

Taller Clase 004: Arqueología Digital

GR1SW

06 DE NOVIEMBRE DEL 2025

JUEGOS INTERACTIVOS

MsC. VICENTE EGUEZ

EQUIPO

Said Luna
Damaris Suquillo



**ESCUELA
POLITÉCNICA
NACIONAL**

Ficha de Análisis de Hito

Era Asignada:

1994 - 2000 (Consolas 3D, CD-ROM, Aceleradoras 3D)

Juego Seleccionado:

Crash Bandicoot (1996, PlayStation 1)

1. Hito Tecnológico Clave:

El almacenamiento masivo en CD-ROM y la Unidad de Procesamiento Geométrico (GTE) del PlayStation. El salto de los cartuchos (pocos MB) a los CD-ROM (650-700 MB) permitió almacenar texturas de alta calidad y audio sin comprimir, mientras que la GTE permitía cálculos matemáticos vectoriales rápidos necesarios para renderizar polígonos 3D en tiempo real, aunque con una memoria RAM extremadamente limitada (2MB).

2. Análisis de Diseño (MDA):

- **Mecánicas (M):**

- **Plataformeo en profundidad:** A diferencia del desplazamiento lateral 2D tradicional, el jugador corre hacia el fondo de la pantalla, requiriendo cálculo de saltos en el eje Z.
- **Ataque de giro:** Una mecánica ofensiva que también sirve para interactuar con cajas y físicas de rebote.
- **Ritmo forzado:** Niveles donde la cámara o un obstáculo persiguen al jugador, forzando decisiones rápidas sin gestión de cámara manual.

- **Estéticas (A):**

- **Desafío:** El juego evocaba una dificultad alta basada en precisión y memoria muscular.
- **Sensación:** La estética visual de dibujo animado jugable que era visualmente superior a casi todo lo que había en consola en ese momento.

3. Innovación Clave (El "Salto"):

Logró gráficos de alta fidelidad y densidad poligonal superior a Super Mario 64, pero sacrificando la libertad de movimiento. Mientras Mario 64 apostó por la exploración abierta con texturas simples por limitaciones de memoria del cartucho de N64, Crash Bandicoot innovó creando el 3D lineal o corredor. Esto demostró que se podía tener una experiencia visualmente densa y rica si se controlaba estrictamente lo que la cámara podía ver.

4. La "Restricción Ingeniosa" (El Desafío de Ingeniería):

Restricción:

La PlayStation 1 tenía solo 2 MB de memoria RAM. Los desarrolladores querían niveles con una densidad de polígonos, texturas y animaciones que excedían por mucho esos 2 MB. Si intentaban cargar un nivel completo como se hacía tradicionalmente, la consola se quedaba sin memoria antes de renderizar la mitad del escenario. Además, el lector de CD era lento para buscar datos.

Solución (El "Hack"):

Paginación dinámica y oclusión agresiva. Los ingenieros Andy Gavin y Jason Rubin crearon un sistema de paginación virtual que iba en contra de las librerías estándar de Sony.

1. **Carga por Chunks:** Dividieron los niveles en piezas pequeñas de 64KB. A medida que Crash corría hacia adelante, el motor borraba de la RAM los polígonos que quedaban atrás y cargaba del CD los que venían adelante exactamente antes de que fueran visibles. El juego estaba constantemente leyendo del disco lo que era algo arriesgado para el hardware de la época.
2. **Cámara pre-calculada:** Como la cámara no era libre, el jugador no podía rotarla, el motor sabía exactamente qué polígonos nunca se verían. Usaron un algoritmo para pre-calcular la visibilidad y eliminar cualquier objeto oculto tras una pared o fuera del ángulo de cámara, liberando ciclos de CPU para dibujar más detalles en lo que sí se veía.
3. **Hack de Lenguaje:** Crearon su propio lenguaje de programación, GOOL - Game Oriented Object LISP, para gestionar la lógica de los objetos de manera más eficiente que el lenguaje C estándar de la consola, empaquetando el código de manera compacta.