

Videojuegos de 8 Bits vs Juegos en la actualidad

Práctica 4, Infografía

Dylan Jitton (62732) - UPB 2023

Resumen y Generación de gráficos en 8 Bits

En la época de los 80 's, el desarrollo de videojuegos, y por ende, el proceso de generación de gráficos era muy distinto al que tenemos en la actualidad, comenzando por el hardware. En la época de los 8 Bits, las consolas de aquella generación contaban con: memoria de 4Kb de RAM (2 para el sistema y 2 para video), cartuchos con 32 Kb de código y 8Kb de datos sobre gráficos, y una PPU (Pictures Processing Unit), un equivalente a las GPU que tenemos hoy en día, con la gran diferencia que no podía ser programada directamente, y contaba con distintas memorias RAM dependiendo como genera gráfico, este se separa en 4 secciones: la primera "Pattern Tables", o mesas de patrones, la que contiene los sprites, y puede almacenar hasta 256 8x8 cuadritos o tiles de pixeles, estas memorias generalmente vienen con los cartuchos de los juegos de entonces. La segunda sección son los "NameTables", o el fondo de los juegos, en el cual es una matriz 32x30 en la que, en cada sección representa un cuadrito 8x8 (1 byte por tile). La 3ra sección está la paleta de colores, en la cual define qué colores de la paleta va usar, ya que, en aquel entonces no se podía usar todos los colores. Finalmente, la 4ta sección es "Object Attribute memory" o OAM, controla los gráficos que se mueven al frente, o los sprites.

Estos procesadores estaban enfocados en los televisores antiguos, los CRT, los de tubos de rayos catódicos, en los cuales, recibían su señal y su intensidad gracias a este co-procesador, y dependiendo de la región en la que se vendían, debido a los estándares de CRT del momento, una consola podía ser más lenta que otra dependiendo de donde la consigas.

Además, en la generación de gráficos, no guardaban los colores exactos en la memoria, sino una locación en la que se encuentran para llamarla, la cual sólo podía almacenar 8 paletas en total, y para el fondo, y 4 para los sprites e interactivos en el programa, cada una con 4 colores. El primer color en cada

paleta siempre será interpretado como el color de transparencia, en especial para la generación de animaciones limpias en el sprite.

Al final de cada sección de las memorias, hay una pequeña región llamada Attribute table, la cual indica qué paletas de colores ejecutar en cada momento del juego.

El OAM puede guardar datos de hasta máximo 64 sprites, y cada entrada consiste en 4 bits, el primero consiste en la coordenada Y del sprite, el segundo explica qué tile hay que renderizar, el tercero controla una serie de atributos de cada sprite, y el cuarto, controla la coordenada horizontal, o X

en algunos casos, para simular las físicas de salto, por ejemplo de mario, había que realizar funciones para hacer cálculos más complejos en base a las limitaciones del procesador de la NES, la cual solo podía sumar, restar, multiplicar x2, y dividir x2, pero como esto añadiría espacio que no hay por las limitaciones de memoria del hardware, hay una técnica de “fingir” una caída “realista, o suave” con una función de sub-pixel.

En la detección de colisiones, en juegos donde hay muchos datos que procesar, como por ejemplo, el impacto de múltiples balas en 4 puntos débiles de un enemigo, o el impacto de las balas del enemigo a 2 jugadores en Contra, la solución es “Point vs rectangle”, el jugador es un “círculo” y el enemigo (o bala enemiga) es un rectángulo que varía sus dimensiones o hitboxes dependiendo de la posición que tenga el jugador.

Resumen y Generación de gráficos en 8 Bits

En resumen, hace no más de 40 años, la tecnología estaba bastante limitada a diferencia de la actualidad, si bien los ordenadores personales recién apenas comenzaban a popularizarse, un aspecto en el mundo del cómputo que comenzó a ganar bastante popularidad fueron los videojuegos, que si bien a día de hoy pueden tener una apariencia bastante simple, en su momento, y su proceso de desarrollo podría considerarse como una proeza de ingeniería, ya que generaban un entorno virtual e incluso con elementos interactivos, con una memoria limitada, diseñando elementos de manera recursiva, con varios tipos de memoria específica para cada acción en el juego. Los desarrolladores de aquella época tenían que priorizar la eficiencia de recursos a toda costa, situación que hoy en día no existe.

Actualmente las computadoras han mejorado a tal punto que, desde la generación de gráficos en 3D, incremento de memoria RAM y almacenamiento interno, hacen que la limitación de recursos no sea un problema, o al menos una prioridad masiva. También existen motores gráficos de libre uso como Unity o Unreal que proveen de tecnología de punto, además de físicas elaboradas que proveen de una experiencia mucho más realista en las experiencias multimedia interactivas.

Sin embargo, volviendo al punto de no priorizar la eficiencia de recursos, genera un problema que no era muy notorio hasta recién entonces, con el hecho de que ya no hay límites en las capacidades computacionales, en especial con los gráficos RTX, o de Ray Tracing, los juegos actuales cada día se hacen más pesados en términos tanto de almacenamiento como de capacidades de procesamiento, no solo porque los juegos actuales llegan a pesar hasta 170 Gb de almacenamiento interno, sino porque muchos juegos solicitan como requerimiento mínimo tarjetas gráficas de cada vez mejor capacidad, y este problema se denota aún más considerando el precio excesivo de tarjetas gráficas de última generación, debido a la inflación y otros factores, por lo que, se obtienen mejores gráficos, pero cada vez menos accesibles para cualquier persona, (sin mencionar el incremento que sufrieron las consolas de última generación, y los efectos de una escasez en la producción de procesadores y semiconductores a nivel global).