

Orbit Camera

Gráficas por Computadora (CC2018) - 2026

Orbit Camera

Semestre 02, 2026

Objetivo

- Distinguir los dos tipos de cámara con los que se trabaja en un motor y ubicar dónde encaja la cámara orbital.
- Describir una cámara con tres vectores, eye, center y up, en lugar de con ángulos sueltos.
- Construir la base de la cámara a partir de esos tres vectores usando producto cruz.
- Entender el cambio de base como el paso de coordenadas de mundo a coordenadas de cámara.
- Mover la cámara con ángulos de Euler y controlar sus límites con clamping.
- Convertir coordenadas esféricas a cartesianas para obtener la posición final de la cámara.

Tipos de cámara

Existen dos tipos de cámara con los que podemos trabajar, y la diferencia entre ambos está en dónde se coloca el observador respecto al personaje.

Free cam

- Es una cámara en primera persona, donde el observador y el personaje ocupan el mismo punto.
- Google Earth funciona así, porque el usuario mueve libremente el punto desde el que mira.
- Es similar al raycaster del maze, donde la cámara es la posición del jugador dentro de la cuadrícula.

Tercera persona

- La cámara sigue al personaje principal desde afuera, manteniéndolo siempre a la vista.
- Lakitu, el personaje que carga la cámara en Super Mario 64, es la representación literal de esta idea.

Lakitu cargando la cámara que sigue a Mario

La cámara orbital es la forma más común de implementar una cámara en tercera persona, y también es la que usan casi todos los editores 3D.

La cámara orbital

Una cámara orbital rota alrededor de un punto objetivo manteniendo una distancia fija, que llamamos radio.

Su movimiento está restringido a la superficie de una esfera alrededor del objetivo. La cámara puede subir, bajar y girar, pero nunca se acerca ni se aleja a menos que cambiemos el radio de forma explícita.

La analogía directa es la órbita de la Tierra alrededor del Sol. El planeta se mueve constantemente, pero siempre a la misma distancia del punto que orbita.

Esa restricción es justamente lo que la hace útil. El objetivo nunca se sale de cuadro, porque la cámara está construida para no perderlo de vista.

Eye, center y up

Una cámara se describe con tres valores, y los tres son vectores.

- Eye: la posición de la cámara en el mundo.
- Center: el punto objetivo, es decir lo que la cámara está mirando. En un juego en tercera persona, es el personaje.
- Up: el vector que indica hacia dónde queda el arriba de la cámara.

El uso de vectores es clave, porque es mucho más manejable que la contraparte, que sería describir la cámara con ángulos sueltos. Con vectores podemos aplicar operaciones entre ellos mismos para encontrar y calcular los valores que nos interesan.

El vector up merece una aclaración. No es el arriba definitivo de la cámara, sino una referencia que le dice al sistema qué dirección considerar como vertical. A partir de esa pista se calcula el arriba real, que sí es perpendicular a la dirección de visión.

Proyecciones

Una proyección es el proceso mediante el cual convertimos una escena 3D en una imagen 2D. Para lograrlo vamos a usar cambios de base, que es cambiar el sistema de coordenadas en el que viven nuestros vectores.

Todo se hace en función de la perspectiva de la cámara, porque lo que se dibuja en pantalla no es la escena en abstracto, sino la escena vista desde un punto concreto.

El mismo vector expresado en dos bases distintas

Un cambio de base no mueve nada. El vector morado no cambió, lo único que cambió es el espacio en el que lo estamos describiendo. Sus componentes son distintas porque los ejes contra los que se miden son otros.

Las nuevas coordenadas son relativas a la posición de la cámara

El objetivo es determinar los valores de la nueva base, y esas nuevas coordenadas deben ser relativas a la posición de la cámara. Una vez que un punto está expresado en el espacio de la cámara, proyectarlo a la pantalla es directo.

Los ejes de la cámara

Vamos a calcular los tres ejes que forman la nueva base. Los tres salen de eye, center y up, y los tres se normalizan.

Eje Z

Es el valor más fácil de calcular. Se obtiene restando center menos eye, que es el vector que va de la cámara hacia el objetivo, también llamado forward.

El signo tiene significado. En la convención de cámara que usamos, el eje Z de la base apunta hacia atrás, en sentido contrario a la dirección de visión, así que la cámara mira hacia el Z negativo.

Eje X

Es el eje que apunta a la derecha de la cámara, y sale del producto cruz entre el forward y el vector up.

La regla de la mano derecha determina la dirección del producto cruz

El producto cruz devuelve un vector perpendicular a los dos operandos, y la regla de la mano derecha es la que nos dice hacia qué lado apunta ese resultado. El orden importa, porque invertir los operandos invierte el sentido del vector.

Eje Y

Es el arriba real de la cámara, y se obtiene con otro producto cruz, esta vez entre el eje X y el forward.

Este paso es el que corrige el up que dimos como referencia. Aunque el up original no fuera exactamente perpendicular a la dirección de visión, el resultado de este producto cruz sí lo es, así que la base queda ortogonal.

Resumen de la base

- Forward, que da el eje Z: $\text{normalize}(\text{center} - \text{eye})$
- Right, que es el eje X: $\text{normalize}(\text{cross}(\text{forward}, \text{up}))$
- Up real, que es el eje Y: $\text{cross}(\text{right}, \text{forward})$

Con los tres ejes calculados y normalizados tenemos una base ortonormal, que es todo lo que hace falta para expresar cualquier punto del mundo en coordenadas de cámara.

Movimiento de la cámara

Para agregar realismo a nuestras escenas vamos a agregar movimiento, y en este caso el movimiento viene desde la cámara y no desde los objetos.

