

PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

UT4. Análisis de motores de juegos

Departamento de Informática y Comunicaciones

CFGS Desarrollo de Aplicaciones Multiplataforma

Segundo Curso

IES Virrey Morcillo

Villarrobledo (Albacete)

Índice

1. Introducción	3
2. Estructura típica de un equipo de desarrollo	3
3. Arquitectura de un juego	4
3.1. Géneros de juegos	5
3.2. Juegos 2D	6
3.3. Juegos 3D	6
3.4. Juegos 2,5D	6
4. Componentes de un juego	7
5. Motores de juegos	8
6. Análisis de diferentes motores de juegos.....	8

1. Introducción

La industria del videojuego ocupa el primer lugar en el ocio audio-visual e interactivo a nivel mundial, por encima de industrias tan potentes como el cine o la música. Como consecuencia directa, existe una gran demanda de profesionales cualificados para diseñar y desarrollar videojuegos no sólo para consolas y ordenadores, sino también para el más que creciente mercado de los dispositivos móviles.

Debido a los distintos avances en el campo de la informática, no sólo a nivel de desarrollo software y capacidad hardware sino también en la aplicación de métodos, técnicas y algoritmos, la industria del videojuego ha evolucionado hasta llegar a cotas inimaginables, tanto a nivel de jugabilidad como de calidad gráfica, tan sólo hace unos años.

A pesar de la vertiginosa evolución de la industria del videojuego, hoy en día existe un gran número de retos que el desarrollador de videojuegos ha de afrontar a la hora de producir un videojuego. En realidad, existen retos que perdurarán eternamente y que no están ligados a la propia evolución del hardware que permite la ejecución de los videojuegos. El más evidente de ellos es la necesidad imperiosa de ofrecer una **experiencia de entretenimiento** al usuario basada en la diversión, ya sea a través de nuevas formas de interacción, como por ejemplo la realidad aumentada o la tecnología de visualización 3D, a través de una mejora evidente en la calidad de los títulos, o mediante innovación en aspectos vinculados a la jugabilidad.

Actualmente la evolución de los videojuegos está estrechamente ligada a la evolución del **hardware** que permite la ejecución de los mismos. Esta evolución atiende, principalmente, a dos factores:

- i) la potencia de dicho hardware y
- ii) las capacidades interactivas del mismo.

En el primer caso, una mayor potencia hardware implica que el desarrollador disfrute de mayores posibilidades a la hora de, por ejemplo, mejorar la calidad gráfica de un título o de incrementar la IA (Inteligencia Artificial) de los enemigos. Este factor está vinculado al **multiprocesamiento**.

En el segundo caso, una mayor riqueza en términos de interactividad puede contribuir a que el usuario de videojuegos viva una experiencia más inmersiva, por ejemplo mediante realidad aumentada, o simplemente más natural, por ejemplo mediante la pantalla táctil de un smartphone.

2. Estructura típica de un equipo de desarrollo

El desarrollo de videojuegos comerciales es un proceso complejo debido a los distintos requisitos que ha de satisfacer y a la integración de distintas disciplinas que intervienen en dicho proceso.

Los **ingenieros** son los responsables de diseñar e implementar el software que permite la ejecución del juego, así como las herramientas que dan soporte a dicha ejecución. Normalmente, los ingenieros se suelen clasificar en dos grandes grupos:

- Los programadores del núcleo del juego, es decir, las personas responsables de desarrollar tanto el motor de juego como el juego propiamente dicho.
- Los programadores de herramientas, es decir, las personas responsables de desarrollar las herramientas que permiten que el resto del equipo de desarrollo pueda trabajar de manera eficiente.

En el mundo del desarrollo de videojuegos, es bastante probable encontrar **ingenieros senior** responsables de supervisar el desarrollo desde un punto de vista técnico, de manera independiente al diseño y generación de código.

