Introducción a la Computación Gráfica

Eduardo Graells-Garrido

9 de marzo de 2026

1 Cuando una consciencia se convierte en universo (tal vez)

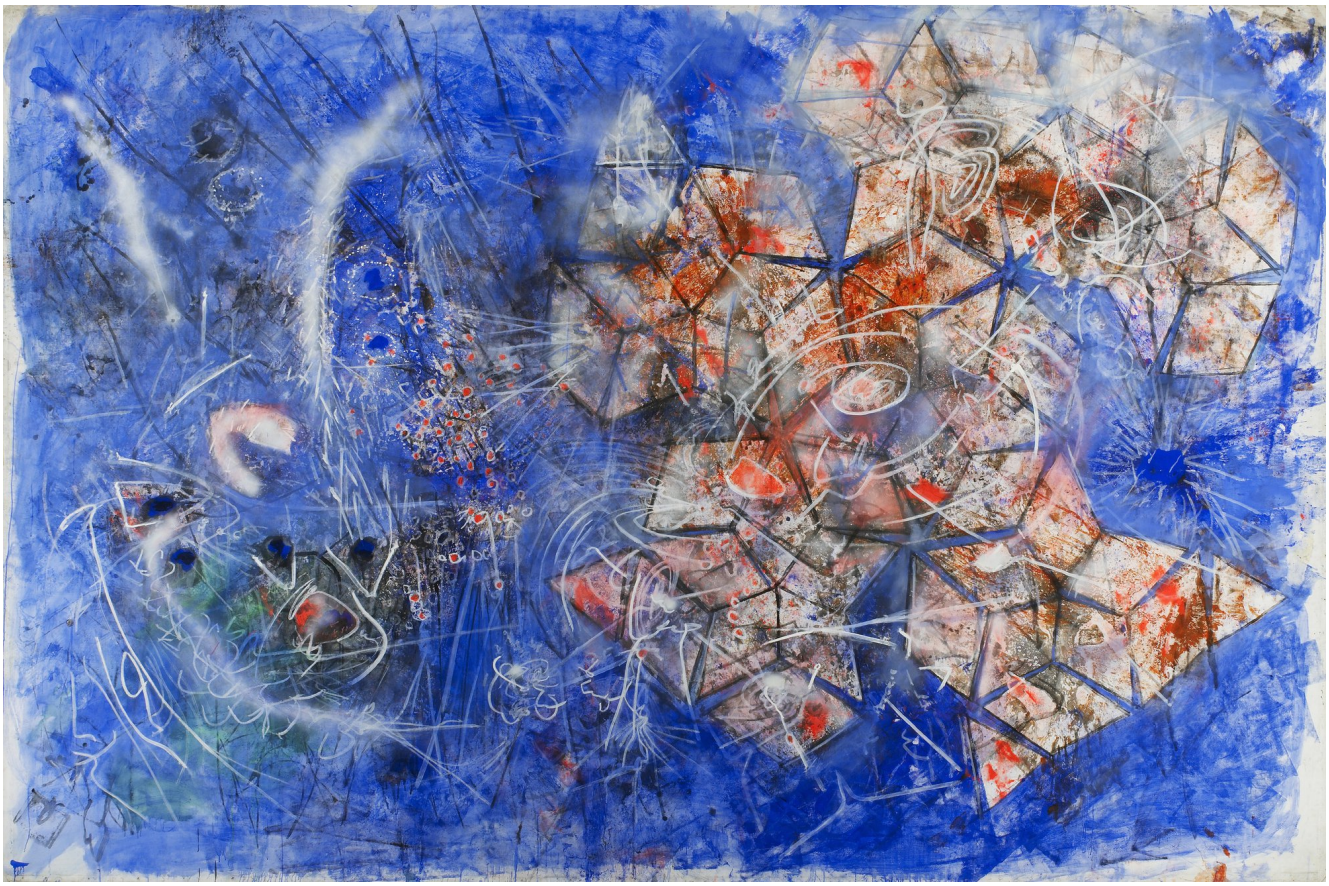

Figura 1. Roberto Matta, *Comment une conscience se fait univers (peut être)*, 1992.

En esta unidad veremos distintos casos de uso que sirven como motivación para aprender qué es Computación Gráfica. Por lo mismo, sé que es extraño comenzarla con una obra abstracta de Roberto Matta (Figura 1). El título de esta sección es una traducción del título original de la obra. Pienso que refleja lo que queremos hacer: convertir algo que no es tangible (la consciencia → el código, las instrucciones, los modelos) en algo que sí podemos percibir: un *universo virtual*.

2 Un mundo virtual

Queremos **simular un mundo usando computadoras**. A qué nos referimos con mundo: ¿a una Realidad Virtual (VR)? ¿A un multiverso? ¿A un videojuego? Usualmente es un mundo que se comporta como nuestro mundo real, pero puede ser otro mundo, con otras reglas.

Lo importante es que esta simulación sea **realista**, para múltiples definiciones de qué es realista: realismo visual, realismo en la física del mundo, realismo en el sonido... incluso realismo en lo que no es realista, como cuando se grafican dibujos animados o estilos artísticos.

Por ello, a CG utiliza conceptos de áreas como:

- Geometría y 3D para definir, organizar y visualizar el mundo virtual.
- Ecuaciones diferenciales (ordinarias y parciales) para modelar el comportamiento de partículas, objectos, fluídos, etc.
- Modelos de iluminación para definir cómo lucen los elementos del mundo virtual.

Y muchísimas más, por ejemplo, modelos de sonido que determinan cómo suenan y cómo se escuchan los elementos del mundo. Pero en este curso introductorio nos enfocaremos en la geometría, el modelamiento con ecuaciones y los modelos de iluminación.

Una **definición** simple y directa es

> La Computación Gráfica comprende el proceso de generación de imágenes que representan escenas computacionales, donde "escena computacional" es una representación geométrica 2D o 3D de distintos elementos dispuestos en un espacio.

