

Introducción

En este primer capítulo, responderemos algunas preguntas básicas acerca de los juegos de video y nos haremos otras, que nos permitirán plantear cuestiones importantes para el desarrollo antes de lanzarnos a la aventura que es la creación de videojuegos.

¿Qué es un videojuego?	16
Un poco de historia	18
De la idea a los hechos	20
Autopsia de un videojuego	22
Características de la programación de videojuegos	22
Componentes de un videojuego 2d	24
Algunos consejos útiles	26
Resumen	27
Actividades	28

¿QUÉ ES UN VIDEOJUEGO?

Podría decir que, a lo largo de mi corta vida, hice y fui muchas cosas. Desde viajar al futuro y ser un astronauta fascinado por visitar otros planetas para luchar contra peligrosas criaturas, hasta visitar el pasado como guerrero egipcio que pelea por el faraón. De eso, se trata jugar: sentir la satisfacción de ser y de hacer lo que queremos en un mundo irreal, sin las consecuencias que ese camino pudiese acarrear en el real. Podríamos caer en lava ardiente y resucitar para intentarlo de nuevo. Y volver a intentarlo tantas veces como sea necesario. Naturalmente, eso no puede hacerse en la realidad.



Figura 1. Como es de suponer, en la realidad, no podríamos montarnos en un dragón para eliminar a nuestros enemigos.

Es un hecho que todo ser humano juega. Tal vez sea más notorio en los niños, pero podemos asegurar que no existe persona que jamás haya jugado. Hemos visto en la plaza desde chicos que jugaban con la pelota hasta ancianos enfrascados en una partida de ajedrez. ¡Hasta los perros juegan! Podríamos decir que es una necesidad innata que se manifiesta durante toda la vida.

Ahora, la respuesta a la pregunta del título. Un videojuego es una aplicación de entretenimiento (software) en la que interactúan una o varias personas con una computadora, consola o arcade. Por lo general, existe una superficie donde se proyecta el juego (monitor, televisor o proyector) y un dispositivo donde ingresamos comandos que puede variar entre una palanca (joystick), ratón (mouse), teclado, pantalla táctil, etcétera. Consiste en una simulación cuyos temas abarcan numerosos ámbitos: deporte, guerra, negocios, fantasía, aventura, ciencia ficción, entre otros.



Figura 2. En el Dance Dance Revolution o el Pump It Up, el dispositivo de entrada es una plataforma sobre la que pisamos siguiendo el ritmo.

Por último, hacemos hincapié en uno de los conceptos más importantes en un videojuego: la **interactividad**. Hay una estrecha relación entre el jugador y el juego: el primero recibe cierta información y estímulos, ya sean visuales o sonoros, y, mediante cierto razonamiento, responde en consecuencia. Esto le genera una sensación de satisfacción y placer.



Figura 3. Cuanto más natural sea la interacción, mayor será la sensación de realismo dentro del juego.

{ UN OSCILOSCOPIO ENTRETENIDO

Uno de los primeros videojuegos fue creado en 1958 por el físico William Higinbotham en un osciloscopio para entretener a los visitantes del Laboratorio Nacional de Brookhaven en Nueva York y tomó el nombre de **Tenis Para Dos**.

No vamos a discutir en este momento si un juego debe ser o no entretenido y divertido, puesto que eso resulta subjetivo según quién lo juegue. Esto nos lleva a una pregunta que mil veces me he formulado: ¿hago juegos para mí o para los demás? Y creo que será respondida de forma automática cuando hayamos realizado el primer videojuego.

UN POCO DE HISTORIA

Remontémonos a los comienzos de los videojuegos de PC. Éstos corrían sobre **DOS** (*Disk Operative System* o **Sistema Operativo de Disco**). En ese tiempo, para poder acceder al hardware con buen rendimiento, debíamos hacerlo de forma directa sin pasar por el sistema operativo. Esto era un gran problema para los programadores, puesto que cada fabricante de hardware tenía su propio set de instrucciones y, por lo tanto, los tiempos de desarrollo aumentaban drásticamente.



Figura 4. Aunque resultaba difícil programar bajo DOS, podíamos disfrutar de excelentes títulos, como uno de mis favoritos: *Out of this World*.

