Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistemas Operativos (0840)

Grupo: 04

Exposición: Optimización de Recursos en Videojuegos

López López Ulysses

Profesor: Ing. Gunnar Eyal Wolf Iszaevich

Fecha: 29 de Octubre de 2019

Optimización de recursos en videojuegos

A lo largo de la historia de la computación, los videojuegos han sido una parte fundamental en éste, ya que los videojuegos llegan a usar el máximo de recursos de las computadoras. Esto conlleva a que la memoria RAM se llene de datos a procesar y en sí, la máquina irá más lento, por lo que la experiencia de usuario será muy mala y esto en vez de generar una buena experiencia, el usuario no querrá jugar más.

A lo largo de la historia de los videojuegos, los desarrolladores de estos han tenido que desarrollar técnicas, trucos y métodos por los cuales se busca optimizar lo máximo de recursos para ofrecer al usuario una experiencia única debido a que las consolas y computadoras han estado limitadas en sus épocas y muchos buscan desarrollar cosas innovadoras pero para ello se han encontrado con problemáticas y es por eso que las siguientes técnicas que se presentarán, han sido utilizadas por muchos videojuegos y desarrolladores para que sus videojuegos corran lo más rápido posible y también ahorrar lo máximo de recursos posibles pero sin afectar la experiencia de usuario.

Antes de empezar a habalr sobre los distintos métodos, se tiene que hablar de una cosa muy importante sobre los videojuegos, la renderización. Básicamente el renderizado es la operación de por la cuál los elementos en la memoria son dibujados físicmente en la pantalla. Para ello tenemos que ver cómo funciona la memoria en un videojuego:

Administrador de Memoria							
M	E	T	A	D	A	T	A

MEMORIA
Vida
Posicion
Vida Enemigo
Pos. Enemigo
Personaje
Personaje
Enemigo
Enemigo
Escenario
Escenario
Escenario

Al crear el videojuego, lo primero que se tendrá es un sistema de administrador de memoria, el cuál solicitará espacio de memoria para que, posteriormente, se carguen los datos de los objetos del juego mediante los metadatos que los objetos tienen, ya que estos nos indicarán que son y cuánto espacio requieren para poder implementarse en memoria y posteriormente, se puedan renderizar o procesar, dependiendo si son obejtos que se mostrarán o sólo datos como vida, la posición, el número de objetos, físicas, etc.

Hay que tener en cuenta que los datos de los videojuegos varian, ya que como bien se sabe, hay videojuegos que ocupan un formato en 2D y otros en 3D, por lo que no es lo mismo hablar de un sprite que es en cierto modo una imagen, a un modelo 3D con polígonos y textura que engloban más espacio de memoria.

Así mismo hay que tener en cuenta que la memoria es muy pequeña e insuficiente para mantener todos los datos en ella, ya que eso provocará sobresaturación y que el juego empiece a verse muy lento debido a tantas cosas por procesar y renderizar, por lo que se tiene que ocupar lo menos posible para que el juego corra bien, pero también que el juego no se vea afectado de manera gráfica, ya que se le quiere dar al jugador una buena experiencia. Y es por eso que se han realizado varios métodos para optimizar recursos.

Estos métodos se pueden categorizar en 4 familias, las cuales son:

