

Generador de entornos virtuales en tiempo real basado en capas

A Real-Time, Layer-Based Virtual Environment Generator

Ómar Correa Madrigal

Universidad de las Ciencias Informáticas, Cuba

✉ ocorrea@uci.cu

Julio Óscar Gutiérrez

Universidad de las Ciencias Informáticas, Cuba

✉ jogutierrez10@uci.cu

Giraldo Andrés González Montoya

Universidad de las Ciencias Informáticas, Cuba

✉ gagonzalez10@uci.cu

ABSTRACT

The generation of virtual environments in real time is presently a highly researched subject. It has great potential to reduce the development time of virtual reality products and to create very large environments with a wide variety of content. These benefits have become very popular in the development of video games and simulators. This work proposes a new generation system—a layer generator that incorporates the best features of such systems with real-time and new generation techniques. To the present, the layer generator has been applied to games, and may potentially be used with other systems.

KEYWORDS: layer generation, real time, virtual environment, development.

El avance alcanzado por el *hardware* y la creciente necesidad de lograr visualizaciones más realistas han permitido obtener resultados impresionantes, a través del gráfico por computadores. Aunque los dispositivos de almacenamiento y de procesamiento permiten aspirar a niveles aceptables de visualización, lo cierto es que la complejidad de los sistemas sigue poniéndolos en crisis, por lo cual continúa latente la necesidad de la eficiencia en la construcción y representación de los entornos virtuales.

La curiosidad de un jugador o de un usuario de cualquier sistema de realidad virtual eleva constantemente la demanda de crear entornos más extensos y con menos restricciones de movimiento. Los desarrolladores, a pesar del avance de las tecnologías de diseño, presentan siempre problemas con el tiempo y el esfuerzo para elaborarlos. Por ese motivo, la *generación automática de entornos* ha devenido siempre como campo de apoyo para resolver tales problemas, pues permite grandes desarrollos.

Gracias a la generación, el tiempo para crear un entorno de interiores de cualquier tamaño puede ir desde dos segundos hasta seis minutos (Adams, 2002), algo que contrasta con la estadística: “la creación de un videojuego con 14 horas de contenidos relevantes puede tomar hasta 1 año de trabajo” (Ince, 1999), si se analiza que la fase de creación de sus con-

tenidos es la que mayor tiempo consume. Por otra parte, la generación brinda otras potencialidades, como es la creación de entornos infinitos (Greuter, Stewart y Leach, 2004; Greuter y Parker, 2003a y 2003b; Blizzard, 2000) en tiempo real con una gran variedad de contenidos, válidos para videojuegos y simuladores.

Tipos de sistemas de generación de entornos virtuales

La generación de entornos virtuales no es un área donde haya muchas clasificaciones establecidas, a pesar de que en ella se ha trabajado bastante. Hasta el momento no se conocen trabajos que intenten congeniar los diferentes vocablos utilizados, con vista a lograr una mejor coherencia en los conocimientos que abarca este tema. Este trabajo no pretende analizar profundamente el tema, pero sí hacer propuestas para una mejor comprensión del resultado que se expone. Según el espacio temporal donde trabajan, los sistemas de generación se pueden clasificar en:

- Generadores en preprocesamiento.
- Generadores en tiempo real (Adams, 2002).

En el primer caso se encuentran las herramientas que permiten la creación y la configuración de los escenarios antes de ser cargados y visualizados por la aplicación en tiempo real. Existen innumerables productos de este tipo, por ejemplo, están las herramientas de creación de contenidos de la compañía Presagis (Corpotatio, 1999), empleadas en el desarrollo de entornos para simuladores, esencialmente. Además, se encuentran las herramientas de edición de niveles para videojuegos, entre las cuales resaltan los editores de la compañía Blizzard (2000), usados en sagas como StarCraft, WarCraft y World of WarCraft (WoW). De igual forma, se pueden enmarcar en esta categoría los sistemas que en el proceso de carga de los contenidos del escenario generan el entorno según ciertas reglas, como pueden ser: la extensión del entorno y la cantidad de usuarios, oponentes, vidas y bonificaciones —por lo general aplicados en videojuegos como (Shoemaket Stainless, 2004)—.

Por otra parte, los sistemas de generación en tiempo real confieren al espacio de ejecución su lugar de acción, lo cual permite construir entornos según sea el avance sobre este. Es importante puntualizar que todos los generadores en tiempo real conocidos (Greuter y Parker, 2003a y 2003b; Blizzard, 2000; Shoemaket Stainless, 2004; Dymchenko, 2002; Strugar, 2007) utilizan o proponen algoritmos y técnicas eficientes que permiten el logro de visualizaciones en tiempo real.