Tiempo más tarde, saldría al mercado **Windows 3.1**, que introdujo el **GDI** (*Graphical Device Interface* o **Interfaz de Dispositivo Gráfico**), proveedor de una serie de funciones para acceder al hardware independizándonos del fabricante. Pero no se lo tomó muy seriamente para el desarrollo de videojuegos por su extrema lentitud y, además, se lograba mejor rendimiento con el acceso directo al hardware. Por lo tanto, los videojuegos que salían al mercado continuaban corriendo bajo DOS.

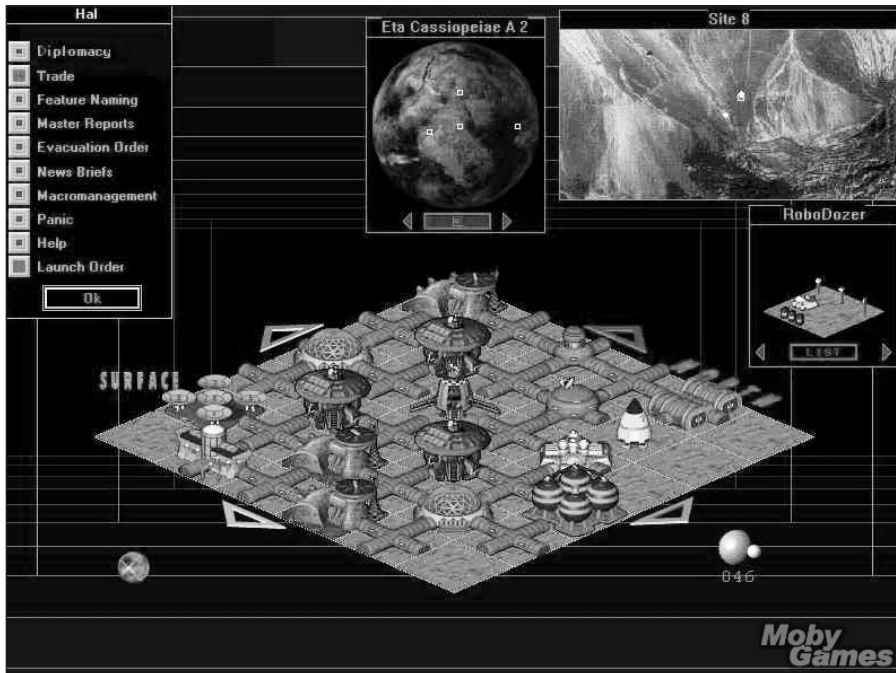


Figura 5. Si bien Windows 3.1 no se tomó muy seriamente para el desarrollo de videojuegos, aparecieron algunos títulos para esta plataforma, como el **Outpost**.

Pasaron unos años, y **Windows 95** vio la luz. Se introdujo el concepto de **Plug & Play** que permitía la fácil instalación de nuevos dispositivos. También se sumó a las mejoras un sistema de multitarea llamado **preemptive** (conocido como multitarea real) que evitaba (a diferencia de la multitarea cooperativa utilizada en Windows 3.1) la monopolización de los recursos por una aplicación.

En este nuevo sistema operativo, para poder comunicarnos con el hardware, debíamos hacerlo a través del propio sistema, aunque continuaba permitiendo el acceso directo para mantener la compatibilidad con viejas aplicaciones.

A pesar de todas estas mejoras, el GDI continuaba siendo demasiado lento para el desarrollo de aplicaciones gráficas, por lo tanto, Microsoft decidió crear el **DirectX**.

III EMULADOR DE DOS

Para quienes les interese revivir los juegos de antaño, podrán descargar gratuitamente un emulador de DOS llamado **DosBox** desde www.dosbox.sourceforge.net.

DirectX es un **API** (*Application Programming Interface* o **I**nterfaz de **P**rogramación de **A**plicaciones) que tuvo como fin hacer de Windows un sistema operativo atractivo para la programación de juegos. Debía ser rápida y de nivel suficientemente bajo como para que se pudiera desarrollar cualquier tipo de juegos. A partir de este momento, el desarrollo de videojuegos cambiaría drásticamente.



