámara.

Aunque, en principio, los destellos de lente son una aberración no intencionada, su incorporación en imágenes y animaciones de síntesis, y especialmente en motores de videojuego y simulación en tiempo real, aportan sensación cinematográfica y de realismo.

### Integración de imagen generada con imagen real

Cada vez es más habitual integrar imagen real y sintética.

En el caso de imágenes basta cuadrar las perspectivas de ambas imágenes, usar mapas de iluminación y reflexión en el render basadas en imágenes del escenario real y utilizar ciertos artificios (materiales matte) para conseguir las sombras y reflexiones del objeto en la imagen real.

En el caso de video se hace necesario usar técnicas de tracking de cámara, esto es, coordinar el movimiento de la cámara virtual con la real. Mediante análisis del video real es posible calcular la posición y movimientos que hizo la cámara para replicarlos.

En el caso de integración en tiempo real (realidad aumentada) es necesario disponer de referencias claras en el escenario real para facilitar al programa las tareas de posicionamiento y orientación de los gráficos sintéticos.



Fig 4.29- Desenfoque de movimiento:

La imagen superior, tomada con una velocidad de obturación muy baja, permite congelar las aspas del helicóptero.

La imagen inferior, con tiempo de exposición prolongado, difumina los objetos en movimiento.



Fig 4.30- La maqueta, virtual, se ha insertado en una imagen real produciendo sombras sobre ésta.

## 5 Realidad Virtual

Nota 1- Se entiende por sentido cinestésico el relativo a la sensación del propio cuerpo, incluyendo la postura, el equilibrio y el movimiento.

Nota 2- Se entiende por dispositivo háptico aquella tecnología que interacciona con el ser humano mediante el tacto, permitiendo sentir “contacto” o presión debido a la interacción con elementos virtuales.

El concepto se puede aplicar, desde a simples joysticks o mandos con vibración incorporada usados en videojuegos, hasta dispositivos para el entrenamiento de operaciones quirúrgicas o simulación de vuelo profesional.

Nota 3- Se entiende por presencia la sensación subjetiva de encontrarse en un lugar. Es objeto de amplio estudio el averiguar que factores intervienen en la sensación de presencia en entornos virtuales.

Nota 4- Se entiende por cascos con sensores de movimiento aquellos que permiten indicar al ordenador los movimientos y giros que hace el usuario con la cabeza, de manera que se pueda modificar el punto de vista del entorno virtual.

### 5.1 Que es la realidad virtual

El término Realidad Virtual se aplica a los entornos tridimensionales e interactivos generados por ordenador en los que el usuario puede sentir presencia física.

Los entornos simulados pueden ser réplicas de entornos reales o ser mundos imaginarios, pero es necesario que presenten una apariencia de “realidad” física que el usuario pueda reconocer (por ejemplo, no se debería “ver” en perspectiva isométrica).

Los entornos simulados deben de poder interactuar con el usuario, o al menos ser vividos de forma dinámica y en tiempo real. Por lo tanto no pueden ser imágenes estáticas ni animaciones predefinidas.

La interacción, en principio, puede producirse por los cinco sentidos para conseguir una sensación de realidad plena. En la práctica, la interacción con el gusto y el olfato no se suele utilizar (aunque se han estudiado en diversas ocasiones a lo largo de la historia) ya que son sentidos con una base de funcionamiento muy compleja de simular correctamente a día de hoy. El sentido del tacto y cinestésico<sup>1</sup> es parcialmente simulable mediante dispositivos hápticos<sup>2</sup>, suelos deslizantes uni o omnidireccionales u otras estrategias (ventiladores, p.e.) dependiendo de las necesidades y el presupuesto.

Los canales más fácil y habitualmente utilizados en Realidad Virtual son la vista y el oído. En ambos, la tecnología permite simular de forma efectiva y tridimensional los entornos mediante dispositivos estereoscópicos.

Un factor importante en el campo de la Realidad Virtual es el de la inmersión y presencia<sup>3</sup>. Un sistema de RV puede ser más o menos inmersivo dependiendo de los dispositivos de comunicación hombre máquina. A modo de ejemplo se indican diversos grados de inmersión de menor a mayor:

- Visualización en pantalla y manejo mediante joystick (p.e. un videojuego común).
- Visualización en pantalla 3D o cascos 3D (sin sensores de movimiento<sup>4</sup>) y manejo mediante joystick.
- Visualización mediante cascos 3D con sensores de movimiento y manejo mediante joystick.
- Visualización mediante cascos 3D con sensores y manejo con guantes o sistemas hápticos.

Obviamente, mayores niveles de inmersión requieren tecnología más costosa.



Fig 5.1 – Sensorama (1962).

Nota 1- Head-Mounted Display, abreviado como HMD, y traducido como “*dispositivo montado en la cabeza*”, es el nombre que se suele dar a los cascos autónomos de realidad virtual, que normalmente incluyen visión y audio estereoscópicos y sistemas de detección de movimiento de la cabeza.

## 5.2 Breve historia de la realidad virtual

Sin contar los murales panorámicos que comenzaron a aparecer en la segunda mitad del siglo XIX, el primer sistema inmersivo de la historia suele ser considerado el Sensorama de 1962, diseñado por Morton Heiling. El aparato (no era un ordenador) consistía en una cabina en la que se proyectaba una película estereoscópica con sonido estéreo, mientras un sistema de ventiladores simulaba viento y se emitían aromas en momentos concretos de la proyección.

Posteriormente, el propio Heiling patentaría un casco con visión y audio estereoscópicos (y sistema de generación de olores), lo que se podría considerar el primer Head-Mounted Display<sup>1</sup>, y diseñaría un teatro inmersivo con dispensadores de olor y butacas con movimiento sincronizado que nunca llegaría a construirse (Carlson, W. 2003).

En 1968 Ivan Sutherland y Bob Sproull presentan su artículo “A head-mounted Three-dimensional Display” donde muestran “la espada de Damocles”, el primer casco 3D. El sistema incluía dos CRTs capaces de mostrar sencillas imágenes de síntesis y un dispositivo para conocer la dirección en la que miraba el usuario.

En aquellas fechas Frederick Brooks, de la Universidad de Carolina del Norte, comienza a trabajar en el proyecto GROPE simulando la visualización de compuestos moleculares. La gran novedad es el diseño del primer sistema háptico, que permite sentir las fuerzas de atracción entre moléculas a los investigadores.

En 1969, investigadores del MIT crean el “Aspen Movie Map”, un recorrido interactivo por Aspen mediante una serie de grabaciones de video de las calles de la ciudad tomadas con cuatro cámaras montadas en un vehículo, de una forma similar a lo que luego ha sido Google Streetview.

A comienzos de los años ochenta la NASA comienza a utilizar tecnología de Realidad Virtual en sus entrenamientos y a finales se populariza el término, apareciendo los primeros sistemas comerciales de RV y las primeras películas que tratan el tema (*Brainstorm*, 1983 o *El cortador de césped*, 1992).

Con el cambio de siglo, la evolución de los ordenadores en general, y de los procesadores gráficos y pantallas LCD en particular, dan un nuevo impulso al desarrollo de sistemas de Realidad Virtual.



Fig 5.2 – Fotograma de la película *El cortador de césped*, de 1992.

Imagen cortesía de New Line Cinema

### 5.3 Funcionamiento de un sistema de Realidad Virtual

Un sistema de Realidad Virtual es, por definición, interactivo, inmersivo, y en tiempo real. Por lo tanto, el funcionamiento básico consiste en una aportación de datos por parte del usuario (posición, movimiento, acción...), un procesamiento de los datos en tiempo real por parte de la unidad central y regeneración de la escena virtual, y por último la presentación de la escena al usuario.

Estos tres pasos implican tres partes diferenciadas del sistema; dispositivos de entrada de datos, unidad central de procesamiento y dispositivos de salida de datos.

La unidad de procesamiento de datos es habitualmente un ordenador de menor o mayor potencia con un software destinado a renderizar las escenas de forma similar a como se ha explicado en los capítulos anteriores. Los dispositivos de entrada y salida de datos se describirán a continuación.

Hay que destacar la importancia de que el proceso transcurra en tiempo real. Esto implica que el conjunto de;

- la adquisición de datos por el dispositivo de entrada
- el envío de datos a la Unidad Central de Procesamiento,
- el procesado de los datos,
- el envío de la respuesta al dispositivo de salida,
- y la presentación de los datos,

han de tener una latencia total (un retardo desde que el usuario realiza la acción hasta que obtiene la reacción) menor a 0,1 segundos para que la sensación sea instantánea.

Por otro lado, la velocidad de refresco ha de ser de 25 a 30 veces por segundo (25-30 Hz) en el caso de estímulos visuales y 500 a 1000 veces por segundo (500-1000 Hz) en el caso de táctiles, para que el usuario los sienta como continuos.

De ahí que sea tan importante como las prestaciones del equipo, la optimización y depuración del mundo virtual (número de polígonos, método de iluminación y estrategias de disminución de los tiempos de renderizado) con objeto de conseguir una experiencia lo más fluida posible.





Fig 5.3 – Ratón 3D SpaceExplorer de la compañía 3DConnexion



Fig 5.4 – Militar de la Armada de Estados Unidos entrenando en una CAVE con una cinta omnidireccional en el Army Research Laboratory.

Cortesía del Gobierno de los Estados Unidos.

## 5.4 Dispositivos de entrada de datos

Son aquellos que permiten al usuario aportar información al sistema.

### Dispositivos de navegación

Permiten el movimiento por el espacio virtual, pueden ser (entre otros):

- TrackBall
- TouchPad
- Ratón
- Ratón 3D (seis grados de libertad)

Las principales ventajas son el precio y la sencillez de uso.

La principal desventaja es que la acción de movimiento no es natural.

Un sistema de navegación que solventa este problema es el Omnidirectional Treadmill, o cinta andadora omnidireccional. Desarrollada para simular la acción de movimiento en el entorno virtual de forma similar al andar humano permite, además, recorrer espacios virtuales de mayor tamaño que el espacio real donde se ubica el sistema.

A pesar de su elevado coste, complejidad y tamaño que solo lo hacen apto para limitados usos, este tipo de dispositivos sigue teniendo limitaciones, como son el saltar y el subir y bajar escaleras o rampas.

Otros sistemas similares, como VirtuSphere, utilizan, en vez de una cinta deslizante omnidireccional, una gran esfera sobre rodamientos en la que se introduce el usuario y se mueve como en una rueda de hámster tridimensional.

Hernández et al. (2011) han estudiado otras formas de permitir la navegación natural (andando) por mundos virtuales grandes moviéndose en recintos pequeños. Se probaron los “pasos de gigante”, que resultaban molestos a los usuarios, y el uso de “cabinas de teletransporte” o “alfombras mágicas voladoras”, que se percibieron como más naturales.

### Dispositivos de Tracking



Fig 5.5 – El PHANTOM Omni Haptic Device de la compañía Sensable es un tracker mecánico que recoge los movimientos en tres dimensiones de un lápiz digital.



Fig 5.6 – Un Citroën C4 se transforma en un robot y realiza un baile.

El coreógrafo Marty Kudelka realiza el baile y le son trackeados los movimientos.

The Embassy VFX los implanta en el modelo virtual del robot.

Por último, el baile del robot insertado en la filmación real a la cual se le ha hecho un tracking de cámara para que coincida la superposición.

Se incluye en este apartado los sistemas que realizan un seguimiento o rastreo (tracking) del usuario. Se pueden clasificar en (Toharia, P 2010):

**Trackers mecánicos:** Dispositivos mecánicos basados en rótulas con potenciómetros que miden la posición y orientación de objetos fijados al propio dispositivo.

El caso más sencillo son los sistemas hápticos tipo lápiz tridimensional. Sistemas más complejos pueden consistir en trajes completos con sistema de tracking mecánico, aunque éstos son poco usados al ser más pesados y aparatosos que los sistemas magnéticos u ópticos.

**Trackers magnéticos:** Dispositivos más pequeños y ligeros que los mecánicos, que miden la posición de los objetos mediante pequeños sensores adheridos y el uso de campos magnéticos.

Sus usos principales son seguimiento del movimiento de material quirúrgico en el cuerpo del paciente, captura del movimiento humano con objetivos biomecánicos o registro de movimientos de la cabeza y las manos de usuarios de realidad virtual.

**Trackers ópticos:** Dispositivos que rastrean la posición y los movimientos sin contacto alguno con el objetivo mediante detección óptica. El sistema suele consistir en una serie de cámaras que toman información del mundo real y un software capaz de interpretar la geometría adquirida. Habitualmente es necesario el uso de marcas para facilitar el rastreo de los objetos.

Algunos ejemplos de trackers ópticos podrían ser:

- Juegos de Realidad Aumentada usando las cámaras de smartphones y dispositivos móviles.
- El periférico Kinect de la Xbox360, desarrollado por Microsoft que, mediante dos cámaras y un sensor de profundidad por infrarrojos permite controlar la videoconsola y jugar a ciertos juegos con gestos y sin necesidad de mandos.
- Programas de tracking de cámara, usados en la industria del cine, que permiten detectar el movimiento realizado por una cámara para clonarlo en una cámara virtual y poder insertar perfectamente personajes digitales en escenas rodadas sin estabilizador de imagen.
- El periférico Leap Motion, aún no comercializado, que capturará el movimiento tridimensional de las manos junto al ordenador, permitiendo navegar, escribir, pintar o esculpir digitalmente con las manos desnudas.



Fig 5.7 – Tintín es interpretado por Jamie Bell, que actúa con un traje que captura su movimiento mientras una cámara trackea sus expresiones faciales.

Toda esta información es utilizada para animar el personaje digital que aparece en la película.

Imagen cortesía de Paramount Pictures y Columbia Pictures.

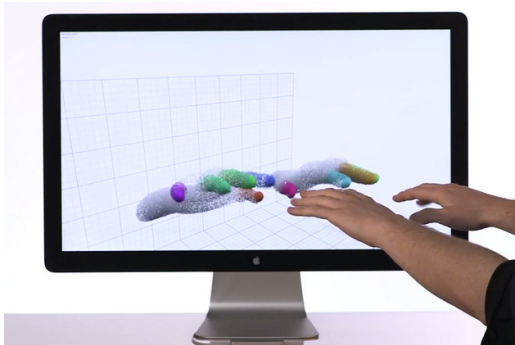


Fig 5.8 – Leap Motion es un pequeño dispositivo que permite capturar el movimiento tridimensional de las manos junto al ordenador.

- Sistemas de Motion Capture utilizados en el rodaje de películas. El actor se viste con unas mallas que tienen una serie de marcadores (bolas de pingpong reflectantes). Una serie de cámaras recogen sus movimientos que posteriormente son implantados a un personaje virtual. De esta manera se han conseguido los movimientos hiperrealistas de los personajes de la película de animación *Las aventuras de Tintín: el secreto del Unicornio* (Steven Spielberg, 2011).

En este caso, la reproducción no es en tiempo real.

Paradójicamente, durante los rodajes de películas de animación basados en Motion Capture, el director no es capaz de “ver” la escena que está grabando, ya que se trabaja en un plató vacío puesto que los escenarios se crearán digitalmente a posteriori. Sin embargo, sistemas de realidad virtual (como el ofrecido por la empresa valenciana Brainstorm Multimedia) permiten visualizar un previo en tiempo real sin calidad fotorrealista, que permite a la dirección intuir el resultado final.

**Trackers inerciales:** Sistemas basados en acelerómetros y giroscopios, que permiten de forma autónoma detectar cambios de posición. Algunos ejemplos conocidos serían:

- El mando de la videoconsola Wii. El wiimote original presenta acelerómetro en los tres ejes. El mando mejorado Wii Motion Plus añade giroscopios en los tres ejes.
- Los sistemas de orientación basados en acelerómetros incorporados en smartphones también se consideran trackers inerciales.

### Dispositivos Gestuales

Dispositivos que permiten la manipulación o gesticulación manual. Se incluye en este subgrupo todos los tipos de guantes de introducción de datos, desde los más sencillos que solo detectan pinzamientos hasta los capaces de interpretar todos los grados de movimiento posible en las manos.

Los avances en este sentido han sido mucho menores que en el caso del tracking óptico. Esto, junto a su precio y aparatosidad es probable que los elimine frente a su competencia directa (sirva como ejemplo las increíbles prestaciones que parece tener el tracker Leap Motion, funcionando con las manos desnudas y a un previsible precio muy competitivo).

El principal futuro del guante gestual está en la inclusión de capacidades hápticas (tacto) como se verá más adelante.

Nota 1:

La visión estereoscópica del ser humano se basa en la captación de imágenes por los dos ojos, que se encuentran separados unos 60mm de media.

Las dos imágenes presentan pequeñas diferencias debidas a la diferencia de punto de vista.

El cerebro interpreta estas diferencias y las fusiona en una única imagen con información de profundidad, mediante el fenómeno del paralaje.

A mayor distancia entre los ojos y mayor cercanía de los objetos, la diferencia entre imágenes es mayor y por tanto la sensación de profundidad.

Cuanto más lejano se encuentra el objetivo, existe menor diferencia entre las imágenes, por lo tanto menor paralaje y por lo tanto menor sensación de profundidad.

A partir de unos 75 metros no se aprecia profundidad en los objetos.

## 5.5 Dispositivos de salida de datos

Son los dispositivos que aportan información a los sentidos del usuario, concretamente a la vista y el oído y en algunos casos al tacto.

### Dispositivos de salida visual

En una primera clasificación, los dispositivos se podrían dividir según si su salida es estereoscópica<sup>1</sup> o no, es decir, si al representar objetos tridimensionales pueden aportar información sobre su profundidad.

A día de hoy la informática de consumo es mayoritariamente no estereoscópica, aunque el aumento de producción tridimensional y la evolución de la tecnología nos están llevando a una situación en la que los monitores, televisores y cines estereoscópicos cada vez son más habituales.

En el caso de la Realidad Virtual, su propia naturaleza inmersiva requiere visualización estereoscópica, por lo que la mayoría de dispositivos de salida gráfica que se van a explicar serán de este tipo.

Sin embargo, no hay que renunciar de los dispositivos no estereoscópicos por una sencilla razón; Las pantallas no estereoscópicas se encuentran, a día de hoy, en toda clase de dispositivos comunes para el público en general (ordenadores, teléfonos móviles y tablets), por lo que suponen una forma muy económica, aunque quizá algo menos inmersiva, de extender el uso de la Realidad Virtual y Realidad Aumentada a la sociedad.

### Representación no estereoscópica

En este caso la sensación de profundidad se consigue de distintas maneras:

**1º Superposición de imágenes:** Cuando un objeto cubre parcialmente a otro el cerebro interpreta automáticamente que se encuentra delante, es decir, más cerca.

**2º Perspectiva:** Una representación en la que líneas o ejes que deberían ser paralelos se muestran confluyentes hacia un punto común (punto de fuga) siguiendo las reglas de la perspectiva geométrica, aportan al cerebro información de profundidad. De igual manera una serie de objetos que se conocen de igual tamaño (p.e. personas) al ser visualizados a distintos tamaños aportan información de su distancia.

**3º Distribución de luces y sombras:** Una distribución de luces y sombras coherente con la percepción de la realidad aporta sensación de volumen o relieve (p.e. círculo contra esfera).

Estos tres efectos pueden encontrarse en cualquier representación tridimensional plasmada en un medio bidimensional, desde pinturas renacentistas hasta películas de animación. Sin embargo, falta un último efecto que se está utilizando en representación tridimensional en tiempo real sobre soporte bidimensional.

**4º Paralaje por movimiento:** Cuando un espectador se mueve lateralmente, la escena captada se modifica de manera que los objetos cercanos se desplazan más (y más rápidamente) que los lejanos. Esta información de diferencia de velocidades es interpretada por el cerebro para medir las distancias. Esto ocurre incluso utilizando un solo ojo.

En un sistema de representación 3D sobre pantalla 2D en tiempo real, si el software es capaz de saber dónde está la cabeza del espectador respecto de la pantalla, podrá simular un punto de vista coherente.

Actualmente es sencillo saber la posición del espectador mediante tracking óptico desde la cámara del móvil, tablet u ordenador, o mediante cualquier otro tipo de sistema. Por ello están comenzando a aparecer aplicaciones que utilizan el paralaje por movimiento en dispositivos móviles.

La principal limitación de esta técnica es que solo funciona para un espectador, ya que simula la perspectiva desde un punto de vista concreto.

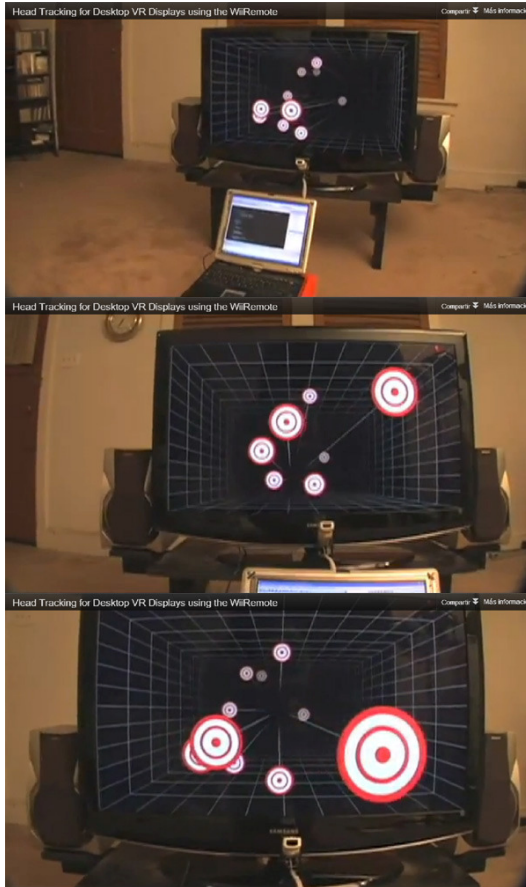


Fig 5.9 – Paralaje por movimiento basado en head tracking usando un WiiMote, usando una pantalla plana convencional.

El usuario lleva un sensor en la cabeza, el WiiMote detecta su posición respecto de la pantalla y modifica el punto de vista de la escena para adaptarlo a dicha posición.

Imágenes extraídas del video de Johnny Chung Lee del Human Computer Interaction Institute.

### Representación estereoscópica



Fig 5.10 – Anaglifo.

Imagen cortesía de Dave Pape.



Fig 5.11 – gafas polarizadas RealD.

Imagen cortesía de Midori iro.

En este caso la sensación de profundidad se ha de conseguir haciendo llegar a cada ojo sendas imágenes con diferencia de paralaje. Algunas opciones:

**1º Proyectar directamente cada imagen en cada ojo:** Puede ser desde el clásico estereoscopio (especie de prismático que muestra diapositivas distintas a cada ojo) de uso lúdico desde 1840, hasta los modernos cascos con dos minipantallas LCD o LED de alta resolución.

**2º Proyectar una imagen doble y aislar para cada ojo:** En este caso, la imagen mostrada es doble y el espectador ha de llevar unas gafas que le permitan enviar la información correcta a cada ojo. La tecnología puede ser de dos tipos:

**2.1 Sistemas pasivos:** Las gafas no presentan ningún funcionamiento activo o dinámico, son un mero filtro, por ejemplo:

- **Anaglifos:** Sendas imágenes son tomadas con filtro azul y rojo respectivamente. Se proyectan ambas a la vez y el espectador lleva unas gafas con sendas lentes también azul y roja. En un ojo la imagen roja se verá filtrada por la lente roja, quedando solamente la imagen azul. En el otro ojo ocurrirá al contrario.

Los principales problemas son la ausencia de color real de las imágenes y la posible presencia de “fantasmas” si los filtros no actúan completamente.

- **Gafas polarizadas:** Evolución del método anterior, usado actualmente en cines 3D. Se emite desde dos proyectores con luces polarizadas en distinto ángulo (90° de diferencia) sobre una pantalla metálica que, además de recibir la proyección, mantiene la polarización. El espectador lleva gafas polarizadas para dejar pasar selectivamente la imagen que le corresponde. En este caso, la imagen se ve a todo color y, para mitigar el efecto oscurecedor de los filtros, se emite con más potencia.

El sistema de proyección de cine RealD consigue el mismo objetivo usando un solo proyector (para abaratar costes de inversión), emitiendo alternativamente frames para cada ojo con distinta polarización. En este caso debe proyectar a 144 fps para que el espectador no perciba parpadeos.

**2.2 Sistemas activos:** Al igual que en el cine RealD, el proyector emite alternativamente frames para cada ojo. En ese caso el proyector es convencional y la emisión no polarizada. Las gafas del espectador llevan un sensor para sincronizar con la emisión y un sistema de obturación que se abre y cierra coordinadamente para cada ojo.

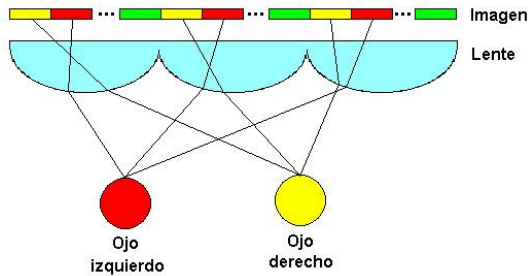


Fig 5.12 – Funcionamiento del sistema autoestereoscópico mediante hoja lenticular.

Fuente: <http://es.wikipedia.org>



Fig 5.13 – Nintendo 3DS, la primera videoconsola 3D autoestereoscópica.

Fuente: <http://nintendo.com>

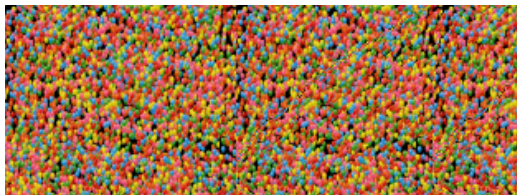


Fig 5.14 – Estereograma de puntos aleatorios.

Fuente: [http:// wikipedia.org](http://wikipedia.org)

La principal ventaja es la ausencia de proyectores polarizadores y pantalla metálica que suponen una gran inversión de actualización desde un cine convencional.