Los sistemas de generación también se conocen por los elementos capaces de generar. Si esto se toma en consideración, los sistemas expertos en crear contenidos (terreno, vegetación, edificios, etc.) son importantes, pues se especializan en la creación procedural de los contenidos (Coelho et al., 2007; Muller y Wonka, 2006). Por otra parte, la generación de espacios virtuales agrupa aquellas herramientas y sistemas expertos en el ensamblado y estructuración de entornos. También están los generadores híbridos, que crean y disponen de los contenidos generados para conformar el entorno. Los generadores en tiempo real son, por lo general, generadores híbridos (Greuter y Parker, 2003a y 2003b).

Sistemas de generación en tiempo real

Los sistemas de generación en tiempo real son muy frecuentes, sobre todo cuando se trabaja en videojuegos y simuladores. En este sentido, se puede encontrar el SpeedTree® (Interactive Data, 2009), un sistema de generación construido por Interactive Data Visualization, de carácter comercial, que repalda la generación procedural de contenidos y de entornos. Ha sido integrado dentro del Epic's Unreal Engine y en otros motores para videojuegos, con buenos resultados. Permite una frecuencia de visualización equilibrada, para lo cual está equipado de buenas técnicas y algoritmos de optimización.

En el plano no comercial se encuentran otros sistemas de generación en tiempo real; pero hasta lo que se conoce ninguno provee el código. En este sentido, se encuentra el generador

propuesto por Greuter y Stewart (2003), el cual está preparado para crear cualquier tipo de entorno, al decir de los autores, aunque las pruebas se basan solo en el establecimiento de ciudades infinitas. Como la mayoría de los generadores estudiados, da forma a contenidos hacia la cámara —en este caso basándose en la estrategia de generación *frustum filling* y construyendo edificios, calles y árboles de manera procedural— con algoritmos que proveen una variabilidad muy alta.

En este sentido, al continuar con el análisis se puede encontrar el generador Scaper (Nocke, 2005), el cual también basa su sistema en la estrategia de *frustum filling*, algo que lo limita en cuanto a flexibilidad. En este caso, no utiliza la generación procedural de contenidos, sólo se centra en su disposición en el entorno, haciendo variaciones en forma (escalado), posición y orientación, para lo cual cuenta con un robusto fichero XML, que respalda toda una variedad de contenidos y alternativas de generación de entornos virtuales.

Es un sistema que genera los entornos en función de la cámara, basándose en estructuras como los generadores, elementos principales que controlan una sección espacial según su topología —la topología identifica el tipo de volumen de encierro del espacio de generación (rectángulo, esfera, cubo, etc.)— y en esta misma distribuyen los contenidos. Sus potencialidades se han probado en entornos de ciudad, paisajes y espacios. Presenta una arquitectura flexible y extensible para la confección de otros entornos.

Todos los generadores estudiados (comerciales o no) presentan una tendencia a la especialización en la generación de ciertos tipos de entornos. Esto presupone de alguna manera que los principios y las estructuras de los generadores no son lo suficientemente genéricos como para permitir crear cualquier entorno basado en diferentes estrategias de generación. Además, desde el punto de vista ingenieril, los generadores no comerciales mencionados no presentan una arquitectura libre para el acople con diferentes motores gráficos, un aspecto muy importante para el presente trabajo.

Generador de entornos virtuales basado en capas

El sistema de generación basado en capas es un enfoque de generación diferente y flexible. Tomando lo mejor de sistemas como Scaper y otros, el sistema propuesto permite el acople con diferentes motores gráficos y brinda las bases para la incorporación de diferentes técnicas de generación. Con el objetivo de entender sus bases es necesario explicar un número de conceptos. Estos son:

- *Contenido*: todas aquellas geometrías que conforman un entorno virtual.
- *Estrategia de generación*: técnicas de generación de contenidos y de entornos virtuales.
- *Capa de generación*: elemento encargado de generar tanto contenidos como una porción de un entorno basado en

alguna estrategia definida, tomando en consideración el espacio que controla según su topología.

