

Motores de Juegos

Un **videojuego** es una aplicación gráfica en tiempo real en la que existe una interacción explícita entre el usuario y el propio videojuego. La expansión del videojuego es tan relevante que actualmente se trata de una industria multimillonaria capaz de rivalizar con las industrias cinematográfica y musical.

El desarrollo de videojuegos se ha beneficiado con la aparición de herramientas que facilitan dicho desarrollo, automatizando determinadas tareas y **ocultando** la complejidad inherente a muchos procesos de bajo nivel.

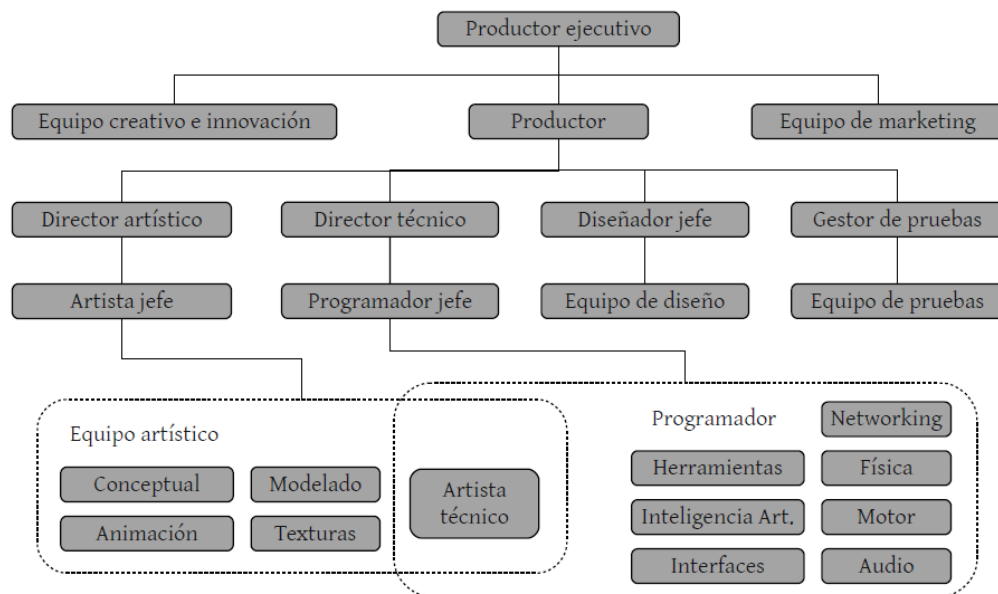


Figura 1.2: Visión conceptual de un equipo de desarrollo de videojuegos, considerando especialmente la parte de programación.

El **término motor de juego** surgió a mediados de los años 90 con la aparición del juego de acción **Doom**, que fue diseñado con una arquitectura orientada a la reutilización mediante la separación en distintos componentes de cada una de sus partes: el sistema de renderizado gráfico, el sistema de detección de colisiones, el sistema de audio, elementos más artísticos ...



¿Qué es un motor de juegos?

Podríamos decir que son un conjunto de herramientas que facilitan las tareas de creación de un juego. Un motor se encarga de asuntos como la gestión de las **colisiones** y las **físicas** del juego, la correcta representación de menús e interfaz, la reproducción de sonidos, o el manejo de la inteligencia artificial (I.A.) de los enemigos, etc.... entre otras cosas. Todas estas herramientas, junto con las rutinas para los gráficos (el **motor gráfico**) conformarán lo que conocemos por un motor de juego.

