

Envío: 20-09-2012

Aceptación: 21-09-2012

Publicación: 24-09-2012

GRÁFICOS A LA MÁXIMA POTENCIA: UNA COMPARATIVA ENTRE MOTORES DE JUEGOS

HIGH PERFORMANCE GRAPHICS: A GAME ENGINE COMPARISON

Jordi Llinares Pellicer¹
Juan Vicente Martínez Pérez²
Arturo Candela Moltó³

1. Doctor en Informática. Instituto Tecnológico de Informática. Universidad Politécnica de Valencia.
2. Ingeniero técnico en informática de Gestión. Instituto Tecnológico de Informática. Universidad Politécnica de Valencia.
3. Máster en software libre. Instituto Tecnológico de Informática. Universidad Politécnica de Valencia.

RESUMEN

El objetivo del presente artículo consiste en presentar una comparativa de prestaciones y características de las más importantes herramientas de generación de juegos y aplicaciones gráficas en tiempo real del momento.

ABSTRACT

The aim of this article consists of showing the results of a comparative analysis of features of the most important current game and real time graphics development engines.

PALABRAS CLAVE

Motores de juegos, gráficos en tiempo real, 3D, 2D, rendimiento.

KEYWORDS

Game engines, real-time graphics, 3D, 2D, performance.

INTRODUCCIÓN

Los juegos y las aplicaciones gráficas interactivas en tiempo real en general, están viviendo uno de los momentos más importantes de su historia. El volumen de negocio de la industria del videojuego ha sobrepasado ya a la del cine, y genera un ecosistema de miles de empresas en todo el mundo. La reciente aparición de los dispositivos móviles (iOS, Android, Windows Phone) ha multiplicado el ya gran mercado y ha convertido en potenciales clientes a prácticamente cualquier ser humano del planeta.

El objetivo del presente trabajo consiste en convertirse en una herramienta que permita a los investigadores y desarrolladores de este tipo de aplicaciones tener un marco de referencia de sus necesidades.

El desarrollo de aplicaciones gráficas que exigen altas prestaciones, en especial juegos, requieren de herramientas de desarrollo altamente especializadas y complejas. Estas herramientas deben garantizar estrictas condiciones, algunas de las más importantes se enumeran a continuación:

- Obtener el **máximo rendimiento gráfico de los dispositivos** sobre los que se ejecuta la aplicación o juego
- Aprovechar las **características más avanzadas de las unidades de procesamiento gráfico** que les permita los mejores resultados visuales y realismo
- Ofrecer un **conjunto de herramientas** que permitan a los desarrolladores aprovechar las características anteriores con el menor esfuerzo posible, con un potente editor, un poderoso lenguaje de programación o scripting, y con una amplia gama de complementos: importación de modelos, imágenes, sonidos y otros medios, motor de físicas, edición de terrenos, sistemas de partículas, herramientas de inteligencia artificial, etc.
- **Multiplataformidad** tanto en las herramientas de desarrollo como en su capacidad de exportar la aplicación final a diversas plataformas: escritorio, web, dispositivos móviles y consolas

Ante este complejo escenario, el desarrollador interesado en generar aplicaciones gráficas en tiempo real y más específicamente juegos, debe elegir convenientemente las herramientas más adecuadas, con características muy dispares e incluso costes muy variados.

Se mostrarán los resultados de un profundo análisis de las opciones más importantes del momento, centrándose en las características más importantes y ofreciendo a potenciales interesados en estas herramientas de precisos criterios de selección en base a los requerimientos de sus desarrollos.

MOTORES DE JUEGOS

Los motores de juegos (game engines) aparecieron prácticamente desde el mismo momento en el que empezó el desarrollo de juegos. Por motor de juegos se entiende un conjunto de herramientas software que permitan asistir a los desarrolladores en la creación de juegos y aplicaciones gráficas interactivas en tiempo real. Son muchas las opciones de las que se dispone en estos momentos, cientos de hecho, y que dificulta enormemente abordar su estudio detallado.

El listado más actual de las distintas posibilidades con las que los desarrolladores se encuentran en estos momentos lo podemos encontrar en [1]. En [2] podemos encontrar también una amplia lista con una descripción breve de sus principales características. Un estudio similar al propuesto en este trabajo pero con un espectro más reducido lo podemos encontrar en [3].