El inconveniente es el uso de unas gafas caras, pesadas y molestas y, como cualquier dispositivo electromecánico, con probabilidad de fallo o rotura. Por eso es un sistema que no se suele utilizar.

**3º Sistemas auto estereoscópicos:** El sistema emite dos imágenes a distancia del espectador que son capaces de llegar discriminadamente a cada ojo sin necesidad de gafas o ningún otro dispositivo. Distintos métodos son:

**3.1 Hojas lenticulares:** Las dos imágenes se dividen en finísimas columnas verticales que se reordenan alternadamente para constituir una sola imagen, fusión de las anteriores. Sobre ésta se adosa una superficie de plástico o vidrio con un texturado formado por una sucesión de delgadísimas protuberancias semicilíndricas.

La dimensión de las franjas de las imágenes está coordinada dimensionalmente con las columnas lenticulares de manera que la refracción de estas últimas muestre a cada ojo la escena que le corresponde cuando el observador está colocado en una cierta posición.

La imagen definitiva puede estar formada por más de dos imágenes, permitiendo formar pequeñas animaciones o puntos de vista que se aprecian al desplazarse lateralmente el observador.

En general la imagen tridimensional se muestra correcta desde muchas posiciones, permitiendo la visualización por más de un espectador, aunque desde ciertos ángulos se ven los pares incorrectos perdiendo toda sensación de tridimensionalidad

**3.2 Barreras de paralaje:** Este método es similar al anterior, a diferencia del sistema para discriminar las imágenes a cada ojo. En vez de una hoja lenticular se utiliza una rejilla que hace de barrera visual.

**3.3 Autoestereograma de puntos aleatorios:** Este efecto óptico, descubierto por Béla Julesz y perfeccionado por Christopher Tyler, permite apreciar profundidad en sucesiones de puntos o manchas no figurativas, especialmente diseñadas, sin necesidad de gafas especiales o sistemas de paralaje, solo fijando la vista más allá de la imagen. Aparte de las importantes implicaciones en el estudio de la percepción humana, se hicieron conocidas hace unos años como divertimento bajo el nombre “magic eye”.

A día de hoy, no tienen mayor conexión con la Realidad Virtual.

### Una clasificación de dispositivos de salida gráfica

Nota 1:

Una CAVE acrónimo recursivo de Cave Automatic Virtual Environment (Cueva, entorno virtual automático) es un sistema de Realidad Virtual inmersiva formado por tres, cuatro, cinco o seis pantallas retroproyectadas que completan un cubo en el que se introduce el usuario.

Éste puede llevar gafas polarizadas para recibir información estereoscópica y sus movimientos pueden ser trackeados ópticamente o con sensores electro magnéticos

A partir de las tecnologías explicadas anteriormente podríamos hacer la siguiente clasificación en tecnologías aplicables a la Realidad Virtual:

#### **1º Pantalla común (smartphone, tablet, ordenador, TV, o proyector).**

Ventajas: Cualquier usuario tiene acceso a ellas. Precio. Navegación vía Web. Posibilidad de trackeo óptico de cara (head tracking) para simular cambios de perspectiva. Posibilidad de varios espectadores (navegación guiada por un solo usuario).

Desventajas: No son estereoscópicas. No son inmersivas.

#### **2º Pantalla 3D autoestereoscópica o con gafas (smartphone, tablet, ordenador, TV, o proyector).**

Ventajas: Es una tecnología que se está expandiendo. Navegación vía Web. Posibilidad de Head Tracking. Posibilidad de varios espectadores (navegación guiada por un solo usuario).

Desventajas: Precio algo más elevado. No son completamente inmersivas.

#### **3º Cascos HMD (Head-Mounted Display).**

Ventajas: Navegación estereoscópica y completamente inmersiva. Sistemas de trackeo avanzados. Posibilidad de representación virtual natural de guantes o dispositivos hápticos

Desventajas: Precio. Uso individual. Uso no al alcance del público general.

#### **4º CAVE<sup>1</sup> (Cave Automatic Virtual Enviroment).**

Ventajas: Navegación estereoscópica y completamente inmersiva. Posibilidad de incluir objetos reales en el entorno virtual. Posibilidad de varios espectadores con visiones independientes (posición controlada por un solo usuario).

Desventajas: Precio. Tamaño y complejidad. Uso no al alcance del público general.



### Localización espacial del sonido

El ser humano es capaz de localizar tridimensionalmente distintas fuentes de sonido debido a su capacidad de escucha binaural (Blauert, J. 1983), o de interpretar las pequeñas diferencias captadas por ambos oídos.

**Localización lateral (izquierda-derecha):** Al disponer los oídos a ambos lados los métodos de localización son relativamente inmediatos, permitiendo distinguir diferencias de un grado para fuentes frontales y quince para fuentes laterales.

- Diferencias de tiempo interaural: El sonido proveniente de un lateral llega antes a un oído que al otro, produciendo retardos en sonidos agudos (longitud de onda corta) y alteraciones de fase en los sonidos graves (longitud de onda larga). Una persona puede detectar diferencias de tiempo de 10 microsegundos (Pitt, Ian. 2010). Incluso con reflexiones y reverberaciones, el sujeto es capaz de interpretar la sensación como proveniente de una sola fuente y localizarla (Efecto Haas).

- Diferencias de nivel interaural: El sonido proveniente de un lateral se escucha más fuerte en el oído cercano que en el lejano, debido a la sombra acústica producida por la cabeza. Este fenómeno se incrementa para sonidos agudos (longitud de onda corta).

**Localización en el plano medio (delante, detrás arriba, abajo):** La oreja, con su característica forma y consistencia cartilaginosa, así como el canal auditivo, actúan como filtros direccionales. Reflexiones y barreras acústicas producidas por el propio cuerpo o el entorno también son tenidas en cuenta para ubicar espacialmente la fuente.

Este tipo de localización es muy subjetiva, pues depende de las características físicas del espectador, de manera que la simulación nunca será perfecta.

**Distancia de la fuente:** Existen distintas pistas para identificar la distancia de la fuente:

- Experiencia: Experiencialmente la persona es capaz de captar la diferencia de nivel, tono y timbre de sonidos conocidos (susurros, voz, gritos) aunque le lleguen con un mismo nivel sonoro.

- Espectro: Con la distancia se atenúan más las frecuencias altas.

- Reflexiones: La diferencia temporal entre el sonido directo y el reflejado ser interpretada para localizar la fuente sonora.

- Paralaje acústico de movimiento: Similar paralaje visual de movimiento explicado anteriormente.

- Efecto Doppler: Si un emisor de sonido (de ondas) se acerca al receptor, éste detecta una compresión de las ondas que se traduce en un aumento de la frecuencia (sonido más agudo). Si el emisor se aleja del espectador, se produce una descompresión de las ondas que implica una disminución de la frecuencia (sonido más grave). Esto puede ser interpretado por el espectador para determinar la dirección de movimiento de la fuente.

### **Dispositivos de salida auditiva**

Conocidos los mecanismos de que dispone el ser humano para ubicar espacialmente los sonidos, se va a esbozar los distintos tipos de emisión que se puede utilizar para simular espacios sonoros tridimensionales.

Los dispositivos de salida auditiva básicamente pueden ser de dos tipos, conjuntos de altavoces o auriculares. Sin embargo, esta diferencia puede ser crucial a la hora de determinar el sistema de espacialización sonora, ya que existen incompatibilidades.

A continuación se describen algunos conceptos:

**Sonido monoaural (mono 1.0):** El sonido que llega a ambos oído es exactamente el mismo (un solo canal de sonido), porque se ha grabado con un solo micrófono o se ha generado de esta manera. No existe sensación espacial. Puede ser escuchado mediante altavoces (todos emiten lo mismo) o mediante auriculares.

Actualmente solo en telefonía o con equipos de audio de un solo altavoz se obtiene sonido mono.

**Sonido estereofónico (estéreo 2.0):** El sonido se ha grabado con dos micrófonos separados o se ha generado y mezclado definiendo diferencias entre dos canales, izquierdo y derecho (se ha modificado el panorama de las pistas).

Al ser escuchado, se percibe una experiencia espacial en la que es posible distinguir sonidos en distintos niveles de lateralidad (izquierda o derecha). Si los dos canales emiten la misma señal (mono) el sonido parece provenir del centro. Sin embargo no existe ninguna sensación de espacialidad en el plano medio (delante, detrás, arriba, abajo).

El sonido estereofónico es actualmente un estándar (más bien un mínimo) en radio, televisión, cine o grabación de CDs musicales.

El sonido estéreo se puede escuchar indistintamente con altavoces o auriculares. La emisión con un altavoz se traducirá a mono y con más de dos altavoces se limitará a dos canales distintos.

**Sonido multicanal:** La creación de sonido (y su posterior emisión) se realiza diferenciando tres o más canales de sonido distintos.

El sistema 2.1. incluye canales izquierdo, derecho y subwoofer para graves (sobre 100 Hz).

El sistema cuadrafónico 4.0 consta de dos canales laterales delanteros y dos traseros.

El sistema 4.1. añade dos canales laterales traseros al 2.1 (o subwoofer al cuadrafónico).

El sistema 5.1. (sistema surround típico de home cinema) añade un altavoz frontal para voz.

El sistema 7.1. incorpora dos laterales para conseguir máxima sensación envolvente.

Obviamente, para una reproducción correcta se ha de escuchar con el número exacto de altavoces que el de canales de emisión. Un número menor no mostrará toda la espacialidad que incluye la grabación y con un número mayor ciertos pares de altavoces emitirán el mismo canal.

En principio, al ser usado este sistema con auriculares convencionales la sensación espacial se reduce a estéreo, aunque existen en el mercado auriculares con capacidad para 5.1.

**Dolby Pro Logic:** Tecnología de codificación de sonido envolvente, desarrollada por Dolby Labs, que permite incorporar información de cuatro canales en los dos canales estéreo convencionales, para posteriormente extraerla con el equipo adecuado. Si el equipo de sonido es estéreo, la señal se oirá en estéreo convencional.

Posteriormente, Dolby Pro Logic II (2000), permitirá codificar y decodificar información 5.1. en una señal estéreo convencional.

**Grabación binaural:** Es un sistema de captación de sonido con capacidades de espacialización tridimensional muy superior al sonido estereofónico.

El método consiste en grabar mediante dos micrófonos direccionales (similares al oído humano) ubicados en los oídos de una cabeza falsa.

A diferencia del estéreo, que solamente espacializa a izquierda y derecha, la grabación con *dummy-head* permite captar las diferencias de tiempo y nivel sonoro interaural, así como la sombra acústica producida por la propia cabeza y los filtros debidos a las orejas.

Para escuchar los sonidos es necesario el uso de auriculares estéreo (para reproducir los dos canales grabados). Este sistema no funciona con altavoces ya que las señales emitidas por cada canal se mezclan antes de llegar a cada oído.

Existe una versión mejorada, la holofonía, patentada por Hugo Zuccarelli, que, en teoría, permite una experiencia más real, e incluso localizar el sonido monoauralmente, aunque sus hipótesis no han llegado a ser demostradas (ni desmentidas). Zuccarelli también ha patentado un sistema de altavoces (de onda cilíndrica) capaces de reproducir sonidos binaurales.

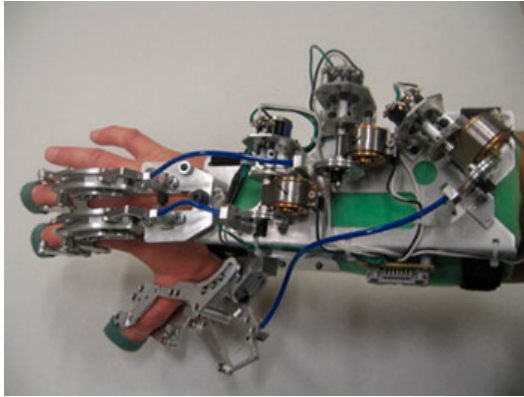


Fig 5.15 – Dispositivo háptico en forma de guante.

Fuente: Research Washington State University Vancouver

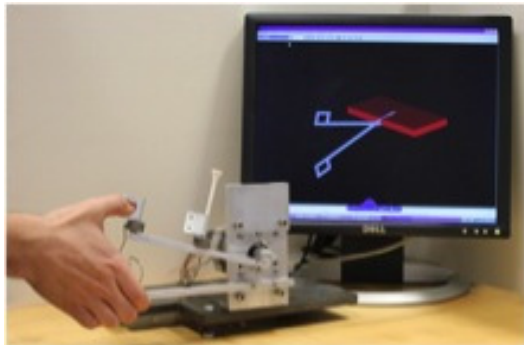


Fig 5.16 – Dispositivo háptico en forma de herramienta (tijera).

Fuente: Research Washington State University Vancouver

**Head-Related Transfer Function:** Las relaciones existentes entre los sonidos captados por ambos oídos en una persona, o en una dummy-head pueden ser caracterizadas por funciones de transferencia que posteriormente podrán ser usadas para sintetizar sonidos y ubicarlos espacialmente.

Algunos programas informáticos permiten, pues, simular sonido tridimensional digital, por ejemplo los sistemas A3C y EAX de Creative Labs (creadores de la gama de tarjetas de sonido SoudBlaster).

Esta tecnología se utiliza ante todo en videojuegos.

### Dispositivos Hápticos

El término háptico (haphthai) hace referencia al tacto. En dispositivos la salida háptica se suele referir a dos aspectos distintos (Burdea, C. 1996):

- Retroalimentación de fuerza: Simula la solidez o dureza de los objetos, viscosidad de líquidos e incluso inercia del movimiento. Necesita sujeciones que absorban las reacciones a las fuerzas transmitidas
- Retroalimentación de tacto: Simula la geometría, aspecto táctil superficial e incluso temperatura. Se basa en sensores en el área de las manos y es de difícil implementación.

Por otro lado, en cuanto a su forma se puede distinguir:

- Forma de guante: Simulan las sensaciones percibidas por las manos, implementación compleja.
- Forma de herramienta: Mandos, lápices, bisturís, etc. conectados a sistemas articulados. A la vez que la herramienta permite captar el movimiento del usuario mediante trackeo mecánico, puede aportar una respuesta háptica.
- Forma específica: Los destinados a un uso concreto pueden ser diseñados con un sistema de control adaptado a la función.

Los dispositivos hápticos han tenido su uso principal en medicina, donde la simulación virtual no depende tanto de la vista y el oído como de la vista y el tacto (por ejemplo, simulación de operaciones quirúrgicas para entrenamiento de cirujanos).

Los dispositivos hápticos tienen unas limitaciones propias, como puedan ser el espacio de trabajo, los grados de libertad de movimientos, velocidad de uso, esfuerzos máximos de resistencia, etc.

Por último indicar que estos sistemas necesitan que el software realice un renderizado háptico a partir de la geometría tridimensional, similar al explicado en capítulos anteriores, pero calculando la apariencia táctil a partir de las propiedades físicas de los objetos (en vez de la apariencia visual), para indicar al instrumento como reaccionar.



Fig 5.17 – Visión con Realidad Aumentada, en la película Terminator (James Cámeron).

Cortesía de Metro Goldwyn Mayer.



Fig 5.18 – Smartphone con software de RA.

Fuente: <http://es.wikipedia.org>

## 5.6 Que es la Realidad Aumentada

La Realidad Aumentada es la visualización del mundo real con información virtual sobreimpresa en tiempo real, normalmente tridimensional. La principal diferencia con la Realidad Virtual es que, mientras ésta última genera un entorno completo que sustituye al real, la RA incorpora información al entorno real, ampliándolo.

Paul Milgram y Fumio Kishino ordenan, en 1994, los distintos grados de realidad:

- 1º Entorno real.
- 2º Realidad Aumentada (con información virtual).
- 3º Virtualidad aumentada (con elementos reales).
- 4º Realidad Virtual.

Ya en los años ochenta el cine de ciencia ficción incorpora lo que podría llamarse “visión con Realidad Aumentada” a los protagonistas cibernéticos de Terminator (James Cámeron, 1984) y Robocop (Paul Verhoeven, 1987), mostrando las posibilidades que ofrece ver la realidad con información adicional sobreimpresa.

Posteriormente, el desarrollo de la Realidad Aumentada vendrá siempre de la mano de la informática gráfica y de la Realidad Virtual, abriéndose su acceso al gran público en el momento en que comienzan a aparecer dispositivos móviles con cámara integrada, y procesadores suficientemente potentes.

La visualización sobreimpresa en pantalla puede ser de varias formas:

- **Dispositivos de mano**, tipo tablet, smartphone o videoconsola portátil con cámara posterior, GPS, trackers inerciales (acelerómetros o giroscopios) y software de reconocimiento de imágenes.
- **Dispositivos de cabeza**: Head mounted Displays, gafas, e incluso lentillas (Larry, G. 2011) con capacidad para sobreimpresionar información de RA están siendo desarrollados actualmente, por lo que es de prever que en el futuro sean habituales.
- **Dispositivos espaciales**: Mediante proyectores se proyecta la información gráfica sobre objetos físicos, como mesas. Esta disposición, junto con el aumento de las dimensiones de trabajo (se pueden coordinar varios proyectores para ampliar la proyección, y la no necesidad de sujetar el dispositivo, favorecen el trabajo colaborativo, ya que varios usuarios pueden interactuar (Contero, Met al. 2012).

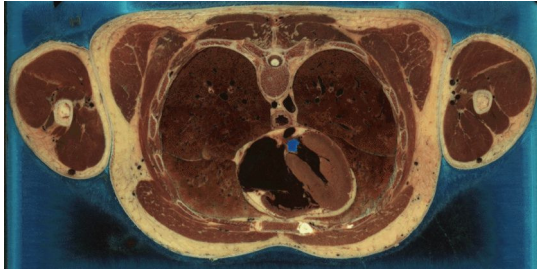


Fig 5.19 – Vista seccionada del Visible Human Project.



Fig 5.20 – Entrenamiento en diagnóstico veterinario mediante sistemas hápticos.



Fig 5.21 – Rehabilitación psicológica de la fobia a las cucarachas mediante realidad aumentada.

## 5.7 Aplicaciones en ciencias de la salud

A continuación se exponen algunos ejemplos del uso de Realidad Virtual y Aumentada.

### Estudio de anatomía

El estudio de anatomía se ha realizado históricamente con cadáveres. En la época actual, la disponibilidad de éstos es escasa y, además, hay que tener en cuenta que el número de usos de cada uno de ellos es escaso (Toharia, P 2010).

The Visible Human Project, de la U.S. National Library of Medicine, es un proyecto de creación de un ser humano virtual orgánica y fisiológicamente completo, que puede ser estudiado en detalle, cortado y visualizado simulando distintas tecnologías médicas.

Diversas Universidades e institutos de investigación han desarrollado aplicaciones para ver el modelo virtual, y proyectos basados en éste (<http://www.nlm.nih.gov/research/visible/>).

### Entrenamiento quirúrgico y de diagnóstico

El entrenamiento quirúrgico y de diagnóstico con pacientes virtuales permite adquirir experiencia en un amplio rango de casos, permitiendo repeticiones y sin riesgo para el paciente.

Existen equipos para la preparación en exámenes endoscópicos (broncoscopia y colonoscopia), palpación de próstata, cirugía vascular, laparoscópica o artroscópica.

### Rehabilitación

Se han utilizado sistemas de Realidad Virtual en rehabilitación física y psicológica.

En el caso de rehabilitación física, el uso de mandos (p.e. guantes de datos) con respuesta háptica permite controlar los movimientos realizados por el miembro en rehabilitación durante terapia de juego virtual.

En las terapias psicológicas, el uso de Realidad Virtual o Aumentada permite exponer al paciente a sus fobias. La realidad Virtual se ha usado en casos de claustrofobia, miedo a volar y a las alturas, mientras que la Realidad Aumentada en terapia de fobia a las arañas, cucarachas u otros insectos.

Estos tratamientos son más baratos, cómodos y seguros que la exposición real a la fuente del miedo.

## 5.8 Aplicaciones en educación y docencia

Son muchas las investigaciones de la RV y RA en el campo de la educación. La posibilidad de introducir al usuario en un entorno simulado, animado e interactivo enfoca la atención y facilita la comprensión frente a métodos docentes tradicionales.

### Estudios de historia del arte, arquitectura e ingeniería

**Visitas de arquitectura:** Imaginemos un alumno que ha de estudiar edificios clásicos lejanos. En el mejor de los casos, realizar la visita supone un coste en tiempo y dinero. Lo habitual, sin embargo, será estudiar a partir de descripciones y fotografías.

Sin embargo, como opina Franco (2011), la fotografía adolece de un punto débil, la falta de “presencia” al tener un punto de vista fijo predefinido. La mirada del arquitecto implica muchos puntos de vista. El recorrido de arquitectura por desgracia es algo personal que no puede ser delegado a la mirada acotada y partidista del fotógrafo. El fotógrafo no es objetivo.

Llegado un punto en el que ya no es posible distinguir una fotografía de una infografía, la Realidad Virtual permite la inspección interactiva, no solo de edificios existentes, sino de los ya destruidos o incluso nunca construidos.

En este sentido podemos encontrar el trabajo de investigación de Takehiko Nagakura, profesor del MIT, creando “fotografías” de proyectos históricos nunca construidos.

**Maquetas de trabajo:** El arquitecto (y el estudiante) ha usado desde siempre la maqueta como ayuda a la hora de comprender los espacios que está proyectando. Sin embargo, en los últimos años esta ayuda comienza a venir principalmente de los modelos digitales.

Como apunta Cazaro (2011), la maqueta tiene una serie de limitaciones; precio y dificultad de ejecución, tamaño, fragilidad o conservación, entre otras.

El modelo digital las suple y permite nuevas posibilidades antes no concebidas, como formas complejas, estudios de materialidad o lumínicos, puntos de vista realistas o inserción en entornos reales. Si se aprovecha el modelo generado para realizar visualizaciones inmersivas e interactivas, la toma de decisiones de diseño se podrá ver enormemente beneficiada.

En este ámbito encontramos un proyecto de investigación que en el futuro puede cambiar la forma de diseñar: Bourdot y su equipo (2010) están trabajando en formas de integrar programas de CAD



Fig 5.22 – Vista interior virtual del Palacio de los Soviets en Moscú, de Le Corbusier. El edificio real nunca se construyó.

Imagen cortesía de Takehiko Nagakura.

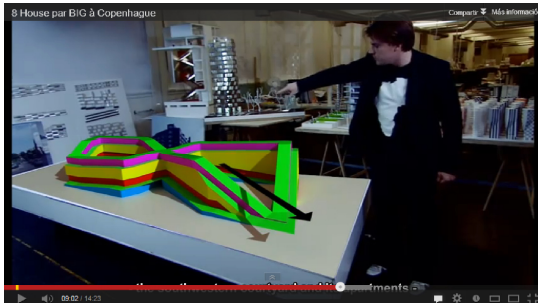


Fig 5.23 – Captura de pantalla de la presentación con maqueta virtual del edificio “8 house” del Bjarke Ingels Group.

Imagen cortesía de Bjarke Ingels Group.

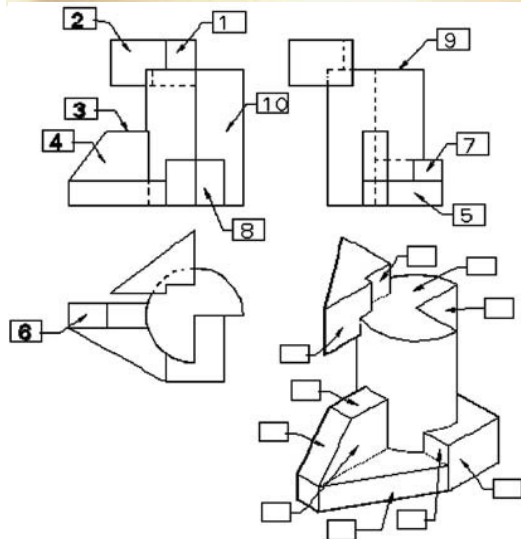
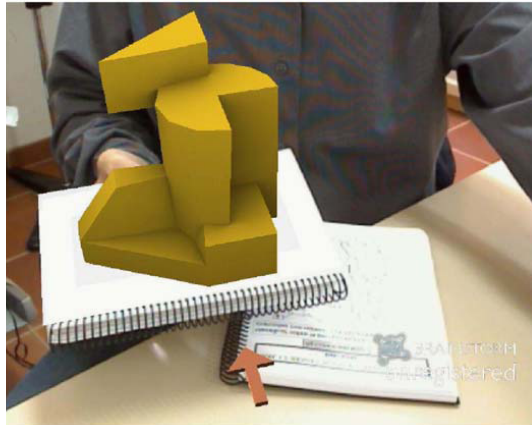


Fig 5.24 – Tecnología de Realidad Aumentada aplicada al desarrollo de habilidades espaciales en alumnos de ingeniería.

Vistas ortográficas e isométrica de una figura (inferior) y superposición del modelo virtual sobre el libro (superior).

con sistemas de Realidad Virtual y dispositivos hápticos, de tal forma que se pueda modificar el diseño en CAD de forma inmersiva e interactiva.

**Mejora de habilidades espaciales:** En este sentido, el uso de las tecnologías de Realidad Virtual y Aumentada, así como de otros avances en comunicación digital, está resultando eficiente en el aprendizaje de materias que requieren habilidades espaciales.

En la Universidad de La Laguna el equipo de Labhuman utilizó un libro con capacidad de Realidad Aumentada para el desarrollo de las habilidades espaciales de estudiantes de primero de ingeniería (Contero, M. et al 2010).

El sistema usa un libro con unos marcadores de RA. Si el alumno pasa las páginas del libro podrá leerlo normalmente y ver unas imágenes de piezas en representación tridimensional. Si el alumno observa el libro a través de un dispositivo de RA (por ejemplo una tablet o un smartphone), verá salir del libro la pieza tridimensional, pudiendo verla desde todos sus ángulos. Durante el estudio los alumnos se sentaban frente a un ordenador con webcam y un software de RA (Brainstorm eStudio).

Como revelaron los cuestionarios de opinión, el alumno de ingeniería joven (19 años) con habilidades espaciales poco desarrolladas, pero conocimiento nativo de la tecnología, no solo aprende mejor, sino que disfruta y se interesa más utilizando esta didáctica frente a los libros tradicionales.