La cámara se desplaza alrededor del objetivo

Usaremos ángulos de Euler, que describen la orientación de un objeto en un espacio 3D mediante rotaciones sobre los tres ejes. Los tres ángulos siempre se manejan en radianes.

- Yaw, que es la rotación sobre el eje Y.
- Pitch, que es la rotación sobre el eje X.
- Roll, que es la rotación sobre el eje Z.

Yaw

El yaw es la rotación de la cámara sobre el eje Y, que es el eje vertical.

Vista top-down del yaw, donde el movimiento se ve como una órbita

Si imaginamos la escena desde arriba, el movimiento se ve claramente como una órbita alrededor del objetivo. Desde ese punto de vista el desplazamiento ocurre en los ejes X y Z, mientras que la altura se mantiene constante.

Para calcular el ángulo necesitamos el radio, que es la magnitud del vector que va del center al eye. Ese vector es el que la cámara está orbitando.

El ángulo se puede representar como el arcotangente de Z sobre X, que en la práctica se simplifica usando la versión de dos argumentos, con el componente Z y el componente X del radio. Esa versión es la que sirve, porque resuelve el signo correcto en los cuatro cuadrantes.

Pitch

El pitch es la rotación sobre el eje X, el que apunta a la derecha de la cámara. Es el movimiento de mirar hacia arriba y hacia abajo.

El pitch convierte la órbita plana en una esfera

Con el yaw solo teníamos una órbita plana, un círculo alrededor del objetivo. Al agregar el pitch esa órbita se convierte en una esfera completa, porque ahora la cámara también puede subir y bajar.

El cateto adyacente aquí no es todo el eje X, sino la proyección del radio sobre el plano horizontal. Por eso el ángulo sale del arcoseno del componente Y del radio dividido entre la magnitud del radio.

Roll

El roll es la rotación sobre el eje Z, el eje que apunta en la dirección de visión. Es el movimiento de inclinar la cabeza hacia un hombro.

Las tres rotaciones sobre los ejes de la cámara

En este curso no lo vamos a volver a mencionar. Una cámara orbital no necesita roll, porque el vector up mantiene el horizonte nivelado, y en la mayoría de los juegos inclinar la cámara solo marea al jugador.

Clamping

Como es posible pasarse de los valores válidos al acumular rotaciones, hace falta limitar los ángulos, y cada uno se limita de forma distinta.

El yaw es cíclico, porque girar 360 grados deja la cámara donde empezó. Si sumamos 20 grados a 350 grados obtenemos 370 grados, que no es un valor válido, así que se aplica un módulo para traerlo de vuelta al rango.

El pitch no es cíclico. Casi ningún juego permite girar 360 grados en la vertical, porque la cámara terminaría de cabeza y el vector up se invertiría. Lo normal es limitarlo a 90 grados hacia arriba y menos 90 grados hacia abajo, y esos extremos ni siquiera se tocan del todo, porque justo ahí el forward queda paralelo al up y el producto cruz colapsa.

De esféricas a cartesianas

Hasta este momento describimos la cámara con coordenadas esféricas, que son un radio y dos ángulos. Para colocarla en el mundo hay que convertir esa descripción a coordenadas cartesianas.

$$x = r \cos(\varphi) \cos(\theta)$$

$$y = r \sin(\varphi)$$

$$z = r \cos(\varphi) \sin(\theta)$$

- r es el radio, es decir la distancia entre la cámara y el objetivo.
- θ es el yaw.
- φ es el pitch.

El resultado de esas tres fórmulas es un desplazamiento respecto al origen, no una posición absoluta. Para obtener la posición final de la cámara hay que sumarlo al punto objetivo.

Esta es la razón por la que la cámara nunca pierde el objetivo. Su posición se calcula siempre a partir del center, así que el objetivo queda en el eje de la órbita por construcción.

Cambio de base

En gráficas por computadora trabajamos con dos sistemas de coordenadas al mismo tiempo.

- Coordenadas de mundo, que son posiciones y direcciones absolutas dentro de la escena.
- Coordenadas de cámara, que son posiciones y direcciones relativas a la cámara.

Los ejes de la cámara son los que conectan ambos sistemas.

$\text{\text{Right}} \rightarrow \text{\text{apunta a la derecha de la cámara}}$

$\text{\text{Up}} \rightarrow \text{\text{apunta hacia arriba desde la cámara}}$

$\text{\text{Forward}} \rightarrow \text{\text{apunta en la dirección de visión}}$

Si tenemos un vector expresado en coordenadas de cámara, podemos pasarlo a coordenadas de mundo como una combinación lineal de los ejes de la cámara.

$\mathbf{v}_{\text{world}} = v_x \cdot \mathbf{right} + v_y \cdot \mathbf{up} + v_z \cdot \mathbf{forward}$

Cada componente del vector dice cuánto avanzar a lo largo de cada eje de la cámara. Sumando esos tres desplazamientos se obtiene el vector en el espacio del mundo.

La operación inversa, que es pasar de mundo a cámara, se hace proyectando el vector sobre cada eje con producto punto. Como la base es ortonormal, esa proyección es todo lo que hace falta y no hay que invertir ninguna matriz.

En el proyecto de gráficas

La cámara orbital es la que nos va a permitir mirar la escena del raytracer desde cualquier ángulo.

- El eye pasa a ser el origen de todos los rayos primarios.
- Los ejes right, up y forward definen hacia dónde apunta cada rayo al atravesar su píxel.
- El center mantiene la escena centrada mientras la cámara se mueve a su alrededor.
- Los ángulos de yaw y pitch son los que se conectan al teclado o al mouse para mover la vista en tiempo real.

El resto del raytracer no cambia. Sigue lanzando un rayo por píxel y buscando el impacto más cercano, solo que ahora ese rayo nace de una cámara que se puede mover.