

Network of above, Kandinsky (1927)

CC3501

Grafos de Escena (Parte 1)

Eduardo Graells-Garrido
Otoño 2023

Esta escena del juego Metal Gear Solid 3 nos muestra una escena llena de flores, pasto, árboles, hojas y pétalos cayendo.

¿Cómo se puede renderizar una escena así de compleja en tiempo real?

¿Cómo se manipula lo que sucede en esta escena?



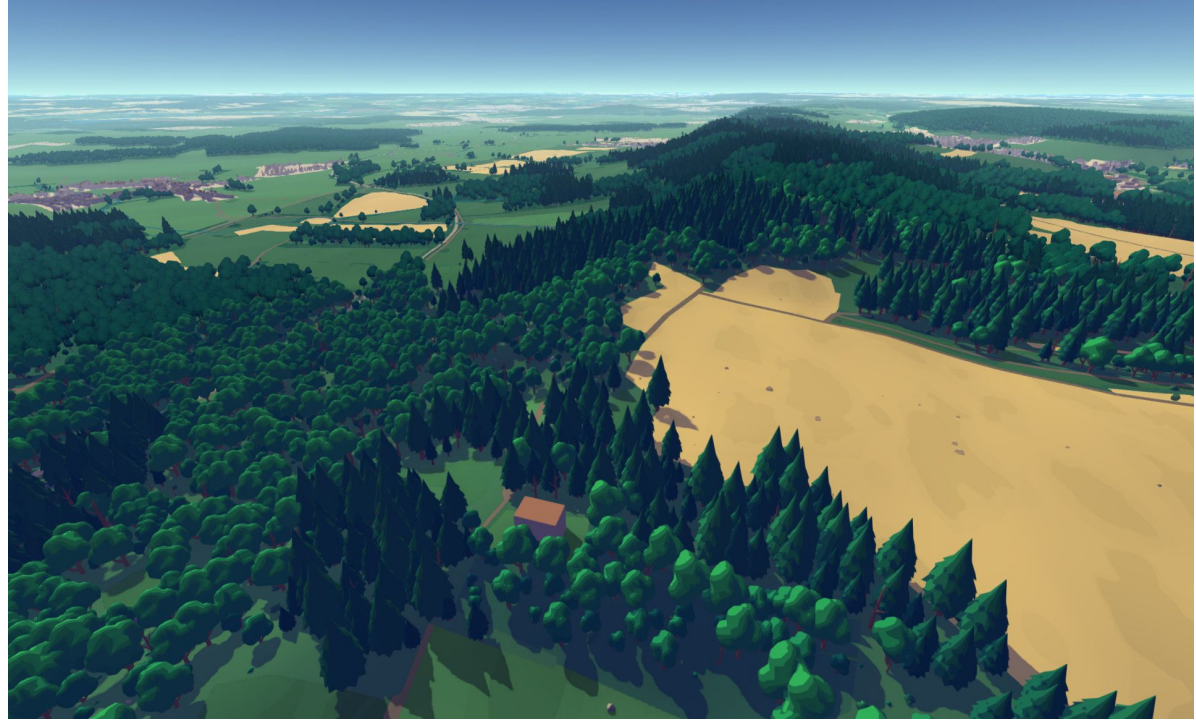
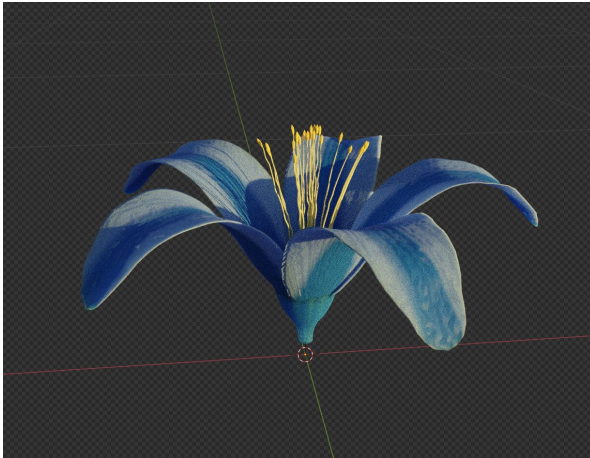
¿Cuántos modelos 3D necesitamos para renderizar un tablero de ajedrez con todas sus piezas?