Los **artistas** son los responsables de la creación de todo el contenido audio-visual del videojuego, como por ejemplo los escenarios, los personajes, las animaciones de dichos personajes, etc. Al igual que ocurre en el caso de los ingenieros, los artistas también se pueden especializar en diversas cuestiones, destacando las siguientes:

- Artistas de concepto, responsables de crear bocetos que permitan al resto del equipo hacerse una idea inicial del aspecto final del videojuego. Su trabajo resulta especialmente importante en las primeras fases de un proyecto.
- Modeladores, responsables de generar el contenido 3D del videojuego, como por ejemplo los escenarios o los propios personajes que forman parte del mismo.
- Artistas de texturizado, responsables de crear las texturas o imágenes bidimensionales que formarán parte del contenido visual del juego. Las texturas se aplican sobre la geometría de los modelos con el objetivo de dotarlos de mayor realismo.
- Artistas de iluminación, responsables de gestionar las fuentes de luz del videojuego, así como sus principales propiedades, tanto estáticas como dinámicas.
- Animadores, responsables de dotar de movimientos a los personajes y objetos dinámicos del videojuego. Un ejemplo típico de animación podría ser el movimiento de brazos de un determinado carácter.
- Actores de captura de movimiento, responsables de obtener datos de movimiento reales para que los animadores puedan integrarlos a la hora de animar los personajes.
- Diseñadores de sonido, responsables de integrar los efectos de sonido del videojuego.
- Otros actores, responsables de diversas tareas como por ejemplo los encargados de dotar de voz a los personajes.

Al igual que suele ocurrir con los ingenieros, existe el rol de **artista senior** cuyas responsabilidades también incluyen la supervisión de los numerosos aspectos vinculados al componente artístico.

Los **diseñadores** de juego son los responsables de diseñar el contenido del juego, destacando la evolución del mismo desde el principio hasta el final, la secuencia de capítulos, las reglas del juego, los objetivos principales y secundarios, etc. Evidentemente, todos los aspectos de diseño están estrechamente ligados al propio género del mismo.

Finalmente, en el desarrollo de videojuegos también están presentes roles vinculados a la **producción**, especialmente en estudios de mayor capacidad, asociados a la planificación del proyecto y a la gestión de recursos humanos. Así mismo, los responsables de **marketing**, de administración y de soporte juegan un papel relevante. También resulta importante resaltar la figura de **publicista** como entidad responsable del marketing y distribución del videojuego desarrollado por un determinado estudio.

3. Arquitectura de un juego

Dentro del mundo del entretenimiento electrónico, un **juego** normalmente se suele asociar a la evolución, entendida desde un punto de vista general, de uno o varios personajes principales o entidades que pretenden alcanzar una serie de objetivos en un mundo acotado, los cuales están controlados por el propio usuario. Así, entre estos elementos podemos encontrar desde superhéroes hasta coches de competición pasando por equipos completos de fútbol. El mundo en el que conviven dichos personajes suele estar compuesto, normalmente, por una serie de escenarios virtuales recreados en tres dimensiones y tiene asociado una serie de reglas que determinan la interacción con el mismo.

De este modo, existe una interacción explícita entre el jugador o usuario de videojuegos y el propio videojuego, el cual plantea una serie de retos al usuario con el objetivo final de garantizar la diversión y el entretenimiento. Además de ofrecer este componente emocional, los videojuegos también suelen tener un componente cognitivo asociado, obligando a los jugadores a aprender técnicas y a dominar el comportamiento del personaje que manejan para resolver los retos o puzzles que los videojuegos plantean.

Desde una perspectiva más formal, la mayoría de videojuegos suponen un ejemplo representativo de lo que se define como aplicaciones gráficas o renderizado en tiempo real.

Desde un punto de vista abstracto, una aplicación gráfica en tiempo real se basa en un bucle donde en cada iteración se realizan los siguientes pasos:

- El usuario visualiza una imagen renderizada por la aplicación en la pantalla o dispositivo de visualización.
- El usuario actúa en función de lo que haya visualizado, interactuando directamente con la aplicación, por ejemplo mediante un teclado.
- En función de la acción realizada por el usuario, la aplicación gráfica genera una salida u otra, es decir, existe una retroalimentación que afecta a la propia aplicación.