Figura 6. Uno de los primeros títulos en salir de la mano de Sega Corporation para Windows 95 y DirectX fue **Comix Zone**.

DE LA IDEA A LOS HECHOS

Hacer videojuegos no es fácil, pero tampoco imposible. Es sólo cuestión de pasión y perseverancia. Sin embargo, hay dos variables que conspiran contra esos requisitos: el incentivo y la frustración.

Por lo general, un videojuego comienza con una idea, una bolsa repleta de imágenes en la mente creativa de quien lo concibe, que muchas veces vuela tan alto que



REFERENCIA PARA EL DESARROLLADOR DE VIDEOJUEGOS

Aquellos que deseen investigar más a fondo acerca del diseño de videojuegos, técnicas avanzadas de programación, el API de DirectX y mucho más, deberán conectarse con www.gamedev.net.

luego no le alcanza el combustible para regresar a tierra; inevitablemente, termina por estrellarse con la dura realidad. Cuando no hay dinero de por medio, el tiempo es un condicionante del incentivo. Por eso, es fundamental que el desarrollo de un juego sin fines de lucro culmine a corto plazo o de lo contrario se corre el riesgo de abandonar el proyecto.

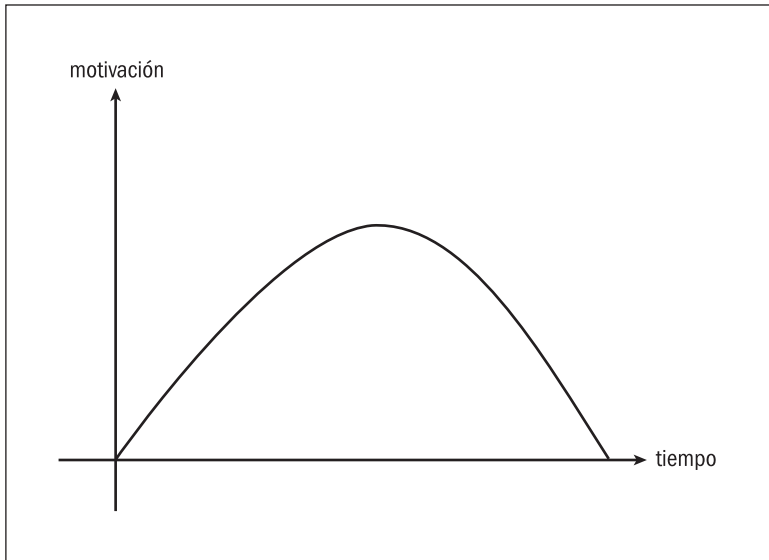


Figura 7. En un principio, la motivación aumenta al ver cómo el proyecto avanza, pero en determinado momento, el tiempo comienza a jugar en contra.

Para combatir la frustración, es importante comenzar con ideas sencillas e ir aumentando la complejidad progresivamente en nuevos trabajos a medida que se adquieren conocimientos. Es imprescindible poner una meta en cada desarrollo. Una vez alcanzada, lo consideraremos finalizado y estaremos listos para encarar un nuevo reto. No es recomendable continuar sobre el mismo proyecto y sumarle complejidad, puesto que de esa manera, probablemente, nunca se logre terminarlo.

El primer paso para poner en marcha la idea es plasmarla en papel y, de esta forma, ver mejor el problema para luego encontrar la solución. Este **documento de diseño** intenta reflejar el concepto del videojuego que se tiene en mente de la manera más detallada, simple y definida posible, con sus límites bien destacados para no caer en el vicio de dilatar el proyecto de manera indefinida.

En el siguiente paso, integraremos un **documento de diseño técnico** compuesto por una explicación minuciosa de todos los aspectos del juego (requerimientos del hardware, si será multitarea o no, si llevará efectos gráficos de algún tipo y cómo serán implementados) y diagramas **UML** (*Unified Modeling Language* o **Lenguaje Unificado de Modelado**) de la estructura del juego. Se trata de diferentes gráficos

destinados a explicar el modo en que se relacionan los módulos de nuestra aplicación entre sí, de forma tal de ver más claramente como será el programa final. Es muy útil en nuestro caso, dado que desarrollamos una aplicación compleja como es un videojuego. Más aún si se trata de un grupo de programadores, dado que sabremos de antemano cómo deberán estar ligados cada uno de los componentes del programa que desarrollará cada parte del equipo.