1.- POP-PUSH

Pop

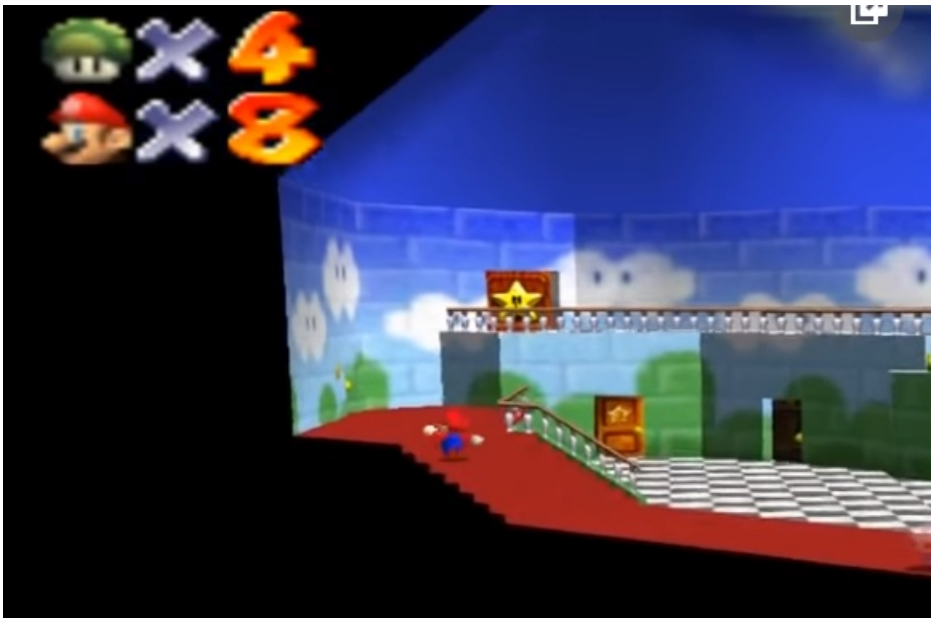
Este método consiste en que, en determinado punto, las coordenadas del jugador estén lo demasiado lejos para no ver o simplemente, el rango de visualización de la cámara no esté al alcance de los objetos, es mejor hacer un POP para poder ahorrar recursos y asignarle estos a algo más importante para el jugador.

Push

El método pushing, utiliza el mismo principio que el poping, solo que en esta ocasión se utiliza cuando el rango de visualización de la cámara está muy cerca, los objetos que están a punto de mostrar en pantalla hacen push, así el jugador no verá que se cargaron los objetos pensando que siempre han estado ahí.

Puntos de carga

Para este método se carga un gran conjunto de elementos y de escenario, en el cuál el jugador se desplazará por él sin cargar más elementos, por lo que el número de recepción de eventos para hacer push y pop se reduce a 1 punto, el punto de carga, este, al llegar el jugador a dicho punto carga lo que sigue del siguiente escenario y hace pop al anterior y en caso de que regrese, se tendrá que cargar de nuevo.



2.- VISTAS

Modelados optimizados

El método consiste principalmente en no detallar los modelos o tenerlos hechos sólo para su propósito, es decir, con partes faltantes del modelo, esto principalmente se da en escenarios o en objetos muy alejados que en jugador no percibe por la distancia que hay entre la posición del jugador y el objeto. En los escenarios principalmente se da en las zonas inaccesibles al jugador o en bordes donde no puede llegar el jugador.

