

## Recursos Visuales del Videojuego

### Relación de Aspecto (Aspect Ratio)

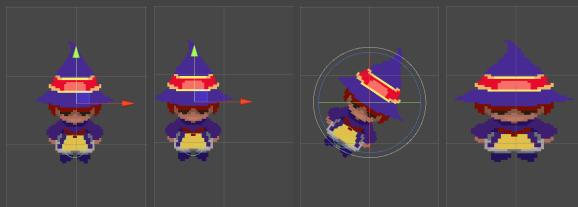
Es una medida que describe la proporción entre el ancho y alto de una imagen o pantalla. Se expresa como una relación numérica, indicando cuántas unidades de ancho hay por cada unidad de alto. Define la forma y la apariencia visual de las imágenes, pantallas, videos y otros elementos visuales. Es importante definir esta relación en base a las dimensiones de la pantalla del dispositivo al que está dirigido (celulares, computadoras, televisores). Generalmente se expresan de dos maneras:

- Se expresa como dos números separados por dos puntos (ancho:alto). Por ejemplo 16:9, que indica que la imagen tiene 16 unidades de ancho por cada 9 unidades de alto.
- Se expresa como un valor decimal que representa la división del ancho por el alto (ancho/alto). Por ejemplo, la relación 16:9 se expresa como 1.78 (16 dividido por 9).

### Ejes cartesianos: posición, rotación y escala.

Los motores gráficos, de videojuegos y de modelado 3D utilizan los ejes cartesianos para indicar la posición, rotación y escala de los objetos en el entorno.

- Posición: El eje X representa la posición horizontal, desde izquierda (-X) a derecha (+X). El eje Y representa la posición vertical, desde abajo (-Y) hasta arriba (+Y). El eje Z representa la profundidad, desde cerca (-Z) hasta lejos (+Z). Cuando ubicamos objetos en un entorno, asignamos valores numéricos a estos ejes para definir su posición en el espacio 3D.
- Rotación: La rotación se describe en grados alrededor de cada eje. Rotar sobre el eje X gira el objeto como si estuviera rodando hacia adelante o hacia atrás, como una rueda de auto. Rotar sobre el eje Y simula un giro hacia los laterales, como un trompo. Rotar sobre el eje Z da la sensación de un giro similar al de un timón o de un reloj.
- Escala: La escala determina el tamaño relativo del objeto en cada eje. Un valor de escala de 1 en todos los ejes mantiene el tamaño original. Valores mayores aumentan el tamaño, mientras que valores menores lo reducen. Aumentar en el eje X es hacerlo más ancho, aumentar en el eje Y es hacerlo más largo.



## Sprite

Un sprite es una representación gráfica bidimensional (2D), donde cada píxel en el mapa de bits determina el color de la imagen. Se representan en un rectángulo y son parcialmente transparentes, para asumir otras formas. Algunas buenas prácticas para lograr la mejor optimización gráfica son:

- Utilizar formatos de imagen comprimidos como PNG para reducir el tamaño del archivo, sin sacrificar la calidad. Además de que admite canales alfa (transparencia).
- Ajustar las dimensiones del sprite al tamaño real que se necesita. No dejar grandes márgenes de píxeles transparentes o en blanco. Esto reduce el tamaño del archivo y mejora la eficiencia del almacenamiento en la memoria.
- Es importante que las dimensiones del sprite sean potencia de dos (por ejemplo 64x64 pixeles, 128x128 pixeles) y en lo posible cuadradas o con el mismo aspect ratio que se utiliza en la cámara del juego (Algunos ejemplos son: 16:9, 4:4, 9:20). Esto acelera el procesamiento de la GPU sobre el sprite por su funcionamiento a nivel de hardware y de algoritmos de renderización. Además, las dimensiones en potencias de dos (POT) son beneficiosas para el mipmapping, un proceso que crea versiones en diferentes niveles de detalle de una textura o sprite. Otro proceso que se beneficia de esta práctica es el proceso de Batching, donde se combinan varios sprites en un solo draw call (llamado de renderización gráfica).
- Habilitar mipmaps (desde el motor gráfico) en los sprites para mejorar la calidad visual frente a diferentes distancias o porcentajes de pantalla. Esto significa que el sprite reduce la cantidad de píxeles (su calidad) en función al porcentaje de pantalla que ocupa (simulando una distancia).
- Modos de filtrado (desde el motor gráfico): los sprites pueden tener filtrado bilineal o trilineal, lo que afecta la calidad de la imagen cuando se escala o cuando se visualizan desde ángulos oblicuos. El filtrado bilineal es una técnica de interpolación horizontal y vertical que calcula el color de un píxel entre cuatro píxeles vecinos en un sprite (ayuda a dar una apariencia suave y menos pixelada). El filtrado trilineal, agrega la interpolación en la dimensión de los niveles de mipmapping (lo que reduce aún más la transición entre los niveles de detalle y a reducir artefactos visuales - ver tema que explica qué y cuáles son-).
- Realizar un “Atlas de Texturas” (Spritesheet) con las imágenes que corresponden a una misma escena o personaje. Esta técnica combina múltiples sprites en un único sprite. Reduce el número de llamadas de renderizado, mejorando la eficiencia del proceso (lo que se conoce como draw call).

Ejemplos de Sprites:

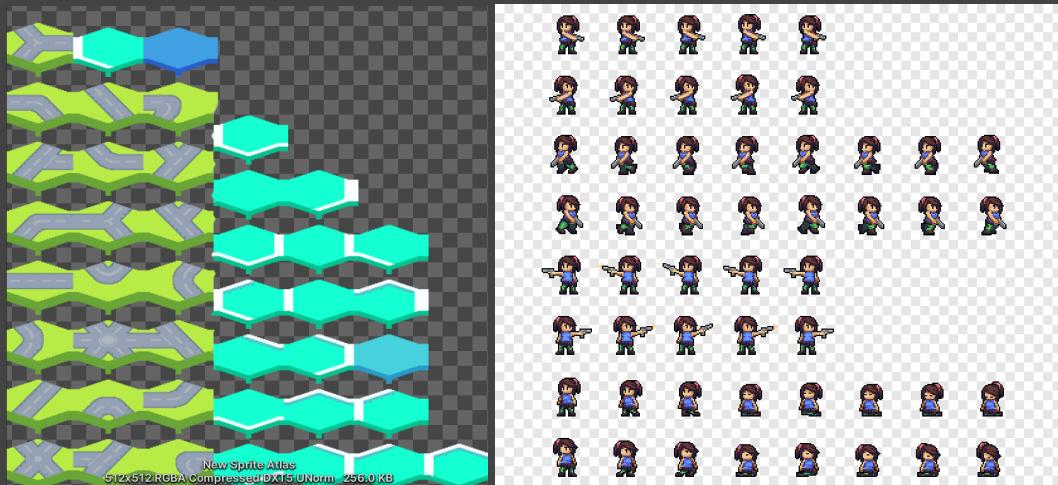


## Spritesheet

Es lo que se conoce como un atlas de sprites o texturas. Deben respetar las mismas buenas prácticas que cualquier sprite. Agrupar sprites sirve para optimizar el proceso de renderizado de la imagen. Es por eso que además de las anteriormente mencionadas, también se siguen las siguientes buenas prácticas:

- Agrupar sprites que estén relacionados entre sí (como animaciones de un mismo personaje u objetos de una misma escena) para minimizar los cambios de textura durante el renderizado. Se puede utilizar herramientas en Unity como el Sprite Packer (que se va a aprender a usar en el módulo 5).
- Para que la agrupación sea aún más eficiente, los sprites deben tener una misma resolución y dimensión.
- Mantener unidades de alineamiento. En Unity, por defecto, una unidad equivale a un metro. Esto significa que si se define un objeto con una escala de 1 en ancho y 1 en alto (no hablamos de píxeles, sino de unidades), tendrás un objeto de un metro de tamaño en todas las direcciones. Es necesario mantener esa escala para todos tus sprites. En cuanto a la cantidad de píxeles por unidad de Unity, depende de la configuración de la cámara y de la resolución de la pantalla. La cámara tiene un tamaño, que representa la mitad de la altura visible en unidades. Por ejemplo, si el tamaño de la cámara es 5, la altura visible es de 10 unidades. Entonces, si la resolución de la pantalla es de 720 píxeles de altura, cada unidad tendrá 72 píxeles ( $720/10$ ). A partir de ello, se pueden calcular los tamaños de los objetos y se debe lograr que la diferencia de tamaños sea realista. Es decir, si un personaje mide un metro y otro dos metros de altura, el primero deberá tener 72 píxeles de alto y el segundo  $72 \times 2$  píxeles de alto (144 píxeles).
- Organizar los sprites en una grilla, donde a cada celda le corresponda un sprite. Esto es ideal para tilemaps. Si se tiene objetos de diferentes tamaños (respetando la regla de la potencia de dos), por ejemplo en objetos de decoración, se puede usar el Sprite Packer para la optimización del atlas.
- En Unity, desactivar el ajuste automático de resolución en las importaciones de texturas. Esto evita que Unity cambie el tamaño de tus Spritesheets de manera automática y potencialmente cause una pérdida de calidad.

Ejemplo de Spritesheet:



## Artefactos Visuales (Artifactoring)

Se refiere a imperfecciones, distorsiones o anomalías no deseadas que pueden aparecer en una imagen. Son el resultado de varios factores y problemas durante el proceso de captura, procesamiento o renderizado de las mismas. Como ser:

- Aliasing: Los bordes dentados o escalonados que se producen cuando se presenta una imagen en una resolución demasiado baja para capturar detalles finos. El antialiasing es una técnica para reducir estos artefactos.
- Moiré: Son patrones de interferencia visual que pueden aparecer cuando se superponen patrones regulares, como líneas o rejillas. El resultado es otro patrón con un aspecto peculiar.
- Blooming: Un halo de luz que se extiende más allá de los límites de un objeto luminoso. Se utiliza para efectos visuales, pero blooming excesivo puede afectar negativamente a la calidad visual.
- Ghosting: Imágenes residuales o “fantasmas” que aparecen cerca de objetos en movimiento rápido, a menudo relacionados con problemas de tiempo de respuesta de pantallas o monitores.
- Comet Trails: Rastros o “colas” dejadas por objetos en movimiento rápido. Puede deberse a problemas de persistencia de la visión o técnicas de renderización inadecuadas.
- Color Banding: Gradientes de color que no se representan de manera suave, sino que muestran bandas de colores discretos. Esto puede ocurrir cuando se utiliza un rango de color insuficiente o cuando se comprimen imágenes de manera agresiva.
- Artifactoring de Compresión: Problemas introducidos durante la compresión de imágenes, especialmente en formatos con pérdida como JPEG. Puede incluir bloques, ruidos o patrones no naturales.

## Fondos con técnica de Parallax y Layers

Son imágenes (Sprites) que se utilizan normalmente para el fondo, aunque no es su única aplicación. Es un efecto visual que simula la percepción de profundidad al

dividir un fondo en varias capas que luego, en el motor de videojuegos, se mueven a diferentes velocidades.

Es la diferencia de velocidad de movimiento de cada capa la que logra la sensación de que cada objeto se encuentra a una distancia mayor y, mientras más distancia, menor velocidad.

Curiosidad: El término ‘Parallax’ proviene de la astronomía, donde se utiliza para describir el cambio aparente en la posición de un objeto cuando se ve desde diferentes puntos de vista.

Este efecto se puede encontrar en el mundo real. Si se observa el paisaje lateral al viajar en auto por una ruta, se percibe que los objetos que se encuentran más cerca del punto de vista (cercos, pasto, arbustos, árboles, otros autos) se mueven a mayor velocidad que los que se encuentran más lejos (montañas, casas, cerros).