Por último, se programará siguiendo a conciencia los documentos antes descriptos. Es normal que, en mitad del desarrollo, haya que modificar la arquitectura del juego y, por lo tanto, el documento de diseño técnico, ya sea por limitaciones de la plataforma encontradas durante este proceso o para elegir una mejor solución al problema planteado. También puede ocurrir que, por las mismas razones, haya que realizar algún cambio en el diseño conceptual del juego. En cualquiera de estos casos, **siempre** hay que volver atrás y plasmar la modificación.

No es el objetivo de este libro entrar en detalle acerca de estos documentos, por lo que se recomienda la búsqueda de bibliografía específica.

AUTOPSIA DE UN VIDEOJUEGO

Bisturí en mano, nos adentraremos en las entrañas de un videojuego para así entender las diferencias con la programación de aplicaciones convencionales. Además, veremos de qué partes se compone y finalizaremos el capítulo con algunos consejos útiles antes de comenzar a teclear nuestras primeras líneas de código.

Características de la programación de videojuegos

Como ya vimos, los videojuegos no son otra cosa que programas. Pero hay que aclarar que son programas **especiales**, puesto que tienen algunas diferencias con respecto a la programación de aplicaciones convencionales. La principal reside en su esqueleto: un juego es un bucle que sólo finaliza cuando deseamos cerrar el programa, dado que debe estar en constante actualización, como por ejemplo de la música.



PRIMERA CONSOLA DE VIDEOJUEGOS

La primera consola de videojuegos que se ofreció al público fue la **Magnavox Odyssey**, en el año **1972**, que permitía acceder a juegos similares al conocido **Pong**.

ca, las animaciones, los personajes, el chequeo de colisiones, la inteligencia artificial, la pantalla, etcétera. En cambio, una aplicación como podría ser un sistema de control de stock, por lo general, está a la espera de eventos entrantes y mientras tanto el programa se encuentra en estado **ocioso**.

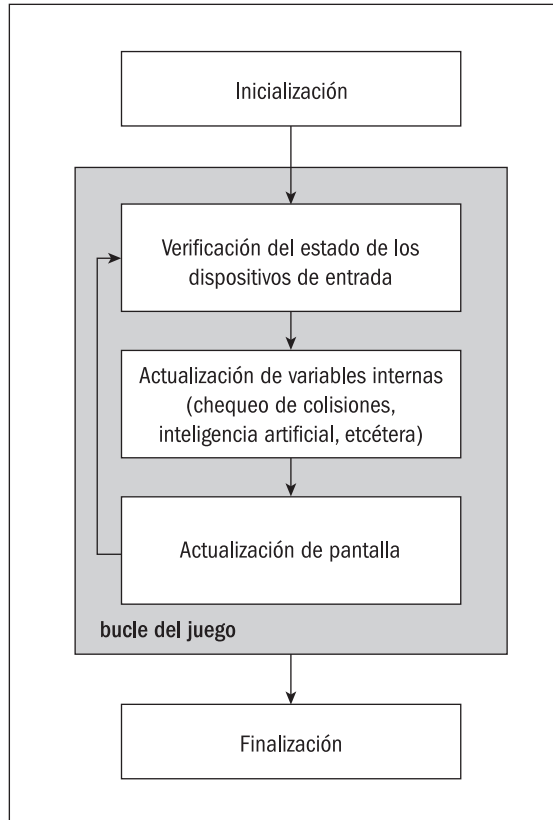


Figura 8. Como podemos ver en el diagrama, un videojuego se mantiene en constante actualización mientras se ejecuta.

Por esta causa, la programación de videojuegos debe mantenerse atenta para optimizar y aprovechar al máximo los recursos, teniendo siempre en mente que la aplicación desarrollada deberá correr a una velocidad aceptable en la mayor cantidad posible de las computadoras existentes en el mercado.