El proceso de "generación de imagen" se llama *rendering*[1]. Existen múltiples paradigmas de *rendering*. En este curso nos enfocaremos en el *rendering* en tiempo real, es decir, aquel que se lleva a cabo en el instante en que se interactúa con un sistema. En cambio, una película de animación generada por computadora no se grafica en tiempo real: el video fue *renderizado* con antelación.

[1] Una posible traducción es *graficación*, pero en la práctica es mejor usar el término original en inglés.

La escena (Figura 2) se define como el conjunto de elementos que aparecen graficados en la imagen generada (Figura 3). Casi como en una obra de teatro o en una película, esos elementos se disponen y se visualizan a través de una cámara. La definición es simple, pero a lo largo del curso estudiaremos muchas complejidades detrás:

- ¿Cómo se define un elemento, un objeto 3D?
- ¿Cómo se expresa matemáticamente una escena?
- ¿Cómo interactúan los objetos de una escena entre sí?
- ¿Cómo grafico una escena lo suficientemente rápido como para ser considerado *rendering en tiempo real*?

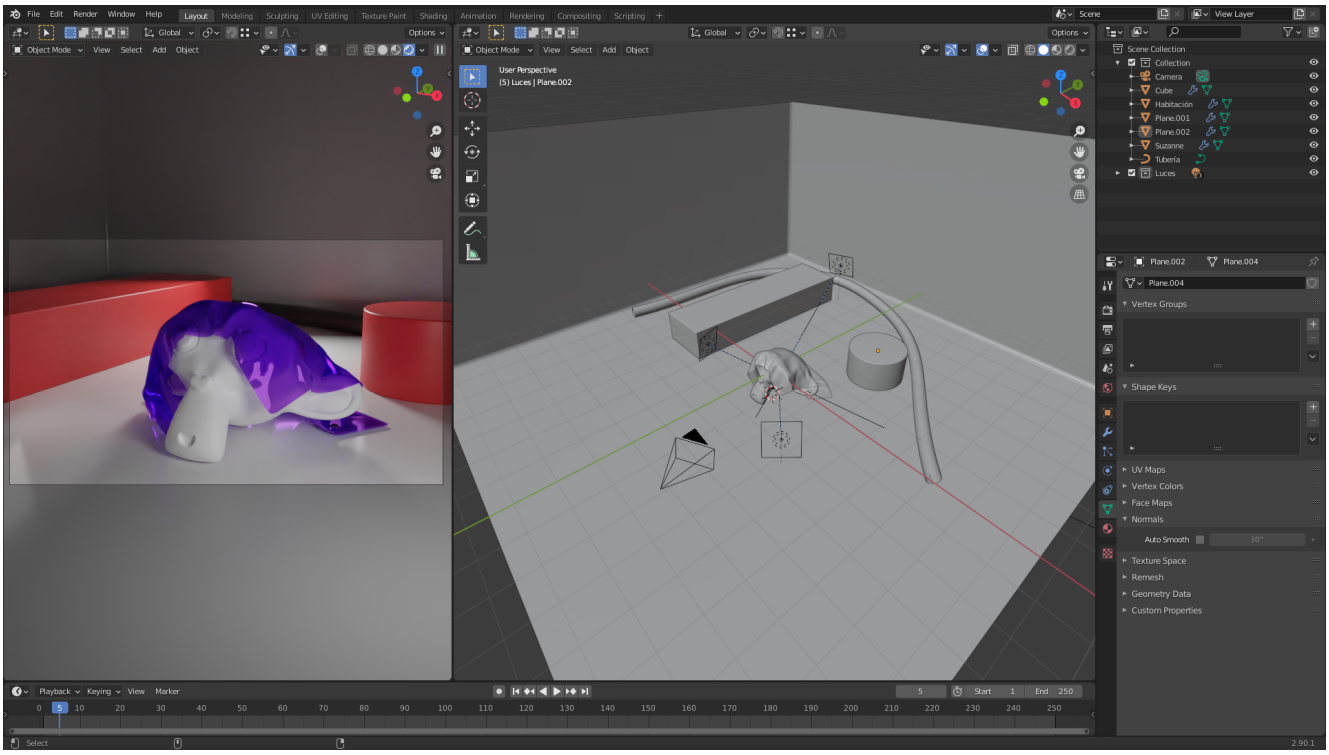

Figura 2. *Render* y Escena. Fuente: *Computer Graphics*, Wikipedia.

Figura 3. ¿Cuántos elementos hay en esta escena? ¿Cuáles son las reglas del mundo que la gobiernan? Un *frame* del videojuego chileno *Metaverso Antártico*, cuyo fin es realizar diseminación científica. Fuente: XR Labs, Universidad de Chile.

Algunas de estas respuestas se responden con **modelación** y otras directamente con **geometría** y graficación. De allí que este curso se titule *Modelación y Computación Gráfica*.

3 Videojuegos

Probablemente los videojuegos son la aplicación de *rendering* en tiempo real más grande en términos de personas que la utilizan y también en términos de mercado. La Figura 4 muestra

una infografía de 2021 sobre los enormes ingresos de múltiples países debido a los videojuegos. Noten que estamos hablando de ingresos, no de gasto. ¡Eso es otro tema, también uno importante! Hoy, incluso los jugadores también comienzan a ganar un dinero considerable en las competencias (Figura 5).