- *Topología de una capa*: encierra la forma que esta tiene y delimita el espacio que controla (puede ser 2D o 3D). Las topologías pueden ser capas 2D con forma de cualquier primitiva (punto, rectángulo, circunferencia o triángulo). De igual forma, las 3D (cubo o esfera).
- *Generador de entornos*: elemento encargado de orquestar todas las capas dentro de un entorno con vista a lograr su generación.
- *Cámara*: representa al usuario en el sistema y es un punto esencial para la generación en tiempo real.
- *Memoria a corto plazo*: es un elemento propio de la inteligencia artificial expuesto con mayor claridad, utilizado por los agentes inteligentes para la toma de decisiones. En este contexto le provee al generador un espacio de almacenamiento de información que se borra cada cierto período y recoge aquellos contenidos significativos para la aplicación, por ejemplo: los elementos de la lógica de un videojuego. Este elemento permite establecer una coherencia estructural en los entornos generados.

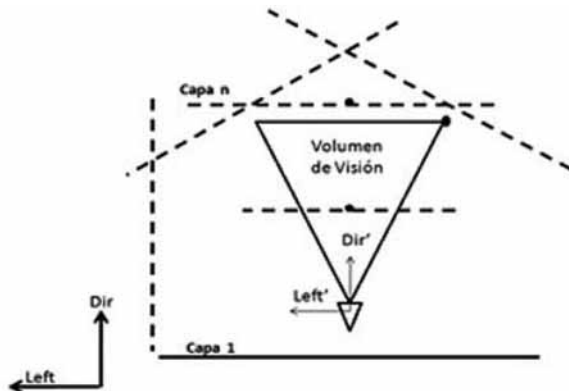


Figura 1. Estructura en capas del generador

Como se puede observar en la Fig. 1, la estructura en capas permite una flexibilidad y potencialidades de generación interesantes. Primero, pueden orientarse respecto a cualquier sistema de referencia e incluso trasladarse; segundo, el sistema de generación siempre se traslada relativamente a la cámara, lo que permite una estabilidad en el número de capas que se van a controlar, y, tercero, generar contenidos en posiciones ventajosas para su visibilidad. Todas las capas son controladas por el generador según la estrategia de generación establecida.

El sistema sigue un proceso en orden genérico, independiente de la estructura, estrategia y el entorno que se desee generar. El orden del proceso es el siguiente:

- Se actualizan todas las capas relativamente a la cámara según la estrategia de generación definida.
- Se revisan las colisiones entre capas. En los casos de colisión se corrigen si la estrategia no la concibe como parte de ella.

- Se le ordenan los contenidos a cada capa que se va a generar y se posicionan solamente según su topología y estrategias definidas. En este punto el generador selecciona si en el espacio de generación se deben posicionar nuevos contenidos o se ponen algunos almacenados en la memoria a corto plazo.
- Se actualiza el tiempo de vida de la memoria. En caso de que este haya llegado a su fin, se limpia la memoria y se condiciona para almacenar nuevos elementos.

Aplicaciones y pruebas de rendimiento

El sistema se encuentra en una primera versión y se ha aplicado en videojuegos. La Fig. 2 muestra una configuración del sistema para crear los entornos del videojuego Meteorix; entre tanto, la Fig. 3 es otra aplicación, pero de carácter de prueba de integración con el motor gráfico Ogre3D.

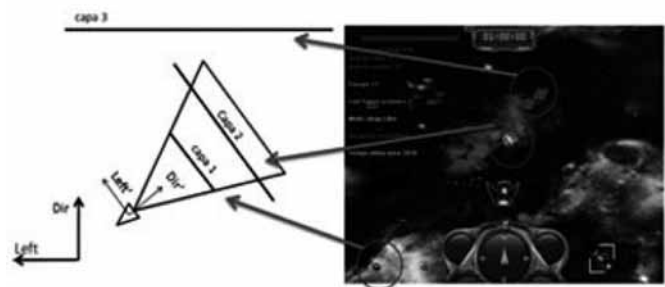


Figura 2. Estructura en capas del generador en Meteorix Game

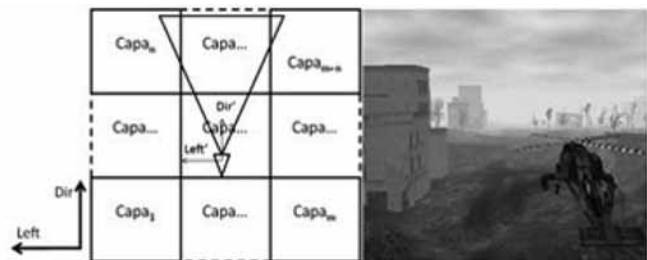


Figura 3. Estructura en capas en un ejemplo de integración con OGRE3D

Se han realizado y probado otros entornos con diferentes estrategias. Los entornos generados hasta el momento han sido: entornos espaciales, de paisajes, interiores y dirigidos. Los entornos de paisajes utilizan una novedosa técnica de generación llamada *DynamicTiles*, y en la Fig. 4 se observa el rendimiento del sistema usándola. La configuración del sistema es como se muestra en la Fig. 3. Todos los entornos generados han dado respuestas en tiempo real.