La acertada decisión de diseñar un sistema que puede ser usado con dispositivos comunes abre puertas a la creación de libros “desplegables” de todo tipo de materias o asignaturas.

### Otros estudios

La experiencia indica que estas tecnologías pueden ser usadas desde edades muy tempranas hasta el ámbito universitario, además de fomentar el aprendizaje colaborativo y el autodidacta por experimentación, relevando en ocasiones al docente al papel de guía, en vez de maestro.

Se han desarrollado aplicaciones de RV en el ámbito de las matemáticas, física, química y biología, así como diversas simulaciones de escenarios históricos y localizaciones geográficas (Escartín, E. 2000).

Se han desarrollado aplicaciones de RA en el ámbito de las matemáticas, geometría, física, química, geografía y en diversas ingenierías (Yu, DG. Et al 2010).



## 5.9 Aplicaciones culturales y lúdicas



Fig 5.25 – Tecnología de Realidad Virtual aplicada a la visualización de la cueva de Santimamiñe.

Captura de la geometría de la cueva (superior) y visita virtual a la misma (inferior).

Imagen cortesía de Virtualware.

En el ámbito cultural y lúdico también se están comenzando a utilizar sistemas más o menos inmersivos que recuerdan a lo que se entiende por Realidad Virtual. Muchos de ellos son simples aplicaciones web, aunque algunos otros utilizan sistemas más avanzados.

Google Earth ha ido incorporando características que lo han ido volviendo cada vez más interactivo e inmersivo: Fotografías de satélite, relieve fotográfico, simulador de vuelo, edificios 3D, street view o posibilidad de navegación con ratones 3D.

Son también muchos los museos y monumentos que comienzan a permitir "visitas virtuales" a través de internet, o los que permiten recibir información adicional in situ vía smartphone leyendo marcas de Realidad Aumentada.

En un nivel superior encontraríamos los sistemas avanzados de RV. Por ejemplo, en las cuevas de Altamira se cerró el acceso al público debido al deterioro producido por las visitas y, posteriormente, se creó una réplica (la neocueva) en el interior de un museo construido al efecto.

Sin embargo, otras cuevas como Santimamiñe, en Vizcaia, o Ardales, en Málaga, han optado por digitalizar las cuevas mediante escáneres 3D y crear sistemas inmersivos en los que el visitante puede recorrer una réplica virtual. Además del menor presupuesto de esta opción frente a una réplica física, existen otras ventajas. Por ejemplo:

- Se salvan todas las barreras arquitectónicas pudiendo tener acceso a todos los puntos de la cueva, incluso a los que solo tendrían acceso espeleólogos o escaladores.
- Se pueden simular condiciones de iluminación apropiadas.
- Se puede hacer la simulación en cualquier museo del mundo.

De hecho, se están estudiando (Puyuelo, M. et al. 2011) las posibilidades que ofrecen la Realidad Virtual y otros recursos técnicos en la accesibilidad al patrimonio cultural, debido a la imposibilidad de eliminar ciertas barreras arquitectónicas sin alterar los bienes protegidos. Igualmente se pueden representar los entornos con unas condiciones de enfoque, cercanía e iluminación capaces de mostrar detalles que un visitante no puede apreciar en la visita al original.

Quizá es el siguiente paso cuando el uso de audio guías y maquetas táctiles es ya habitual.



Fig 5.26 – Vistas virtual (superior) y real (inferior) del edificio de oficinas estudiado por Westerdahl y su equipo.

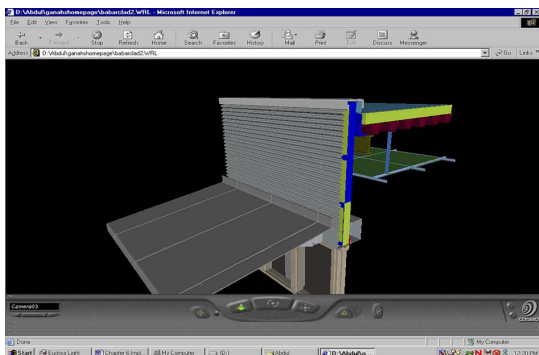


Fig 5.27 – Modelo virtual no inmersivo, programado en VRML, de un detalle constructivo que puede ser vinculado a planos 2D para su posterior inspección y comprensión (Bouschlaghem, D. et al 2005).

## 5.10 Aplicaciones en arquitectura y edificación

El ámbito del diseño en general se ha beneficiado extraordinariamente de los avances en representación tridimensional, ya que permiten “ver” objetos proyectados y aún no construidos.

En el caso particular de la arquitectura, el uso de la Realidad Virtual puede ser aún más provechoso por tres motivos (Fumarola, M. 2011):

- Primero, porque permite, no solo “ver”, sino “recorrer” y “sentir” un objeto (el edificio) que en sí mismo es muy complejo, y por tanto difícil de imaginar en todas sus situaciones, desde todos sus puntos de vista, en reposo, en movimiento, a distintas horas, con distinta iluminación...
- Segundo, porque permite la toma de decisiones a un bajo coste.
- Tercero, porque permite el trabajo colaborativo y facilita la comunicación entre las partes.

Frost y Warren (2000) han estudiado la importancia de la RV en el diseño arquitectónico de unos laboratorios al permitir apreciar las visuales, volumen de los espacios, altura correcta de las ventanas o disposición del mobiliario de una forma que no se podría hacer con representaciones en 2D.

De igual manera, Westerdahl y su equipo (2006) utilizaron sistemas de RV durante el diseño de un edificio de oficinas para Ericsson, para mostrar a los futuros usuarios las distintas opciones. Se propusieron tres tipos de oficinas, clásica (cubículos), landscape (espacios abiertos) y mixta.

Los empleados fueron guiados en un recorrido virtual por el exterior y los interiores del edificio, y rellenaron unas encuestas en las cuales indicaban sus preferencias y su evaluación del sistema de RV. Dos años después, con las obras acabadas, los empleados realizaron el mismo recorrido por el edificio real. Una segunda encuesta puso de relieve que en general, la experiencia con la RV les había ayudado a entender el edificio y a tomar decisiones sobre sus espacios de trabajo.

Se encuentran, también, ejemplos de uso de Realidad Virtual no inmersiva usando ordenadores convencionales con conexión a internet. A partir de modelos de AutoCAD o 3Ds Max, y mediante lenguajes de programación VRML y Java, diversos autores han creado escenarios interactivos de arquitectura e interiorismo (Bouschlaghem, D. et al 2005) (Shaoliang, Q. 2012).



Fig 5.28 – Modelo virtual (superior) de la biblioteca municipal de Göteborg, creado por el equipo de Sunesson durante el concurso, y captura de pantalla de Google Street View con el edificio construido.

### 5.11 Aplicaciones en urbanismo y paisajismo

La visualización tridimensional por ordenador en urbanismo y paisajismo lleva unos veinte años usándose e investigándose (Paar, P. 2006), junto con el uso de los Sistemas de Información Geográfica (GIS) (P. Ghadirian, P. Bishop, D. 2008) (Shen, S. et al. 2001).

Conseguir representaciones fotorrealistas en tiempo real no es fácil debido a la complejidad propia de las escenas (cantidades grandes de plantas, alto número de polígonos y texturas complejas) aunque, según Perrin (1991, citado por Paar, P. 2006), esto no es importante, ya que el fotorrealismo no aporta excesivas ventajas durante el proceso de diseño. Como ejemplo, el software no realista Sketchup está teniendo gran aceptación en el desarrollo de visualizaciones de paisajismo.

Sunesson (et al. 2008) realizó un estudio muy a tener en cuenta. Durante el concurso para el diseño de la biblioteca pública de Göteborg, se utilizaron sistemas de Realidad Virtual para mostrar las propuestas al jurado técnico y las autoridades municipales.

Se pidió a los tres estudios de arquitectura preseleccionados que facilitasen archivos de CAD a los artistas 3D para que desarrollaran los modelos. Tras la experiencia, se realizaron una serie de encuestas a los cuatro grupos mencionados, donde cabe destacar algunas conclusiones:

- Las autoridades municipales (que no habían tenido contacto anterior con la RV) vieron gran potencial y auguraron un prometedor futuro a este tipo de visualización.
- Entre los arquitectos ninguno tenía experiencia en RV y, aunque algunos tenían conocimientos de trabajo en 3D, solían externalizar las infografías. En dos despachos se mostraron preocupados por la falta de control sobre su modelo, al ser realizado por un artista 3D ajeno. Todos los participantes admitieron que la RV afectará a su trabajo en un futuro.
- Los artistas 3D predijeron que la RV se volverá común en la presentación de proyectos y sostuvieron que los arquitectos tendrán que adaptarse y aprender a hacer sus propios modelos.

Por otro lado, la RV puede ser utilizada para simular la accesibilidad del entorno urbano.

En el Laboratorio de Entornos Virtuales del grupo ABACUS del Departamento de Arquitectura de la Universidad de Strathclyde (Maver, T. Petric, J 2003) se ha diseñado un dispositivo de navegación con forma de silla de ruedas con respuesta háptica a la inclinación de los espacios simulados.

Mediante este sistema se puede recorrer espacios urbanos detectando sus problemas de accesibilidad durante el proceso de diseño.



Fig 5.29 – Simulacro virtual de incendio en el interior de un edificio (Smith, S. Trenholme, D. 2009).



Fig 5.30 – Simulación de entorno laboral para el aprendizaje de seguridad y salud por parte de los operarios (Hongling, G. et al 2012).

## 5.12 Aplicaciones para el uso, mantenimiento y seguridad de las construcciones

Un estudio (Shih, N.-J. Lin, C.-Y. Yang, C.-H. 2000) demostró que el tiempo de evacuación en caso de incendio es mayor en simulaciones de RV que mediante los métodos tradicionales y, además, la elección de los recorridos de evacuación no es correcta.

Esto es debido a que los métodos de cálculo de evacuación contra incendio no tienen en cuenta la visibilidad real de cada edificio concreto. Una simulación de emergencia con un modelo virtual puede ayudar a entender a los proyectistas que puntos conflictivos existen en el diseño.

En este sentido, otras investigaciones han tratado la simulación mediante tecnología de videojuegos y RV de situaciones de incendio (Smith, S. Trenholme, D. 2009) u otras emergencias.

En la Universidad de Darmstadt se ha investigado la creación de un "serious game" que permitiera, a partir de un modelo BIM de Revit y mediante un motor de físicas, crear un simulador de rescate que pueda ser usado para entrenamiento en distintas situaciones de emergencia (Rüppel, U. Schatz, K 2011).

Aysu Sagun (et al 2011) ha simulado la evacuación de multitudes con objeto de determinar tiempos de evacuación según distintos parámetros de diseño, así como identificar posibles modificaciones en las directrices de diseño actuales para mejorar la seguridad.

También se ha investigado esta tecnología en la formación especializada en Seguridad y Salud de personal en obras de construcción o instalaciones industriales (Li, H. et al. 2012; Hongling, G. et al 2012).

Por otro lado, en cuanto a las ayudas en el uso de entornos y edificios, se han estudiado sistemas de realidad aumentada, junto con dispositivos con GPS, para orientar e indicar direcciones al usuario (Liarokapis, F et al. 2007).

En el caso de uso urbano, en vez de visualizar las indicaciones sobre un plano esquemático, éstas se sobrepresionan en pantalla sobre la imagen real, pudiendo incluso ofrecer información interactiva vía internet. En este sentido se puede destacar la nueva gama de GPS para vehículos Pioneer AVIC-VH99HUD, que muestra la información proyectada en el parabrisas.

Esta tecnología puede ser usada para guiar en el interior de grandes edificios (hospitales, oficinas...) para guiar hasta una habitación concreta o para encontrar la salida más rápida en caso de

emergencia (Malkawi, A. et al. 2004). Solo haría falta un repetidor de señal en el interior del edificio y un dispositivo (smartphone o tablet) con GPS.

La realidad aumentada se ha utilizado dentro de edificios para sobreimpresionar in-situ elementos ocultos tales como la estructura o instalaciones, tal como se explica en “Architectural anatomy” (Feiner, S. et al. 1995).

Esta tecnología se podría aplicar durante el proceso construcción (Webster, A. et al. 1996), sobreimpresionando replanteos de instalaciones, tabiquerías o acabados sobre la estructura desnuda del edificio, y aportando información sobre el espacio donde se encuentra el usuario (materiales de suelo, paredes y techo, colores, notas y recordatorios).

En el ámbito del mantenimiento del edificio, sobre los paramentos terminados se podrían sobreimpresionar elementos estructurales e instalaciones, informando de sus dimensiones y características (Kensek, K. et al. 2000).



Fig 5.31 – GPS Pioneer AVIC-VH99HUD.

La información se sobreimpresiona en el parabrisas del automovil.

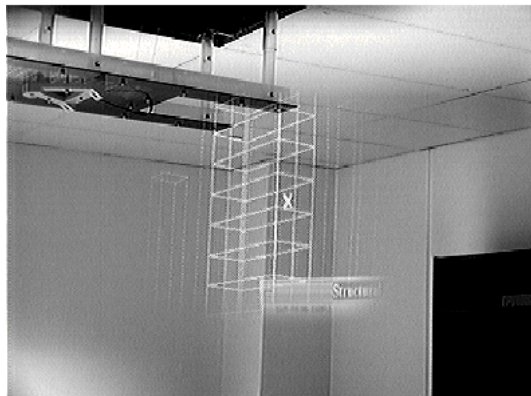


Fig 5.32 – Sobre impresión de elementos ocultos en un edificio mediante Realidad Aumentada (Webster, A. et al. 1996).

### 5.13 Aplicaciones en Rehabilitación del Patrimonio Construido.

Tanto la Realidad Virtual como la Realidad Aumentada se han utilizado para visualizar en su estado original elementos patrimoniales destruidos o modificados.

Habitualmente, el paso previo e indispensable es la obtención del modelo tridimensional fidedigno. Esto se ha llevado a cabo, in-situ, mediante el uso de escáneres 3D o modelado basado en imágenes para elementos existentes o, en el caso de elementos inexistentes, mediante un estudio exhaustivo de la geometría y materialidad y su posterior modelado a mano por técnicos o artistas gráficos.

En este sentido, Rua y Alvito (2011) han recreado virtualmente la villa romana de Freiria, Cascais, Portugal, hoy desaparecida mediante un estudio minucioso de sus huellas sobre el lugar, modelado tridimensional a la manera clásica mediante 3ds Max e inserción del modelo en un entorno interactivo mediante un motor de videojuegos.

Athanasios D. Styliadis (2007) reconstruye la basílica de St. Achilleios, lake Prespes, Grecia, mediante fotogrametría 3D a partir de fotografías y posteriormente exporta el modelo al programa de CAD MicroStation, desde el que puede generar imágenes fotorrealistas.

El modelo tridimensional puede servir incluso de “copia de seguridad”. En 2007 se realizó un modelo virtual completo del Palacio de los Condestables de Casalarreina (La Rioja), a partir de levantamientos de planos y demás documentación gráfica detallada tomada en 1992, así como registros fotográficos y cartografía urbana del entorno. En 2008, durante las labores de rehabilitación, un tercio del edificio se derrumbó, quedando el modelo virtual como la fuente más directa y fidedigna hoy en día para su estudio y futura restitución (Fernando de Fuentes, A, et al. 2010).



Fig 5.33 – Reconstrucción virtual mediante fotogrametría de la basílica de St. Achilleios (Athanasios D. 2007)

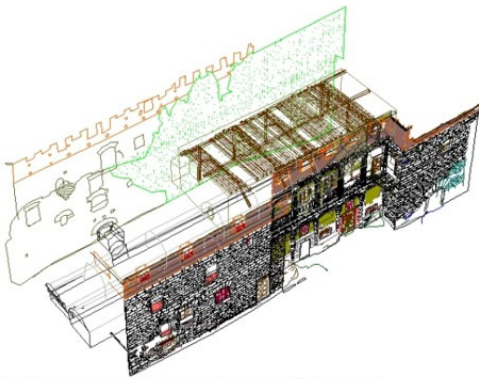


Fig 5.34 – Reconstrucción mediante fotogrametría y topografía del Palacio de los Condestables de Casalarreina (Fernando de Fuentes, A, et al. 2010).



Fig 6.1 – Dos capturas de pantalla del videojuego *Grand theft auto IV*, desarrollado por Rockstar North.



Fig 6.2 – Captura de pantalla del "serious game" *Zero Hour*, para entrenamiento ante emergencias, de la empresa Virtual Heroes.

## 6 Game Engines

Como ya se comentó, gran parte de la evolución de la informática gráfica es debida al motor económico que suponen el mundo del cine y el del videojuego. En un origen, los desarrollos científicos y tecnológicos permitieron crear imágenes (y sonidos) nunca antes vistos. Los espectadores comenzaron a pagar por disfrutar de estos espectáculos y, poco a poco, las tornas se invirtieron. El público empezó a solicitar a la ciencia nuevas formas de ser sorprendidos.

En el caso del cine, se ha desarrollado una tecnología tal, que ha llevado a la creación del subgénero de "cine de efectos especiales" en el cual el objetivo es lograr el "más difícil todavía", lo nunca visto. En este sentido, el último párrafo del capítulo segundo explica las últimas tendencias.

En el caso del videojuego, su evolución ha sido mucho más importante para la informática gráfica en parte por tres razones:

- **La interactividad:** A diferencia de una película, los videojuegos funcionan en tiempo real. Como ya se explicó, esto significa que los cálculos han de ser procesados instantáneamente. Por lo tanto, mejorar los videojuegos implica mejorar la potencia de cálculo de las máquinas y optimizar los algoritmos de renderizado.

- **El equipo:** Mientras que en el cine la tecnología es usada por profesionales, el usuario de videojuegos necesita de un equipo para hacerlos funcionar. Esto implica que los equipos (ordenadores, videoconsolas, dispositivos móviles...) no solo han de ser potentes, sino también accesibles económicamente a la gran mayoría. Además, el interés por sorprender al usuario ha llevado al desarrollo de nuevas tecnologías y periféricos como mandos inalámbricos o con detección de movimiento (Wimote), control sin mandos (Kinect), pantallas 3D autoestereoscópicas (Nintendo 3DS), posibilidad de utilizar Realidad Aumentada...

- **La sinergia:** En estos últimos años, la tecnología informática ha encontrado un nuevo camino de acceso a los usuarios: Internet, la conectividad y las redes sociales. A día de hoy llega a ser difícil diferenciar que es un teléfono móvil de lo que es un ordenador, que es una web de lo que es una aplicación, o que es un videojuego de lo que es una red social.



Fig 6.3 – Captura de pantalla de Doom (idSoftware, 1993), uno de los primeros juegos de disparos en primera persona (first person shooter, FPS).



Fig 6.4 – Escenas de simulación de una plataforma petrolífera creada por RealVisual.

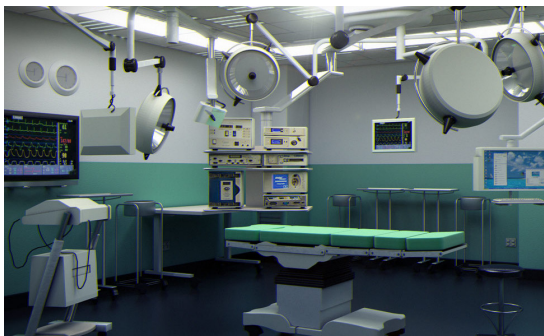


Fig 6.5 – Escenas de simulación de un quirófano creada por RealVisual.

## 6.1 Que es un Game Engine

Se conoce como *game engine* o *motor de videojuego* el software que permite la creación y ejecución de un videojuego. Los motores de videojuego pueden trabajar con gráficos 2D y 3D, sonido o físicas entre otros.

Por otro lado, estos programas permiten el desarrollo de software compatible con multitud de plataformas (PC, MAC, Android, Web, Wii, PS3, Xbox...).

Antes del desarrollo de los game engines, los juegos se programaban como aplicaciones individuales creadas desde cero. Posteriormente, algunas grandes compañías crearon programas para la elaboración de videojuegos, para su uso propio, de forma que les facilitase la elaboración de distintos juegos con una base común.

En la época de los primeros shooters 3D en primera persona (Wolfstein3D, Doom...) comienzan a aparecer los primeros motores disponibles para terceras compañías. Esto permitía, en vez de programar completamente el videojuego, crear escenarios, personajes y demás contenido, delegando la coordinación del conjunto al motor.

En los últimos años se han desarrollado una serie de motores de videojuego orientados a un tipo de creación sencilla y visual, permitiendo que personas sin formación avanzada en programación puedan desarrollar aplicaciones. Esto ha coincidido con el aumento de pequeñas compañías independientes que desarrollan sencillas aplicaciones, por ejemplo para dispositivos móviles.

### Game Engines como motores de Realidad Virtual Low Cost

La facilidad de creación de juegos con estas herramientas, y su bajo coste, han permitido que disciplinas ajenas al negocio del entretenimiento (Medicina, psicología, educación, ciencia e industria ) puedan desarrollar aplicaciones con muy diversos propósitos.

Los dos usos principales son la creación de simuladores y de serious games ("juegos" destinados al aprendizaje o mejora de habilidades, o tratamiento de problemas).





Fig 6.6 – La compañía Vizerra ha desarrollado un software, basado en el motor Unity3D, para representación arquitectónica en tiempo real.

En la imagen, vistas del proyecto de una estación en Barcelona (barrio de Sagrera), del centro histórico de Praga, plaza de San Marcos de Venecia, Coliseo de Roma y Machu Pichu (de arriba abajo).

**Las principales ventajas del uso de los game engines son:**

- Coste económico, frente al desarrollo de aplicaciones de forma autónoma.
- Sencillez de diseño y creación, no siendo necesarios conocimientos avanzados de programación para desarrollar aplicaciones, permitiendo su uso a personal de diversos ámbitos de estudio.
- Posibilidad de aprovechar modelos tridimensionales creados con otro software.
- Posibilidad de crear aplicaciones ejecutables en equipos de uso común, lo cual amplía el ámbito de uso.
- Su evolución y desarrollo avanza al ritmo de la industria del videojuego.

**Las principales limitaciones o inconvenientes son:**

- En principio no están diseñados para trabajar con dispositivos hápticos ni con sistemas estereoscópicos, por lo que en principio no se consigue inmersión total.
- Las limitaciones particulares de cada game engine. Es difícil salirse de las opciones que permite el programa.

En concreto, la visualización arquitectónica se puede ver enormemente beneficiada en el futuro por el uso de Game Engines por los siguientes motivos (entre otros):

- 1º La arquitectura es un ámbito que se presta, por su propia naturaleza, a ser inspeccionada por el usuario. Moverse libremente, frente a una imagen estática o una animación predefinida, es un gran salto adelante en cuanto a la sensación de presencia.
- 2º Los entornos arquitectónicos, urbanos o paisajísticos son elementos imprescindibles en cualquier videojuego y, por tanto, están convenientemente considerados en los motores de juego. Trabajar con arquitectura es nativamente sencillo (p.e. más sencillo que con moléculas).
- 3º Conceptos como visualización en primera persona, iluminación y materialidad realista, sombras, etc. que son importantes en la visualización arquitectónica, son también importantes en los videojuegos y, por lo tanto, también están correctamente tratados en los motores de juego.
- 4º Otros aspectos como el sonido o la interacción con objetos y personajes que hasta ahora no han tenido importancia en la visualización arquitectónica, pero están implementados en los motores de juego, pueden ser utilizados. Oír el agua al acercarse a una fuente, abrir una puerta, encender o apagar luces, o escuchar a un guía son algunas de las posibilidades que se abren.

## 6.2 Algunos ejemplos de aplicación

Los game engines han sido utilizados para mejorar la visualización de entornos arquitectónicos (Moloney, J, et al. 2003), así como para entrenar a trabajadores de la construcción (Juang, J.R, 2011; Fumarola, M. 2011)).

También se ha utilizado tecnología de juegos para la presentación de promociones inmobiliarias (Miliano, V. 1999). De hecho, Reika y Weimin recrearon una vivienda en la que el usuario podía disponer los muebles a su antojo y comprobar el resultado.

En urbanismo se han utilizado para mostrar a grandes grupos de personas simulaciones de entornos, para conocer sus preferencias (Bishop, I, 2011). Ejemplos de ello son las actuaciones urbanas en Victory Avenue, Shenyang, o el Plan Urbano de Nanhu (Reika, S, Weimin, G, 2011).

Quizá una de las investigaciones más interesantes la están llevando a cabo Wei Yan, Charles Culp, y Robert Graf de la Universidad de Texas (2011). Han integrado BIM con visualización basada en un Game Engine de manera que permite a los propios diseñadores navegar por sus creaciones durante el proceso de desarrollo de manera interactiva, en tiempo real y con calidad fotorrealista.

Su trabajo, que conecta disciplinas tan dispares como la arquitectura e ingeniería con programación, visualización y desarrollo de juegos, toca casi todos los puntos tratados en este trabajo: modelado arquitectónico, materialidad, iluminación, animación, detección de colisiones o simulación de efectos físicos, entre otros.

Por otro lado, también se han utilizado motores de juego en disciplinas distintas a la arquitectura.

Dell, Starbucks, Samsung o Suzuki han utilizado juegos que, además de mejorar la experiencia del cliente, permiten recopilar información que puede ser utilizada para incrementar las ventas.

Para la mejora de habilidades directivas y empresariales existen "serious games" como AirwaySim Game, Business Tycoon, Capitalism, Virtonomics, Industry Player (Kosmadoudi, Z, et al. 2012).

En medicina, Triage Trainer, Pulse o Dental Implant Training Simulation permiten mejorar las habilidades de los facultativos.

De hecho, muchas de las aplicaciones descritas en el apartado de Realidad Virtual (apartados 5.7 a 5.13) podrían ser simuladas mediante game engines, dentro de sus posibilidades.



Fig 6.7 – Imágenes de los estudios de Reika y Weimin. Vivienda amueblable por el usuario (arriba) y vistas del Plan Urbano de Nanhu (centro y abajo).

### 6.3 Game Engines

En este apartado se describen tres Game Engines habituales en la creación de juegos 3D en primera persona.