Todo videojuego, de una manera u otra, necesita comunicarse con el hardware para transmitir información al usuario y, de él, recibir comandos. Por ejemplo, cuando el usuario mueve el mouse o presiona una tecla, el programa recibe un evento para actuar en consecuencia, como así también los gráficos que vemos en pantalla, que se logran induciendo al hardware para que éste encienda pixeles de colores en el dispositivo. Para esto, la aplicación deberá comunicarse con la librería DirectX, la cual hará las veces de **punto** entre ésta y el hardware.

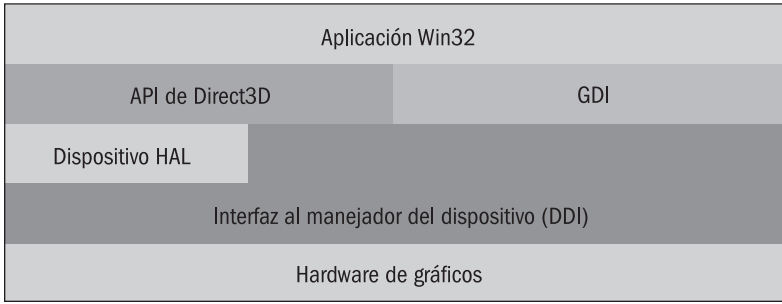


Figura 9. En el diagrama, se puede ver cómo una aplicación se comunica con el hardware a través de las librerías del sistema operativo.

Componentes de un videojuego 2d

Este libro se centrará sobre el desarrollo de juegos bidimensionales y no tridimensionales, dado que si bien tienen componentes en común como los dispositivos de entrada o la música, se basan en tecnologías y lógicas muy diferentes entre sí.

Los videojuegos 2d se caracterizan por representar el mundo y sus elementos a través de figuras bidimensionales planas. Sus componentes elementales son:

- Píxeles
- Bitmaps
- Sprites
- Mapas
- Interfaz

Toda imagen que se muestra en pantalla está compuesta por **puntos** de diferentes colores. A estos puntos, se los denomina **píxeles**. El pixel es la unidad mínima de color programable. Su tamaño físico en pantalla depende de la resolución a la que se trabaje. Estas resoluciones pueden variar, pero las más comunes son **640x480**, **800x600** y **1024x768**.

Los **bitmaps** (o mapas de bits) son imágenes estáticas formadas de píxeles codificados bajo algún patrón que conforman, como base elemental, a los demás objetos antes mencionados. Este patrón o formato de pixel indica cómo debe interpretarse cada unidad de la imagen. Algunos ejemplos de bitmaps son los archivos **bmp**, **png**, **jpg**, **gif**, entre otros.



PRIMERA PATENTE DE UN VIDEOJUEGO

El origen de los videojuegos se remonta al año **1948** cuando la idea de un videojuego como tal fue concebida y patentada por **Thomas T. Goldsmith Jr.** y **Estle Ray Mann**.



Figura 10. El fondo, los personajes, la interfaz y todos los elementos gráficos de un videojuego están formados por bitmaps.

Los **sprites** son los elementos gráficos (generalmente animados) formados por uno o más bitmaps que pueden ser trasladados, rotados, escalados, y que habitualmente producen eventos en la lógica del juego al colisionar unos con otros. Por ejemplo, el Pac-Man es un sprite tanto como lo son los fantasmas. Si uno colisiona con el otro, el programa disparará un efecto que será la muerte de nuestro avatar.

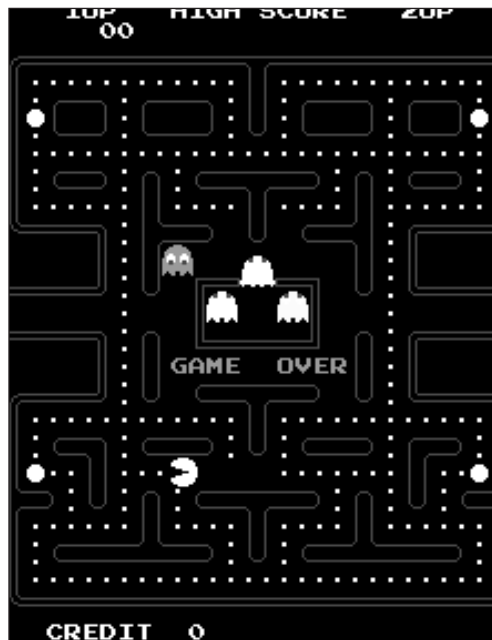


Figura 11. Tanto el Pac-Man, como los fantasmas y las pastillas son sprites que pueden colisionar entre sí y disparar eventos.