En el caso de los videojuegos, este ciclo de visualización, actuación y renderizado ha de ejecutarse con una frecuencia lo suficientemente elevada como para que el usuario se sienta inmerso en el videojuego, y no lo perciba simplemente como una sucesión de imágenes estáticas. En este contexto, el frame rate se define como el número de imágenes por segundo, comúnmente fps, que la aplicación gráfica es capaz de generar. A mayor frame rate, mayor sensación de realismo en el videojuego. Actualmente, una tasa de 30 fps se considera más que aceptable para la mayoría de juegos. No obstante, algunos juegos ofrecen tasas que doblan dicha medida.

En otras palabras, los juegos son aplicaciones interactivas que están marcadas por el tiempo, es decir, cada uno de los ciclos de ejecución tiene un deadline que ha de cumplirse para no perder realismo.

Aunque el componente gráfico representa gran parte de la complejidad computacional de los videojuegos, no es el único. En cada ciclo de ejecución, el videojuego ha de tener en cuenta la evolución del mundo en el que se desarrolla el mismo. Dicha evolución dependerá del estado de dicho mundo en un momento determinado y de cómo las distintas entidades dinámicas interactúan con él.

Por otra parte, un juego también está ligado al comportamiento del personaje principal y del resto de entidades que existen dentro del mundo virtual. Un videojuego ha de contemplar las distintas acciones realizadas por el mismo, considerando la posibilidad de decisiones impredecibles a priori y las consecuencias que podrían desencadenar.

3.1. Géneros de juegos

Los motores de juegos suelen estar, generalmente, ligados a un tipo o género particular de juegos. Por ejemplo, un motor de juegos diseñado con la idea de desarrollar juegos de conducción diferirá en gran parte con respecto a un motor orientado a juegos de acción en tercera persona.

Probablemente, el género de juegos más popular ha sido y es el de los **juegos en primera persona (FPS o First-person shooter)**, abreviado tradicionalmente como shooters, representado por juegos como Quake, Half-Life, Call of Duty o Gears of War, entre muchos otros. En este género, el usuario normalmente controla a un personaje con una vista en primera persona a lo largo de escenarios que tradicionalmente han sido interiores, como los típicos pasillos, pero que han ido evolucionando a escenarios exteriores de gran complejidad.

Otro de los géneros más relevantes son los denominados **juegos en tercera persona**, donde el usuario tiene el control de un personaje cuyas acciones se pueden apreciar por completo desde el punto de vista de la cámara virtual. Aunque existe un gran parecido entre este género y el de los FPS, los juegos en tercera persona hacen especial hincapié en la animación del personaje, destacando sus movimientos y habilidades, además de prestar mucha atención al detalle gráfico de la totalidad de su cuerpo. Ejemplos representativos de este género son Resident Evil, Metal Gear, Gears of War o Uncharted, entre otros.

Otro género importante está representado por los **juegos de lucha**, en los que, normalmente, dos jugadores compiten para ganar un determinado número de combates minando la vida o stamina del jugador contrario. Ejemplos representativos de juegos de lucha son Virtua Fighter, Street Fighter, Tekken, o Soul Calibur, entre otros. Actualmente, los juegos de lucha se desarrollan normalmente en escenarios tridimensionales donde los luchadores tienen una gran libertad de movimiento. Sin embargo, últimamente se han desarrollado diversos juegos en los que tanto el escenario como los

personajes son en 3D, pero donde el movimiento de los mismos está limitado a dos dimensiones, enfoque comúnmente conocido como juegos de lucha de scroll lateral.

Otro género representativo en el mundo de los videojuegos son los **juegos de conducción**, en el que el usuario controla a un vehículo que normalmente rivaliza con más adversarios virtuales o reales para llegar a la meta en primera posición. En este género se suele distinguir entre simuladores, como por ejemplo Gran Turismo, y arcade, como por ejemplo Ridge Racer o Wipe Out.

Otro género tradicional son los **juegos de estrategia**, normalmente clasificados en tiempo real o RTS (Real-Time Strategy)) y por turnos (turn-based strategy). Ejemplos representativos de este género son Warcraft, Command & Conquer, Comandos, Age of Empires o Starcraft, entre otros. Este tipo de juegos se caracterizan por mantener una cámara con una perspectiva isométrica, normalmente fija, de manera que el jugador tiene una visión más o menos completa del escenario, ya sea 2D o 3D. Así mismo, es bastante común encontrar un gran número de unidades virtuales desplegadas en el mapa, siendo responsabilidad del jugador su control, desplazamiento y acción.