#### CryENGINE 3

Posee uno de los motores gráficos más potentes, permitiendo el cálculo de iluminación indirecta en tiempo real, tanto en elementos móviles como inmóviles. Permite también el uso de grandes cantidades de luces y acepta iluminación HDR, sombras y Ambient Occlusion.

Permite mapas de imagen, de normales, de desplazamiento y de opacidad. Admite reflexiones y subsurface scattering.

Simula profundidad de campo y desenfocado de movimiento.

Integra vegetación, creación de terrenos, fluidos, sistemas de partículas y niebla volumétrica.

Permite simulación física de colisiones, solido rígido, deformable y rompible, así como cinemáticas directas e inversas.

Incluye sonido dinámico y envolvente.

Permite visión estereoscópica.

Pero para procesar semejante cantidad de datos, CryENGINE 3 pide unos requerimientos exigentes al equipo. Sistema operativo de 64 bits, procesador i7 de al menos 3GHz, 4GB de RAM y tarjeta gráfica con al menos 2GB de memoria dedicada. Estos requisitos no los cumple a día de hoy cualquier ordenador.

Aparte de la licencia para la creación de juegos, CryENGINE ofrece una licencia especial para visualización y trabajo de investigación.

ENODO en Francia e Inreal en Alemania utilizan CryENGINE para la visualización arquitectónica.



Fig 6.8 – Tres imágenes de ejemplo generadas con CryEngine3 y mostradas en su página web.

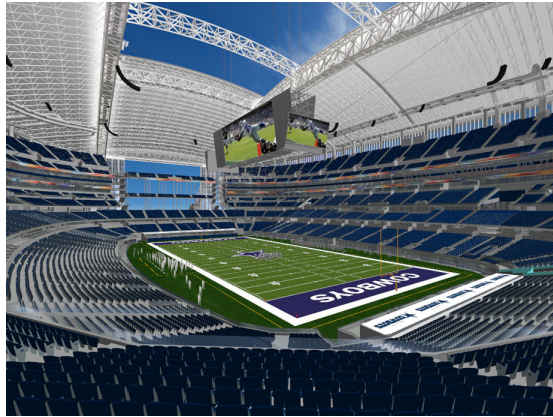


Fig 6.9 – Estadio de los Dallas Cowboys.

Imagen creada por HKS usando UDK.



Fig 6.10 – Vivienda unifamiliar desarrollada por el estudiante Luigi Russo usando UDK.

### Unreal Development Kit UDK

Tiene unas características similares a CryENGINE, quizá con unas capacidades gráficas ligeramente inferiores.

Por otro lado, sus requerimientos, siendo importantes, no lo son tanto como en CryENGINE.

La licencia de UDK es gratuita para usos educativos o no comerciales. Para la creación de juegos UDK cobra una licencia de 100\$ y, a partir de unos ingresos de 50.000\$ por la venta del juego, un porcentaje del 25%. Las compañías que utilizan UDK para visualización arquitectónica se limitan a pagar la licencia inicial.

Entre los usos distintos al mercado del videojuego, Yost Engineering, Inc. (YEI Technology) ha desarrollado sistemas de captura del movimiento corporal y trackeo de la cabeza para controlar un Head-Mounted Display utilizando el motor UDK.

En visualización arquitectónica, la firma americana HKS, con un volumen de casi 200 proyectos anuales, tiene entre su equipo cuatro arquitectos dedicados a crear simulaciones virtuales de sus proyectos más emblemáticos, usando UDK. Encontramos ejemplos en el hotel W de Dallas o los estadios Dallas Cowboys, Indianápolis Colts o England Soccer Stadium de Liverpool.

Destacable es también el proyecto arquitectónico de una vivienda unifamiliar desarrollado por Luigi Russo, estudiante de la Facultad de Ingeniería de la Universidad Federico II de Nápoles (Italia), modelado con el software gratuito Blender e insertado en una versión educacional de UDK, como parte de un trabajo universitario y sin conocimientos previos.

En simulación y creación de "serious games", la empresa norteamericana Virtual Heroes ha creado diversas aplicaciones; Creado en colaboración con la NASA, Moonbase Alpha es un simulador de entrenamiento para astronautas. Pamoja Mtaani es un juego destinado a educar en la prevención contra el SIDA a los jóvenes de Nairobi. Tanto HumanSim como Zero Hour son simuladores para el entrenamiento de personal sanitario, tanto civil como militar, ante emergencias.

### Unity 3D



Fig. 6.11 – Distintas vistas del centro urbano de Dubuque (Iowa) creadas por ArchVirtual utilizando el software Unity3D Pro.

Unity 3D no alcanza las capacidades gráficas de los dos motores ya mencionados pero, por el contrario, los requerimientos del equipo son bastante inferiores y además resulta relativamente más fácil de aprender y por lo tanto ha alcanzado gran popularidad.

Otro de sus aspectos destacados es que permite exportar el juego a multitud de plataformas, incluyendo PC y MAC, Linux, Xbox 360, PlayStation 3, Wii, Wii U, iPad, iPhone, Android (compatible con la mayoría de tablets y smartphones) e incluso incrustarse en páginas web para jugar desde el navegador.

Otra ventaja es que Unity tiene una versión completamente gratuita con la que se pueden crear y comercializar juegos. Si el desarrollador supera unos beneficios de 100.000\$ o desea más altas prestaciones (sombras en tiempo real, light mapping, efectos...), deberá comprar Unity Pro por unos 1500\$.

Este programa ha sido utilizado ampliamente en usos distintos al del entretenimiento:

ArchVirtual (<http://archvirtual.com>) utiliza Unity3D Pro para la visualización arquitectónica. Su proyecto más importante es la simulación del entorno urbano del centro de Dubuque (Iowa), de forma que se pueda incorporar información municipal y de comercios, de forma que se pueda llegar a obtener una réplica virtual interactiva de la comunidad.

RealVisual (<http://www.real-visual.com/>) es una empresa inglesa que se basa en Unity3D para ofrecer sus servicios de visualización, simulación en tiempo real y serious games. Simula desde portaaviones hasta plataformas petrolíferas.

Heartwood Inc. (<http://hwd3d.com/>) también desarrolla serious games para el uso en el aprendizaje de maquinaria e instrumental técnico, mediante aplicaciones de entrenamiento y manuales de usuario 3D operables desde dispositivos móviles.

Vizerra (<http://vizerra.com/>) desarrolla entornos urbanos para visualización de proyectos (reconstrucción del distrito de Sagrera, en Barcelona) y con propósitos turísticos. En este ámbito han recreado la Plaza Roja de Moscú, la Plaza de San Marcos de Venecia, el centro histórico de Praga, el Coliseo romano, el Taj Mahal, Machu Pichu o Petra.

## 7. Caso Práctico: Unity 3D.

En este capítulo de investigación, se va a probar la usabilidad de un motor de videojuego como sistema de bajo coste de acceso a la visualización arquitectónica interactiva en tiempo real.

Se ha elegido el programa Unity3D (<http://unity3d.com/>) ya que permite su descarga gratuita, no tiene unos requerimientos de equipo elevados, permite exportar a múltiples plataformas incluyendo dispositivos móviles y tiene tutoriales de aprendizaje disponibles on-line.

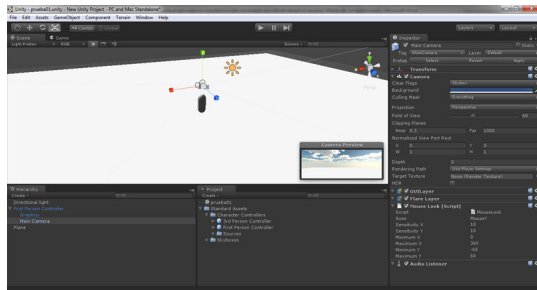


Fig. 7. 1 – interfaz de usuario de Unity3D Pro.

### 7.1 Comenzando a trabajar.

#### Interfaz de usuario

Al abrir el programa encontramos la pantalla principal dividida en las siguientes áreas (al estar el programa disponible únicamente en inglés se irán traduciendo las distintas partes):

- Scene (escena): Un área de visualización en perspectiva de la geometría y objetos pertenecientes al proyecto abierto.
- Game (juego): Una pantalla donde se permite simular el juego en cualquier momento de la creación, para comprobar su correcto funcionamiento.
- Menú Hierarchy (jerarquía): Aquí se crean y listan todos elementos que aparecen en el juego, como geometría, cámaras, luces, sonidos, efectos...
- Menú Project (proyecto): Aquí se crean y listan los assets o recursos que se pueden utilizar para crear el videojuego, modelos 3D, líneas de programación, shaders de texturas, skyboxes, sonidos... que después pueden ser instanciados y colocados en el juego.
- Menú Inspector: Al seleccionar cualquier elemento de los dos menús anteriores, aquí nos aparece su descripción y posibilidades de configuración detalladas.
- Menús superiores: Además de los clásicos *File*, *Edit*, *Window* y *Help*, encontramos *Assets* (recursos), que permite crear o importar elementos para la creación del juego (p.e. modelos 3D), *Game Object*, que permite crear o importar otro tipo de elementos (geometrías, sonidos, efectos, cámaras, fuerzas físicas...), *Component*, que permite insertar otro tipo de elementos y *Terrain* (terreno), que permite crear y modificar terrenos.

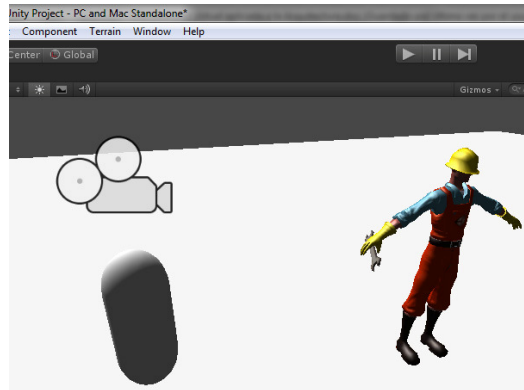


Fig. 7. 2 – Person controllers. First person a la izquierda y third person a la derecha.

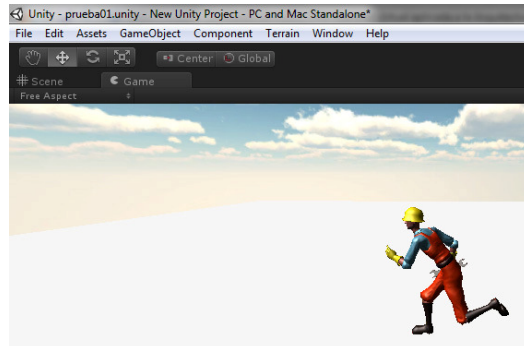


Fig. 7. 3 – Punto de vista del Third person controller en un momento de simulación de juego.

La visión en tercera persona no es apropiada para visualización arquitectónica.

Por otro lado, al iniciar un nuevo proyecto, el programa nos permite cargar una serie de assets:

- Character controller (controladores de personaje).
- Glass refractor.
- Image effects.
- Light cookies...

Lo cual, junto a que no existen a primera vista herramientas de creación de geometría (líneas, objetos...) lleva a pensar que el modo de trabajo es más de ensamblaje de elementos creados por otro software que de creación desde cero.

Por otro lado, en una primera impresión, la interfaz de usuario es poco amigable para un usuario habitual de CAD (arquitecto) que nunca ha utilizado un motor de videojuegos, ya que hace referencia a conceptos a los que no estamos acostumbrados.

### Person Controller

Debería ser el primer objeto a crear. Un *Person Controller* es el gizmo que representa al personaje que se controla en el juego. Existen dos tipos; First person controller (primera persona) y third person controller (tercera persona) según el tipo de visión.

Es lógico que para visualización arquitectónica se trabaje siempre en primera persona.

El Person controller lleva asociado:

- Avatar: en el caso de visión en tercera persona será un personaje visible. En el caso de primera persona (no visible) puede ser simbolizado por un objeto.
- Cámara: Que es la que da el punto de vista de la escena. En el caso de primera persona, la cámara está ubicada en los ojos del personaje, mientras que en tercera persona le sigue desde atrás.
- Controles de movimiento: El personaje se puede mover mediante teclado y ratón pudiéndose configurar los controles de movimiento.
- Propiedades dinámicas y físicas: Aquí se definen la velocidad, altura de salto, gravedad, pendientes máximas que puede subir, etc que afectan al personaje.

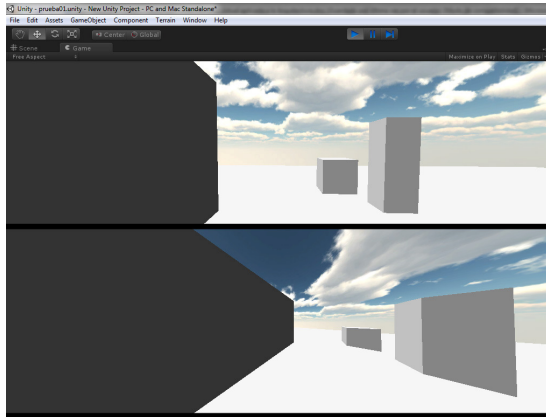


Fig. 7. 4 – Punto de vista del First person controller.

La imagen superior muestra el campo de visión por defecto de 60° y la inferior se ha forzado a 120°.

## Cámara

Es un elemento muy importante porque controla como se ve el entorno virtual. Permite elegir:

- Fondo del entorno, pudiendo usar skyboxes que aportan realismo a la escena. Se deberían usar siempre (ver Cube Mapping en el apartado 4.3).
- Tipo de proyección, perspectiva u ortogonal. Para visualización inmersiva debería usarse perspectiva, aunque para ciertas aplicaciones puede ser de utilidad la perspectiva ortogonal.
- Campo de visión (en grados). Por defecto está definido en 60° que equivale al campo de visión humana formal (no periférica) aunque se puede variar. Una mayor amplitud puede ser útil en interiores aunque deforme la perspectiva y no sea natural.
- Clipping planes: son dos planos, frontal y trasero, entre los que renderiza la cámara. Todo lo que quede fuera de ellos no es representado.
- Tipo de renderizado, con tres tipos de calidades.

## 7.2 Insertar contenido arquitectónico

El programa permite crear unas pocas primitivas (prismas, esferas...) que no son en ningún caso suficientes para crear entornos arquitectónicos, por lo tanto cualquier modelo se deberá crear con un programa de modelado 3D y posteriormente será insertado.

Los formatos admitidos son los siguientes según la guía de uso on-line;

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

- Importación nativa desde los programas Maya, Cinema 4D, 3ds Max, Cheetah3D, Modo, Lightwave y Blender. De estos programas, en el ámbito de la arquitectura es conocido 3ds Max (aunque no es fácil de usar) y Blender está ganando adeptos al ser potente y gratuito.
- Importación de los formatos fbx, dae, 3ds, dxf y obj. Esto nos permitirá importar desde otros programas de uso habitual como puedan ser Sketchup, Rhinoceros o Revit.



### Insertar modelos desde AutoCAD

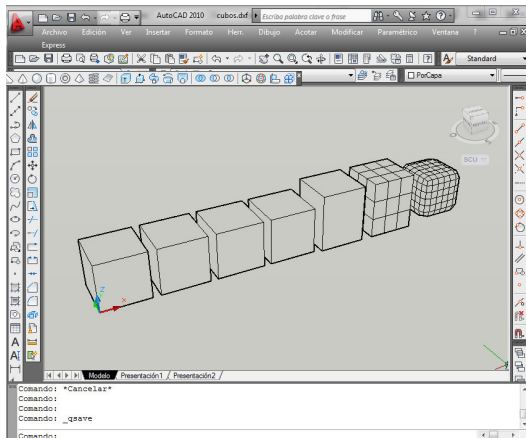


Fig. 7. 5 – Los siete cubos creados en AutoCAD por distintos métodos para su exportación a formato DXF.

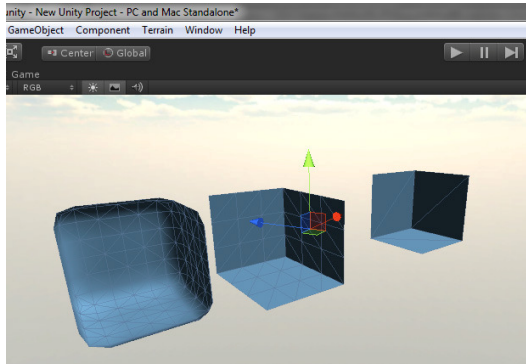


Fig. 7. 6 – Solo los tres cubos formados por caras 3D son reconocidos por el programa pero, aún así, las normales de las caras se han importado al revés.

Al tener Unity3D por defecto la opción de backface culling activada, solo se ven las caras ocultas de los objetos.

El formato nativo de AutoCAD es el DWG. También permite exportar en dxf, sat, stl, eps, dxx, bmp y dgn, de los cuales solo dxf es compatible con Unity3D.

Hay que hacer constar que hoy en día, de las tres opciones disponibles de modelado en AutoCAD (sólidos, mallas y 3D caras) se usan generalmente sólidos (opción 1). Puede ser usado alguna vez el modelado mediante malla (opción 5) (más en diseño que en arquitectura porque es un sistema más preparado para controlar curvas), y prácticamente nunca 3D caras al ser un sistema primitivo.

Por otro lado, se pueden convertir sólidos a mallas y viceversa, y se pueden descomponer mallas en 3D caras sin opción inversa.

Para comprobar la exportación de distintos elementos desde AutoCAD se crean una serie de cubos en AutoCAD por distintos métodos:

1º Cubo creado mediante una primitiva de sólido.

2º Cubo creado por regiones a partir de la descomposición del anterior.

3º Cubo creado mediante superficies planas.

4º Cubo creado mediante caras 3D.

5º Cubo creado mediante una primitiva de malla.

6º Cubo creado por caras 3D a partir de la descomposición del anterior.

7º Cubo creado por caras 3D a partir de la transformación del sólido (1º) en malla (5º) y posterior descomposición (6º).

Al exportar como dxf directamente a Unity 3D se observa que el programa solo ha reconocido las caras 3D, es decir los cubos 4º, 6º y 7º. Por otro lado, en la importación las normales de cada cara se han interpretado en el sentido contrario, y como el shader del material tiene por defecto la ocultación de caras traseras (backside culling) solo se ven las caras ocultas.

Este problema se puede corregir programando un script en C# que invierta las normales de las caras, o programando un shader que en vez de ocultar las caras traseras de los objetos oculte las delanteras. La primera opción parece más práctica ya que permite voltear solo las normales de las mallas que sea necesario, e independientemente de su material.

El siguiente script, creado por Joachim Ante y colgado en el foro de la comunidad de usuarios de Unity 3D permite voltear las normales:

```
using UnityEngine;
using System.Collections;
[RequireComponent(typeof(MeshFilter))]
public class ReverseNormals : MonoBehaviour {
    void Start () {
        MeshFilter filter = GetComponent(typeof (MeshFilter)) as MeshFilter;
        if (filter != null)
        {
            Mesh mesh = filter.mesh;
            Vector3[] normals = mesh.normals;
            for (int i=0;i<normals.Length;i++)
                normals[i] = -normals[i];
            mesh.normals = normals;
            for (int m=0;m<mesh.subMeshCount;m++)
            {
                int[] triangles = mesh.GetTriangles(m);
                for (int i=0;i<triangles.Length;i+=3)
                {
                    int temp = triangles[i + 0];
                    triangles[i + 0] = triangles[i + 1];
                    triangles[i + 1] = temp;
                }
                mesh.SetTriangles(triangles, m);
            }
        }
    }
}
```

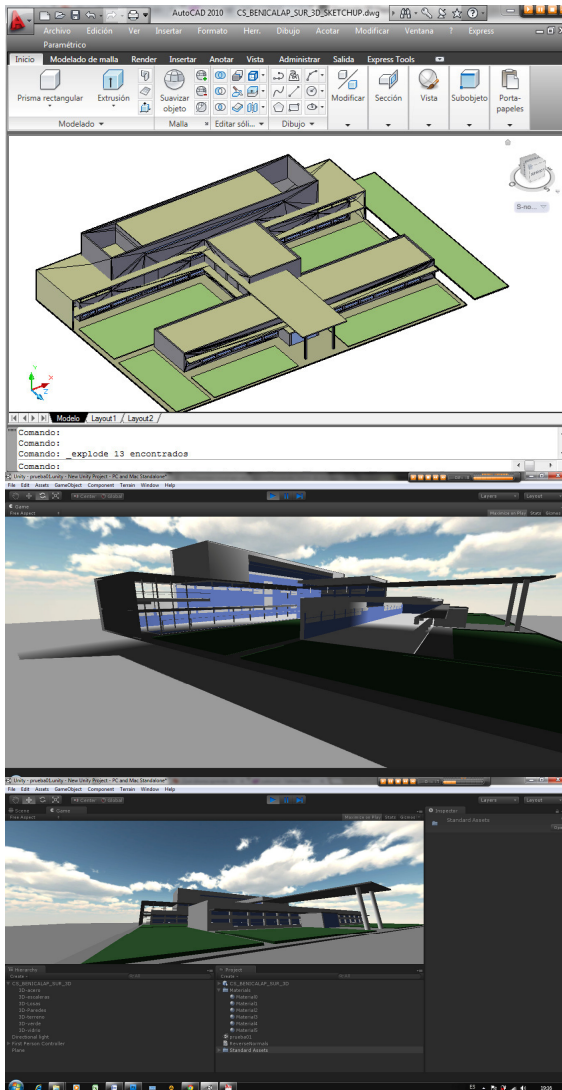


Fig. 7. 7 – Al importar de AutoCAD un modelo más complejo (arriba) se aprecia mejor la visualización incorrecta de las caras (centro).

El script “reverse normals” corrige la visualización (abajo), pero se generarán errores al utilizar iluminación indirecta mediante lightmaps.

Por tanto, el proceso más directo para importar desde AutoCAD (modelado sólido) sería:

- 1º Convertir el modelo sólido en modelo de malla, en AutoCAD.
- 2º Descomponer la malla en 3D caras, en AutoCAD.
- 3º Guardar el archivo en formato dxf, en AutoCAD.
- 4º Importar el dxf en Unity 3D.
- 5º Aplicar el script “reverse normals” a todas las mallas importadas.

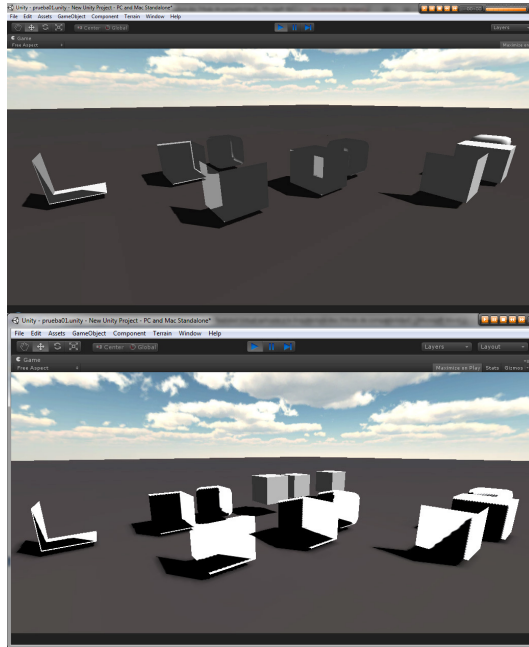


Fig. 7. 8 –

Imagen superior, iluminación directa:

- izquierda: cubos con normales invertidas.
- centro: cubos con script "cull off".
- derecha: cubos con script "reverse normals"

Imagen inferior, usando lightmaps:

- izquierda: cubos con normales invertidas.
- centro: cubos con script "cull off".
- derecha: cubos con script "reverse normals"
- fondo: cubos de referencia.

Se aprecia que la visualización de los objetos importados en ningún caso es correcta.

Sin embargo, se aprecian problemas en las sombras y en la iluminación. Más aún si se utiliza lightmapping para "tostar" iluminación indirecta (como se verá más adelante) para conseguir un mayor realismo (ver apartado 4.2 "lightmaps").

Se han creado pruebas de visualización de los cubos de testeo sin y con iluminación indirecta tostada y los resultados son desastrosos comparados con la visualización de cubos de referencia creados directamente en Unity 3D.

En las imágenes se pueden apreciar:

- izquierda: Cubos importados directamente desde AutoCAD con las normales invertidas.
- centro: Cubos a los que se les ha modificado el material añadiendo la orden "Cull Off" en la programación del shader, para que muestre las caras traseras (que en este caso, al estar invertidas las normales, son las caras delanteras).
- derecha: Cubos a los que se les ha aplicado el script "reverse normal" para corregir las normales en la visualización, y que con iluminación directa se visualizan relativamente bien pero al calcular los lightmaps producen errores.
- fondo: en la segunda imagen se ven los cubos de referencia creados en Unity 3D.

Por lo tanto, vistos los resultados (y sin haber entrado a probar la importación de materiales y texturas), parece claro que no es una opción factible insertar directamente modelos desde AutoCAD a Unity 3D. Las opciones principales pasarían por exportar el DWG a 3ds Max (conocido entre los arquitectos y también de Autodesk) o Blender (gratis), y de aquí nativamente a Unity 3D, o bien exportar el DWG a Sketchup (gratis y fácil de usar) o Rhinoceros (conocido entre los diseñadores y también de Autodesk).