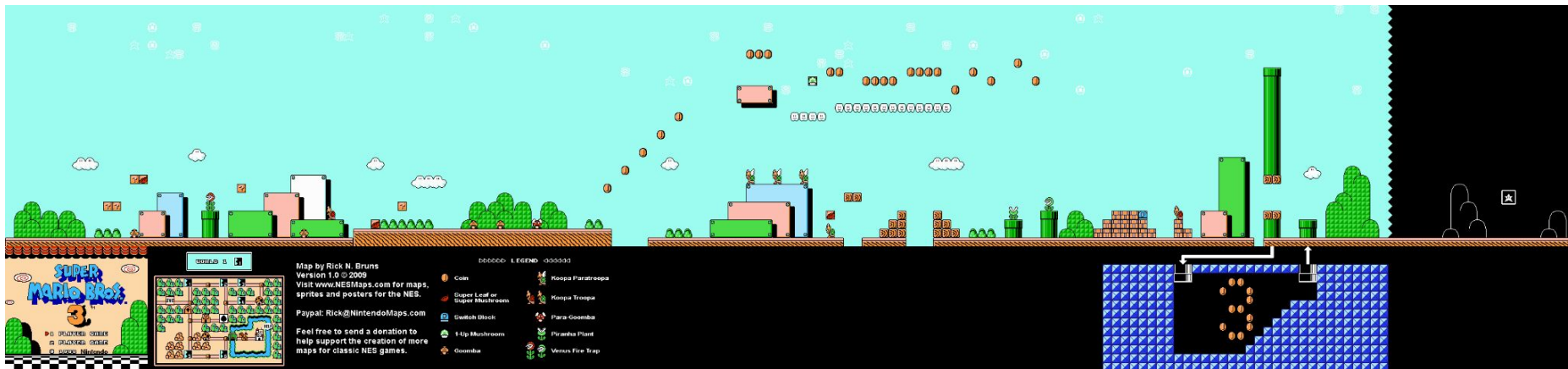


Geometry Instancing

Una escena 3D puede tener múltiples objetos.

A veces esos objetos son distintas **instancias** de un modelo existente.





La idea de repetir los mismos objetos en la pantalla no es propia de 3D. Antes con gráficos 2D ya se hacía.

Grafo de Escena

El grafo de escena es la estructura de datos que nos permite:

- **Describir** la escena de una manera generalizable.
- **Administrar** las instancias de nuestros modelos 3D.

Y, como consecuencia de lo anterior:

- **Componer** transformaciones.
- **Separar** la escena a graficar del código que la grafica.*

*: importante en otras áreas. Por ej., ciencia de datos.

Motivación 🙄

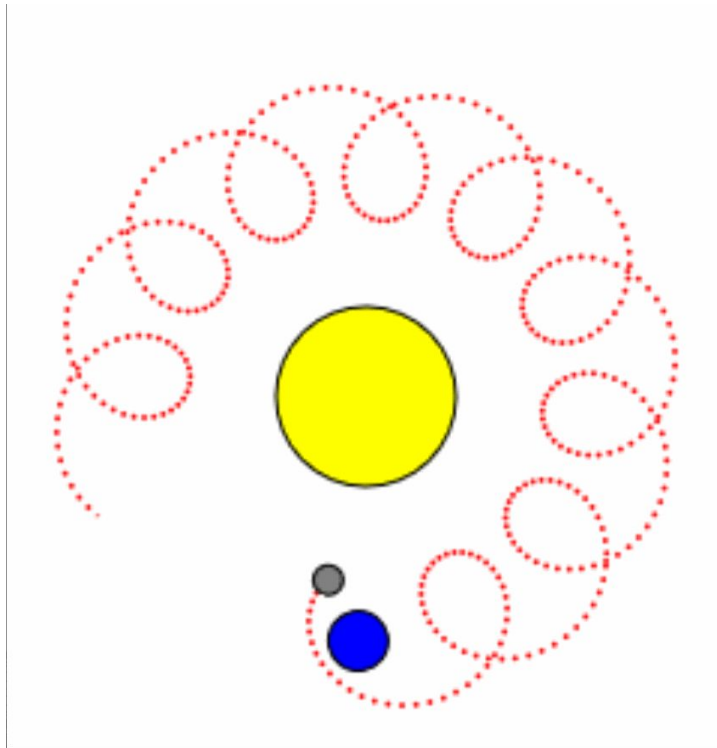
Cuando hablamos de organización no solamente nos referimos a tener los objetos en una lista.