A continuación, en los GIF anteriores (hacer click en las imágenes para abrir el gif), se puede observar este efecto en funcionamiento. En la primera imagen, el punto de vista de la cámara se desplaza en un movimiento horizontal (Eje X) y en la segunda imagen, en un movimiento vertical/profundidad (Eje Y o Z dependiendo del motor). Si se tuviera que representar estas situaciones en un videojuego bidimensional, donde no se cuenta con el eje Z de profundidad y, por lo tanto, las imágenes están “superpuestas” en una misma posición, se podría identificar dos capas: la capa dónde se dibuja la pared con la ventana y la capa donde se dibujan los edificios con el cielo (ignorando los gatos). En el primer ejemplo, en movimiento horizontal, la primera capa se mueve más rápido que la cámara y la segunda más lenta, dando la sensación de que el edificio se mueve junto con la cámara. En el segundo ejemplo, dónde el movimiento es “vertical” o de “acercamiento”, la cámara no se mueve en el eje Z, sino que las imágenes se escalan en diferentes velocidades. La primera capa, más cercana, se escala en menor velocidad que la

segunda capa, dando la sensación de que el edificio se “acerca” o se “aleja” de la cámara, como si de un “zoom” se tratara.

En estos ejemplos, se puede comprender cómo el efecto parallax da dinamismo a las escenas. Un ejemplo donde se utilizan estos dos tipos de desplazamientos, es en el juego Gris (recomendado) y es ideal para cambios de encuadre y generar diferentes sensaciones al jugador.

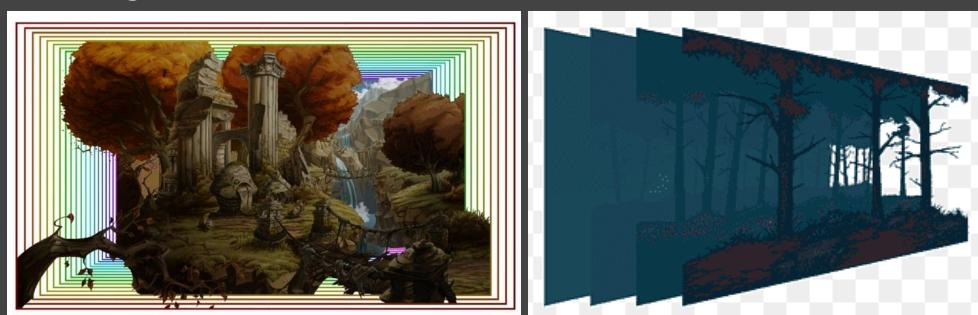
A continuación, se describen diferentes métodos de aplicación de este efecto:

- Método de Capa: Es el más utilizado. El desplazamiento de capas es como trabajar con transparencias en varias hojas de acetato. Cada capa de fondo se puede mover independientemente en direcciones horizontal y vertical. Las capas que se desplazan más rápido se perciben como más cercanas a la cámara. Este método es excelente para simular la profundidad y dar vida a los fondos en juegos de plataformas y aventuras.
- Método de Sprite: Crear pseudo capas con sprites ofrece control individual sobre objetos móviles. Por ejemplo, campos estelares animados se pueden lograr usando sprites. Esto es especialmente útil para juegos de disparos o aventuras donde quieras resaltar elementos específicos del fondo.
- Método de Patrón de Repetición/Animación: Este método utiliza mosaicos individuales animados para crear un efecto de paralaje. Animar mapas de bits de mosaicos individuales da la ilusión de capas adicionales. Es perfecto para representar efectos atmosféricos como lluvia o nieve en juegos de aventuras.
- Método Ráster: Aprovechar las características de gráficos con trama es como trabajar en una tela pixelada. Al dividir la pantalla en tiras horizontales, cada una con su posición y velocidad, puedes lograr efectos sorprendentes. Desde la profundidad de campo hasta el desplazamiento de filas y columnas, el método ráster es versátil y se usa en juegos icónicos como Sonic the Hedgehog y Street Fighter II (Retro).

Buenas prácticas para la optimización de imágenes con Efecto Parallax:

- Ajustar las dimensiones de las imágenes para que se adapten al tamaño real que aparecerán en pantalla. Respetando el Aspect Ratio de la pantalla del dispositivo al que está dirigido.
- Utilizar formatos de imagen eficientes en términos de tamaño y calidad. Por ejemplo: utilizar formatos comprimidos como JPEG para fondos con transiciones suaves y PNG para aquellos con transparencias.
- Uso de Sprites para fondos con detalles, creando subcapas entre las capas.