### Insertar modelos desde 3ds Max

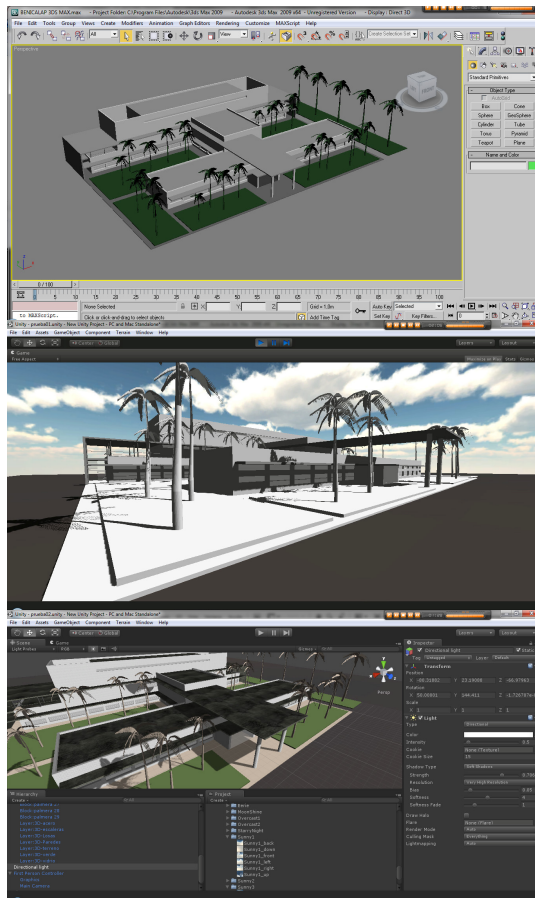


Fig. 7. 9 – Modelo de AutoCAD importado correctamente en 3ds Max (arriba).

Inserción del archivo MAX en Unity3D (centro), y aplicación de algunos materiales (abajo).

3ds Max permite guardar los archivos por defecto en formato MAX, el cual es aceptado directamente por Unity3D conservando la siguiente información:

- Todos los nodos con su posición, rotación y escala.
- Nombres de los objetos.
- Mallas con color de vértice y normales.
- Material con textura difusa y color de base (múltiples materiales por malla).

Por otro lado, 3ds Max permite exportar en todos los formatos aceptados por Unity3D, FBX, DAE, DXF, OBJ y 3DS, siendo aconsejable, según el manual on-line, exportar en formato FBX.

Para la prueba de exportación se ha importado previamente el modelo de edificio generado en CAD (en formato DWG) en 3ds Max. A partir de ahí, se ha exportado en todos los formatos posibles.

Hay que comentar que el archivo original importado DWG tiene un tamaño de 701 KB, el archivo MAX 864 KB, el FBX 13.274 KB, el DAE 14.214 KB, el DXF 2.394 KB, el OBJ 6.920 KB y el 3DS 400 KB, siendo recomendable, a igual calidad, el de menor tamaño ya que requiere menos recursos para ser procesado posteriormente.

Al exportar en formato MAX y los resultados son correctos. La escala y ejes del modelo MAX coinciden con los de Unity3D, además las normales se han importado correctamente y no se producen artefactos al utilizar iluminación. Al probar materiales en los objetos importados, estos se muestran correctamente, manteniendo la escala de mapeado asignada en 3ds Max.

La inserción desde formato FBX (recomendado en el manual de usuario) también se realiza correctamente, escala y texturas incluidas.

También la inserción desde archivo OBJ aunque, en este caso, el archivo ha tardado más tiempo en ser insertado y ha aparecido a escala 100/1 (lo cual se corrige fácilmente).

La inserción desde 3DS también ha sido correcta, siendo éste el archivo más ligero. La inserción desde formato DAE ha dado es incorrecta, pues no se ve la geometría.

La inserción desde DXF genera los mismos problemas de visualización que desde AutoCAD, en lo relativo a caras con normales invertidas.

### Insertar modelos desde Sketchup

Es interesante estudiar la importación de modelos desde Google Sketchup.

Éste es un modelador 3D a partir de líneas y caras que permite trabajar con polígonos facetados creando desde cero o importando entidades desde AutoCAD.

El programa no trabaja con sólidos ni curvas tipo NURBS ni genera renders por si mismo pero tiene una serie de características lo hacen muy ventajoso para los arquitectos o estudiantes de arquitectura frente a sus competidores, entre ellas:

- El programa es gratuito (aunque existe una versión profesional de pago).
- El manejo es muy intuitivo, casi un juego de niños. Completamente distinto al manejo de 3ds Max, o incluso AutoCAD 3D, por ejemplo.
- Permite geolocalizar modelos arquitectónicos en Google Earth.
- Gracias a la colaboración desinteresada de un gran número de usuarios, existe una colección de bloques prácticamente infinita, tanto arquitectónica como de mobiliario o cualquier otro tipo de objeto.
- Permite crear sencillas animaciones de recorrido o estudios solares, lo cual es suficiente (a nivel de animación) para el uso habitual en arquitectura.
- Permite exportar a CAD, BIM y motores de render.

Además de su formato propio SKP, el programa permite la exportar en los formatos DAE, OBJ, 3DS y DXF compatibles con Unity3D, además de DWG y WRL utilizado en el lenguaje de Realidad Virtual VRML.

Se inserta en Sketchup un modelo del Museo Príncipe Felipe de Valencia, descargado de la galería 3D online del programa, con geometría compleja y texturas incorporadas.

La exportación en formato FBX se produce correctamente, generando un archivo con la geometría y un directorio con todas las imágenes de textura. Al copiar todos los archivos a la carpeta del proyecto de Unity3D, la exportación se hace correctamente.

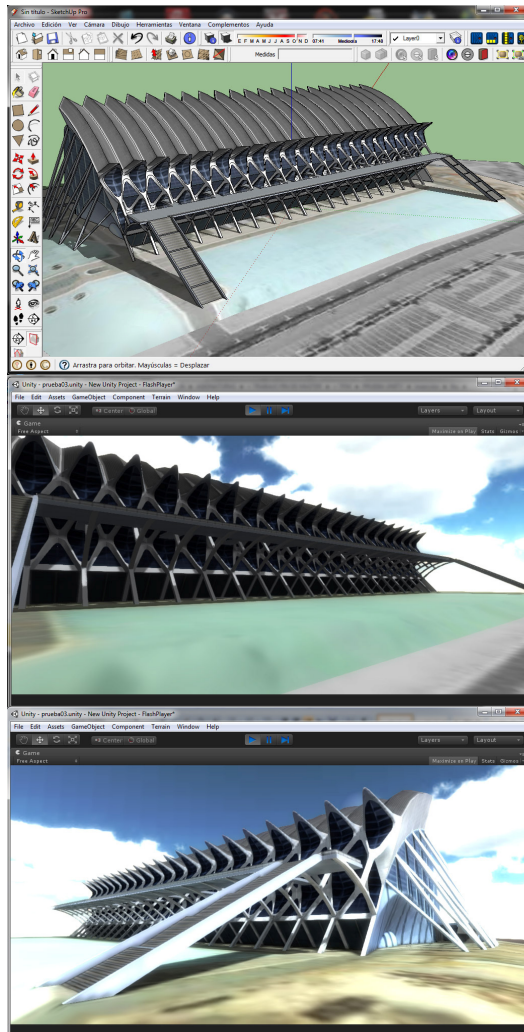


Fig. 7. 10 – Modelo de SketchUp importado desde la Galería 3D (arriba).

Inserción del archivo FBX creado por SketchUp en Unity3D (centro) y visualización tras “tostar” los lightmaps para conseguir iluminación indirecta (abajo).

### Insertar modelos desde Revit

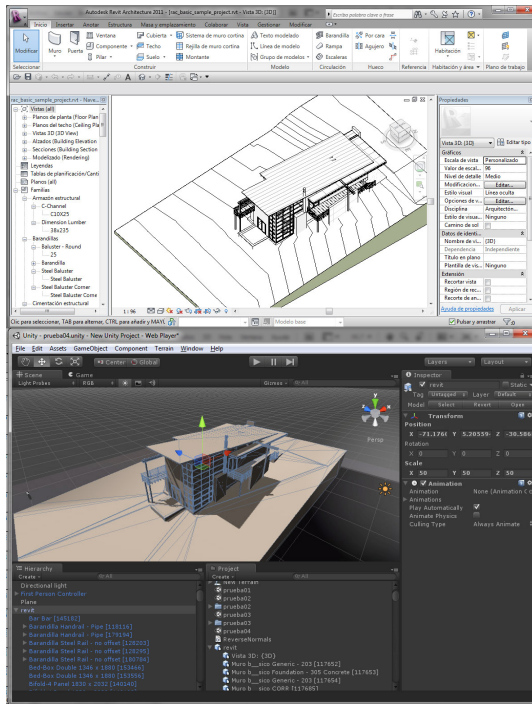


Fig. 7. 11 – Proyecto de Revit (arriba).

Insertación del archivo FBX creado por Revit en Unity3D (abajo). Las texturas se han perdido en la inserción y habrá que introducirlas a mano.

Es interesante, también, estudiar la inserción de modelos desde un programa BIM ya que, además de tener amplio uso en países de Europa y en Estados Unidos, parece ser que en el futuro los programas de CAD les irán cediendo cada vez más protagonismo a éstos.

En este caso se estudia la inserción desde Revit Architecture, de Autodesk, que es uno de los tres grandes BIM.

La inserción se realiza sobre un proyecto completo de ejemplo que incluye Revit en su instalación. Consiste en una vivienda unifamiliar aislada amueblada.

Se abre el archivo en Revit y se comprueba que el proyecto se visualiza correctamente.

Revit permite guardar en su formato nativo RVT y exportar en el formato genérico BIM IFC, en los formatos CAD DWG y DXF y en el formato FBX compatible con Unity3D.

Se genera la exportación sin problemas en Revit y se importa correctamente el archivo en Unity3D. La escala del modelo insertado es extraña pero puede ser debido a que Unity3D trabaja en metros y el modelo de ejemplo estaba desarrollado en unidades anglosajonas.

Al importar en Unity3D, el modelo de Revit aparece como un bloque que contiene todos los elementos arquitectónicos individuales en su interior, de manera que se pueden modificar sus partes (por ejemplo, se pueden mover o borrar muebles contenidos dentro del modelo), o se pueden asignar materiales individualmente a cada elemento.

Por otro lado, al importar un FBX desde Revit, se pierde la información de texturas por elementos, de modo que éstas deberán ser importadas en Unity3D posteriormente y aplicadas individualmente a cada material de la escena.

Otra opción consiste en exportar de Revit a 3ds Max (ambos en la misma versión para ser completamente compatibles) y posteriormente, desde 3ds Max, exportar normalmente en formato MAX a Unity3D.

### 7.3 Iluminación.

Unity3D permite iluminación directa e indirecta.

La iluminación directa la ofrecen cuatro tipos de luces y se calcula en tiempo real, permitiendo modificar la intensidad y el color de la luz, si produce sombras (fuertes o difusas) o si simula destellos de lente (ver apartado 4.3). En una misma escena se pueden utilizar varias luces a la vez, con distintas tonalidades. Los distintos tipos de luz son:

- Point light (puntual): Emite en todas las direcciones (como una bombilla).
- Spot light (foco): Emite en una dirección con una amplitud de cono dada.
- Directional light: Emite en una dirección desde el infinito (rayos paralelos) simulando al sol.
- Area light: Area luminosa que solo se tiene en cuenta en el cálculo de iluminación indirecta.

La iluminación indirecta se realiza mediante lightmapping o tostado de texturas, tal como se explicó en el apartado 4.2. En este caso, la iluminación se precalcula y se simula “pintando” los objetos. Por lo tanto, no consume recursos en tiempo real, pero solo se puede aplicar a objetos estáticos (de lo contrario, los objetos “se llevarían” la iluminación al moverse).

El motor de lightmapping de Unity3D permite seleccionar que objetos y luces son estáticos, la calidad del mapeado, el número de fotones y el número de rebotes de éstos en las superficies, e incluso el color e intensidad de la bóveda celeste.

La visualización arquitectónica tiene la ventaja de que prácticamente todos los objetos son estáticos, por lo que se puede usar en todos ellos iluminación indirecta, que es la que aporta verdadero realismo a la escena.

A continuación se muestran algunas capturas de pantalla de las pruebas realizadas.

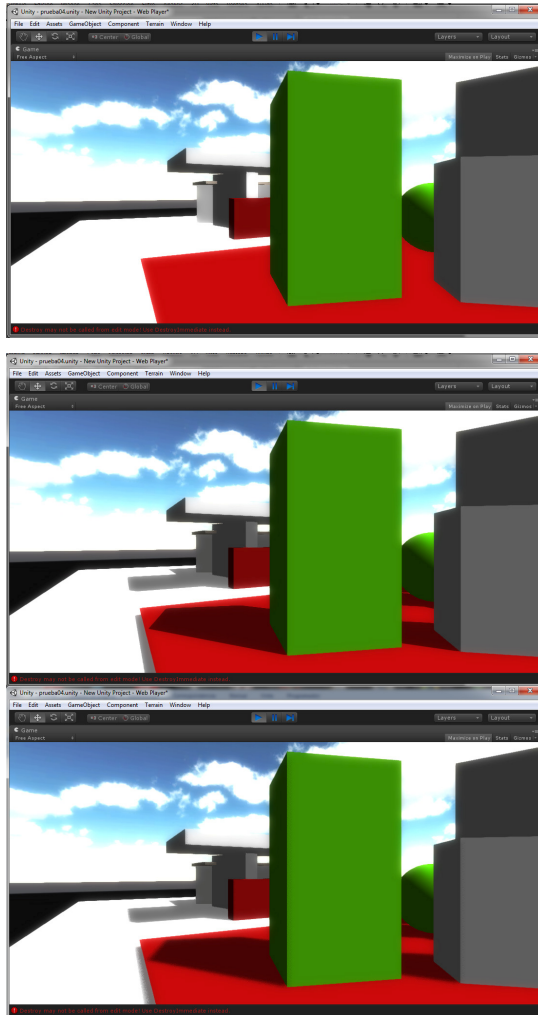


Fig. 7. 12 – Tipos de sombras en tiempo real.

Escena iluminada mediante una luz direccional que no genera sombras (arriba), genera sombras definidas (centro) y genera sombras difusas (abajo). Este último caso es el que consume más tiempo de cálculo.

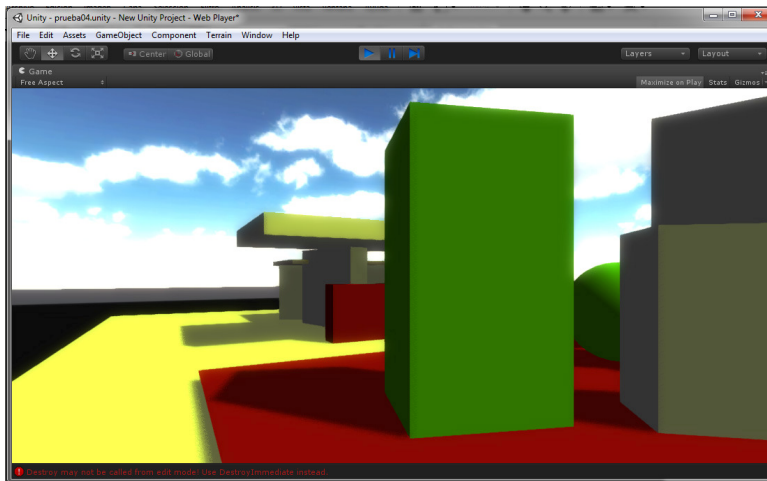


Fig. 7. 13 – Iluminación directa mediante luz de color.

Se pueden crear efectos diurnos o nocturnos eligiendo adecuadamente el tono de las fuentes de luz.

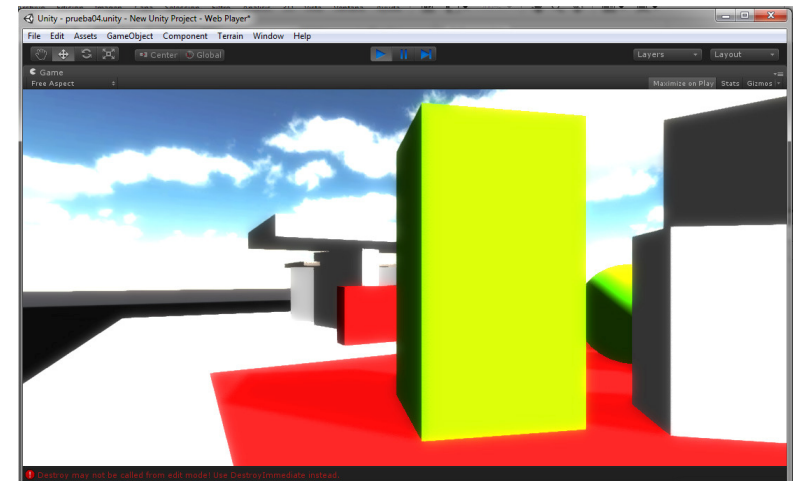


Fig. 7. 14 – La modificación de la intensidad luminosa permite ajustar de forma inmediata la exposición de la escena.

Posteriormente se pueden aplicar filtros para un ajuste más preciso.

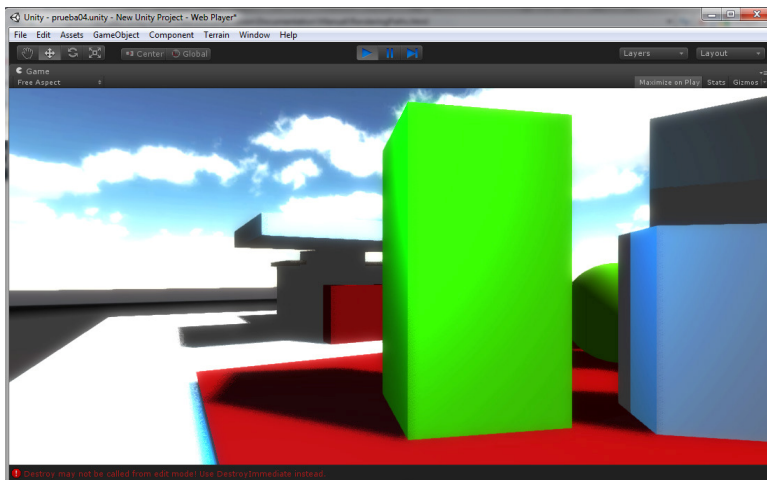


Fig. 7. 15 – Iluminación añadiendo una segunda luz (puntual) de color azul que tiñe y genera sombras en una zona de la escena. Varias luces puntuales pueden simular la iluminación interior en un edificio.

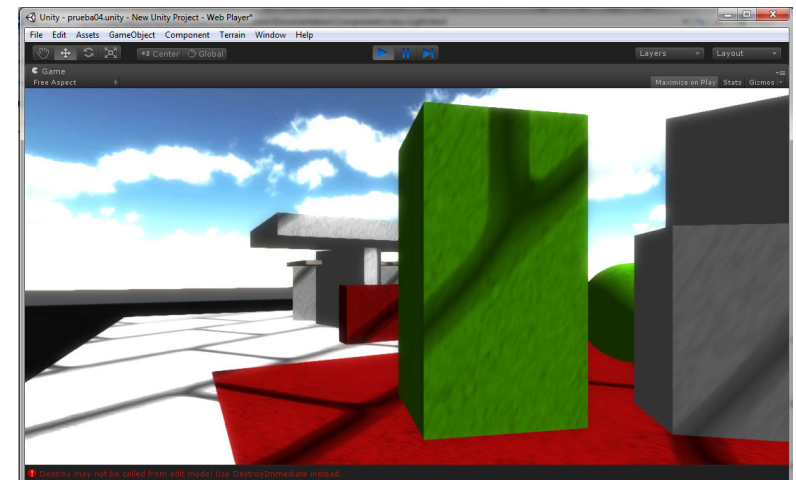


Fig. 7. 16 – La iluminación usando gobos (filtros en el proyector) genera sombras proyectadas en los objetos aumentando la sensación de tridimensionalidad.

Se pueden animar para simular sombras de nubes.



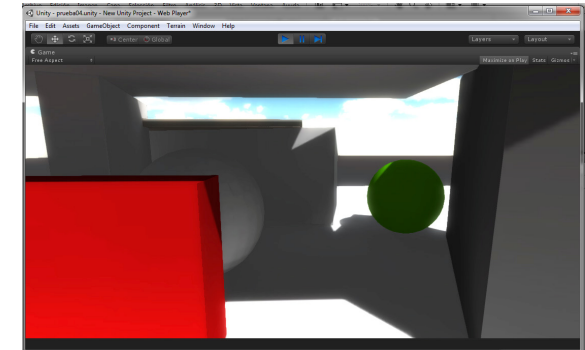
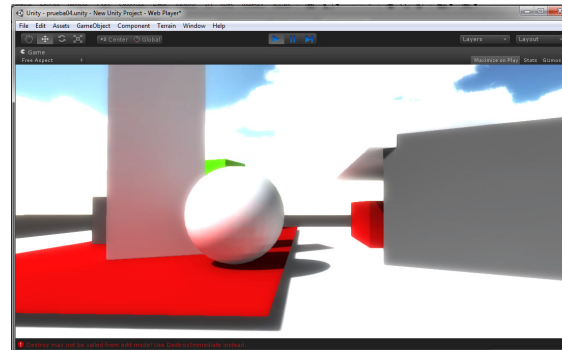
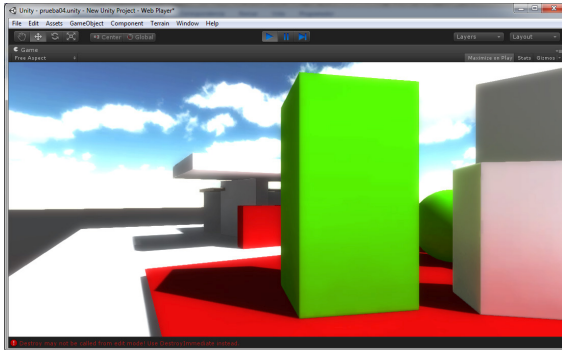


Fig. 7. 18 – Iluminación indirecta mediante lightmapping. Los fotones han efectuado un rebote en las superficies.

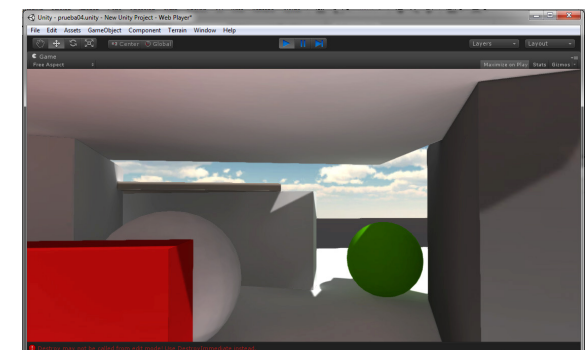
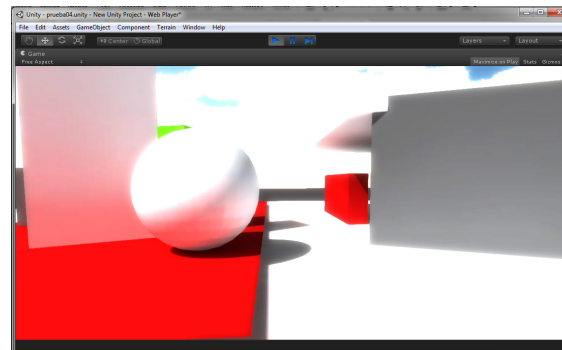
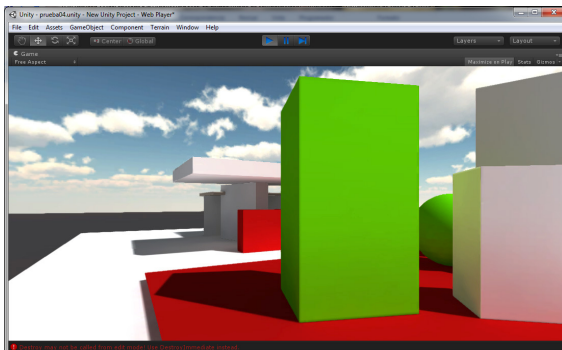


Fig. 7. 19 –Iluminación indirecta mediante lightmapping. Los fotones han efectuado cuatro rebotes en las superficies.

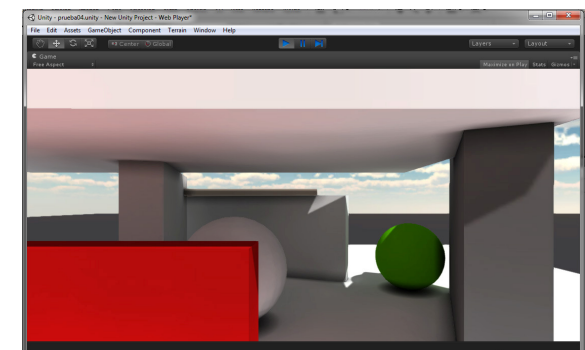
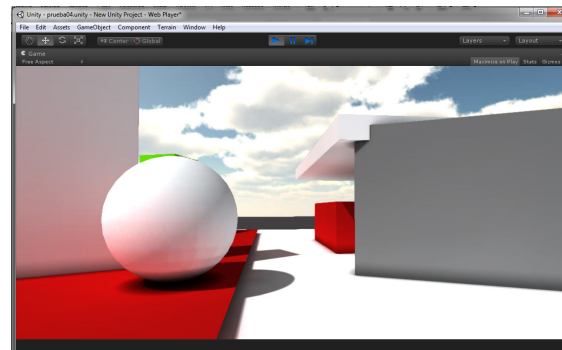
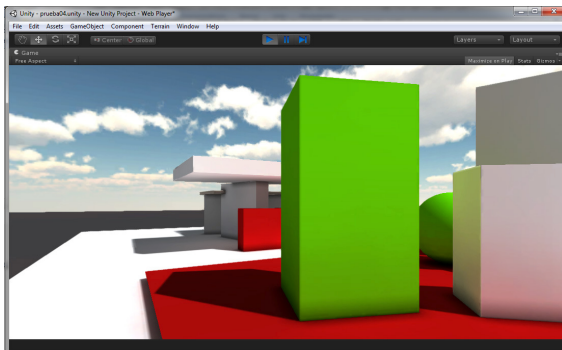


Fig. 7. 20 – Iluminación indirecta mediante lightmapping. Cuatro rebotes más una pasada de Ambient Occlusion que genera sombras en los encuentros.