También queremos hacernos cargo de las dependencias entre ellos:

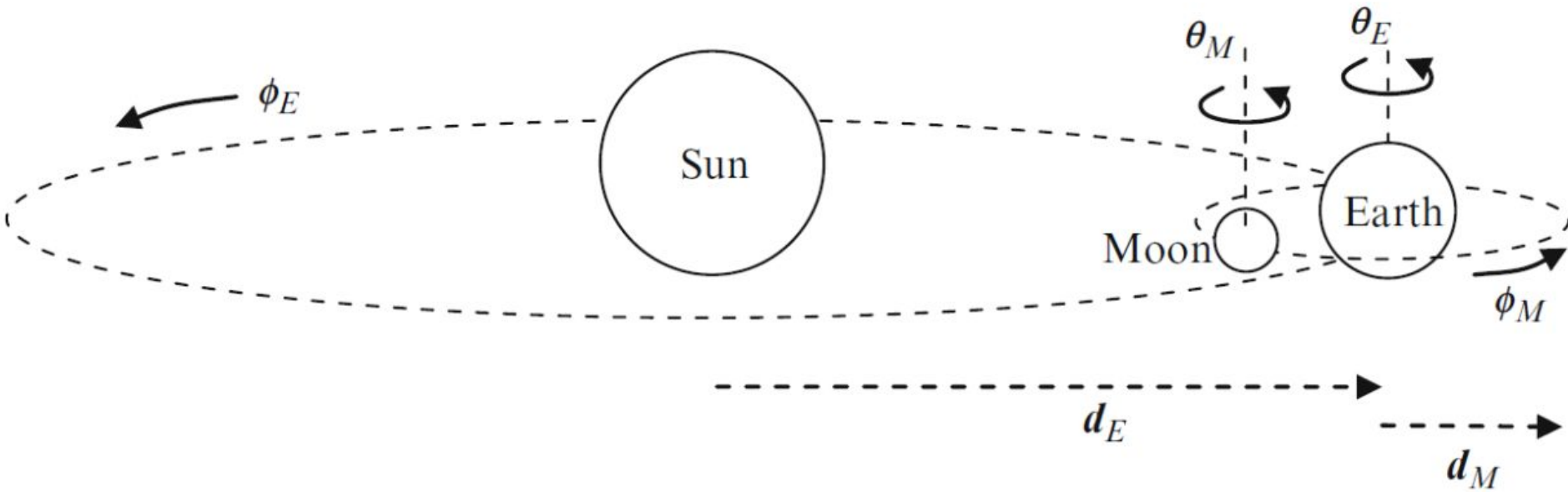
Objetos contenidos dentro de otros

Objetos cuya posición depende de otros

Antes hablamos de composición de transformaciones. Pero, en una escena con miles de objetos, **¿cómo asegurarnos que trabajamos sin equivocarnos en el orden de las transformaciones?**



<https://threejs.org/manual/#en/scenegraph>



Un esquema de las distintas transformaciones que debemos considerar.