Ejemplo de Imágenes con efecto Parallax:



## Tipos de Grilla y Tilesets

La Grilla es una estructura bidimensional que organiza elementos gráficos en filas y columnas. En videojuegos, las grillas se utilizan para crear Tilemaps (grilla exclusivamente de Tiles), que son grillas en donde cada celda (espacio de fila y columna) está ocupada por un tile o mosaico, esencialmente un sprite. El conjunto de estos tiles se define como Tileset y se suele utilizar para la creación de mapas y niveles. Un Tileset es lo mismo que un Spritesheet, solo que exclusivamente de Tiles.

Las Grillas o Tilemaps pueden adoptar diversas características, como ser de vista isométricas u ortográficas y de forma hexagonales o rectangulares, según las necesidades particulares de un juego. Por lo tanto, los tiles pueden variar en forma y orientación para adaptarse a este tipo de grillas. Podemos definir:

- Forma de los Tiles: rectangulares, cuadrados, hexagonales, triangulares.
- Orientación de los Tiles: vista isométrica, ortográfica, frontal, lateral, top down.

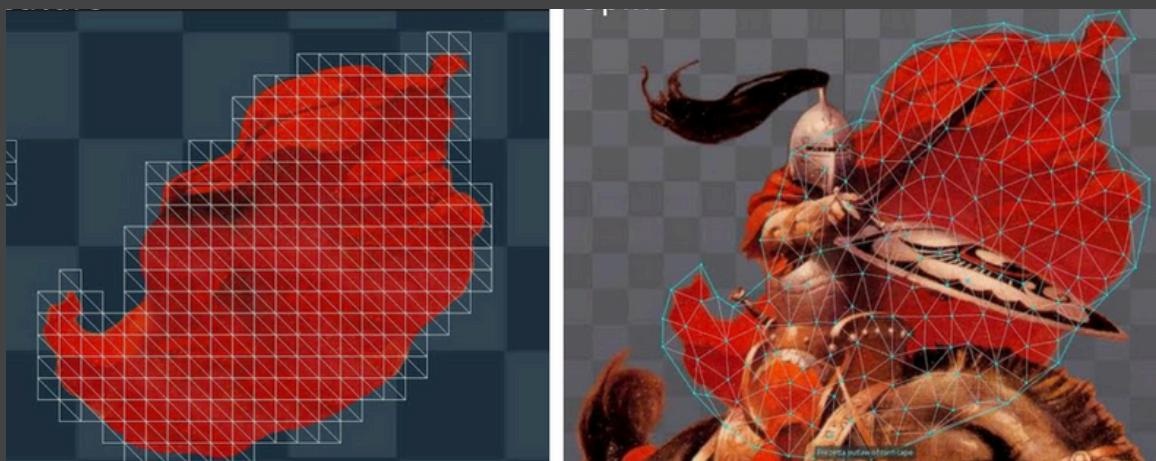
Ejemplos de Tiles:



## Mesh, Texturas y Shader

Estos son conceptos que se suelen aprender normalmente en modelado 3D, pero a medida que avanza la tecnología, se utilizan estas mismas técnicas para juegos en 2D. El uso de estos conceptos permite realizar técnicas de Wrapping, de iluminación, de relieves, entre otras. A continuación, se escribe la definición de cada uno de estos conceptos.

- Mesh (Maya o Forma): es una malla compuesta por polígonos y por lo tanto, por vértices, aristas y caras y define la forma de un objeto.