## 7.4 Materiales

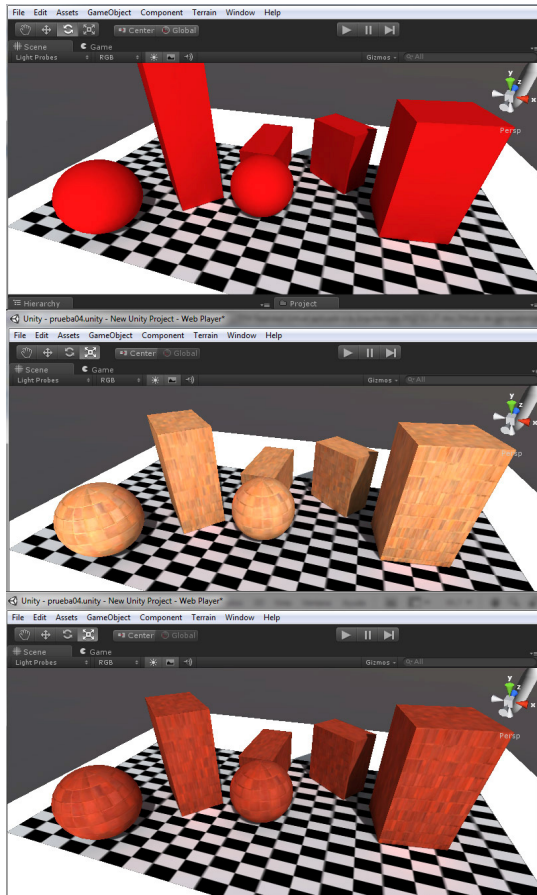


Fig. 7. 21 – Material difuso básico definido por un color (arriba), una textura (centro) y mezcla de ambas (abajo).

En Unity3D los materiales se interpretan mediante shaders, que son las líneas de programación que indican cómo ha de ser la superficie de los objetos a los que son aplicados.

Por defecto, el programa trae instalados una serie de shaders habituales en programas de visualización, para materiales con o sin textura, relieve, opacidad, autoiluminación, etc. que se van a testear en este apartado.

Por otro lado, al ser Unity un motor de programación de videojuegos permite, precisamente, programar nuevos shaders (siempre que se tengan los conocimientos de C# necesarios) para conseguir la apariencia buscada.

### Normal shader Family

Es la familia de shaders básica, y es apropiada para la mayoría de los objetos opacos.

**Diffuse:** Material básico. Permite elegir un color o una textura (imagen 2D), o mezclar ambas.

En arquitectura es válido para modelos con acabado sencillo “tipo maqueta”.

**Specular:** Permite utilizar un color o una textura y definir zonas brillantes del material.

Si la imagen de textura (en formato TIFF) incluye un canal alfa (ver capítulo 4), éste actúa como “gloss map” indicando que zonas son más o menos reflectantes.

En visualización arquitectónica es válido para superficies pulidas (pavimentos) o metálicas. La creación de un canal alfa es un trabajo extra que el diseñador habrá de realizar en photoshop.

**Normal map:** Permite crear materiales con relieve. Permite utilizar un mapa de textura y, además un mapa normal (ver capítulo 4) para indicar el relieve.

En visualización arquitectónica es apropiado para la gran mayoría de materiales que presentan irregularidad superficial. Su uso debería ser habitual.

**Normal map specular:** Combina material especular con relieve (los dos anteriores).

**Parallax:** Es una versión avanzada del Normal Map, consiguiendo un efecto de profundidad más acusado. Requiere un mapa difuso, un mapa normal y un mapa de profundidad. Consume bastantes recursos de sistema.

**Parallax specular:** Combina material especular con el parallax.

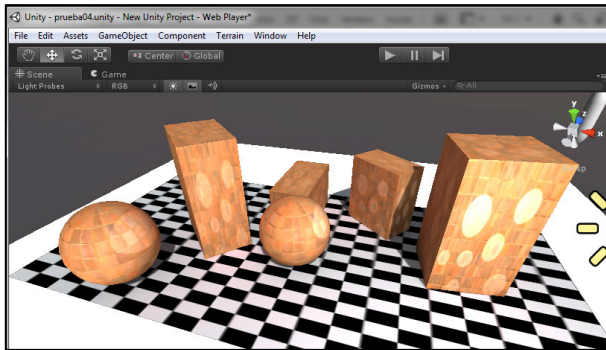


Fig. 7. 22 – Material especular.

La imagen TIFF incorpora, además de la textura cerámica visible, información en un canal alfa en escala de grises para definir el nivel de brillo de la superficie.

En este caso se han simulado zonas redondeadas brillantes.

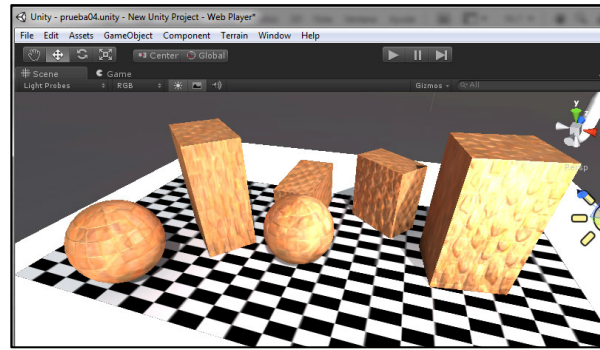


Fig. 7. 23– Material NormalMap.

El mapa de normales (ver apartado 4.1) aporta falsa sensación de tridimensionalidad superficial ya que corrige la orientación de cada punto de la superficie, de cara a calcular su iluminación.

En las dos imágenes se aprecian objetos con el mismo mapa de normales iluminados desde dos puntos diferentes. El mapa aporta información para generar sombras coherentes a la posición de la luz.

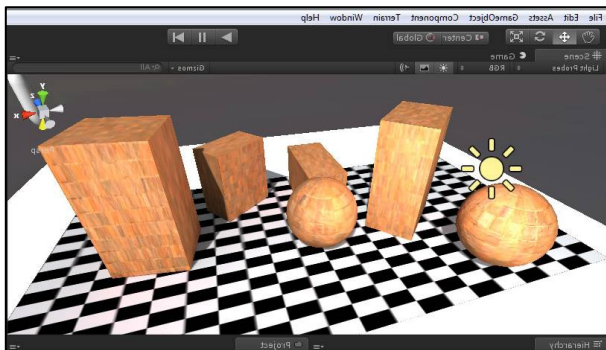
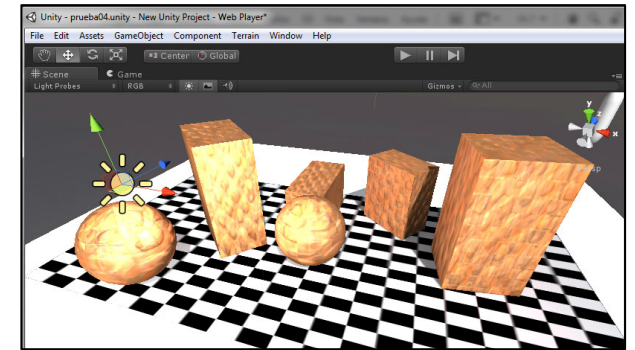


Fig. 7. 24 – Material parallax.

Añade un canal en escala de grises para indicar la profundidad de los puntos de la superficie.

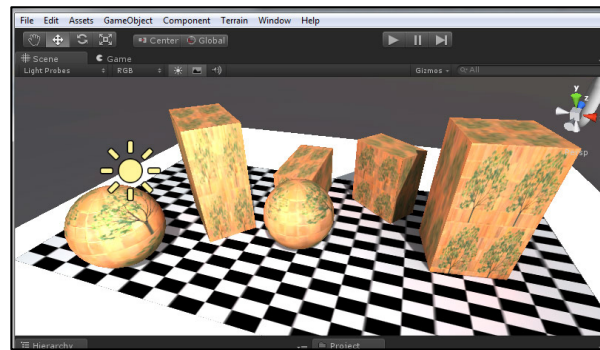


Fig. 7. 25 – Material Decal.

Ha permitido sobreimpresionar dos texturas, una al fondo continua y otra delantera con canal alfa para indicar zonas de transparencia.

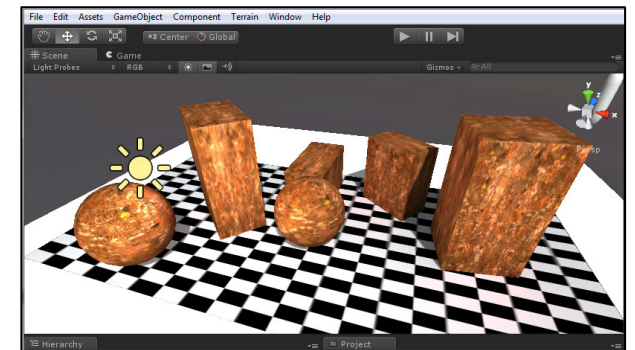


Fig. 7. 26 – Material diffuse detail.

Ha permitido fusionar dos texturas.

En la imagen se han seleccionado texturas contrastadas aunque su uso habitual es mezclar una textura con buena apariencia lejana con otra textura que aporte granulado o irregularidades en las distancias cortas, añadiendo detalle a la primera.

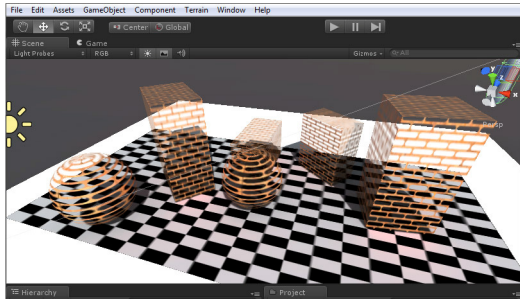


Fig. 7. 27 – Material transparent.

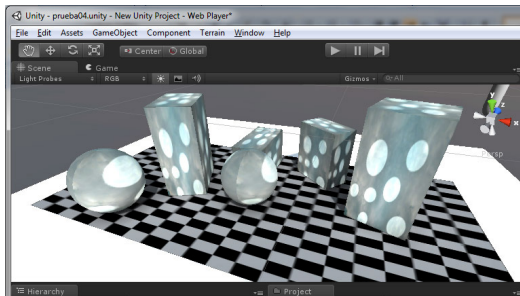


Fig. 7. 28 – Material self-illuminated.

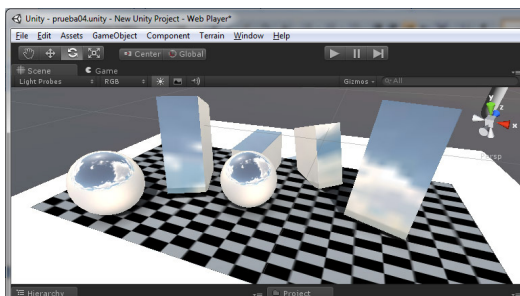


Fig. 7. 29 – Material reflective.

**Decal:** Este material permite superponer a una textura de fondo una segunda imagen. La segunda imagen debe contener un canal alfa que indique las zonas de transparencia.

En visualización arquitectónica, este material permite superponer carteles, graffitis, etc. sobre objetos texturados.

**Diffuse detail:** Es una versión ampliada del shader diffuse. Permite mezclar dos imágenes en una misma textura. Con ello, la primera imagen puede aportar la textura base y una segunda imagen aporta las imperfecciones o detalles de la superficie cuando el observador se acerca.

### Transparent shader family

Permite crear materiales con zonas total o parcialmente transparentes mediante canales alfa incluidos en la textura base. Dentro de esta familia se encuentran los shaders diffuse, specular, normal map y parallax explicados anteriormente, con las propiedades adicionales de transparencia.

Los materiales transparentes son muy útiles en visualización arquitectónica para crear elementos con decalados, como puedan ser enrejados, barrotes o lamas, sin tener que modelar físicamente cada pieza por separado, lo cual tendría mayor coste de modelado y de procesado durante la simulación.

### Transparent Cut Out shader family

Permite crear materiales con zonas totalmente transparentes mediante canales alfa incluidos en la textura base. Dentro de esta familia se encuentran los shaders diffuse, specular y normal map explicados anteriormente, con las propiedades adicionales de transparencia.

### Self-illuminated shader family

Este shader permite definir, mediante una imagen adicional con canal alfa, que partes del material tienen autoiluminación, es decir, están iluminadas aunque el entorno se encuentre oscuro. De esta manera se puede simular elementos luminosos en el material, o incluso objetos que brillan completamente en ambientes oscuros. Dentro de esta familia se encuentran los shaders diffuse, specular y normal map explicados anteriormente, con las propiedades adicionales de autoiluminación.

Sin embargo, hay que aclarar que, aunque el material se muestra más luminoso que lo que le correspondería según la iluminación de su entorno, en ningún caso emite luz hacia el exterior. Es decir, no permite crear objetos emisores de luz.

Para la visualización arquitectónica, este shader, en principio, no tiene especial interés de aplicación.

### Reflective shader family

Este shader permite vincular un Cube Map (ver apartado 4.3) que definirá las reflexiones del objeto. Lo habitual es utilizar como CubeMap de reflexiones, el mismo skybox que se utiliza como fondo de entorno (cielo) de manera que los objetos parezcan reflectantes.

En visualización arquitectónica este material tiene uso en todo tipo de materiales muy cromados, e incluso en acristalamientos.

Dentro de esta familia se encuentran los shaders diffuse, specular, normal map y parallax explicados anteriormente, con las propiedades adicionales de reflexión.

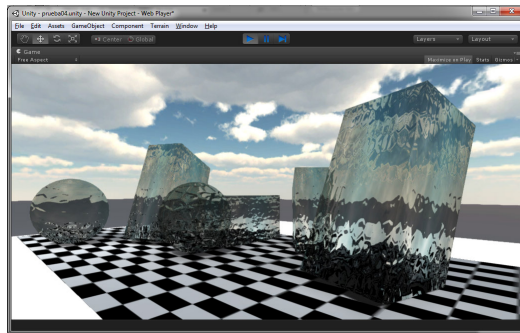


Fig. 7. 30 – Material glass refraction.

### Glass refraction shader

Este material permite simular la refracción del vidrio, así como efectos de vidrio esmerilado o moldeado.

En visualización arquitectónica puede tener mucho interés a la hora de enfatizar la presencia de vidrios en interiores, teniendo en cuenta que el reflective shader usualmente trabaja con cubemaps exteriores con lo que su uso en interiores no es apropiado, y considerando que un simple efecto de transparencia no suele quedar realista en la simulación de vidrios.

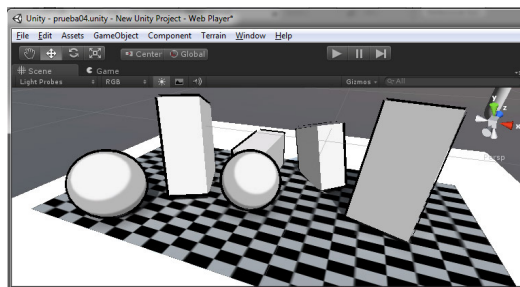


Fig. 7. 31 – Material Toon.

### Toon shader

Este shader permite dar una apariencia de dibujos animados a los objetos.

No tiene mayor interés en representación arquitectónica, ya que no permite transparencias y relieves por lo que su uso se limitaría a casos anecdóticos. Parece estar más enfocado a la creación de juegos.

### Otros materiales

Unity3D permite programar shaders para personalizar materiales. La comunidad de usuarios comparte en <http://wiki.unity3d.com/> sus scripts, de manera que cualquiera que no sepa programar puede aprovecharlos. Algunos como los materiales espejo, agua o terreno pueden ser de aplicación en la visualización arquitectónica.

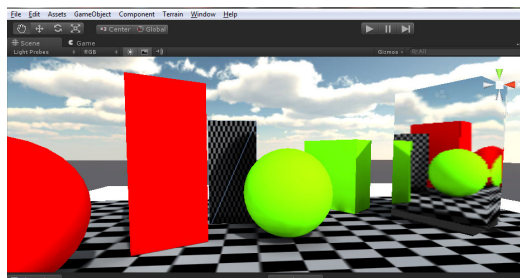


Fig. 7. 32 – Material mirror (a la derecha) reflejando otros objetos en la escena.

## 7.5 Entorno (terreno y vegetación)

Cuando se recrea virtualmente un entorno arquitectónico (o urbanístico) tridimensional es interesante disponer de herramientas para crear espacios exteriores naturales, más allá de los que puedan ser importados desde otros programas.

Unity3D incorpora un motor de creación de terrenos bastante intuitivo y rápido (mucho más que modelar terrenos desde AutoCAD, por ejemplo), si bien no tiene gran precisión para simular entornos topográficamente exactos (que se deberían importar).

Inicialmente, al crear un terreno, se crea una superficie plana de unas dimensiones configurables (por defecto 2000x2000 metros). Al seleccionar éste, una serie de herramientas permiten su manipulación para crear relieves, añadir texturas superficiales o crear distintos tipos de vegetación.

### Modelado

El relieve se crea “pintando” con una serie de pinceles cuya forma, tamaño y dureza se puede definir. Al “pintar sobre la superficie, ésta se eleva o hunde a voluntad. También se puede definir una altura específica a alcanzar.

Diferentes tamaños, tipos y durezas de pincel permiten generar distintos tipos de rugosidad, desde colinas hasta terrenos abruptos.

### Importación de información planimétrica

El programa, además permite una opción muy interesante en el ámbito de la arquitectura: La importación de mapas de altura (en formato RAW 16 bits) basados en escalas de grises, para definir el relieve de los terrenos. Estos llamados “heightmaps” permiten introducir información de altura de terrenos reales desde imágenes.

### Texturizado

Una vez se ha conseguido modelar el relieve, el programa permite texturizar las superficies seleccionando imágenes 2D como patrón y pintando sobre las superficies. Se puede regular la opacidad de la pintura sobre capas inferiores, de manera que el resultado final sea una textura compuesta (por ejemplo, se puede mezclar tierra con zonas verdes en distintas proporciones).

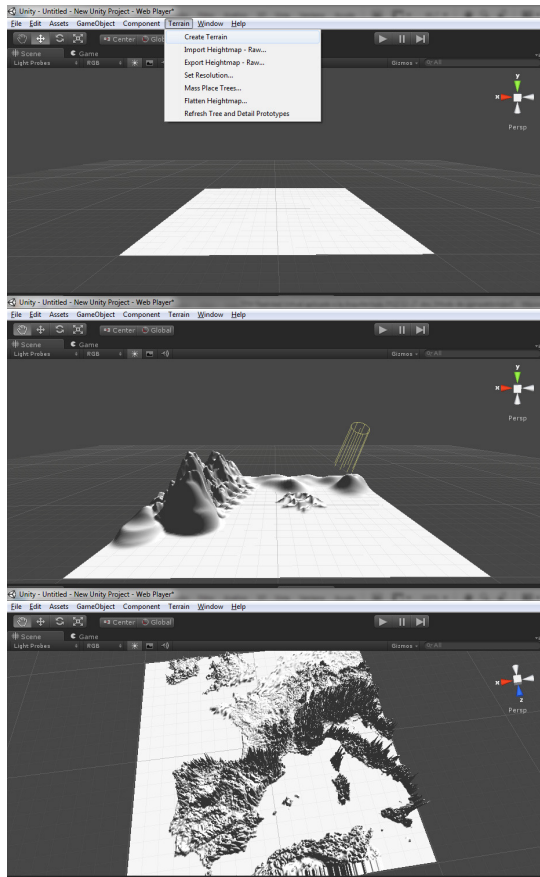


Fig. 7. 33 – Capturas de pantalla de distintas etapas de modelizado de terreno:

- 1º Inserción del plano de trabajo (arriba).
- 2º Creación de relieves mediante pinceles (centro).
- 3º Inserción de un mapa de relieve en escala de grises para definir alturas. En este caso se ha usado un mapa planimétrico de Europa (abajo).

### Creación de arbolado

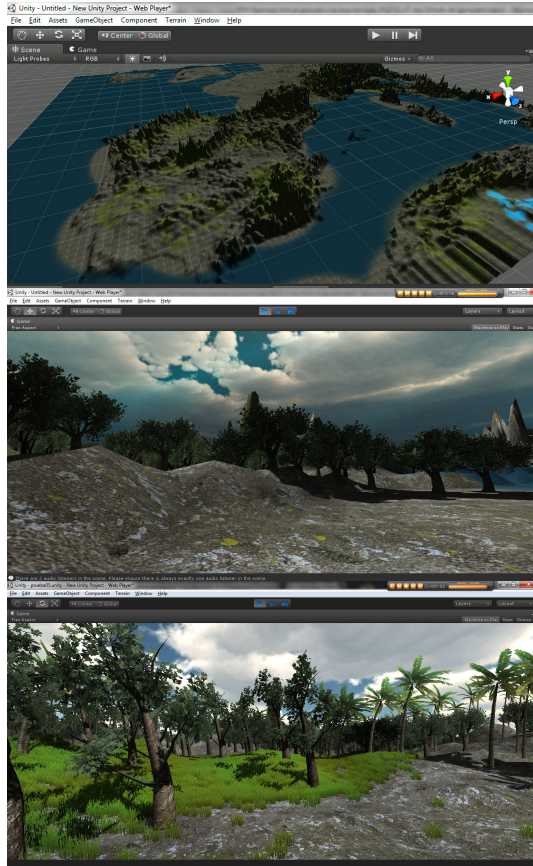


Fig. 7. 34 – Capturas de pantalla de distintas etapas de la creación del terreno:

- 1º Texturizado de las superficies (arriba).
- 2º Inserción de árboles prediseñados (centro).
- 3º Creación de distinto arbolado paramétrico con "tree creator" y adición de césped tridimensional animado (abajo).

Unity3D incorpora un generador de árboles que permite dotar de realismo a las escenas de naturaleza incluyendo cientos o miles de árboles de una forma rápida. Esto es una ventaja ya que muchos programas de modelado usados por los arquitectos no incorporan este tipo de ayudas por lo que cada árbol ha de ser colocado individualmente en el entorno, lo cual se convierte en una tarea tediosa.

Para optimizar el renderizado en tiempo real, el programa utiliza árboles 3D en las distancias cercanas y los sustituye por imágenes 2D orientadas a la cámara en las distancias lejanas. De esta forma se ahorran cálculos y el espectador no lo aprecia.

Los conjuntos de árboles también se "pintan" mediante pinceles para lo cual, el primer paso, es seleccionar que modelo de árbol será clonado al "pintar". Los pinceles permiten definir la densidad de arbolado, la diferencia de altura entre distintos bloques del mismo árbol e incluso la diferencia de color. De esta manera se evita la habitual situación de mostrar exactamente el mismo árbol clonado cientos de veces, lo cual produce un efecto artificial y poco realista.

Los árboles se pueden insertar desde otros programas o crear desde la herramienta "tree creator" de Unity3D. Ésta es una herramienta procedural que permite parametrizar tipos de tronco, rama y hojas, con sus formas y colores para crear distintas variedades de árboles y plantas.

Otro elemento que trabaja junto con los terrenos y que afecta a los árboles son los objetos "wind force" que crean fuerzas de viento que mecen las ramas. Se pueden definir wind forces que afecten a toda la escena o solo a un área dada. Igualmente, se pueden definir otros parámetros como la fuerza del viento, la velocidad o la turbulencia para conseguir distintos efectos.

### Césped y elementos de detalle

Esta herramienta permite crear césped tridimensional. Consiste en una serie de elementos bidimensionales verticales a los que se les puede adherir una textura con mapa de opacidad.

El método de creación e inserción es similar al del arbolado, y el césped también puede ser animado para simular la acción del viento.

Igualmente se puede seleccionar cualquier pequeño objeto tridimensional (piedras o troncos, por ejemplo) para ser dispersados por la superficie, incrementando la complejidad y realismo de la escena.

## 7.7 Audio

Unity3D permite integrar audio dentro de la simulación. La gran ventaja respecto a una animación prediseñada a la que se le ha añadido audio en una edición posterior, es que éste se encuentra perfectamente integrado en el entorno tridimensional consiguiendo sensación espacial auditiva.

El primer paso consiste en cargar un archivo de audio. El programa soporta archivos wav, mp3 ogg o aiff con salida mono, estéreo, o con hasta ocho canales.

Los componentes de un sistema de sonido en Unity3D son las fuentes de sonido, los receptores de sonido, los filtros de sonido y las áreas de reverberación.

Las fuentes de sonido se insertan en el espacio 3D y emiten como sonido el archivo de audio que tengan enlazado.

Al cargar el archivo de sonido, éste puede ser definido como sonido 3D. En ese caso, cuando se inserte la fuente de sonido en la escena, el programa tiene en cuenta la posición de ésta para simular la sensación auditiva del personaje controlando el volumen, el panorama y el efecto Doppler en caso de movimiento entre el emisor y el receptor. Existen varios controles para ajustar y afinar la sensación 3D.

Si el clip de audio no está marcado como 3D el sonido se oirá de fondo en todo el espacio tridimensional, como una banda sonora.

El receptor de sonido es el micrófono virtual que capta los sonidos del entorno y los presenta al usuario (es sus oídos en el entorno virtual). Lógicamente, solo puede haber uno en el escenario y lo normal es que vaya incrustado al person controller.

Por otro lado existen una serie de filtros de sonido (paso alto, paso bajo, eco, distorsión, reverb y chorus) que pueden ser aplicados a las distintas fuentes de sonido o al receptor de sonido (en cuyo caso afecta a todos los sonidos del entorno).

Por último, las áreas de reverberación son zonas esféricas en las que, al entrar el personaje, escucha los sonidos con reverberación. Su uso es principalmente para la entrada a lugares con eco.

En visualización arquitectónica, el sonido puede ser usado para generar sensación de ambiente, tanto con sonido (o música) de fondo, como incluyendo sonidos 3D a elementos de la escena, como puedan ser fuentes, vehículos o personajes. Con un nivel más avanzado de programación se podrían crear personajes con los que interactuar (por ejemplo, guías de museo).



## 7.8 Sistemas de partículas

Los sistemas de partículas se utilizan en Unity3D (como en otros programas de renderizado) para recrear nieve, lluvia, humo, fuego, o cualquier elemento que esté compuesto por pequeñas partículas móviles.

El generador de partículas es un guizmo que se puede situar en cualquier punto del escenario virtual.

El sistema de control permite manipular paramétricamente gran cantidad de opciones para modificar la apariencia y comportamiento de las partículas, como la velocidad de generación, cantidad de partículas, color, gravedad, dirección y forma de difusión, así como variaciones de estas propiedades a lo largo de la vida útil de las mismas. Muchas de estas propiedades se pueden controlar mediante gráficas (lineales, curvas sinodales o logarítmicas...).