Estructura: DAG

Directed Acyclic Graph (red dirigida acíclica)

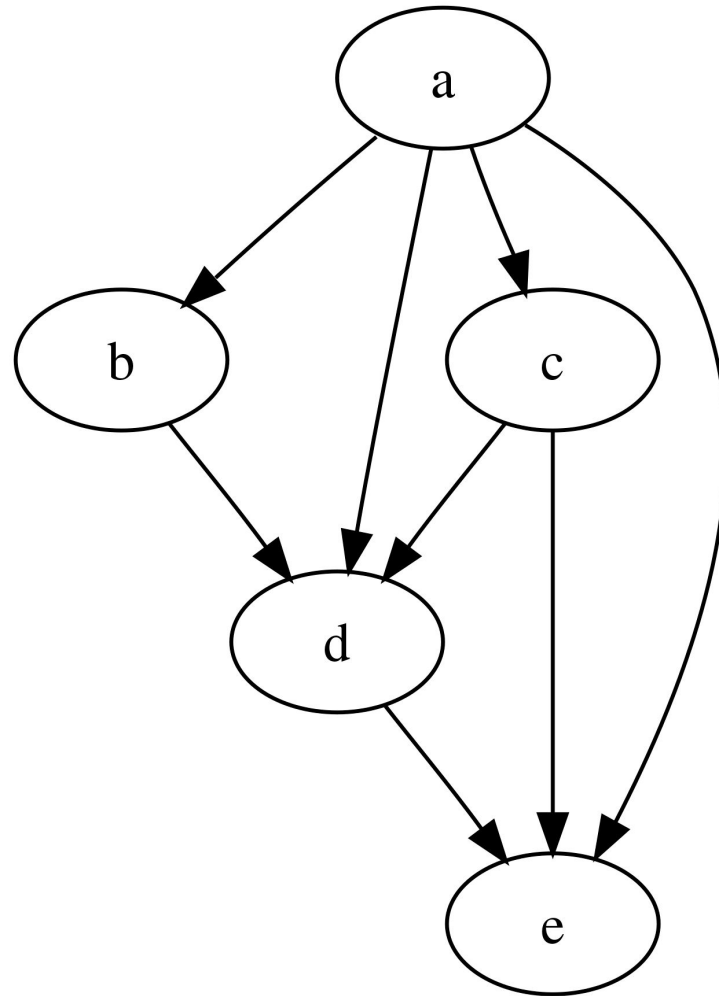
Esta estructura tiene nodos y aristas.