- Shader: es un programa especializado que se ejecuta en la GPU para controlar cómo se renderizan los gráficos. Se utilizan para definir la



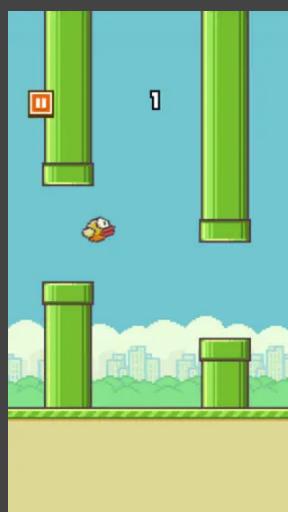
- Roughness: Ajusta cómo interactúa la luz con la superficie, influenciando la apariencia de brillo y reflejos. Ajusta cómo interactúa la luz con la superficie, influenciando la apariencia de brillo y reflejos. En un mapa de oclusión ambiental, los colores generalmente representan la intensidad de la sombra. Las áreas más oscuras indican regiones más sombreadas y menos expuestas a la luz ambiental.
- Metallic: Ajusta cómo interactúa la luz con la superficie, influenciando la apariencia de brillo y reflejos. Ajusta cómo interactúa la luz con la superficie, influenciando la apariencia de brillo y reflejos. En un mapa de oclusión ambiental, los colores generalmente representan la intensidad de la sombra. Las áreas más oscuras indican regiones más sombreadas y menos expuestas a la luz ambiental.



### Técnica de Mapeo Envolvente (Wrapping)

El Wrapping se refiere a la forma en la que se mapea una textura sobre una superficie bidimensional o tridimensional y cómo se gestionan las coordenadas de textura cuando excedan los límites estándar de 0 a 1.

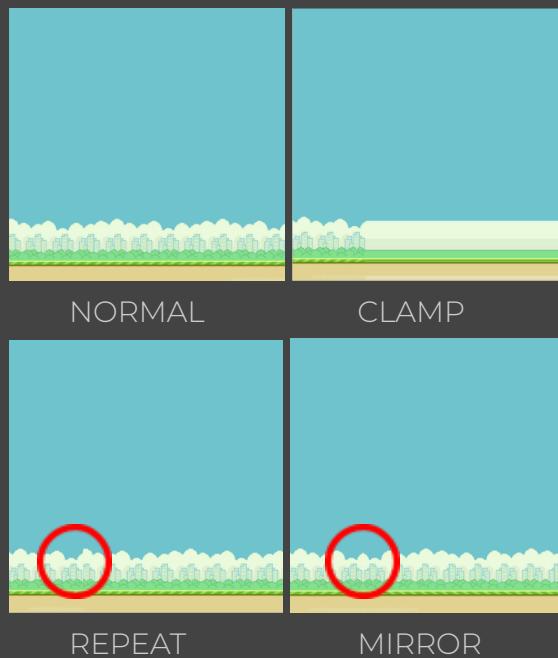
En algunos juegos donde la cámara y el personaje son estáticos (por ejemplo un Runner), pero se desea dar la sensación de que el personaje está avanzando, se utiliza la técnica de Wrapping y de Parallax.



Las imágenes no se mueven en diferentes velocidades con respecto al movimiento de la cámara, sino que se mueven continuamente en diferentes velocidades, así la

cámara es estática. Y este movimiento se hace usando el Wrapping, para mover una textura en un plano sin modificar la posición de ese plano. Entonces, cuando las coordenadas de textura en la dirección horizontal/vertical superan el rango de 0 a 1 (es decir se mueven de izquierda a derecha o de , saliendo del plano), la técnica de envoltura determina cómo se maneja esto. Las opciones comunes son:

- Repeat (Repetir): La textura se repite continuamente, creando un patrón de mosaico. Por ejemplo, si la coordenada s es 1.2, se verá como si fuera 0.2.
- Clamp (Sujetar): La textura se sujetará y no se repite. Si la coordenada s es mayor que 1, se utilizará la textura en el borde (1).
- Mirror (Espejo): La textura se repite, pero con un efecto de espejo en cada repetición. Esto puede crear un efecto de espejo suave en los bordes.



Una buena práctica para el uso correcto de esta técnica es tener en cuenta que al repetir la imagen, no existan cortes o desperfectos. Como se ve en el ejemplo de Flappy Bird, el fondo funciona tanto con repeat como con mirror, los pixeles conectan de manera suavizada.