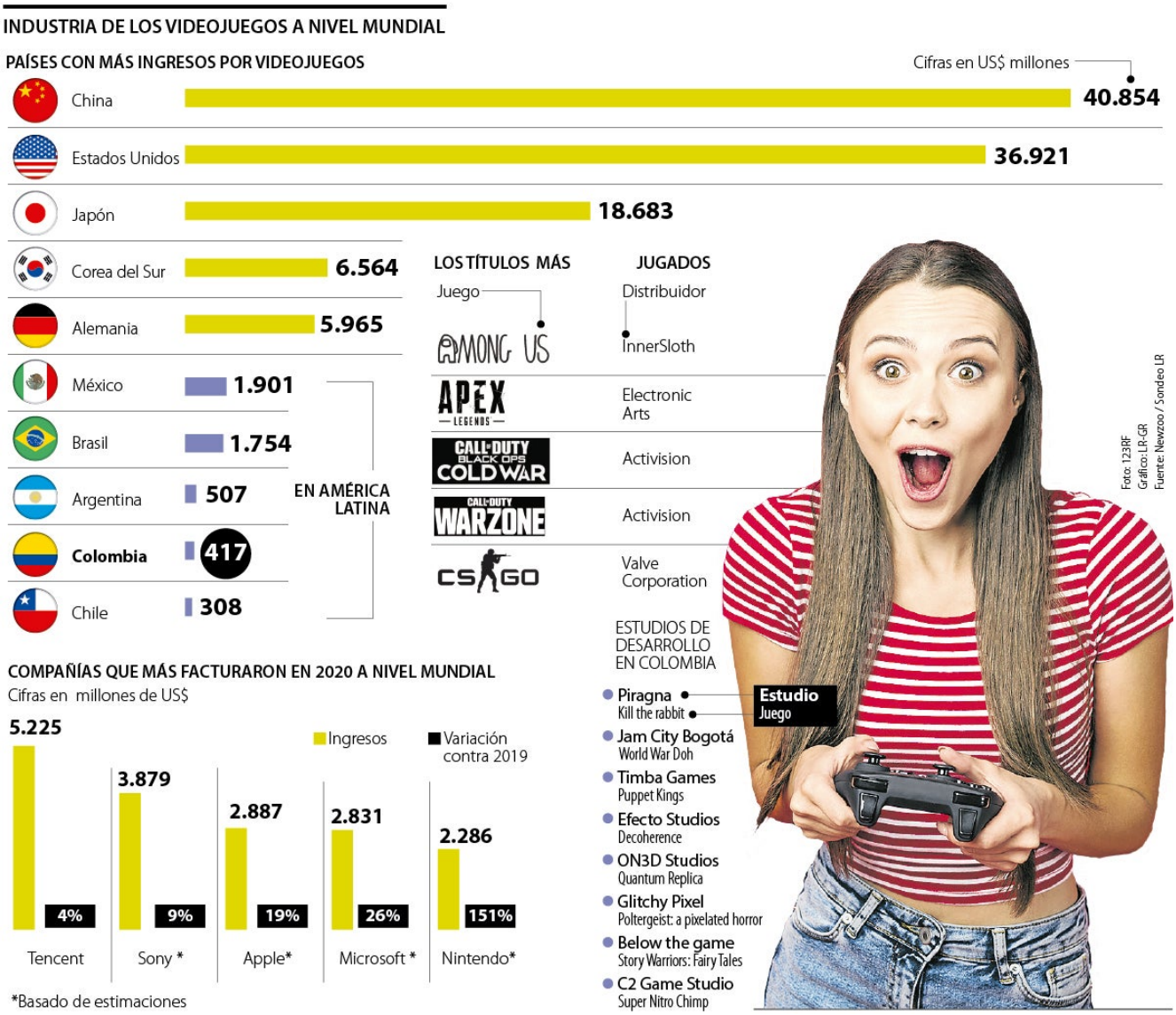

Figura 4. La industria de los videojuegos mueve solo en cinco países casi US$109.000 millones, Fuente: La República (Colombia), 2021.

En Chile hay empresas que desarrollan videojuegos, principalmente para el mercado internacional, como AOne Games y ACE Team. Además, hay empresas o estudios más pequeños que realizan juegos para empresas más grandes.

Los juegos actuales cuentan con motores avanzados de *rendering* y de simulación física. Parte de la CG es hacernos creer que

Figura 5. Una imagen de *Street Fighter 6*, en la competencia CAPCOM CUP 2025. El chileno Derek Blaz, de quince años, obtuvo el segundo lugar y con ello se llevó un premio de 100 mil dólares.

Figura 6. El videojuego *Stray* es un simulador de "gato callejero" con una estética y físicas realistas en tiempo real. Mira su trailer.

son completamente realistas. Por ejemplo, ¿el juego *Stray* simula completamente la física del pelaje de un gato, o parece hacerlo (Figura 6)? Mucho del trabajo en CG no es realizar la simulación a un nivel perfecto, sino *aparentar* el realismo a través de modelos más simples o de atajos que nos permitan alcanzar un buen desempeño en tiempo real.

No siempre se busca el realismo visual. El llamado *rendering no-fotorrealista* (NPR: *Non-Photorealistic Rendering*) busca simular la apariencia de estilos pictográficos (ver *Okami* en Figura 7). En ocasiones, también hay juegos cuya estética no es realista pero las reglas físicas sí lo son. Un caso emblemático es *Angry Birds*, que tiene un motor de física realista en 2D, incluso para el espacio exterior.

Figura 7. El videojuego *Okami* no tiene estética fotorrealista, pero sí busca expresar el estilo Ukyo-e de manera fiel a la técnica real. Mira su trailer.

4 Cine y animación

Este año (2025) la película que ganó el *Oscar* de la categoría "Mejor película animada" fue *Flow* (Figura 8), protagonizada por un gato[2] y que tiene varias particularidades. No solo es una película 3D[3], sino que su parte gráfica fue creada íntegramente en *Blender*, un *software* de modelado 3D de código abierto y licencia libre.