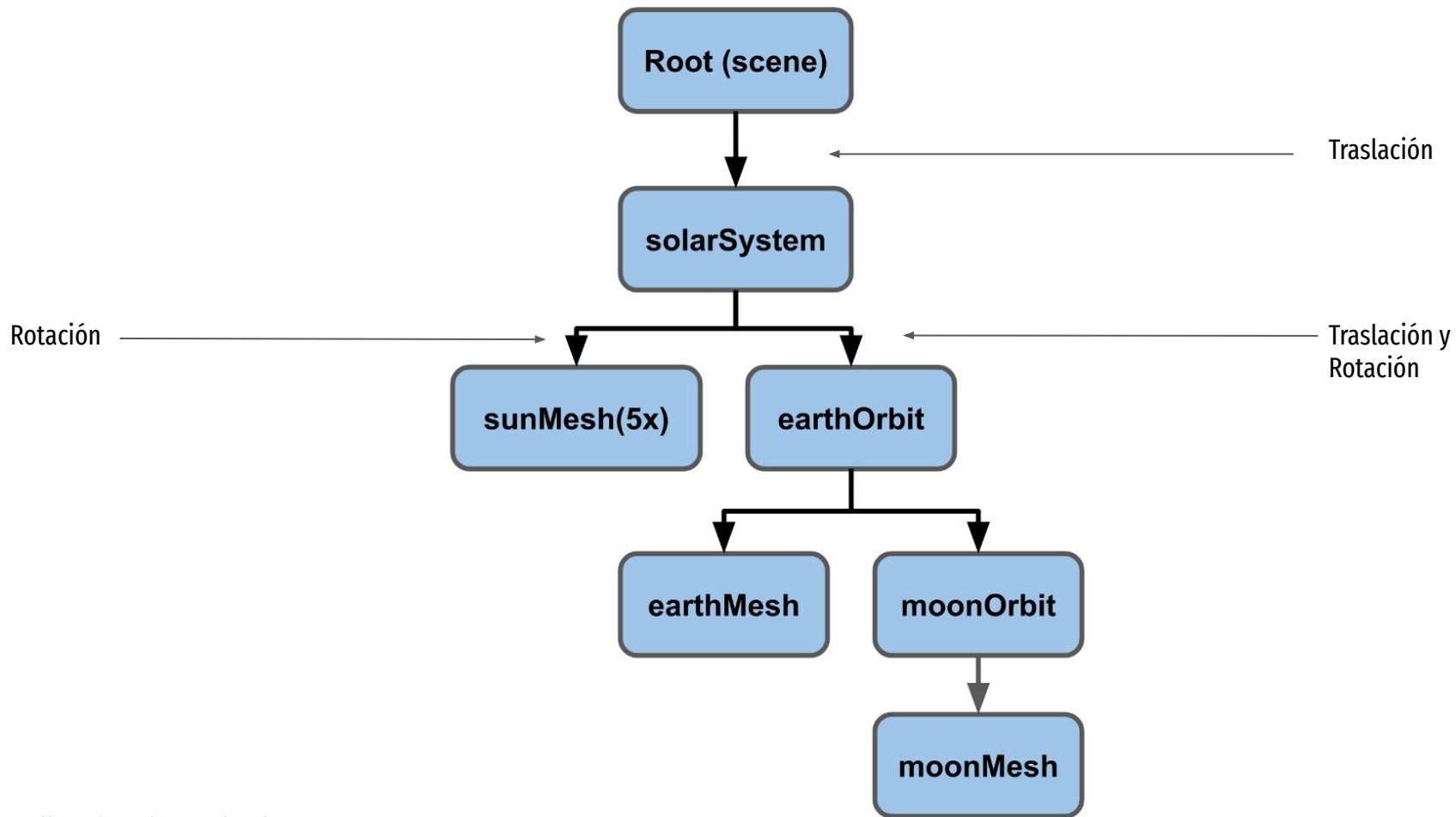
Los nodos son **elementos**, es decir, un objeto 3D o un conjunto de objetos 3D.

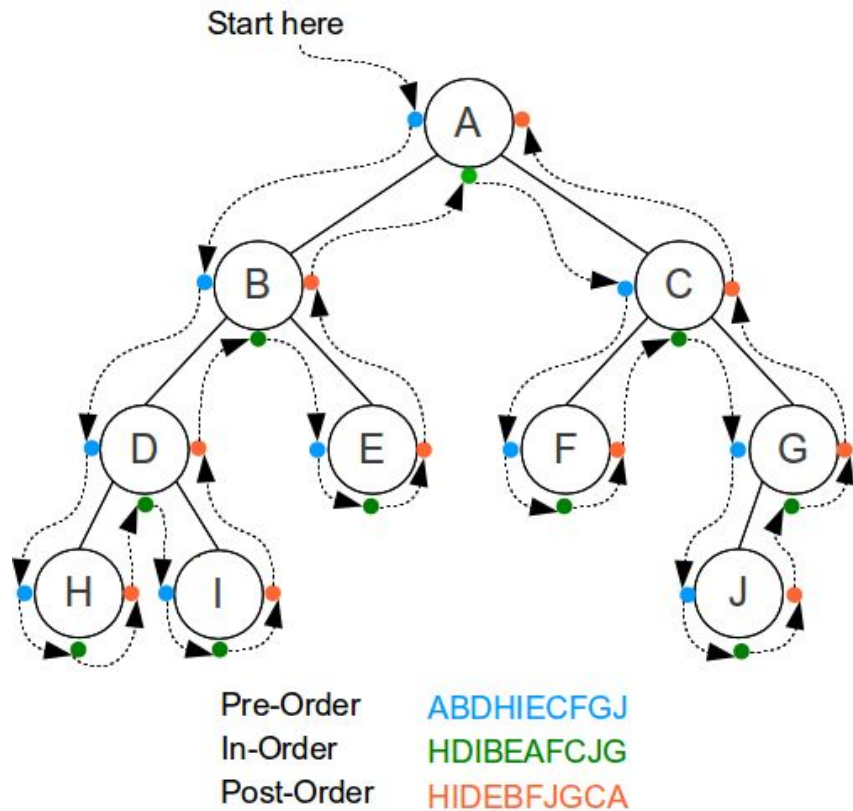
Las aristas, arcos o conexiones son **relaciones**.