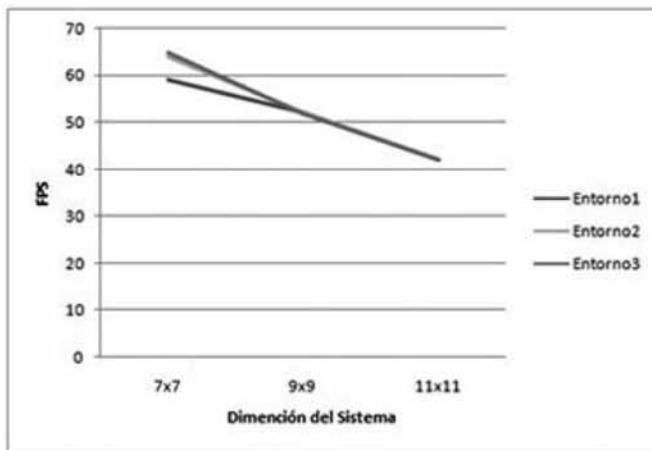


Figura 4. Pruebas de rendimiento del sistema con la técnica DynamicTiles

Conclusiones

El estudio permitió identificar que la generación de entornos virtuales es un área de pocas clasificaciones establecidas, lo cual es necesario para su mejor desarrollo y comprensión. También se identificó que existen pocas publicaciones de sistemas de generación en tiempo real, lo cual demuestra que es un área de trabajo en la cual se puede continuar trabajando. El sistema resultante constituye una novedosa solución, genérica tanto por su arquitectura como por las estrategias de generación de entornos que encapsula. Así, un sistema multi-técnicas, independiente del motor de visualización e inteligente, resume sus potencialidades. A modo de recomendación, el sistema debe seguir aumentando el número de técnicas y extender su aplicación a otras áreas donde se requiera una mayor fidelidad.

Referencias

- Adams, D. (2002). *Automatic generation of dungeons for computer games*. Recuperado el 22 de septiembre de 2010, de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.1445&rep=rep1&type=pdf>.
- Blizzard, C. (2000). *World of Warcraft*. Recuperado en el 2009, de <http://www.blizzard.com/war3x/>.
- Coelho et al. (2007). Expedient modelling of virtual urban environments with geospatial L-systems. *Computer Graphics*, 26 (4), 769-782.
- Corpotatio, P. (1999). *Content creation*. Recuperado en el 2009, de http://www.presagis.com/products/content_creation
- Dymchenko, L. (2002). Polygonal technology: weaknesses and alternative. Recuperado el 22 de septiembre de 2010, de <http://ixtblabs.com/articles/virtualrayengine/index.html>.
- Greuter, S. y Parker, J. (2003a). *Undiscovered worlds - towards a framework for real-time procedural world generation*. s. l.: MelbourneDAC.
- (2003b). *Real-time procedural generation of pseudo infinite cities*. Melbourne: RMIT University.
- Greuter, S., Stewart, N. y Leach, G. (2004). Beyond the horizon: Computer generated, three-dimensional, infinite virtual worlds without repetition in real-time. En *Image text and sound*. Melbourne: RMIT Publishing.
- Ince, S. (1999). *Automatic dynamic content generation for computer games*. Disertación no publicacda, University of Sheffield.
- Interactive Date, V. (2009). SpeedTree. Recuperado de <http://www.idvinc.com/html/speedtreert.htm>.
- Muller, P. y Wonka, P. (2006). *Procedural modeling of buildings*. Recuperado de http://www.vision.ee.ethz.ch/~pmueller/documents/mueller.procedural_modeling_of_buildings.SG2006.web-version.pdf.
- Nocke, F. (2005). *Scaper: real-time generation of infinite environment*. Recuperado de <http://sciencestage.com/d/1137214/scaper-?-real-time-generation-of-infinite-environments.html>.
- Shoemaket Stainless, S. (2004). Random map generation for strategy games. En *AI Game Programming. Wisdom II* (po. 405-412). s. l.: Charles River Media.
- Strugar, F. (2007). *Advantage terrain library*. Recuperado el 22 de septiembre de 2010, de <http://www.advantageterrain.com/>.