Una serie de módulos conectables a voluntad permiten realizar efectos avanzados, como la emisión de subpartículas por parte de las partículas iniciales (para simulación de fuegos artificiales, por ejemplo).

En la visualización arquitectónica, mediante los sistemas de partículas (acompañados de sonido) se puede simular lluvia o nieve. Por otro lado, por paralaje, estos efectos ayudan a acrecentar la sensación espacial del espectador.

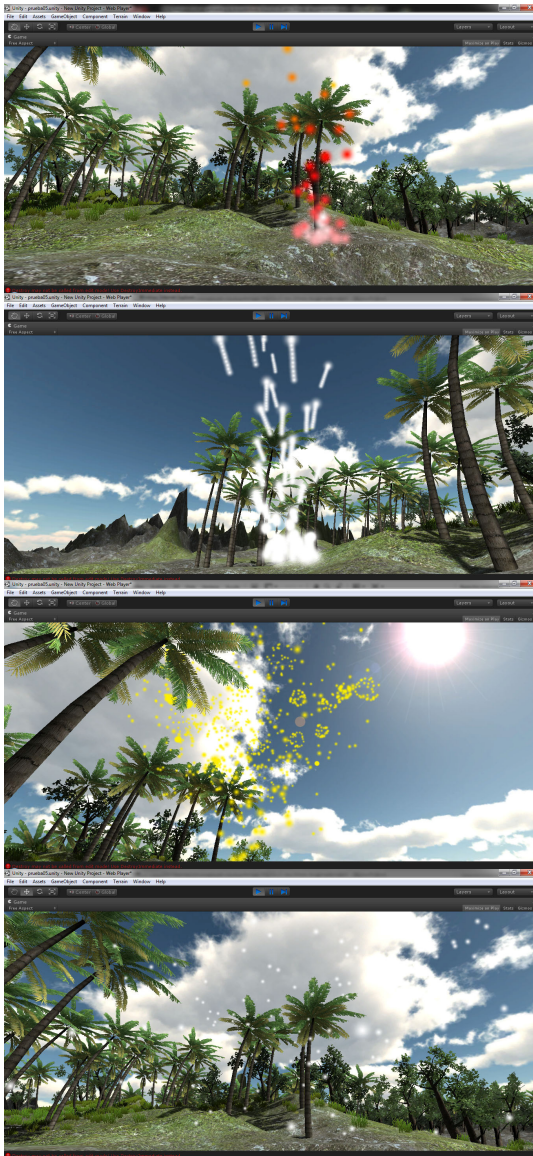


Fig. 7. 35 – Capturas de pantalla de sistemas de partículas con distintas combinaciones.

## 7.9 Efectos de imagen

Los efectos de imagen de Unity3D sustituyen el paso de postprocesado de imagen (ver apartado 3.2 “postprocesado” y 3.3 “retoque digital de la imagen”).

Los efectos de imagen se incrustan en la cámara activa, son el último elemento en renderizar y afectan a como se visualiza toda la imagen mostrada. Cada uno de ellos contiene distintos parámetros modificables para ajustar los efectos.

Entre los efectos más destacables en el campo de la visualización arquitectónica están:

**Antialiasing:** difumina la representación de los bordes de los objetos eliminando el efecto “dientes de sierra” creado por contraste entre píxeles. Este efecto debería estar siempre incluido pues supone una mejora visual importante en cualquier ámbito de visualización.

**Bloom and flare effects:** Crea aureolas alrededor de los objetos luminosos. Además, si la cámara tiene activada la función HDR (ver apartado 4.2 “iluminación”) el control de luminosidades se adapta a cómo ve el ojo humano, expandiendo el rango de oscuridad a luz de la escena por encima de 0 a 1 y ajustando el nivel de exposición a lo que se está viendo actualmente.

**Color correction curves:** Permite ajustar las curvas de rojo, verde y azul (RGB) de salida de la imagen, permitiendo corregir o manipular el tono y color final de la escena.

**Depth of field:** La profundidad de campo, el efecto de tener zonas enfocadas y desenfocadas a distintas distancias, como se explicó en el capítulo 4.3, es propio del ojo humano ya que las imágenes de síntesis no necesitan lentes (que son las que crean este efecto) al crear la imagen. Unity3D permite simular efectos de profundidad de campo, y definir sus propiedades, para hacer la experiencia más cercana a la realidad tal como la vemos los seres humanos.

**Screen Space Ambient Occlusion effect:** Crea efecto de oclusión ambiental en tiempo real, mejorando la apariencia de interrelación entre objetos de la escena. Si no resiente la fluidez de juego, debería estar siempre activado en visualización arquitectónica, como el antialiasing, para aumentar el realismo.



Fig. 7. 36 – Capturas de pantalla de efectos de imagen.

1º Bloom and flare effects (arriba) crea aureolas alrededor de las nubes y demás elementos luminosos.

2º Color correction curves (centro) permite modificar el tono de la imagen final. En este caso se ha buscado un tono verdoso.

3º Depth of field (abajo) simula el desenfoco de objetos lejanos.

### 7.10 Interactividad – animaciones

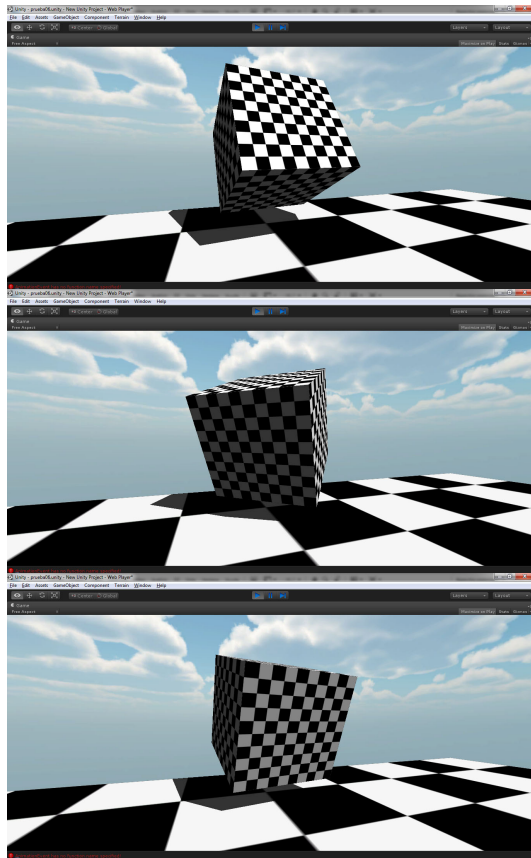


Fig. 7. 37 – Capturas de pantalla de un cubo animado rotando en el espacio.

Unity3D permite crear e importar animaciones para obtener elementos móviles dentro de la escena.

Aunque en visualización arquitectónica la mayor parte de los objetos pueden ser estáticos (a excepción del punto de vista), ciertos elementos, como puertas que se abren y cierran, luces que se encienden y apagan (o se mueven o cambian de intensidad y color), ascensores que funcionan, vehículos que circulan, e incluso personajes que pasean, pueden contribuir a aumentar el realismo de la escena.

Las animaciones pueden ser creadas con software específico e importadas en Unity3D en formato FBX como un modelo normal. En este caso, el gestor de animaciones de Unity3D permite “partir” el modelo en secciones de fotogramas con las distintas etapas de la animación. Por ejemplo, al importar la animación de una puerta abriéndose y cerrándose, se podrían “rescatar” las animaciones e la puerta abriéndose, entreabriéndose, cerrándose o entrecerrándose, dependiendo de los fotogramas seleccionados.

Por otro lado, el programa también permite programar sus propias animaciones, lo cual queda fuera del ámbito de estudio de este capítulo.

Por último, existe un visor de animaciones que permite crear modificaciones animadas en ciertas propiedades de los objetos (posición escala, giro, material, etc...) a lo largo de una línea temporal mediante curvas.

### 7.11 Interactividad – físicas

El motor de cálculo de fuerzas físicas de Unity3D permite incorporar comportamiento dinámico a los objetos del entorno.

En primer lugar, hay que indicar a los objetos que detecten colisiones. Si se activa esta opción los objetos se tendrán en cuenta entre sí y no se traspasarán mutuamente. Existe la posibilidad de adherir colisionadores “colliders” a objetos que no los tengan por defecto. Estos “colliders” son volúmenes de contacto para calcular colisiones.

En segundo lugar, se puede añadir un “rigid body” a cualquier objeto. A partir de ese momento al objeto se le asigna una masa y una inercia lineal y angular y es atraído por la gravedad.

En tercer lugar, se puede crear un “physic material” que define la fricción estática y dinámica y el coeficiente de rebote de los objetos a los que es añadido.

En cuarto lugar, a los objetos se les puede aplicar tanto fuerzas direccionales como momentos de giro.

En quinto lugar se pueden definir uniones entre objetos. Estas uniones pueden ser fijas, tipo bisagra, tipo muelle y personalizada, definiendo ejes de giro permitidos, etc.

En sexto lugar, existe la posibilidad de crear objetos blandos (skinned cloth) o telas con comportamiento dinámico (interactive cloth).

Por último, a un nivel avanzado se pueden programar fuerzas, acciones o comportamientos no existentes en el catálogo por defecto.

Con todos estos elementos, a nivel de visualización arquitectónica se puede montar un sistema físico en el que es posible empujar mobiliario o puertas, deslizarse por rampas o tener unas cortinas movidas por el viento.

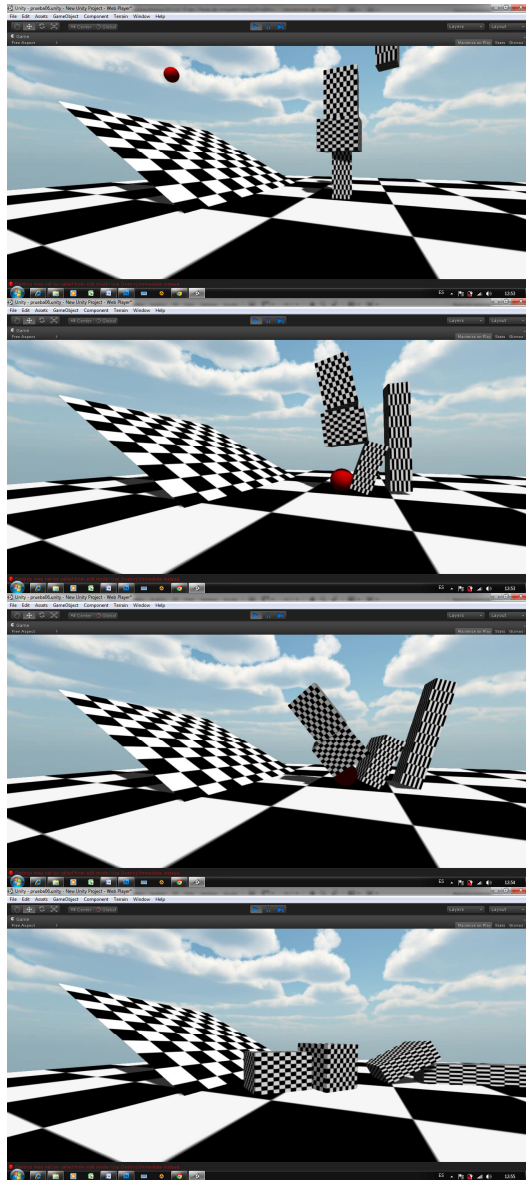


Fig. 7. 38 – Capturas de pantalla de colisiones entre objetos con propiedades físicas.

### **7.11 Interfaz gráfica de usuario (GUI)**

La interfaz gráfica de usuario (Graphical User Interface) está compuesta por los elementos 2D superpuestos en pantalla para dar información al usuario. Esta información puede consistir en texto o imagen.

Esta información, que en un videojuego puede incorporar las vidas o munición disponible, en visualización arquitectónica puede tener una gran diversidad de usos:

- Superponer un plano de planta indicando la posición del usuario.
- Superponer información automática sobre la estancia en la que se encuentra el usuario. Por ejemplo, nombre y características principales en un edificio público de interés arquitectónico.
- Superponer información de determinados elementos a petición del usuario. Por ejemplo, información sobre obras de arte en un museo virtual.

La interfaz permite, igualmente, definir los distintos controles que se pueden ejercer sobre la escena, como por ejemplo encender y apagar luces, cambiar materiales de acabado, etc...

## **8. Conclusiones y aspectos finales.**

### **8.1 Conclusiones**

El objetivo de este trabajo consistía en dar una visión de las tecnologías informáticas de representación gráfica utilizadas (o utilizables) por el arquitecto, y dar un paso más allá; investigar las bondades de los motores de creación de videojuegos para conseguir acercar el mundo de la Realidad Virtual a la visualización arquitectónica interactiva y en tiempo real a un bajo coste.

En este último apartado se analizan los resultados obtenidos durante la elaboración del trabajo.

#### **Sobre la informática aplicada a la arquitectura**

1º En su momento el CAD supuso un gran salto en productividad respecto al delineado tradicional. Se minimizó el tiempo y el personal dedicado a dibujar y sin embargo, se incrementó la calidad del dibujo y se redujeron los errores.

Parece ser que el BIM supondrá otro salto en productividad similar a aquel. Tanto en reducción de tiempos y personal (todas las vistas de un proyecto se autocorrijen al modificar una de ellas, las mediciones, acotaciones, cuadros de superficies y planos de carpinterías son cada vez más automáticas) como en incremento de la calidad y reducción de errores (todos los documentos están coordinados al estar referidos a una base de datos común, se pueden definir fases y tiempos de construcción a los distintos elementos dibujados, el modelo arquitectónico se coordina con el modelo estructural y con las instalaciones para evitar incoherencias o solapes).

2º El salto del delineado tradicional al CAD obligó al delineante a reciclarse para ser competitivo. El salto al BIM tarde o temprano forzará al proyectista a aprender una nueva manera de trabajar.

De igual manera, otro software (cálculo de estructuras e instalaciones, presupuestos...) aumenta la productividad del proyectista enormemente en los casos habituales. Por ello, éste, además de formarse en conocimientos técnicos, cada vez ha de ser más hábil en el manejo de los programas informáticos.

3º El aumento de posibilidades, complejidad y parametrización que permiten las herramientas informáticas y la tecnología han abierto el campo de la arquitectura hacia diseños más sostenibles (posibilidad de hacer estudios solares, lumínicos, de ventilación, termodinámicos) o formalmente

complejos (Gehry, Zahid, Libeskind) desembocando en un nuevo tipo de arquitectura muchas veces basado en formas gratuitas y en el “más difícil todavía”.

4º Actualmente, existe una rama del negocio arquitectónico basada en el modelado tridimensional y renderizado fotorrealista de imágenes de edificios.

El uso del BIM (que autogenera el modelo tridimensional), la irrupción de los game engines (que permiten visualización interactiva) y la proliferación de dispositivos móviles (laptops, smartphones, tablets) cada vez más ligeros y potentes es muy probable que nos dirija a una nueva manera de visualizar los entornos arquitectónicos.

### **Sobre la Realidad Virtual, Realidad Aumentada y Game Engines**

5º La RV y la RA implican interactividad y funcionamiento en tiempo real. Ello supone un mínimo de potencia de los equipos. Cuanta mayor sea la potencia, mayor será el número de polígonos, texturas, luces y efectos procesados, aumentando el realismo. Las tendencias en hardware auguran incrementos de potencia cada vez mayores, así que en los próximos años la RV y RA pueden ser cada vez más comunes.

Igualmente, los sistemas de control han dado un paso de gigante desde el clásico joystick a mandos inalámbricos (Wiimote, wii balance board) e incluso control gestual sin mandos (Kinect, Leap motion). Es de prever que toda esta tecnología tenga un uso común en el futuro más allá del ámbito del videojuego.

6º La arquitectura es una disciplina muy adecuada para la visualización interactiva y en tiempo real.

De hecho, de entre todas las tecnologías disponibles, incluso las más sencillas son aptas para sentir la presencia en entornos virtuales. Con un buen sistema de visualización (incluso no siendo estereoscópico), audio 3D muy sencillo de implementar y controles rudimentarios (joystick, teclado o ratón) se puede navegar cómodamente. Por ello es muy fácil de implementar en equipos no especializados.

Sistemas de trackeo avanzados, entornos CAVE, dispositivos hápticos, etc. más costosos, complejos y escasos, podrían ser aprovechados para ciertos usos, pero en ningún caso son una necesidad.

7º La visualización arquitectónica puede tener, a su vez, múltiples objetivos:

- Docencia de asignaturas de historia del arte o de la arquitectura por medio del acceso a edificios lejanos, destruidos o nunca construidos.
- Maquetas virtuales en las que testear los diseños.
- Mejora de las habilidades espaciales y geométricas.
- Difusión del conocimiento y del Patrimonio Arquitectónico, Cultural, Artístico o Histórico por medio de visitas virtuales a edificios emblemáticos o históricos, museos, cueva, etc.
- Diversas ayudas en la rehabilitación de edificios (visualización de diversos estados o intervenciones, “copia de seguridad” del edificio original...”
- Aplicaciones de fomento o de apoyo al turismo.
- Ayuda al proceso de diseño de edificios y entornos urbanos: Toma de decisiones por parte del propietario, promotor, proyectista, autoridades. Trabajo colaborativo y comunicación entre las partes, opinión del ciudadano.
- Ayuda a la evaluación de riesgos laborales, accesibilidad o situaciones de emergencia o incendio.
- Ayuda al proceso de construcción (modelos de replanteos, acabados, estructura o instalaciones sobreimpresionados mediante Realidad Aumentada).
- Ayuda al mantenimiento del edificio o al aprendizaje y entrenamiento en el uso de instalaciones.
- Ayuda a la orientación de personas en entornos arquitectónicos (vía GPS)

8º Los Game Engines son una opción económica de implementar entornos de Realidad Virtual interactivos y ya se están usando con éxito.

#### **Sobre la elaboración de un entorno virtual mediante un Game Engine**

9º Con conocimientos de arquitectura, modelado 3D y renderizado se pueden llegar a realizar entornos virtuales efectivo mediante Game Engines.

Para aplicaciones con elementos interactivos o de una cierta complejidad existe la barrera de la programación en javascript o C#. Para crear visualizaciones complejas se requiere la colaboración de arquitecto, artista 3D y programador.



10° En cuanto al modelado para visualización arquitectónica en tiempo real, parece necesario simplificar al máximo la geometría y obtener el realismo y detalle mediante texturizado e iluminación, si se quiere conseguir fluidez de movimientos.

Los objetos deberían tener activadas las opciones de backface culling para reducir el número de polígonos en escena. El uso de cube mapping o skyboxes es muy apropiado para conseguir realismo en el entorno sin incrementar los cálculos de geometría.

11° En cuanto a materialidad, en visualización arquitectónica es indispensable contar con shaders y mapas difusos, de relieve y de opacidad (que exigen relativamente pocos recursos). Otros shader útiles son los reflectantes (consumen más recursos) y simuladores de agua. La refracción, autoiluminación, cartoon y subsurface scattering apenas tienen aplicación en arquitectura.

12° En cuanto a iluminación, es básico contar con iluminación directa e indirecta si se quieren resultados realistas. El cálculo de la iluminación indirecta en tiempo real no es factible hoy en día en la mayoría de los casos. El uso de lightmaps y pases de Ambient Occlusion prerrenderizados son los métodos habituales para conseguir iluminación compleja realista.

La iluminación basada en imagen, que consigue grados de realismo excepcionales, no está implementada por defecto, probablemente por el costo de su cálculo. Quizá en el futuro se implemente.

Las causticas no están implementadas por defecto, consumirían muchos recursos no son necesarias en la mayoría de las ocasiones en visualización arquitectónica.

13° Ciertos efectos como antialiasing, destellos de lente o profundidad de campo son muy útiles a la hora de aportar realismo con relativo bajo consumo de recursos.

14° En Unity 3D no es factible importar directamente desde AutoCAD. Las opciones son pasar previamente por 3ds Max o SketchUp para crear los archivos MAX o FBX (respectivamente) que reconoce Unity3D.

Se puede importar directamente desde Revit pero se pierden las texturas. Las opciones son pasar previamente por 3ds Max (misma versión que Revit) o reasignar los materiales desde Unity3D.

15° La creación de terrenos naturales con arbolado y césped, la creación de sistemas de partículas, la inserción de efectos sonoros de fondo o ubicados en el espacio tridimensional y la creación de efectos de imagen (correcciones de tono, saturación, contraste, etc.) y físicas son relativamente fáciles de realizar sin conocimientos previos.

16° La inclusión de animaciones se puede realizar de tres maneras: Importando la animación desde el software específico, programándola en Unity3D o mediante el visor de animaciones (para sencillas transformaciones de posición, giro, escala o cambio de material). Para realizar animaciones correctamente en necesario, al menos, tener algunos conocimientos de programación.

## **8.2 Interés formativo del trabajo realizado**

Se ha estudiado una tecnología gráfica que probablemente en los próximos años se haga habitual en el campo de la arquitectura y edificación, y que presenta una serie de bondades indudables que se han ido explicando a lo largo de los capítulos anteriores.

## **8.3 Beneficios para la sociedad**

La frontera entre la informática y el resto de disciplinas técnicas se está desvaneciendo forzosamente debido a las virtudes de la simbiosis resultante. En un área como la arquitectura que combina lo gráfico con la técnica, la física, o la gestión, la optimización de recursos y el abanico de nuevas posibilidades que abre el uso de la informática gráfica.

En la conclusión número 7° se citan una serie de usos de la Realidad Virtual y Realidad Aumentada en el ámbito de la arquitectura, haciendo referencia a los apartados 5.8 a 5.13.

De igual manera, en el capítulo 6 se describen una serie de casos de éxito de la aplicación de los Game Engines en la elaboración de visualizaciones arquitectónicas y herramientas de aprendizaje o entrenamiento de distintas habilidades “juegos serios”.

## 8.4 Futuros trabajos

Este trabajo ha supuesto una breve introducción a la informática gráfica aplicada a la arquitectura, a los conceptos, técnicas y métodos habituales y al Estado del Arte en cuanto a tecnologías y usos para la visualización arquitectónica interactiva en tiempo real.

Por otro lado, ha supuesto una breve experiencia en el uso de un motor de creación de videojuegos , por parte de un arquitecto sin conocimientos en programación, y con el objetivo de averiguar los puntos fuertes y débiles de este tipo de software a la hora de realizar sencillas visualizaciones interactivas de bajo coste.

Se abren a partir de ahora múltiples campos de estudio:

- Evaluar el límite de realismo alcanzable desde un punto de vista arquitectónico o como mejorar la experiencia visual del usuario.
- Evaluar las posibilidades de interactividad o complejidad a las que se puede llegar con conocimientos profundos de programación, y la utilidad de estas posibilidades.
- Evaluar las distintas plataformas de funcionamiento de la aplicación (web player, ordenador, tablet, smartphone, videoconsola...).
- Estudiar la aplicación de esta tecnología en cualquiera de los campos descritos en la conclusión 7º:
  - Aplicación en docencia.
  - Aplicación en cultura, conocimiento y turismo
  - Aplicación en arquitectura, construcción, uso y mantenimiento del edificio.

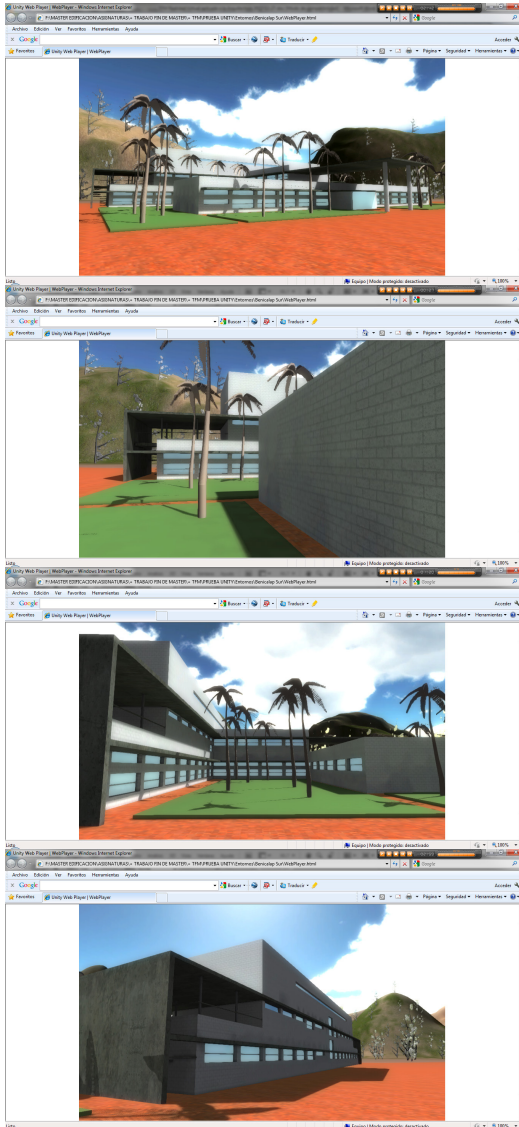


Fig. 8. 1 – Capturas de pantalla de la visualización desde Internet Explorer del Centro de Salud Benicalap Sur (Valencia) proyectado por el autor.