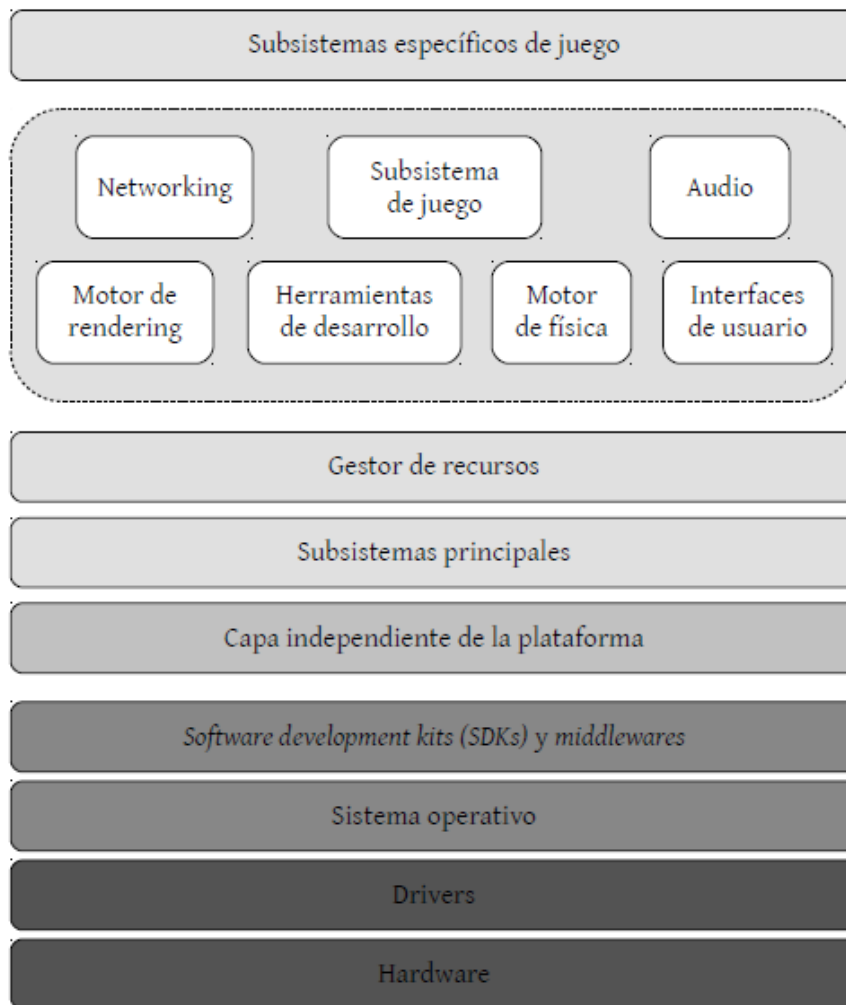
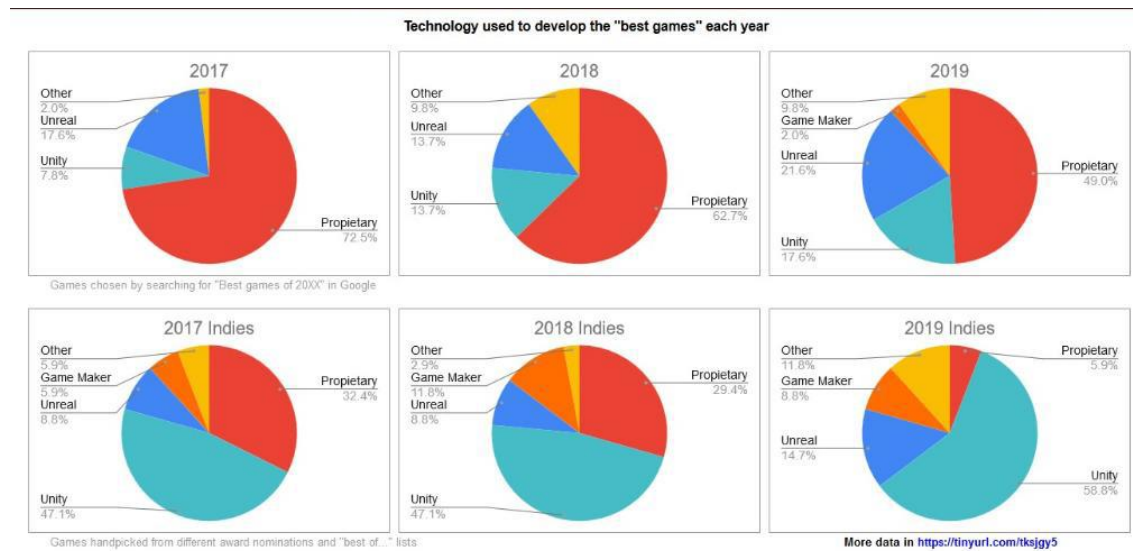


Figura 1.9: Visión conceptual de la arquitectura general de un motor de juegos. Esquema adaptado de la arquitectura propuesta en [5].