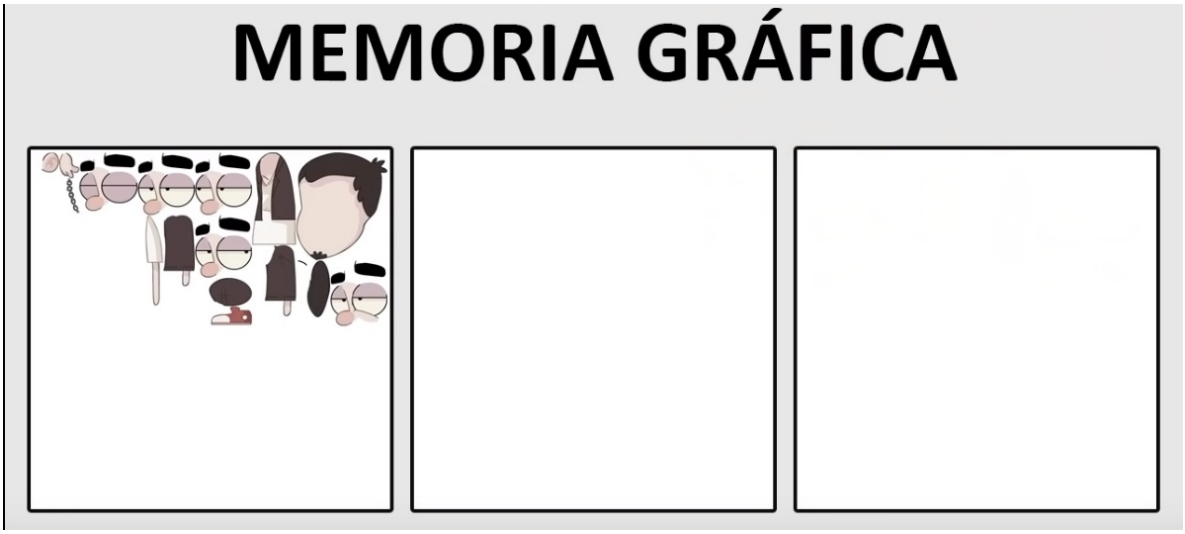
Texture Mapping

Este método consiste en que los elementos del mapa, escenario ya están fijos, lo único que no están cargados son las texturas. A medida que el jugador va explorando las zonas, las que la cámara está visualizando se carga la textura mientras que en las que no está viendo, estas sólo son los objetos sin textura.

3.- SPRITES

Atlas de texturas

Este método consiste en ocupar una imagen con todos los modelos de un personaje y dividirla en sectores, por lo que al ir cambiando de animación, en vez de cargar otra imagen, sólo se va colocando la misma imagen, pero sector diferente, por lo que se ahorra mucha memoria.



Skyboxes

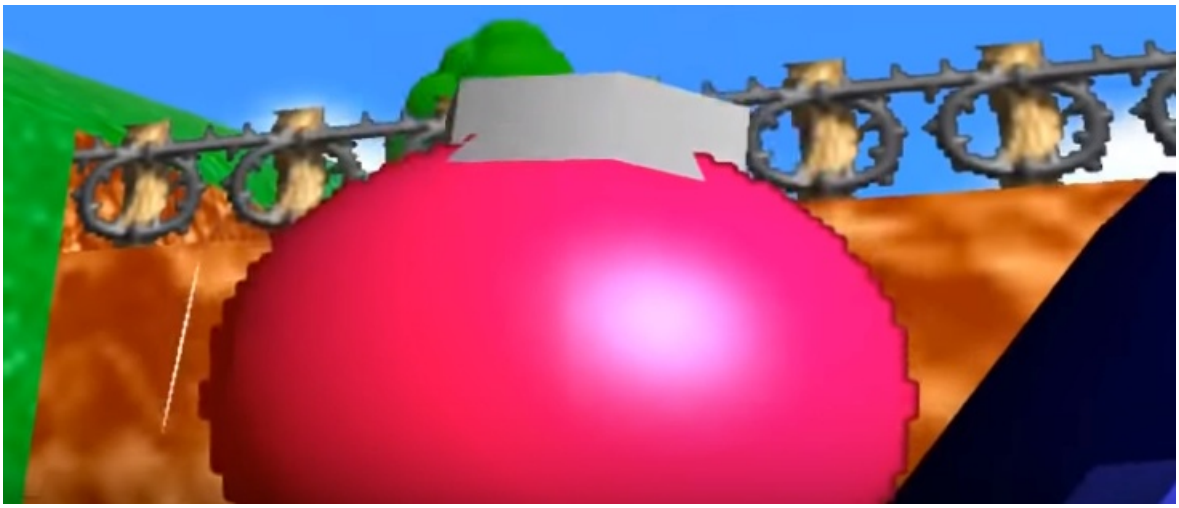
Estos son recursos ayudan a no implementar un fondo o un escenario con tantos recursos y así, obtener la misma experiencia, consiste en colocar una imagen con la resolución adecuada para que no se vea pixelada o que no se vea “rara” y así mostrar un fondo adecuado y sin que el juego se muestre lento.