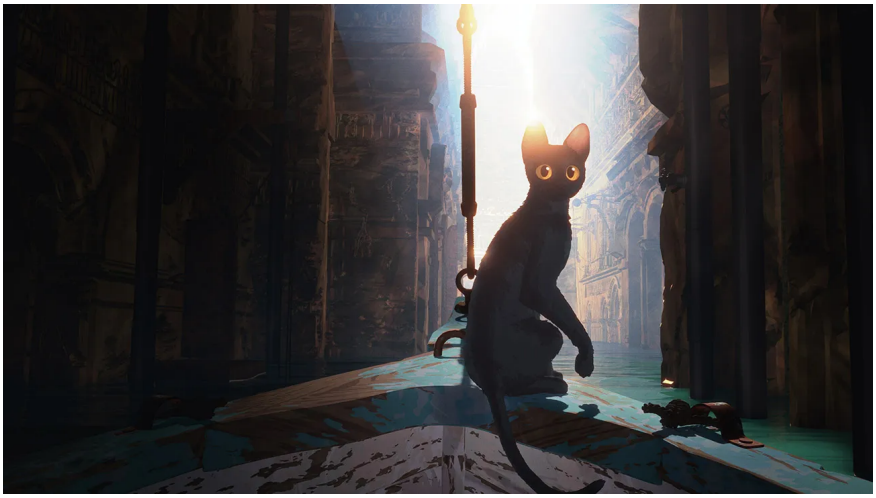

Figura 8. Imagen de *Flow*. Esta película independiente ganó un premio que tradicionalmente obtenían estudios enormes como Pixar. Mira el trailer.

Las películas animadas se caracterizan por escenas complejas y personajes llenos de detalles, con una iluminación que usualmente no puede alcanzarse en tiempo real. Para generar un cuadro de animación se utilizan múltiples máquinas, y para gene-

[2] No es el primer ejemplo con gatos. Como ustedes verán en este curso, soy fan de los pajaritos, no de los gatos. Pero casi todo el mundo es fan de los gatos. Así que es más fácil encontrar ejemplos de gatos que de pajaritos.

[3] ¿Cuándo se dejaron de producir películas dibujadas a mano? Incluso aquellas que parecen utilizar estilos tenían estudios enormes computadora, utilizando técnicas de NPR.

rar una película, una granja completa. En *Shrek* la estética tiene tintes fotorrealistas, como el modelo de iluminación, pero las formas y las expresiones son exageradas e incluso irreales, todo en pos de la expresividad y de la historia (Figura 9).

Figura 9. *Shrek* es una historia de fantasía cuyo *rendering* es realista en las reglas físicas de su mundo.

Más adelante veremos la evolución en el *hardware* dedicado a CG. Eso ha tenido un impacto enorme en la calidad gráfica de los videojuegos, al ser aplicaciones en tiempo real. Pero, ¿y la animación? Al no requerir tiempo real, ¿ha tenido saltos así de marcados? La respuesta es sí. Antes de responder el por qué, necesitamos conocer algo de historia.

La primera animación fue hecha el año 1972 (Figura 10) por Edwin Catmull[4] y Fred Parke. Se llamó *A Computer Animated Hand* (*Una mano animada por computadora*)[5] y consistió justamente en lo que menciona su título. En ese tiempo no existían las interfaces gráficas o las herramientas actuales para modelar. De hecho, probablemente incluso la visualización de un modelo 3D era algo complejo: Ed Catmull tuvo que dibujar polígonos sobre su propia mano para conocer las coordenadas de la mano a graficar; Henri Gouraud tuvo que dibujar mallas geométricas en el rostro de Sylvie Gouraud para modelar rostros a ser usados en modelos de iluminación (Figura 11).

Fast Forward: la primera película animada fue *Toy Story* (1995) de Pixar (ve información de su *making-off* en las Figuras 12 y 13). ¡Ustedes todavía no habían nacido! Pero yo sí, y puedo decirles fue una revolución. Gran parte de eso también se debe a su historia encantadora, y justamente ese es el punto: la gráfica no era del todo realista (hoy incluso al lado de un juego parece rudimentaria) pero eso no era necesario, puesto que era lo que necesitaba la historia.

Entonces, ¿por qué, al ser en tiempo no real, no se tenía la calidad de hoy? Eso se debe a los otros aspectos que también

[4]"Ed" Catmull fue uno de los fundadores de Pixar.

[5]Aquí hay una historia interesante al respecto.