Los juegos se desarrollan teniendo en cuenta su potencial lanzamiento en diversas plataformas. Por ejemplo, un título se puede desarrollar para diversas consolas y para PC al mismo tiempo. En este contexto, es bastante común encontrar una capa software que **aisle** al resto de capas superiores de cualquier aspecto que sea **dependiente de la plataforma**. Dicha capa se suele denominar capa independiente de la plataforma.

Un equipo de desarrollo puede optar por crear su propio motor de juego o usar uno ya existente. Para decidirse por una u otra opción habrá que tener en cuenta bastantes factores, pero los más determinantes son el tiempo y los costes de desarrollo que requieren un motor propio.

Normalmente, solo grandes compañías desarrollan sus propios motores de juego, y unas pocas empresas se dedican a ofrecer y mantener este tipo de herramientas (es su producto) para segundas empresas. Por ejemplo, el **Unreal Engine** de la empresa Epic Games ha sido utilizado en multitud de juegos diferentes, e incluso es posible descargarse su versión 4 de forma gratuita. Mario García Lázaro, Lead Gameplay Programmer en Mercury Steam, incluso mostró su preocupación sobre hasta qué punto motores de juego libres como Unity le estaban comiendo terreno al desarrollo propietario.



Lenguajes de programación más usados

Aunque hay docenas de lenguajes de programación que puedes utilizar, si quieres desarrollar un proyecto profesional y trabajar en videojuegos, los más importantes son:

C++ - Este lenguaje presenta un tiempo de ejecución muy bajo, permite a los desarrolladores tener control sobre el hardware y, además, es compatible con la mayoría de los entornos de programación. Su curva de aprendizaje es bastante corta y, si lo dominas, podrás programar fácilmente en otros lenguajes.

C# - Aunque es más fácil de aprender que C++, también es más limitado. No obstante, a pesar de no ser tan flexible ni tan compatible, algunos motores como Unity permiten programar con él. El hecho de dominar C# puede abrirte muchas puertas en el ámbito profesional, ya que es uno de los lenguajes más utilizados en entornos de Windows.

Java - Este lenguaje se caracteriza por ser multiplataforma y tener una enorme flexibilidad. Su utilidad y su diseño son similares a C++, pero, a diferencia de este lenguaje, Java se ejecuta dentro de su máquina virtual, lo cual supone una pérdida de rendimiento a la hora de ejecutar juegos.

JavaScript - Como es uno de los lenguajes de programación más utilizados en el ámbito del desarrollo web, presenta una gran variedad de recursos. La mayoría de los motores de videojuegos son compatibles con JavaScript, por lo que, si lo dominas, podrás usarlo para crear todo tipo de scripts. Funciona muy bien para juegos online y tiene múltiples Frameworks para la creación de programas 3D.

Python - Se trata de un lenguaje de programación multiplataforma muy potente y flexible. Uno de los principales beneficios que ofrece es su framework Pygame, que permite crear prototipos de videojuegos de forma rápida y sencilla.

Además, también se pueden usar otros lenguajes de programación para crear videojuegos. Los lenguajes de marcas, como HTML5 y CSS3, son también bastante utilizados a la hora de crear juegos, especialmente si queremos que estos sean multiplataformas y se puedan ejecutar dentro de cualquier navegador web, independientemente del sistema operativo.

Enlaces de Interés

Algunos enlaces para que trasteéis:

- <https://www.hyperhype.es/motores-graficos-y-de-juego-definicion-tipos-y-mo-delos-de-negocio/>
- <https://www.xataka.com/videojuegos/me-voy-a-meter-a-estudiar-videojuegos-que-donde-esta-futuro-mito-realidad-vayamos-partes>

¿Dónde puedo aprender a hacer videojuegos?

- https://tartanga.hezkuntza.net/es/formacion/ciclos/visualizador?p_p_id=KAIOAContenido_WAR_w24p007ContenidoWARportlet&_KAIOAContenido_WAR_w24p007ContenidoWARportlet_idArticulo=7009584&_KAIOAContenido_WAR_w24p007ContenidoWARportlet_idPlantilla=7001856