**REFERENCIAS**

- Bishop, I, 2011, "Landscape planning is not a game: Should it be?", Landscape and Urban Planning nº100 pags 390–392
- Blauert, J. 1983 "Spatial hearing: the psychophysics of human sound localization" MIT Press, Cambridge, Massachusetts.
- Bourdot, P. et al. 2010 "VR\_CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models" Computer-Aided Design nº42, pags 445-461.
- Bouschlaghem, D. et al 2005 "Visualisation in architecture, engineering and construction (AEC)", Automation in Construction nº14, pags 287-295
- Burdea, C. 1996 "Force and touch feedback for virtual reality", John Wiley & Sons, Inc., New York, NY
- Carlson, W. 2003 A Critical History of Computer Graphics and Animation. The Ohio State University
- Cazaro, E. 2011 "Maqueta o modelo digital, la pervivencia de un sistema" REVISTA EGA expresión gráfica arquitectónica nº17
- Contero, M, Alcañiz, M, Salvador, G, et al. 2012 "HumanTop: a multi-object tracking tabletop" Multimedia Tools and Applications.
- Contero, M. et al 2010 "Design and validation of an augmented book for spatial abilities development in engineering students" Computers & Graphics nº34, pas 77-91.
- Debevec, P. 2002 "A Tutorial on Image-Based Lighting " IEEE Computer Graphics and Applications.
- Escartín, E. Junio 2000 "la realidad virtual, una tecnología educativa a nuestro alcance" Pixel-bit nº 15
- Feiner, S. et al. 1995 "Architectural anatomy" In Presence, 4(3), pags. 318-325
- Fernando de Fuentes, A, et al. 2010 "Model of sources: a proposal for the hierarchy, merging strategy and representation of the information sources in virtual models of historical buildings" Computer Applications and Quantitative Methods in Archeology - CAA'2010
- Franco, J.A. 2011 "Sobre perspectiva, fotografía e infografía" REVISTA EGA expresión gráfica arquitectónica nº17.
- Frost, P. Warren, P. 2000 "Virtual reality used in a collaborative architectural design process", Information Visualization. Proceedings. IEEE International Conference on Practical, Interactive Inst., Malmo" Univ. Coll., Sweden.

- Ghadirian, P. Bishop, D. 2008 "Integration of augmented reality and GIS: A new approach to realistic landscape visualisation", *Landscape and Urban Planning*, nº86, pags. 226-232
- Hernandez, L, et alt. 2011 "la percepción del espacio en la visualización de arquitectura mediante realidad virtual inmersiva"" *REVISTA EGA expresión gráfica arquitectónica* nº18.
- Hobsbawm, E. 1962 "The Age of Revolution", (London 1962; repr. 2008), p.45.
- Hongling, G. et al 2012 "Using game technologies to improve the safety of construction plant operations" *Accident Analysis and Prevention* nº48, pags 204– 213
- Ifrac, G. 2001, *The Universal History of Computing: From the Abacus to the Quantum Computer*, New York: John Wiley & Sons.
- Juang, J.R, Hung, W.H, Kang, S.C, 2011, "Using game engines for physical – based simulations – a forklift, Special Issue Use of Gaming Technology in Architecture", *Engineering and Construction, ITcon* 16 pags 3-22
- Kagan, A. 1980 "Ben F. Laposky: A Midwestern Pioneer of Absolute Light Form", *Arts Magazine*
- Kensek, K. et al. 2000 "Augmented Reality: an application for architecture", *Proc. 8th International Conference on Computing in Civil and Building Engineering, ASCE, Stanford, CA*, pags. 294-301
- Kosmadoudi, Z, Lim, T, Ritchie, J, Louchart, S, Liu, Y, Sung, R, 2012 "Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements". *Computer-Aided Design*
- Krull, F.N. 1994 "The origin of computer graphics within General Motors". *IEEE Annals of the History of Computing* 16
- Larry, G. 2011 "Computerized Contact Lenses Could Enable In-Eye Augmented Reality" *Scientific American*.
- Lertlakkhanakul, J. et al, 2005 "Using the Mobile Augmented Reality Techniques for Construction Management", *Proc. of the 10th International Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2005)*, pags 396-403
- Li, H. et al 2012 "Visualizing safety assessment by integrating the use of game technology" *Automation in Construction* nº22, pag 498–505
- Liarokapis, F et al. 2007 "Exploring Urban Environments Using Virtual and Augmented Reality, *Journal of Virtual Reality and Broadcasting*", nº3, pags. 1-13

- Lonsing, W. 2004 "Augmented Reality as Tool in Architecture", Proc. of the 22nd eCAADe Conference: Architecture in the Network Society, pags 495-499
- Malkawi, A. et al. 2004 "Interactive, Immersive Visualization for Indoor Environments: Use of Augmented Reality, Human-Computer Interaction and Building Simulation", Proc. of the Eighth International Conference on Information Visualization, pags. 833-838
- Maor, E. 2009. e: The Story of a Number. Princeton University Press. p. 16
- Marguin, J. 1994, Histoire des instruments et machines à calculer, trois siècles de mécanique pensante 1642-1942. Hermann.
- Miliano, V. 1999, "Unreality: Application of a 3D Game Engine to Enhance the Design, Visualisation and Presentation of Commercial Real Estate", Proceedings of the 5th International Conference on Virtual Systems and Multimedia in Dundee, Scotland.
- Moloney, J, Amor, R, Zealand, N, 2003, "StringCVE: Advancecs in a game engine based collaborative virtual environments for architecture design, In: Proceedings of CONVR 2003 Conference on Construction Applications of Virtual Reality, pags 156–168.
- Otxotorena, J. 2011 "Arquitecturas y "blandografías"" REVISTA EGA expresión gráfica arquitectónica nº17.
- Paar, P. 2006 "Landscape visualizations: Applications and requirements of 3D visualization software for environmental planning", Computers, Environment and Urban Systems nº30, pags 815–839
- Pitt, Ian. 2010 "Auditory Perception". <http://www.cs.ucc.ie/~ianp/CS2511/HAP.html>
- Puyuelo, M, et al. 2011 "Representaciones virtuales y otros recursos técnicos en la accesibilidad al patrimonio cultural" REVISTA EGA expresión gráfica arquitectónica nº17.
- Reika, S, Weimin, G, 2011, "The Use of 3d Design Tools With Game-Like Functionality and Physiological Information in Urban Planning and Architectural Design", Procedia Environmental Sciences nº10, pags 170 – 177
- Rua, H, Alvito, P. 2011, "Living the past: 3D models, virtual reality and game engines as tools for supporting archaeology and the reconstruction of cultural heritage e the case-study of the Roman villa of Casal de Freiria", *Journal of Archaeological Science*, nº38.
- Rüppel, U. Schatz, K 2011 "Designing a BIM-based serious game for fire safety evacuation simulations" *Advanced Engineering Informatics* nº25, pags 600–611

- Sagun, A. et al. 2011 "Computer simulations vs. building guidance to enhance evacuation performance of buildings during emergency events" *Simulation Modelling Practice and Theory* nº19, pags 1007–1019
- Shaoliang, Q. 2012 "Virtual Interior Design Based On VRML AND JAVA" *Physics Procedia* nº33, pags 1614-1620
- Shen, S. et al. 2001 "Urban Planning Using Augmented Reality", *Journal of Urban Planning and Development*, pagsp. 118-125
- Shih, N.-J. Lin, C.-Y. Yang, C.-H. 2000 "A virtual-reality-based feasibility study of evacuation time compared to the traditional calculation method", *Fire Safety Journal* 34, pags 377– 391
- Smith, S. Trenholme, D. 2009 "Rapid prototyping a virtual fire drill environment using computer game technology" *Fire Safety Journal* nº44, pags 559–569
- Styliadis, A. 2007, "Digital documentation of historical buildings with 3-d modeling functionality" *Automation in Construction* nº16 pags 498–510
- Sunesson, K. et al. 2008 "Virtual Reality As a New Tool in the City Planning Process" *Tsinghua Science And Technology* Vol13 nºS1, octubre, pp255-260
- Sutherland, I. 1963 "Sketchpad: A man-machine graphical communication system". Massachusetts Institute of Technology. Tesis Doctoral. Reedición Universidad de Cambridge 2003.
- Sutherland, I. 1968 "A head-mounted three-dimensional display". *Proceedings of AFIPS* 68, pp. 757-764
- Toharia, P 2010 "Realidad Virtual e interacción" *Temario del Master en Informática Gráfica, Juegos y Realidad Virtual*, Universidad Rey Juan Carlos.
- Webster, A. et al. 1996 "Augmented reality in architectural construction, inspection and renovation", *Proc. ASCE Third Congress on Computing in Civil Engineering*, Anaheim, CA, pags. 913-919
- Westerdahl, B. et al. 2006 "Users' evaluation of a virtual reality architectural model compared with the experience of the completed building" *Automation in construction* nº15, pags 150-165.
- Yan, W, Culp, C, Graf, R, 2011, "Integrating BIM and gaming for real-time interactive architectural visualization", *Automation in Construction* nº20, pags 446–458
- Yu, DG. Et al. 2010 "A Useful Visualization Technique: A Literature Review for Augmented Reality and its Application, limitation & future direction", *visual Information Communication* Pages: 311-337

**ÍNDICE DE IMÁGENES**

Figura	Página	Fuente
Fig 1.1	p 6	<a href="http://commons.wikimedia.org/wiki/File:CampidoglioEng.jpg">http://commons.wikimedia.org/wiki/File:CampidoglioEng.jpg</a>
Fig 1.2	p 6	Elaboración propia
Fig 1.3	p 7	<a href="http://unity3d.com/">http://unity3d.com/</a>
Fig 1.4	p 8	Rua, H, Alvito, P. 2011
Fig 2.1	p 10	<a href="http://commons.wikimedia.org/wiki/File:Boulier1.JPG">http://commons.wikimedia.org/wiki/File:Boulier1.JPG</a>
Fig 2.2	p 10	<a href="http://es.wikipedia.org/wiki/Archivo:Arts_et_Metiers_Pascaline_dsc03869.jpg">http://es.wikipedia.org/wiki/Archivo:Arts_et_Metiers_Pascaline_dsc03869.jpg</a>
Fig 2.3	p 11	<a href="http://en.wikipedia.org/wiki/Jacquard_loom">http://en.wikipedia.org/wiki/Jacquard_loom</a>
Fig 2.4	p 11	<a href="http://en.wikipedia.org/wiki/File:Classic_shot_of_the_ENIAC.jpg">http://en.wikipedia.org/wiki/File:Classic_shot_of_the_ENIAC.jpg</a>
Fig 2.5	p 12	<a href="http://es.wikipedia.org/wiki/Archivo:FortranCardPROJ039.agr.jpg">http://es.wikipedia.org/wiki/Archivo:FortranCardPROJ039.agr.jpg</a>
Fig 2.6	p 12	<a href="http://es.wikipedia.org/wiki/Computadora_Whirlwind">http://es.wikipedia.org/wiki/Computadora_Whirlwind</a>
Fig 2.7	p 13	<a href="http://design.osu.edu/carlson/history/images/sage_f1693.jpg">http://design.osu.edu/carlson/history/images/sage_f1693.jpg</a>
Fig 2.8	p 13	<a href="http://dada.compart-bremen.de/node/2687">http://dada.compart-bremen.de/node/2687</a>
Fig 2.9	p 14	<a href="http://design.osu.edu/carlson/history/images/pages/ivan-sutherland_jpg.htm">http://design.osu.edu/carlson/history/images/pages/ivan-sutherland_jpg.htm</a>
Fig 2.10	p 14	<a href="http://en.wikipedia.org/wiki/File:Sketchpad-Apple.jpg">http://en.wikipedia.org/wiki/File:Sketchpad-Apple.jpg</a>
Fig 2.11	p 15	<a href="http://sophia.javeriana.edu.co/~ochavarr/computer_graphics_history/historia.htm">http://sophia.javeriana.edu.co/~ochavarr/computer_graphics_history/historia.htm</a>
Fig 2.12	p 15	<a href="http://riskwiki.bishopphillips.com/index.php?title=Real_Learning_in_Virtual_Worlds_-_CHAPTER_2:_Literature_Review.htm">http://riskwiki.bishopphillips.com/index.php?title=Real_Learning_in_Virtual_Worlds - CHAPTER 2: Literature Review.htm</a>
Fig 2.13	p 15	<a href="http://cinepoeme.blogspot.com.es/2011/01/computing-photographs.html">http://cinepoeme.blogspot.com.es/2011/01/computing-photographs.html</a>
Fig 2.14	p 16	<a href="http://www.graphics.cornell.edu/online/box/compare.html">http://www.graphics.cornell.edu/online/box/compare.html</a>
Fig 2.15	p 16	<a href="http://area.autodesk.com/forum/autodesk-maya/dynamics/terminator-t-1000-effect/">http://area.autodesk.com/forum/autodesk-maya/dynamics/terminator-t-1000-effect/</a>
Fig 2.16	p 17	<a href="http://bibwp.ulpgc.es/mediatecaulpgc/page/2/">http://bibwp.ulpgc.es/mediatecaulpgc/page/2/</a>



Fig 2.17	p 17	<a href="http://theillusveonesreviews.blogspot.com.es/2010/11/jason-and-argonauts.html">http://theillusveonesreviews.blogspot.com.es/2010/11/jason-and-argonauts.html</a>
Fig 2.18	p 17	<a href="http://www.behance.net/onursenturk/frame/4003411">http://www.behance.net/onursenturk/frame/4003411</a>
Fig 2.19	p 18	<a href="http://www.cinemaniablog.com/tag/toy+story">http://www.cinemaniablog.com/tag/toy+story</a>
Fig 2.20	p 18	<a href="http://royalmediapro.com/3d.html">http://royalmediapro.com/3d.html</a>
Fig 3.1	p 20	Elaboración propia
Fig 3.2	p 21	<a href="http://www.urbanity.es/foro/edificios-en-general-mad/10018-madrid-ampliacion-del-banco-de-espana-por-moneo.html">http://www.urbanity.es/foro/edificios-en-general-mad/10018-madrid-ampliacion-del-banco-de-espana-por-moneo.html</a>
Fig 3.3	p 22	Elaboración propia
Fig 3.4	p 23	Elaboración propia
Fig 3.5	p 24	Elaboración propia
Fig 3.6	p 25	Elaboración propia
Fig 3.7	p 26	Elaboración propia
Fig 3.8	p 27	Elaboración propia
Fig 3.9	p 27	<a href="http://www.3dtotal.com/index_tutorial_detailed.php?id=1227#.UOh2Im-ZSSo">http://www.3dtotal.com/index_tutorial_detailed.php?id=1227#.UOh2Im-ZSSo</a>
Fig 3.10	p 28	Elaboración propia
Fig 3.11	p 29	Elaboración propia
Fig 3.12	p 29	Elaboración propia
Fig 3.13	p 30	<a href="http://vimeo.com/7809605">http://vimeo.com/7809605</a>
Fig 3.14	p 32	Elaboración propia
Fig 3.15	p 32	Elaboración propia
Fig 3.16	p 33	<a href="http://www.twinmotion.com/twinmotion2/specifications/realistic-rendering.html/?_store=xtwlsb&amp;_from_store=vefmsyx">http://www.twinmotion.com/twinmotion2/specifications/realistic-rendering.html/?_store=xtwlsb&amp;_from_store=vefmsyx</a>
Fig 3.17	p 34	Elaboración propia y Ramón Llacer (escalera)
Fig 3.18	p 35	Elaboración propia
Fig 3.19	p 36	Elaboración propia

Fig 3.20	p 36	Elaboración propia
Fig 3.21	p 37	Elaboración propia
Fig 3.22	p 38	Elaboración propia
Fig 3.23	p 39	Elaboración propia
Fig 3.24	p 39	Elaboración propia
Fig 3.25	p 40	Elaboración propia
Fig 3.26	p 40	Elaboración propia
Fig 3.27	p 41	Elaboración propia
Fig 3.28	p 41	Elaboración propia

Fig 4.1	p 42	Elaboración propia
Fig 4.2	p 43	Elaboración propia
Fig 4.3	p 44	Elaboración propia
Fig 4.4	p 45	Elaboración propia
Fig 4.5	p 45	Elaboración propia
Fig 4.6	p 46	Elaboración propia
Fig 4.7	p 47	Elaboración propia
Fig 4.8	p 47	Elaboración propia
Fig 4.9	p 48	Elaboración propia
Fig 4.10	p 48	Elaboración propia
Fig 4.11	p 49	Elaboración propia
Fig 4.12	p 49	Elaboración propia
Fig 4.13	p 49	Elaboración propia
Fig 4.14	p 50	<a href="http://spaceflight.nasa.gov/gallery/images/apollo/apollo15/html/as15-86-11603.html">http://spaceflight.nasa.gov/gallery/images/apollo/apollo15/html/as15-86-11603.html</a>

Fig 4.15	p 50	Elaboración propia
Fig 4.16	p 51	Elaboración propia
Fig 4.17	p 52	Elaboración propia
Fig 4.18	p 52	Elaboración propia
Fig 4.19	p 52	Elaboración propia
Fig 4.20	p 53	Elaboración propia
Fig 4.21	p 53	Elaboración propia
Fig 4.22	p 54	Elaboración propia
Fig 4.23	p 54	Elaboración propia
Fig 4.24	p 56	Elaboración propia
Fig 4.25	p 56	Elaboración propia
Fig 4.26	p 57	Elaboración propia
Fig 4.27	p 58	Elaboración propia
Fig 4.28	p 58	Elaboración propia
Fig 4.29	p 59	Elaboración propia
Fig 4.30	p 59	Elaboración propia
Fig 5.1	p 61	<a href="http://sweb.cityu.edu.hk/sm2203/oldclass2007/lab1/toc02/index.html">http://sweb.cityu.edu.hk/sm2203/oldclass2007/lab1/toc02/index.html</a>
Fig 5.2	p 62	<a href="http://www.imdb.com/media/rm2602929664/tt0104692">http://www.imdb.com/media/rm2602929664/tt0104692</a>
Fig 5.3	p 63	<a href="http://www.3dconnexion.es/products/spaceexplorer.html">http://www.3dconnexion.es/products/spaceexplorer.html</a>
Fig 5.4	p 63	<a href="http://en.wikipedia.org/wiki/File:ARL_ODT.jpg">http://en.wikipedia.org/wiki/File:ARL_ODT.jpg</a>
Fig 5.5	p 64	<a href="http://www.bienetec.es/bienetec/jsp/web/catalogo/productos/productos_familias/index.jsp?P_FAMILIA=FFH">http://www.bienetec.es/bienetec/jsp/web/catalogo/productos/productos_familias/index.jsp?P_FAMILIA=FFH</a>
Fig 5.6	p 64	<a href="http://www.youtube.com/watch?v=C4d109Re_0">http://www.youtube.com/watch?v=C4d109Re_0</a>
Fig 5.7	p 65	<a href="http://lamiradaencendida.wordpress.com/2012/01/12/las-aventuras-de-tintin-el-secreto-del-unicornio/">http://lamiradaencendida.wordpress.com/2012/01/12/las-aventuras-de-tintin-el-secreto-del-unicornio/</a>

Fig 5.8	p 65	<a href="http://www.xatakaciencia.com/tecnologia/leap-motion-usando-el-ordenador-como-tom-cruise-en-minority-report">http://www.xatakaciencia.com/tecnologia/leap-motion-usando-el-ordenador-como-tom-cruise-en-minority-report</a>
Fig 5.9	p 67	<a href="https://www.youtube.com/watch?v=Jd3-eiid-Uw">https://www.youtube.com/watch?v=Jd3-eiid-Uw</a>
Fig 5.10	p 68	<a href="http://en.wikipedia.org/wiki/File:Stereograph_as_an_educator_-_anaglyph.jpg">http://en.wikipedia.org/wiki/File:Stereograph_as_an_educator_-_anaglyph.jpg</a>
Fig 5.11	p 68	<a href="http://ookaboo.com/o/pictures/picture/12523518/Resembling_sunglasses_polarized_glasses">http://ookaboo.com/o/pictures/picture/12523518/Resembling_sunglasses_polarized_glasses</a>
Fig 5.12	p 69	<a href="http://es.wikipedia.org/wiki/Autoestereoscopia">http://es.wikipedia.org/wiki/Autoestereoscopia</a>
Fig 5.13	p 69	<a href="http://www.nintendo.es/Nintendo-3DS/Nintendo-3DS-94560.html">http://www.nintendo.es/Nintendo-3DS/Nintendo-3DS-94560.html</a>
Fig 5.14	p 69	<a href="http://en.wikipedia.org/wiki/Autostereogram">http://en.wikipedia.org/wiki/Autostereogram</a>
Fig 5.15	p 74	<a href="http://research.vancouver.wsu.edu/dr-hakan-gurocak/haptic-interfaces">http://research.vancouver.wsu.edu/dr-hakan-gurocak/haptic-interfaces</a>
Fig 5.16	p 74	<a href="http://research.vancouver.wsu.edu/dr-hakan-gurocak/haptic-interfaces">http://research.vancouver.wsu.edu/dr-hakan-gurocak/haptic-interfaces</a>
Fig 5.17	p 75	<a href="http://gizmodo.com/5861749/heads+up-display-contact-lenses-are-one-step-closer-after-passing-safety-tests">http://gizmodo.com/5861749/heads+up-display-contact-lenses-are-one-step-closer-after-passing-safety-tests</a>
Fig 5.18	p 75	<a href="http://en.wikipedia.org/wiki/Augmented_reality">http://en.wikipedia.org/wiki/Augmented_reality</a>
Fig 5.19	p 76	<a href="http://www.fundabiomed.fcs.uc.edu.ve/project.html">http://www.fundabiomed.fcs.uc.edu.ve/project.html</a>
Fig 5.20	p 76	<a href="http://www.sensable.com/ViewHapticApp.htm?5">http://www.sensable.com/ViewHapticApp.htm?5</a>
Fig 5.21	p 76	<a href="http://shishka4u.wordpress.com/category/human-factors/augmented-reality/">http://shishka4u.wordpress.com/category/human-factors/augmented-reality/</a>
Fig 5.22	p 77	<a href="http://cat2.mit.edu/projects1/biznes/">http://cat2.mit.edu/projects1/biznes/</a>
Fig 5.23	p 77	<a href="https://www.youtube.com/watch?v=L8ONVseGL2E">https://www.youtube.com/watch?v=L8ONVseGL2E</a>
Fig 5.24	p 78	Contero, M. et al 2010
Fig 5.25	p 79	<a href="http://virtualwaregroup.com/cueva-virtual-de-santimamine/">http://virtualwaregroup.com/cueva-virtual-de-santimamine/</a>
Fig 5.26	p 80	Westerdahl, B. et al. 2006
Fig 5.27	p 80	Bouschlaghem, D. et al 2005
Fig 5.28	p 81	Sunesson, K. et al. 2008
Fig 5.29	p 82	Smith, S. Trenholme, D. 2009
Fig 5.30	p 82	Hongling, G. et al 2012
Fig 5.31	p 83	<a href="http://www.tecmovia.com/2012/05/10/pioneer-acerca-la-realidad-aumentada-al-parabrisas-de-nuestro-vehiculo/">http://www.tecmovia.com/2012/05/10/pioneer-acerca-la-realidad-aumentada-al-parabrisas-de-nuestro-vehiculo/</a>

Fig 5.32	p 83	Webster, A. et al. 1996
Fig 5.33	p 84	Athanasios. D 2007
Fig 5.34	p 84	Fernando de Fuentes, A, et al. 2010
Fig 6.1	p 85	<a href="http://www.gtagaming.com/gtaiv/screenshots.php?page=23">http://www.gtagaming.com/gtaiv/screenshots.php?page=23</a>
Fig 6.2	p 85	<a href="http://www.unrealengine.com/zh/showcase/simulation_training/zero_hour/">http://www.unrealengine.com/zh/showcase/simulation_training/zero_hour/</a>
Fig 6.3	p 86	<a href="http://www.scenicreflections.com/download/412973/Doom_Screenshot_Wallpaper/">http://www.scenicreflections.com/download/412973/Doom_Screenshot_Wallpaper/</a>
Fig 6.4	p 86	<a href="http://www.real-visual.com/">http://www.real-visual.com/</a>
Fig 6.5	p 86	<a href="http://www.real-visual.com/">http://www.real-visual.com/</a>
Fig 6.6	p 87	<a href="http://vizerra.com/">http://vizerra.com/</a>
Fig 6.7	p 88	Reika, S, Weimin, G, 2011
Fig 6.8	p 89	<a href="http://www.mycryengine.com/">www.mycryengine.com/</a>
Fig 6.9	p 90	<a href="http://www.unrealengine.com/showcase/visualization/cowboys_stadium/">http://www.unrealengine.com/showcase/visualization/cowboys_stadium/</a>
Fig 6.10	p 90	<a href="http://www.unrealengine.com/showcase/visualization/architectural_visualization_1/">http://www.unrealengine.com/showcase/visualization/architectural_visualization_1/</a>
Fig 6.11	p 91	<a href="http://archvirtual.com/portfolio/absolute-moon-youtube/">http://archvirtual.com/portfolio/absolute-moon-youtube/</a>
Fig 7.1	p 92	Elaboración propia
Fig 7.2	p 93	Elaboración propia
Fig 7.3	p 93	Elaboración propia
Fig 7.4	p 94	Elaboración propia
Fig 7.5	p 95	Elaboración propia
Fig 7.6	p 95	Elaboración propia
Fig 7.7	p 96	Elaboración propia
Fig 7.8	p 97	Elaboración propia

Fig 7.9	p 98	Elaboración propia
Fig 7.10	p 99	Elaboración propia
Fig 7.11	p 100	Elaboración propia
Fig 7.12	p 101	Elaboración propia
Fig 7.13	p 102	Elaboración propia
Fig 7.14	p 102	Elaboración propia
Fig 7.15	p 102	Elaboración propia
Fig 7.16	p 102	Elaboración propia
Fig 7.18	p 103	Elaboración propia
Fig 7.19	p 103	Elaboración propia
Fig 7.20	p 103	Elaboración propia
Fig 7.21	p 104	Elaboración propia
Fig 7.22	p 105	Elaboración propia
Fig 7.23	p 105	Elaboración propia
Fig 7.24	p 105	Elaboración propia
Fig 7.25	p 105	Elaboración propia
Fig 7.26	p 105	Elaboración propia
Fig 7.27	p 106	Elaboración propia
Fig 7.28	p 106	Elaboración propia
Fig 7.29	p 106	Elaboración propia
Fig 7.30	p 107	Elaboración propia
Fig 7.31	p 107	Elaboración propia
Fig 7.32	p 107	Elaboración propia
Fig 7.33	p 108	Elaboración propia

Fig 7.34	p 109	Elaboración propia
Fig 7.35	p 111	Elaboración propia
Fig 7.36	p 112	Elaboración propia
Fig 7.37	p 113	Elaboración propia
Fig 7.38	p 114	Elaboración propia
Fig 8.1	p 121	Elaboración propia