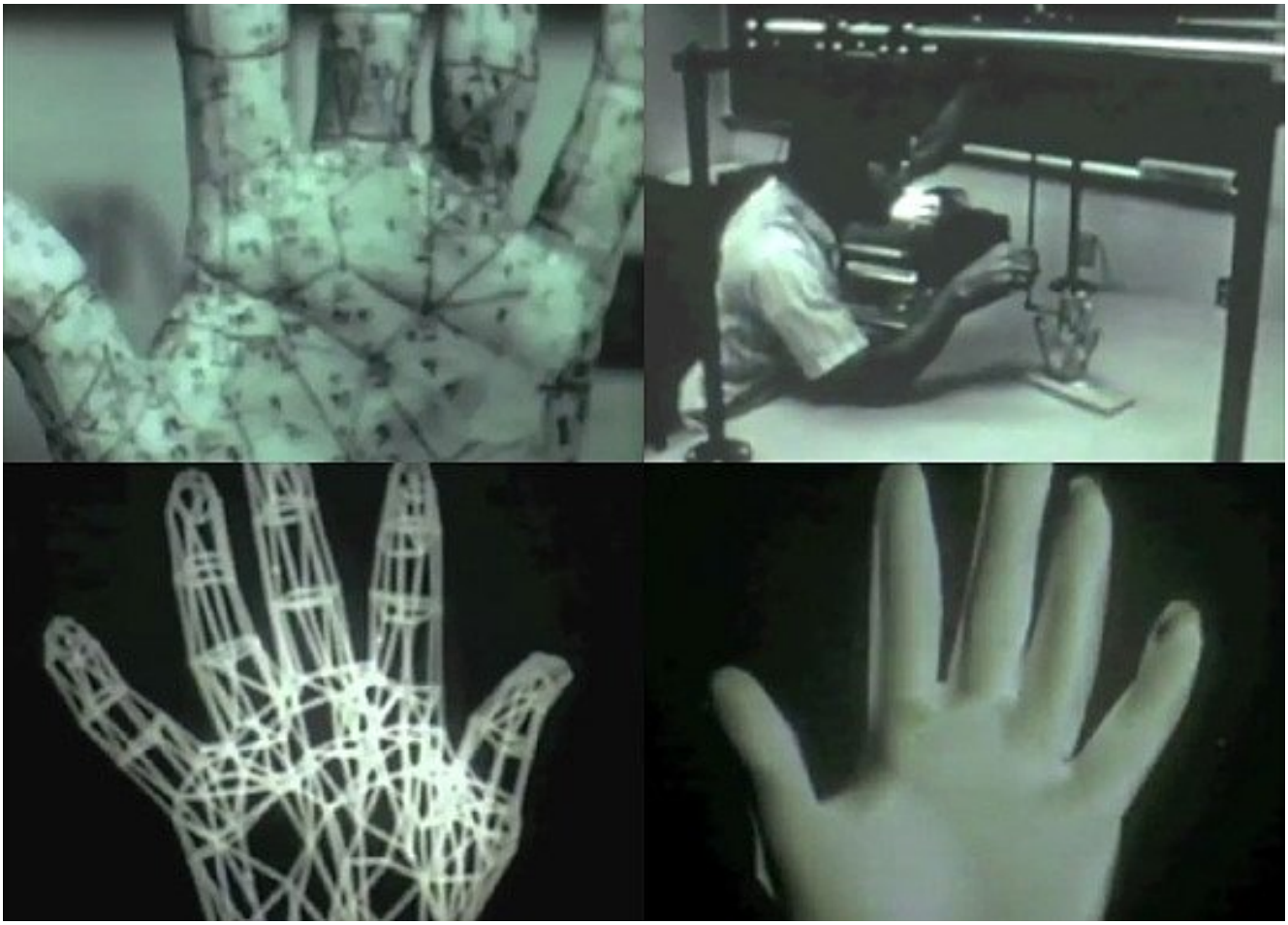

Figura 10. *A Computer Animated Hand (1972)*. Puedes ver este cortometraje en YouTube.

deben ser realistas: la simulación física y el modelo de iluminación. Ambas cosas son sumamente caras y difíciles de lograr, y por tanto, incluso aunque se *renderice* en tiempo no real, hay presupuestos de costo: tiempo de desarrollo, tiempo de ejecución, energía requerida, almacenamiento requerido, tiempo comprometido de entrega... entre otros. Para dar un ejemplo: simular de manera realista la apariencia y el comportamiento del cabello requiere enormes cantidades de primitivas geométricas y de modelamiento de superficies, de resistencia de materiales, de interacciones físicas, de reflexión de la luz. Y no siempre existen ecuaciones o modelos que expliquen fenómenos como ese. En ocasiones, para poder graficar algo, primero debemos investigar cómo modelarlo. Y eso puede tomar años.

Fast Forward de nuevo: la serie *The Mandalorian* también utiliza animación. ¿Cómo, si está protagonizada por Pedro Pascal? No solo para crear el modelo 3D y la animación de Grogu, sino también para el entorno. Muchas series y películas se graban

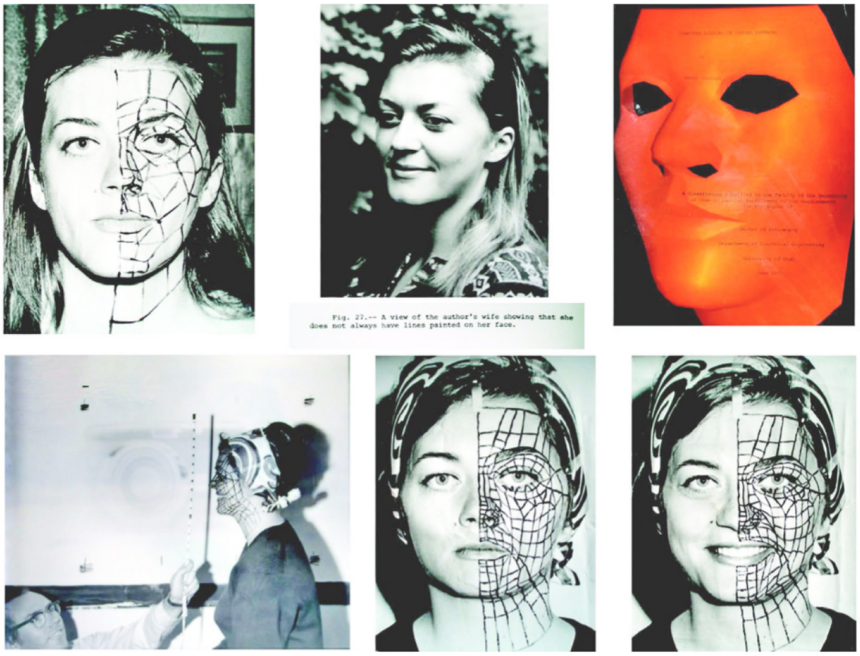

Figura 11. Sylvie Gouraud utilizando su rostro para hacer un modelo 3D y aplicar un modelo de sombreado para aplicar un modelo de iluminación, técnica que veremos más adelante en el curso. La imagen es de 1971. Recuperada de un trabajo de Simone C. Niquille titulado *What does the Graphical User Interface want?* que discute la historia de las interfaces gráficas.