Sprites repetidos

Por lo general este tipo de método se utiliza más en juegos de dos dimensiones, se ocupa para optimizar memoria y no gastar tantos recursos en varios modelos, lo único que se hace es cambiar el color de paleta para el objeto en común y así ya se obtienen varios objetos o hasta personajes.

Sprites “3D”

Este método consiste en jugar con la perspectiva del usuario para presentarle un objeto en “3D”, el Sprite en vez de estar fijo, siempre estará en dirección de la vista de la cámara principal, por lo que se pretende visualizar que es un objeto de tres dimensiones y así no se gasta tantos recursos en ello, sirve principalmente para objetos que se repiten mucho y tienen el mismo modelo, así no se gasta tanta memoria para generar muchos objetos tridimensionales.



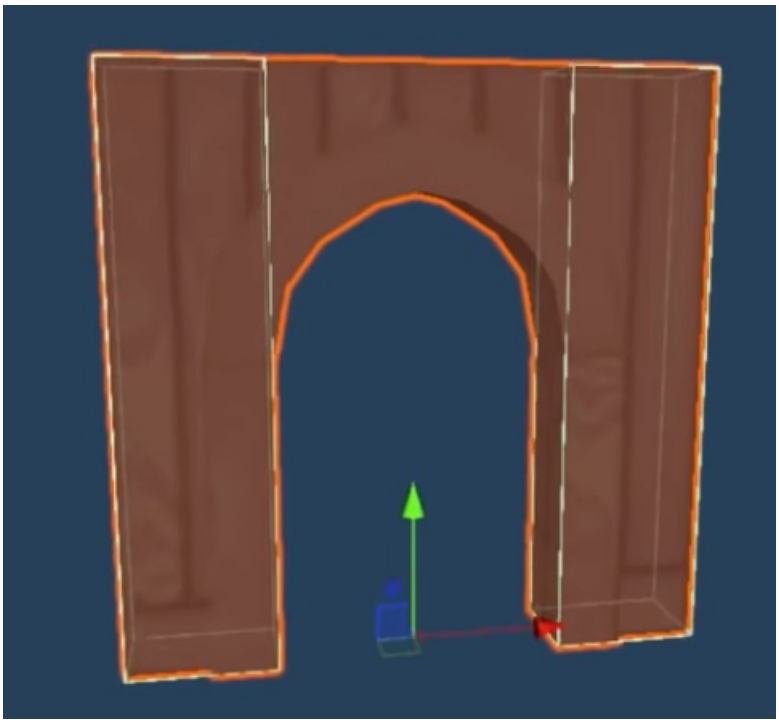
4.- OTROS

Proyectiles

Principalmente se utiliza para las balas de armas de fuego, ya que, al ser tan veloces casi ni se ven, por lo que se opta por sólo calcular la trayectoria de la bala y ver en donde colisiona para poner una animación de impacto, pero no se agregan partículas de balas porque esto gastaría más recursos.

Box collider

Los box collider son muy útiles al desarrollar un videojuego, sin embargo, utilizar muchos harán que el juego no sea optimo, por lo que se tiene que analizar la forma del modelo y reducirlo a una geometría más compacta, por lo que se tiene que definir cómo serán usados los Box Collider.



Tiempo después se crearon los motores de videojuegos que buscan hacer juegos de calidad con una administración de recursos optima, pero ese es otro tema aparte ya que se necesita más profundización en este tema ya que en un sólo párrafo es imposible cubrir todo lo que conlleva.

La relación que conlleva este tema de exposición con la asignatura de sistemas operativos es sobre el manejo de memoria y cómo se han creado varias técnicas y trucos para poder llevar a cabo una administración optima para que no se ocupe mucha memoria y así poder disfrutar de una jugabilidad buena para el jugador.

Referencias

1.- https://www.youtube.com/user/GuinxuVideos

La mayoría de la investigación ha sido por experiencia propia al desarrollar videojuegos.