De este enorme listado se ha seleccionado un conjunto inicial de opciones en base los siguientes criterios:

- Importancia en el mercado (penetración y número de desarrolladores que los utilizan)
- Potencia de las herramientas que ofrecen a los usuarios
- Plataformas sobre las que los usuarios pueden distribuir sus productos al utilizar dicho motor

En base a los criterios anteriores, los principales motores del momento son los siguientes:

- Unity3D (<http://unity3d.com>)
- UDK (<http://www.udk.com>)
- Corona SDK (<http://www.anscamobile.com/corona>)
- Shiva3D (<http://www.stonetrip.com>)
- Torque3D (<http://www.garagegames.com/products/torque-3d>)
- Hero Engine (<http://www.heroengine.com>) (MMO)
- Big World Technology (<http://www.bigworldtech.com>) (MMO)
- Cry Engine (<http://mycryengine.com>)
- Marmalade (<http://www.madewithmarmalade.com>)
- Realm Crafter (<http://www.realmcrafter.com>) (MMORPG)
- GameSalad (<http://gamesalad.com>)
- Panda3D (<http://www.panda3d.org>) Open Source
- Irrlicht (<http://irrlicht.sourceforge.net>) Open Source
- Havok (<http://www.havok.com>) Havok Vision Engine

- Esenthel Engine (<http://esenthel.com>)
- Game bryo (<http://gamebryo.com>)
- Source Engine (<http://source.valvesoftware.com>)
- OGRE (<http://www.ogre3d.org>) Open Source
- XNA (<http://msdn.microsoft.com/en-us/centrum-xna.aspx>) Librería

Aunque a priori menos interesantes por su implantación y extensión, se consideran muy interesantes la siguiente lista de motores:




- SIO2 Engine (<http://sio2interactive.com>)
- Bork3D Engine (<http://www.bork3d.com/engine>)
- SDL (<http://93i.de/cms/products/game-engines/sdl-game-engine/>)
- SFML (<http://www.sfml-dev.org>)
- RPG Maker (<http://www.rpgmakerweb.com>)
- The Game Creators (<http://www.thegamecreators.com>) -> FPS Creator, 3D Game Maker, agk app game kit (<http://www.appgamekit.com>), X-Quad Editor, PureGDK etc.
- 3D GameStudio (<http://www.3dgamestudio.com>)
- Leadwerks Game Engine (<http://www.leadwerks.com/werkspace/page/Products/le2>)
- The Shoot'Em Up Kit (<http://www.tallstudios.com/products/shoot-em-up-kit>)
- Euphoria (<http://www.naturalmotion.com/products/euphoria/>) (Librería)
- Crystal Space (<http://www.crystalspace3d.org>) Open Source
- Ika (<http://sourceforge.net/projects/ika/>) Open Source
- Nebula Device (<http://sourceforge.net/projects/nebuladevice/>) Open Source
- Genesis Device (<http://sourceforge.net/projects/genesisdevice/>) Open Source
- JMonkey (<http://jmonkeyengine.com>) Open Source
- Sploder (<http://www.sploder.com>)
- PyGame (<http://www.pygame.org>) Open Source
- Bajaengine (<http://www.bajaengine.com>)
- Moai (<http://getmoai.com>)




ANÁLISIS




La comparativa que se ha llevado a cabo entre esta selección de motores de juegos se ha basado en el estudio de sus posibilidades en base a los siguientes criterios:




- Soporte 2D y/o 3D.
- Herramientas de edición (Editor), que permitan la edición de mundos, escenas en 2D o 3D. Es importante también sobre qué plataformas funciona el editor.
- Plataformas sobre las que se puede distribuir el juego o producto utilizando el motor. La multiplataformidad es un elemento esencial.
- El lenguaje de programación utilizado.
- Dos complementos altamente recomendables: motor de físicas y herramientas de IA (Inteligencia Artificial, como las técnicas de búsqueda de caminos o Path Finding).
- Características avanzadas, como occlusion culling, lightmapping, shaders, editor de terrenos, sistemas de partículas etc. En este apartado se hará mención de las que su existencia se haya podido constatar en el análisis.
- Licencias. Importante si el motor está basado en software libre (y el tipo de licencia utilizada) o una licencia comercial y las condiciones económicas derivadas de su uso.





Los resultados del análisis de los motores más importantes se pueden ver en la siguiente tabla:

Nombre	2D/3D	IDE	Lenguaje	Plataformas de Distribución	Motor Físico	Herramientas de IA	Características Avanzadas	Licencia	Datos de Interés
 Unity3D	2D y 3D	Sí (Mac y Windows)	C#, Boo, JS	Mac, Windows, Linux, iOS, Android, PS3, XB360, PSP, Wii, PSVita, Flash, Web	PhysX	NavMesh y Path Finding	Forward y Deferred rendering, Occlusion Culling, light mapping, light probing, LOD discreto, edición de terrenos, sistema de partículas	Comercial. Versión básica gratuita (desktop y web) y versiones PRO y plataformas móviles y consolas con un pago único sin royalties	Motor con mayor número de desarrolladores registrados del momento
 UDK/Unreal Engine 3	2D y 3D	Sí	Unreal Script	Mac, Windows, iOS, Android, PS3, XB360, PSVita, Wii, Flash	PhysX	Path Finding	Deferred rendering, occlusion culling, parallax mapping, Per Object Motion Blur	Comercial. Versión libre y pago de royalties sobre producto vendido al superar un umbral de facturación. La versión Unreal Engine 3 dispone de más funcionalidades y una licencia comercial más cara.	Motcon el motor grafico más potente del momento según los expertos
 Corona SDK	2D	No	LUA	iOS, Android, Nook, Kindle Fire	Box2D	No. Deben ser programadas desde cero en LUA	Interfaz simple de Lua para la programación de física	Comercial. Versión Indie para desarrollar para una única plataforma (iOS o Android) y versión Pro para todas las plataformas	OpenGL, OpenAL, Google Maps, Facebook Connect, Game Center, in-app purchase

 Shiva3D	2D y 3D	Sí	C++, LUA	Windows, Mac OS, Linux, iOS, Android, BlackBerry QNX, HP WebOS, Bada, Symbian, Palm, Wii, Web	PhysX. Incluye soporte para ODE	AIModel	Per-pixel lighting, terrains, vegetation, oceans, particles, Video & Data streaming	Comercial. Versiones: Web, Basic, Advanced, Educational	Herramientas específicas MMO (Massive Multiplayer Online)
 Torque3D	3D	Sí. Sólo para Windows	TorqueScript	Windows, Web	PhysX. Además cuenta con una API física abstracta para integrar cualquier otro motor físico.	Tactical AI Kit	Per-pixel dynamic lighting, extensible shaders, normal & parallax occlusion mapping, Screen Space Ambient Occlusion	Comercial. Próximamente liberado bajo licencia MIT	Arquitectura cliente/servidor incluso para juegos de un único jugador. Cliente: elementos GUI, menús, rendering... Servidor: lógica del juego, definición de objetos...
 Hero Engine (MMO)	3D	Sí	Hero Script, C++ (PRO)	Windows	PhysX	Path Finding	Forward rendering	Comercial. Precio por año.	Herramientas específicas MMO (Massive Multiplayer Online)

 Big World Technology (MMO)	3D	Sí	Python, C++ (PRO)	Windows	Motor propio	Path Finding	Forward rendering, parallax mapping	Licencia. Precios por año, versión indie y Pro.	Herramientas específicas MMO (Massive Multiplayer Online)
 Cry Engine	3D	Sí	VisualScript	Windows, XB360, PS3	Motor propio	Path Finding	Occlusion culling, Deferred rendering, parallax mapping, Per Object Motion Blur	Comercial. Es gratis para uso no comercial	CryENGINE Cinebox - Herramienta para reducir el Time to Market de los creadores de Contenido
 Marmalade	2D y 3D	No, Windows y Linux	C, C++, HTML5, CSS3, JS	iOS, Android, Blackberry Playbook OS, bada, Windows, Mac, LG Smart TV	Open Dynamics Engine. Se puede integrar cualquier otro motor	No. Se deben programar desde 0		Licencia. Precios por año e instalación. Versiones community, indie y pro.	Disponible EDK para acceso directo a las APIs nativas de iOS y Android
 Realm Crafter (MMORPG)	3D	Sí	C#	Windows	PhysX	Path Finding	Dynamic lighting, renderizado ambiente con nubes dinámicas y sombras suaves, ambiente con efectos atmosféricos y ciclo día/noche, superficies de agua reflexivas y refractivas	Comercial. Versión Standar, Professional y Combo con un único pago	Varios editores: General Editor, Scripting Editor (con Quest Wizard), Armoury y un editor de terreno básico

 vGameSalad	2D	Mac y Windows	"Drag&Drop"	Windows, Mac, iOS, Android, HTML5	Motor propio	Addons disponibles en GameSalad MarketPlace		Licencia. Versión Basic gratuita con limitaciones. Versión Pro con precio por año.	Tienda MarketPlace con muchos recursos disponibles gratis y de pago para nuestros proyectos
 Panda3D	2D y 3D	Sí. Para Mac, Windows y Linux	Python, C++	Windows, Mac y Linux	Motor básico propio. Soporta también: PhysX, ODE y Bullet	Steering behaviors: Seek, Flee, Pursue, Evade, Wander, Flock, Obstacle Avoidance, Path Following; Path Finding; Navigation Mesh generator for EGG files	Posibilidad de uso de lenguajes de alto nivel para shaders: Cg, GLSL. Sistema de partículas propio.	Open Source	
 Irrlicht	2D y 3D	Sí. Para Windows y Linux	C++	Windows, Mac y Linux	Motor básico propio. Integración con Tokamak (http://sourceforge.net/projects/tokamak/)	X	Utiliza un CLOD para terreno integrado	Open Source	Soporta la importación de muchos formatos de geometría 3D