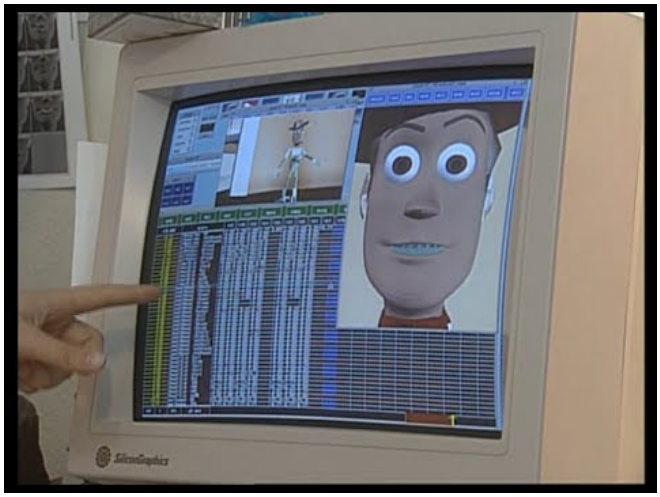

Figura 12. ¿Cómo se hizo *Toy Story*? Se requiere tanto *hardware* como *software* especializado. Aunque lo especial hace treinta años probablemente ya se puede llevar a cabo en un *smartphone* hoy. Fuente: YouTube.

con fondos verdes, un escenario o cortinaje que después es reemplazado por una imagen, un video o un *render*. En el caso de *The Mandarolian*, se tenía un ambiente completo en el cual no se utilizaba un fondo verde, sino que había pantallas donde se mostraba *rendering* en tiempo real del escenario, un "Virtual Set" graficado con el motor Unreal Engine[6] (Figura 14). ¿Por qué es mejor que "el fondo verde"? Porque, entre otras cosas, el fondo verde no crea reflejos o luces acorde al lugar en el que sucede la acción, y eso tiene repercusiones visuales. Esto es impresionante no solo a nivel técnico, sino también porque muestra cuánto ha avanzado el *rendering* en tiempo real y los videojuegos. Hablaremos más sobre eso al final de esta unidad.

[6] Pueden saber más de este sistema en Polygon.

Figura 13. Otra imagen del *Making of Toy Story*. Me encanta el escritorio noventero. Me recuerda mi juventud, quizás más que la película en sí misma.

Figura 14. El "Virtual Set" de *The Mandalorian*. Puedes ver un video del set en acción. Fuente de la imagen: TechCrunch.

5 Medicina

Hay usos importantes en medicina (Figura 15), que apoyan la labor de personal médico en realizar diagnóstico y seguimiento de pacientes. Esto incluye técnicas de visualización y herramientas de exploración de resultados de escáner (resonancias magnéticas, tomografías, etc.), así como el modelamiento y simulación de componentes internos (arterias y órganos) e incluso de partículas (como proteínas).

Esta área está creciendo. Sus usos futuros incluirán *rendering* y modelamiento en tiempo real de simulación de respuestas biológicas a tratamientos o medicina, o métodos para realizar cirugía de manera remota[7].

[7] Puedes leer más información en Health Care IT News.

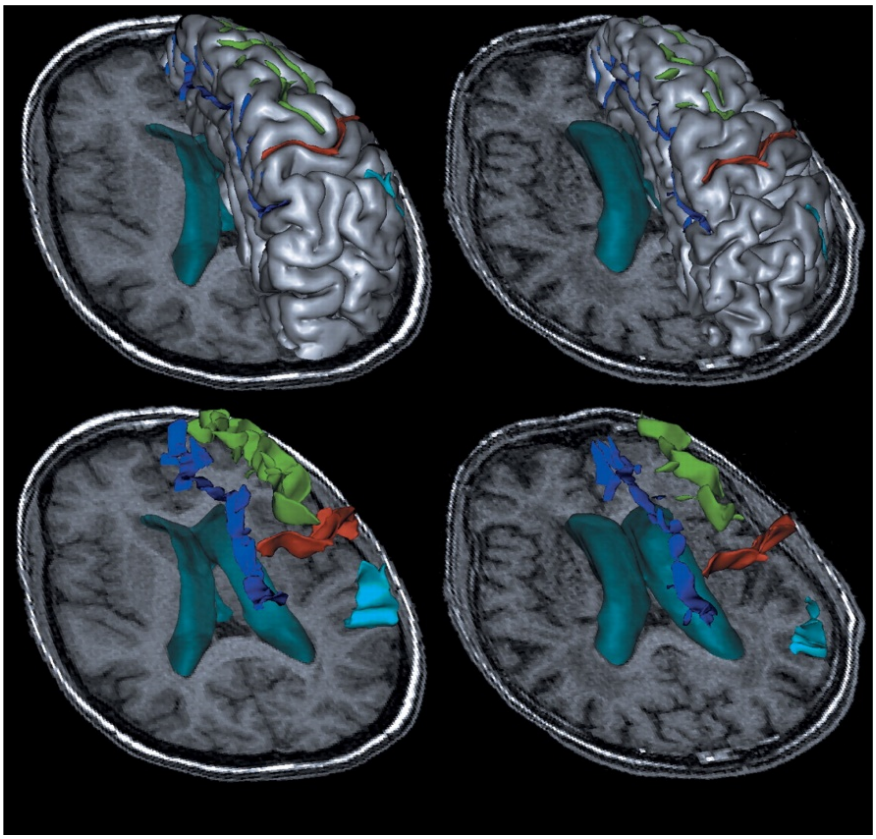

Figura 15. Comparación del estado de los cerebros de dos pacientes, utilizando escáner. Fuente: *Cortical changes in cerebral small vessel diseases: a 3D MRI study of cortical morphology in CADASIL*.

6 CAD (*Computer Aided Design*)

Los sistemas de Diseño Asistido por Computadora (Figura 16) son utilizados para fabricar piezas críticas, diseñar infraestructura, vehículos, arquitectura, etc. En general, cuando se requiere dibujo técnico, se utiliza CAD. Veremos que en este tipo de herramientas es común que haya múltiples maneras de construir y modelar objetos, a veces muy diferentes de las utilizadas en videojuegos y animación. ¿Por qué? Porque las computadoras trabajan con números discretizados (incluso los números de punto flotante son una aproximación del número real), pero en una pieza real, como una turbina, un error de aproximación a causa de una representación inexacta puede causar un accidente.