Finalmente, en los últimos años ha aparecido un género de juegos cuya principal característica es la posibilidad de jugar con un gran número de jugadores reales al mismo tiempo, del orden de cientos o incluso miles de jugadores. Los juegos que se encuadran bajo este género se denominan comúnmente **juegos MMOG (Massively Multiplayer Online Game)**. El ejemplo más representativo de este género es el juego World of Warcraft. Debido a la necesidad de soportar un gran número de jugadores en línea, los desarrolladores de este tipo de juegos han de realizar un gran esfuerzo en la parte relativa al networking, ya que han de proporcionar un servicio de calidad sobre el que construir su modelo de negocio, el cual suele estar basado en suscripciones mensuales o anuales por parte de los usuarios.

3.2. Juegos 2D

Los juegos 2D utilizan gráficos planos, a los que se llama sprites, y no tienen una geometría tridimensional. Se trazan en la pantalla como imágenes planas y la cámara (cámara ortográfica) no tiene perspectiva.

Estos son algunos ejemplos de juegos 2D típicos:



3.3. Juegos 3D

Los juegos 3D normalmente hacen uso de la geometría tridimensional, con Materiales y Texturas renderizados sobre la superficie de GameObjects para hacerlos aparecer como ambientes, personajes y objetos sólidos que conforman el mundo de juego.

Los juegos 3D normalmente renderizan la Escena aplicando perspectiva, por lo que los objetos se ven más grandes en la pantalla a medida que se acercan a la cámara.



3.4. Juegos 2,5D

Algunos juegos 2D emplean geometría 3D para el ambiente y los personajes, pero restringen la mecánica del juego a 2 dimensiones, por ej., la cámara puede mostrar una vista de desplazamiento

lateral, pero el jugador se mueve únicamente en 2 dimensiones. Para este tipo de juegos, el efecto 3D tiene un propósito más visual que funcional.

Hay juegos que emplean geometría 3D, pero utilizan una cámara ortográfica en lugar de una de perspectiva. Es una técnica común que brinda al jugador la vista aérea de la acción del juego, y a la que con frecuencia se hace referencia como 2,5D.



4. Componentes de un juego

Los componentes más importantes que definen a los motores de juegos actualmente podrían ser las siguientes:

✓ Gráficos.

Los motores de juegos modernos pueden dar vida a gráficos impresionantes y ayudar a facilitar la producción e importación de assets de diferentes plataformas, de manera que no tienes que esperar mucho tiempo cada vez que realizas una importación.

Para la mayoría de los juegos, los motores de juegos pueden proporcionarte una arquitectura con capacidades generales de renderización de alto rendimiento y con acceso a gráficos API rápidos, de manera que puedes lograr la máxima fidelidad visual para tus juegos.

✓ Sonido.

Puedes optar por diseñar tu propio sonido desde cero, empezando por la composición musical, los efectos de sonido y la actuación de voz y terminar con la postproducción. También puedes optar por comprar assets de sonido de plataformas enfocadas en assets, tales como Tienda de Activos de Unity, que pueden combinarse y mezclarse de varias maneras.

Los motores de juegos pueden ayudarte a integrar estos sonidos en forma nativa o a través de aplicaciones de terceros, de manera que puedas enfocarte en la composición de sonidos y la colocación de sonidos en forma creativa en el contexto de las escenas del juego, así como asignar sonidos a eventos de una manera que encaje con tu visión.

✓ Networking.

A muchos jugadores les gusta conectarse con sus amigos sobre los juegos y las aplicaciones, ya sea a través de una computadora, una consola o un dispositivo móvil. Para un tipo específico de juego, la interacción en red requiere servidores de mucha potencia, lo cual puede dar lugar a muchos flujos de trabajo y procesos complejos. Por suerte, muchos motores de juegos cuentan con flujos de trabajo, componentes controlados mediante scripts y API implementados que puedes optimizar de la manera que desees para los juegos en línea o multijugador.