 Havok	3D	Visual Studio + herramientas propias	C++ y Havok Script	Xbox 360®, PlayStation 3®, PC Games for Windows, PlayStation® Vita, Wii™, Wii U™, Windows 8, Android™, Apple Mac, and iOS	Havok Physics	Havok AI y Havok Behavior	Havok Destruction permite diseñar objetos teniendo en cuenta que van a ser destruidos. Havok Cloth Para el diseño y efectos de ropa	Comercial, Tienen una versión limitada gratuita para juegos que se vendan por debajo de 10\$ solo para pc	
 Esenthel Engine	3D	Sí	C++, Esenthel Script	Windows, Mac, Android, iOS	PhysX Bullet	Path Finding	Occlusion culling, Deferred and Forward rendering, parallax mapping, Per Object Motion Blur	Comercial. Versión demo y Full con pago único	Herramientas específicas MMO
 Game bryo	3D		C++	Windows, XB360, PS3, Wii	PhysX	Path Finding	Forward rendering, parallax mapping	Comercial	
 Source Engine	3D	Sí	C++	Windows, Mac, XB360, PS3	Motor propio	Path Finding	Forward rendering	Comercial	



 OGRE	2D y 3D	Mediante las herramientas de desarrollo de la plataforma	C++, .NET (MOGRE), python (Python-Ogre), java (Ogre4j), Lua (Lugre)	Windows, Mac y Linux, iOS	Integración con OgreBullet, OgreNewt, NXOgre (PhysX Wrapper), OgreODE No funcionan en todas las plataformas sobportadas por ogre	Utilizando el motor físico Newton (OgreNewt) se puede implementar Path Findinf	Se gestionan de forma automática los objetos transparentes de la escena, Herramienta para buscar memory leaks en los juegos	Open Source - LGPL con excepciones. A partir de la versión 1.7 licencia MIT (http://opensource.org/licenses/mit-license.php)	
 XNA - Librería	2D y 3D	XNA Game Studio	C# Oficial y .NET	Windows, Windows Mobile Phone, Xbox 360	Integración con JigLix (3D/2D), Farseer (2D)	Librería Externa: XNA Pathfinding Library	En un par de clicks desde el editor se genera el archivo instalable para plataformas Windows	Comercial. Para distribuir aplicaciones en las plataformas de Microsoft, hay que ser miembro del Microsoft XNA Creator Club, que cuesta 99\$/Año	

Tabla 1. Resultados de análisis de los motores más importantes. Fuente: Elaboración propia.

CONCLUSIONES

En el presente trabajo se ha llevado a cabo un exhaustivo estudio de los motores de juegos y aplicaciones gráficas interactivas en tiempo real más importantes del momento. De una inmensa colección inicial, se han seleccionado las 19 principales opciones en el mercado actual.

Son muchas las peculiaridades de cada una de las opciones, pero encontramos claramente 3 grupos de herramientas que destacamos a continuación:

- Entornos Open Source, que bajo distintas licencias de código abierto, permiten el desarrollo de juegos y aplicaciones sin el sometimiento a licencias comerciales o costes de ningún tipo. Aunque con capacidad de llegar a desarrollar productos muy competitivos, carecen de algunas de las herramientas que las soluciones principales proporcionan. Uno de sus más claros exponentes es OGRE.
- Herramientas muy dirigidas a desarrolladores no avanzados o con necesidades no muy exigentes. Es el caso de Corona o GameSalad, centrados en juegos 2D, con lenguajes de scripting muy sencillos, pero con posibilidades muy limitadas.
- Finalmente, hay un conjunto de herramientas muy poderosas, con grandes diferencias en prestaciones y sistema de licencias, pero casi todas con capacidad de desarrollar juegos de altísimas prestaciones. La elección entre ellas depende de muchos criterios:
 - Prestaciones concretas que el producto a elegir requiere
 - Plataformas sobre las cuales el producto debe ser generado
 - Coste o inversión económica necesaria para el uso de las herramientas de desarrollo

En este grupo destacan Unity3D (muy horizontal en plataformas de desarrollo y editor para Mac y Windows), UDK y su hermano mayor Unreal Engine (que destaca generalmente por ofrecer las más altas prestaciones y características en las diferentes plataformas), Havok Vision Engine, Cry Engine, Source Valve, Gamebryo, Torque y Shiva.

REFERENCIAS

- [1] <http://devmaster.net/devdb/engines>
- [2] http://en.wikipedia.org/wiki/List_of_game_engines
- [3] http://www.gamasutra.com/blogs/MarkDeLoura/20090302/581/The_Engine_Survey_General_results.php