A veces estos sistemas también incluyen componentes de simulación. Por ejemplo, conociendo la composición y resistencia de un material, es posible poner a prueba su funcionamiento en condiciones críticas. Si el diseño supera las pruebas virtuales, entonces se vuelve factible hacer una prueba física.

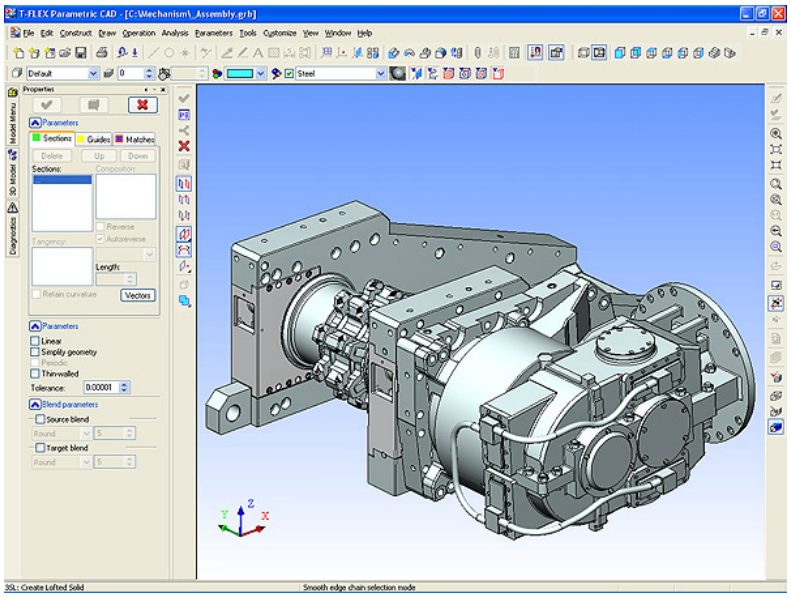

Figura 16. Imagen del *software* T-FLEX, un sistema de Diseño Asistido por Computadora (CAD).

7 Visualización Científica

En esta área, también conocida como *SciVis*, se busca representar fielmente el comportamiento de un fenómeno. El realismo está en su modelamiento, no necesariamente en cómo luce visualmente. Se utiliza para apoyar la toma de decisiones y para comunicar resultados científicos. Por ejemplo, el proyecto EXCELLERAT realiza diferentes modelaciones aerodinámicas y de fluidos (Figura 17), que son comunicadas de manera efectiva utilizando técnicas de CG. ¡Pero lo que grafica no necesariamente es lo que ve un ojo humano! Esto también se puede aplicar a múltiples fenómenos naturales, cuyas consecuencias pueden ser catastróficas para la humanidad, y que necesitan ser visibles (Figura 18).

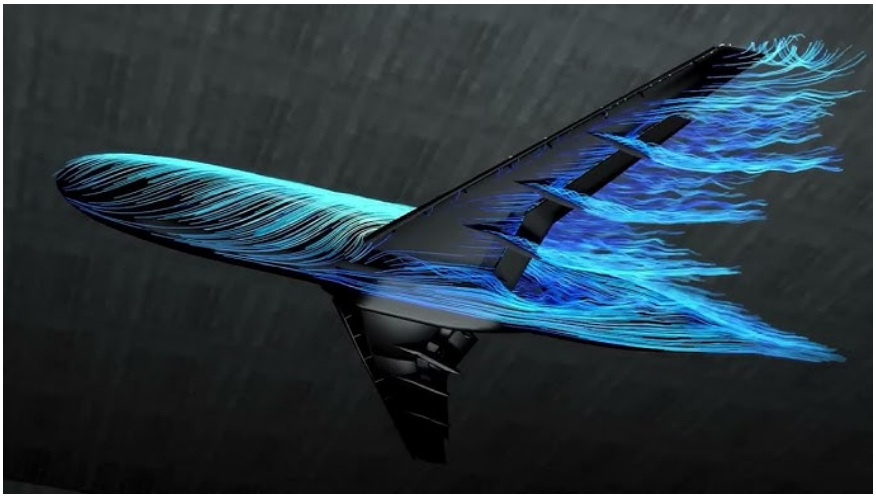

Figura 17. Proyecto EXCELLERAT. A menos que un avión esté incendiándose con un fuego mágico, esas líneas de colores no son visibles en la realidad. Pero lo que representan sí que es real. Ve un video del proyecto.

Figura 18. Un video con el trabajo del Grupo de Visualización Científica del Barcelona Supercomputing Center. Puedes verlo en YouTube.

Una área de aplicación es la de visualización de datos urbanos. Un simulador y visualizador de clima y contaminación puede usar estos datos como insumo. Afortunadamente muchos datos urbanos son descargables de manera abierta, por lo que es posible experimentar, con la característica de que este tipo de datos nos es más cercano. Existen herramientas gráficas especializadas en este tipo de visualización, como `deck.gl`.

Figura 19. ¿Qué es lo que ven los automóviles autónomos? Fuente: Waymo.

La empresa Waymo opera vehículos autónomos en San Francisco. Estos vehículos están equipados con múltiples cámaras LiDAR (*Light Detection and Ranging o Laser Imaging Detection and Ranging*) que capturan nubes de puntos que representan los objetos, personas e infraestructura urbana alrededor del vehículo.

Esa nube de puntos debe ser procesada y analizada en tiempo real para que un modelo de Inteligencia Artificial decida cómo moverse (o no hacerlo, en caso de ser necesario). Y lo que "ve" el vehículo (Figura 19) se despliega en tiempo real en una pantalla dentro de él, para que quienes estén dentro sepan que están en un vehículo seguro.

8 ¿Cómo hacemos todo esto?