¿Cómo implementar esta estructura de datos?

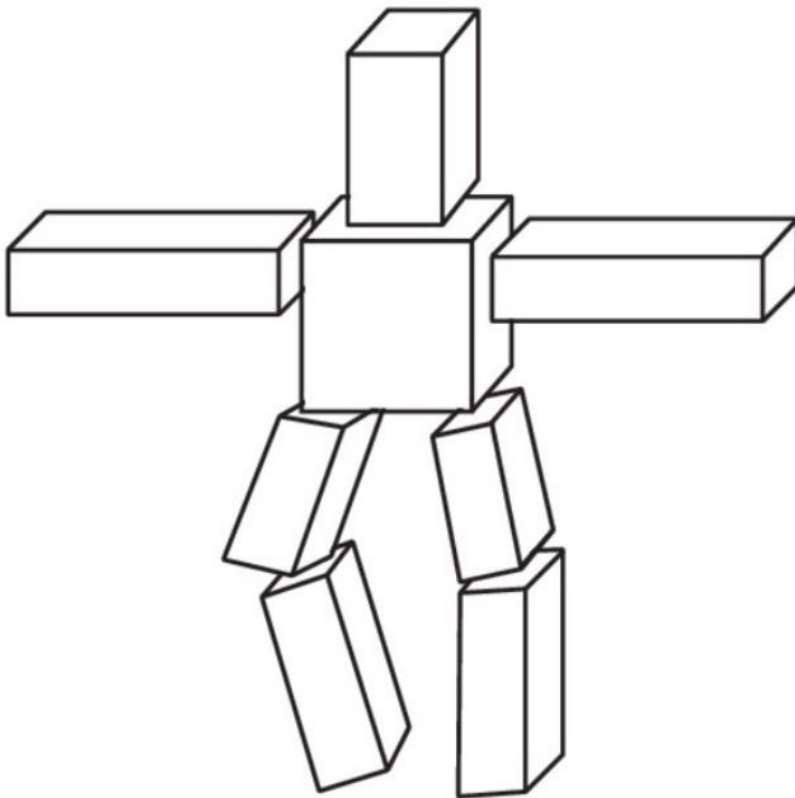
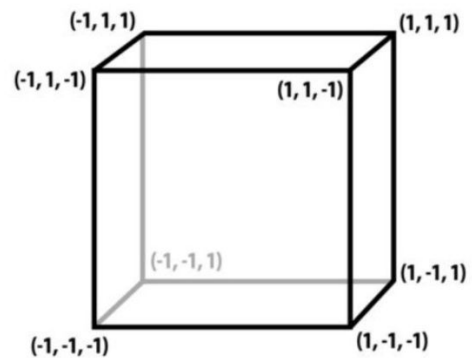
Cada arista tiene una matriz de transformación asociada. A medida que recorremos el árbol desde la raíz, las matrices se van multiplicando.