✓ Física.

El sistema de Física de un motor de juegos te proporciona los componentes que manejan la simulación física por ti. No tienes que hacer todo el código pesado tú mismo, ni codificar cada movimiento creado en la escena por los elementos del juego o las colisiones entre los componentes de tu juego. Los motores de juegos te permiten crear objetos que pueden comportarse de una manera realista tan solo con unas cuantas líneas de código.

✓ Interfaces gráficas de usuario (GUI).

Muchos juegos cuentan con su propia GUI que encaja con el tema del juego, la dirección del arte y la narrativa. Esta es la razón por la cual los motores de juegos ofrecen a los creadores herramientas para que construyan sus propias interfaces gráficas de usuario, como botones y menús desplegables, barras deslizantes y medios para combinar elementos de interacción diferentes.

✓ Scripting.

Una parte crucial de los motores de juegos está representada por los scripts que puedes utilizar para definir la lógica de los componentes de tu juego, agregando comportamientos. Con los scripts prefabricados, una solución que proporcionan algunos motores de juegos, puedes controlar fácilmente la cámara, la relación entre los elementos de tu juego y el sistema de animación que puede utilizarse para controlar la animación de los personajes.

5. Motores de juegos

El término motor de juego surgió a mediados de los años 90 con la aparición del famosísimo juego de acción en primera persona Doom, desarrollado por la compañía id Software bajo la dirección de John Carmack. Doom fue diseñado con una arquitectura orientada a la reutilización mediante una separación adecuada en distintos módulos de los componentes fundamentales, como por ejemplo el sistema de renderizado gráfico, el sistema de detección de colisiones o el sistema de audio, y los elementos más artísticos, como por ejemplo los escenarios virtuales o las reglas que gobernaban al propio juego.

Este planteamiento facilitaba enormemente la reutilización de software y el concepto de motor de juego se hizo más popular a medida que otros desarrolladores comenzaron a utilizar diversos módulos o juegos previamente licenciados para generar los suyos propios. En otras palabras, era posible diseñar un juego del mismo tipo sin apenas modificar el núcleo o motor del juego, sino que el esfuerzo se podía dirigir directamente a la parte artística y a las reglas del mismo.

A día de hoy, una gran parte de compañías de desarrollo de videojuego utilizan motores o herramientas pertenecientes a terceras partes, debido a que les resulta más rentable económicamente y obtienen, generalmente, resultados espectaculares.

Obviamente, la separación entre motor de juego y juego nunca es total y, por una circunstancia u otra, siempre existen dependencias directas que no permiten la reusabilidad completa del motor para crear otro juego.

Un motor de juegos es una estructura para los desarrolladores de juegos que reúne varias áreas básicas. Puedes importar arte y assets o recursos, en 2D y 3D, de otros tipos de software, tales como Maya o 3ds Max o Photoshop; ensamblar estos assets para formar escenas y entornos; agregar iluminación, audio, efectos especiales, física y animación, interacción y lógica de mecánica del juego; y editar, depurar y optimizar el contenido para tus plataformas de destino.

6. Análisis de diferentes motores de juegos

No es posible desarrollar un videojuego sin un motor gráfico. Es por tanto una elección que cualquier programador de videojuegos ha tenido que hacer cada vez que se haya planteado un proyecto así.

Dentro del mercado existen varias opciones, algunas más potentes y otras más flexibles. La elección depende en gran medida del tipo de desarrollo y también de las necesidades de los profesionales que se vayan a encargar de su conceptualización y diseño.

Dentro de todas estas opciones, hay varias que son totalmente gratuitas, con un gran rendimiento para desarrolladores independientes que quieren hacerse un hueco dentro de este sector. Básicamente son tres:

- Unity.
- Cryengine.
- Unreal Development Kit (UDK).

Además, existen otras alternativas de pago que han dado un gran salto en los videojuegos de última generación:

- Unreal Engine.

- MT Framework.
- Samaritan.
- Titanium.
- Rockstar Advanced Game Engine (RAGE).
- Frostbite.
- Havok.
- Anvil Engine.
- 4A Engine.