Como hemos visto, hoy existen muchas maneras de hacer CG. Nosotres lo haremos programando, puesto que, desde una perspectiva ingenieril, es necesario entender de *software* y *hardware* para diseñar e implementar soluciones de CG. Por ello, en este curso estudiaremos la teoría base, programaremos *software* y aprenderemos a utilizar *hardware* especializado en gráficos: la GPU. Utilizaremos principalmente el lenguaje de programación Python y la biblioteca `pyglet`.

Existen herramientas que permiten desarrollar aplicaciones gráficas mediante la carga de insumos (*assets*) y la configuración de las reglas del mundo. Estos programas se llaman *game engines* porque se utilizan principalmente para hacer videojuegos, pero se pueden aplicar a cualquier área.

Figura 20. Con Unreal Engine puedes hacer una aplicación realista con una ciudad tipo Matrix. Video: https://www.youtube.com/watch?v=WU0gvPcc3jQ.

Existen motores especializados, con el estado del arte en CG, como *Unreal Engine* (Figura 20), otros cercanos a este estado, pero más asequibles tanto en términos monetarios como conceptuales, como *Unity*. Hay algunos libres, como *Godot* https://godotengine.org/, que no se quedan atrás en flexibilidad, y que,

al ser libres, democratizan la creación de aplicaciones y animaciones (Figura 21).

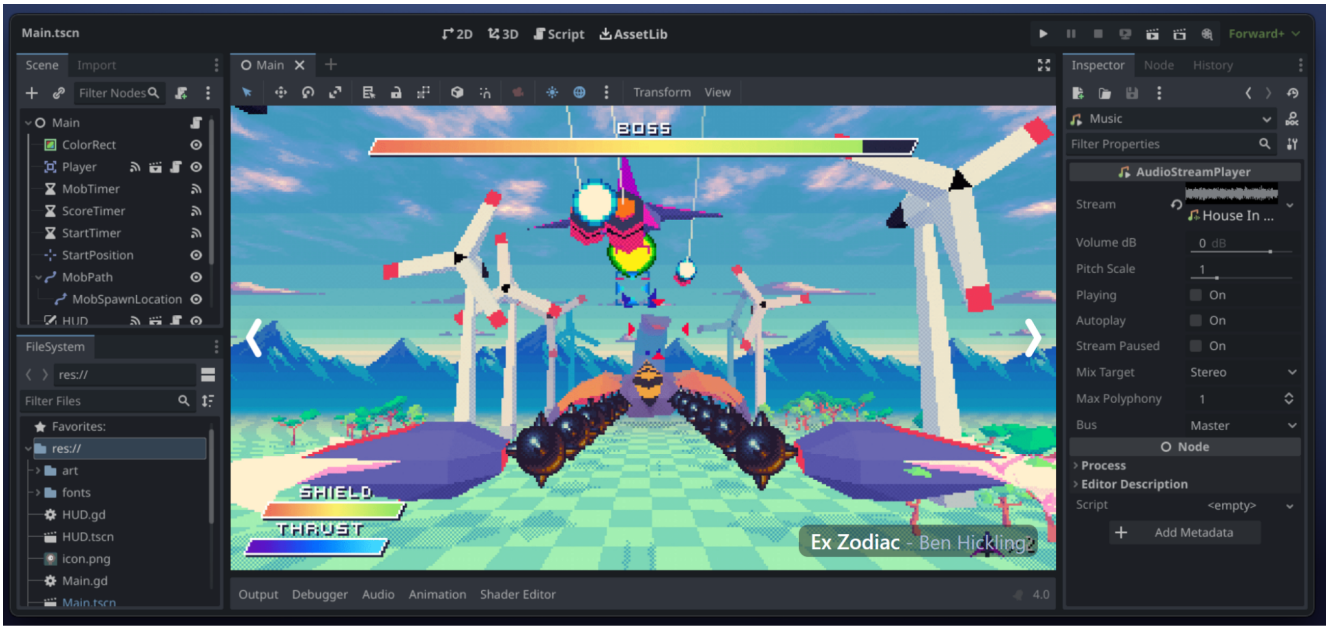

Figura 21. Godot Engine en el desarrollo de un juego con la estética de *Star Fox*, el primer juego 3D de Super Nintendo. En ese entonces, no cualquier *hardware* podía hacer cálculos 3D, por lo que era necesario tener chips especiales, como el *SuperFX*. Aquí puedes ver parte de la historia de este chip.

Hay mercado laboral para todos estos enfoques. Quienes trabajen haciendo videojuegos probablemente lo harán mediante un *game engine* como Unreal, Unity o Godot. Quienes realicen visualización científica pueden hacerlo mediante *scripting* de Python en *Blender*, *Houdini* o *Maya*. Habrá quienes hagan visualizaciones interactivas en el navegador utilizando WebGL a través de Javascript. Y habrá quienes trabajen directamente implementando sus propias herramientas, motores y visualizaciones. En estos últimos casos es probable que se utilicen lenguajes de programación como C++ o, últimamente, Rust. Python también es una alternativa, siempre que "por debajo" se sigan utilizando bibliotecas de alto rendimiento.

Sea cual sea tu camino, este curso te entregará los fundamentos. Y, en caso de que decidas no seguir el camino gráfico, aprenderás a trabajar con sistemas y modelos virtuales de fenómenos reales. Independiente del área a la que te dediques, el ser capaz de modelar un fenómeno y comunicar los resultados de una solución es una habilidad necesaria en ingeniería.

9 Recomendación de bibliografía

- La ACM tiene al menos dos listas de *papers* seminales en Computación Gráfica (primera, segunda).
- El grupo SIGGRAPH (*Special Interest Group in Computer Graphics*) organiza la conferencia homónima que reúne los avances anuales más importantes en gráfica y simulación. Se suelen subir videos con resultados destacados por año, por ejemplo: 2023, 2024, 2025.
- Otros videos interesantes son *Crash Course: Screen & 2D Graphics* y A brief history of graphics.