Los **mapas** están constituidos por porciones de bitmaps llamadas **mosaicos** (*tiles*) que forman el mundo por el cual se mueven los personajes del juego. Por lo general, se crean a partir de herramientas especialmente concebidas para tal fin, por ejemplo **AnaConda Map Editor** o **Mappy**.

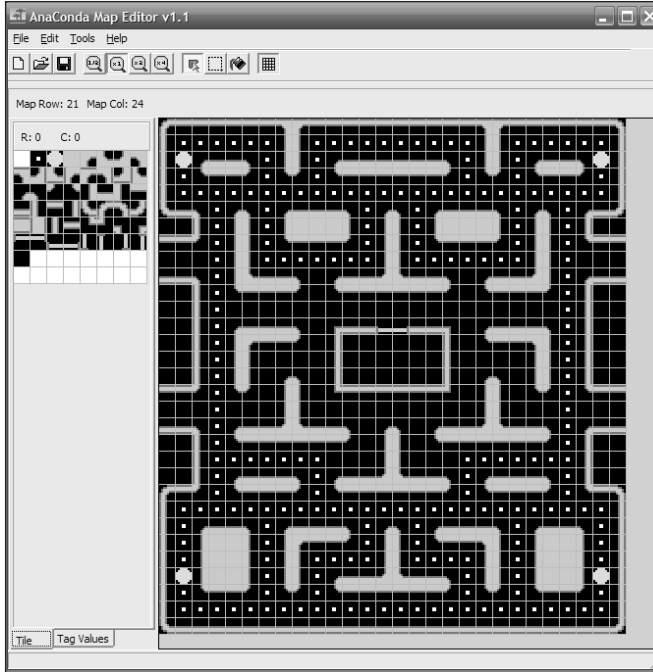


Figura 12. Aquí vemos los mosaicos de un mapa de un juego similar a Pac-Man, editado con AnaConda Map Editor.

ALGUNOS CONSEJOS ÚTILES

Antes de poner las manos sobre el teclado, no está de más mencionar algunos consejos útiles para quienes recién comienzan:

- Tratar siempre de **esquivar de antemano los problemas que no afecten al gameplay**, por ejemplo, si la idea original es que el personaje salte de plataforma en plataforma realizando tres vueltas hacia atrás y esto representa un inconveniente a la hora de programarlo, la mejor solución es que, simplemente, salte hacia delante.
- Si en mitad del desarrollo se encuentra un comportamiento no deseado que sólo se reproduce con una combinación de eventos determinada y conocida, **priorizar**

el evitar que se genere el problema antes que localizarlo y solucionarlo, que por lo general toma más tiempo. Por ejemplo, si un enemigo al movilizarse queda bloqueado por algún elemento en particular, es preferible quitar el obstáculo del nivel antes que reprogramar el comportamiento.

- Es importante que **cada cambio que se realice tanto en los documentos de diseño como en el código fuente se guarden como versiones diferentes** ordenadas por número o por fecha. Esto facilita la vuelta atrás en cualquier momento que sea necesario.

RESUMEN

Ya definimos lo que es un videojuego, sus inicios en PC, las características de su programación, de qué partes se compone y cómo empezar el desarrollo cuando sólo se tiene una idea en mente. En los próximos capítulos, pondremos las manos sobre el teclado para darles forma a nuestros proyectos. ¿Estamos listos?



ACTIVIDADES

1 ¿Qué es un videojuego?

2 ¿Por qué es tan importante la interactividad?

3 ¿Cómo nació DirectX y para qué sirve?

4 ¿Qué diferencia hay entre programar un videojuego y una aplicación convencional?

5 ¿Qué componentes definen un juego 2d?

6 ¿Qué pasos hay que seguir para comenzar el desarrollo de un videojuego?

7 Investigar acerca de los tres tipos de documentos de diseño: High Concept, Game Treatment y Game Script.

8 ¿Qué son los diagramas UML?

9 Investigar los tipos de diagramas UML que existen y para qué son utilizados.