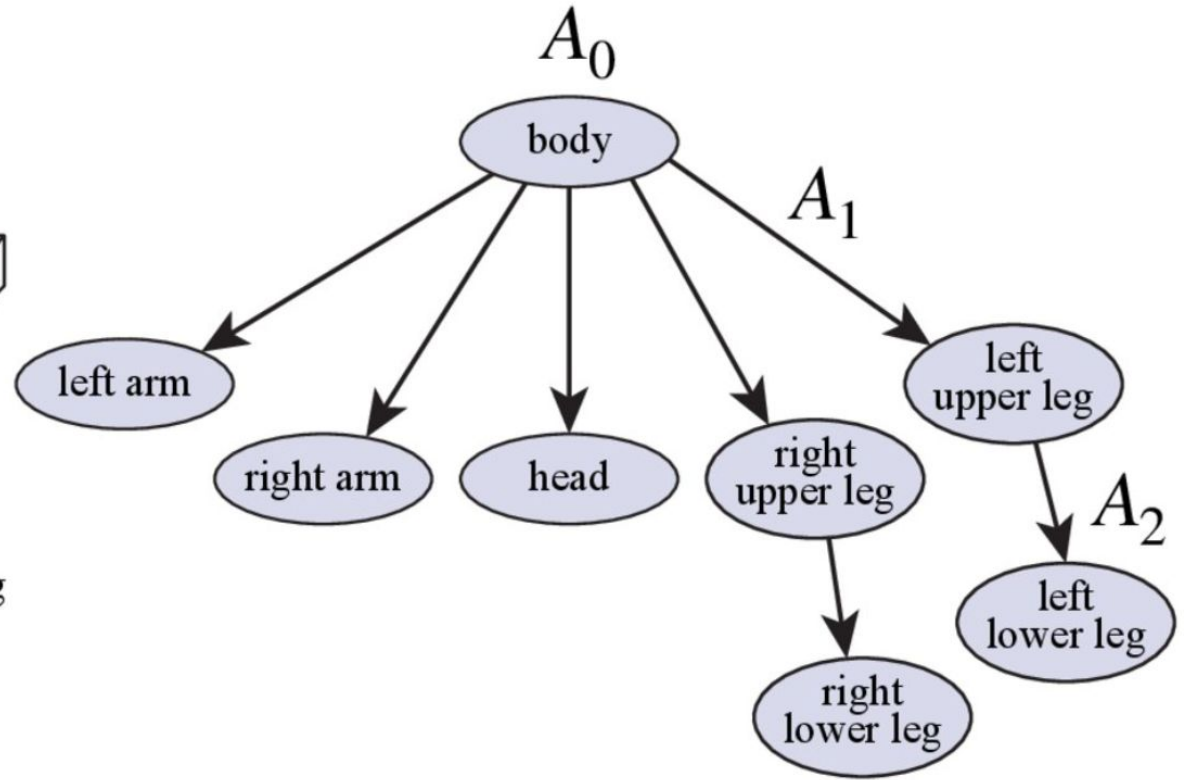
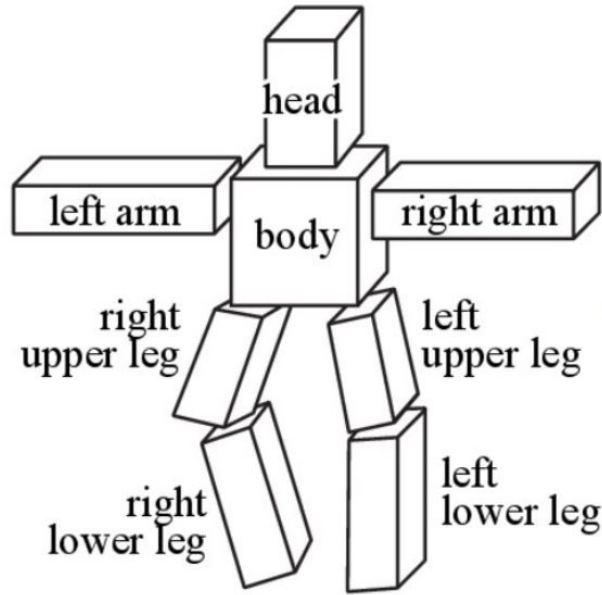




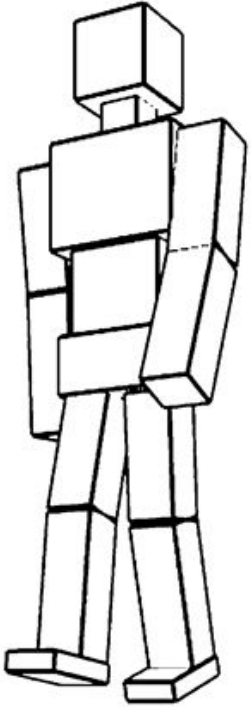
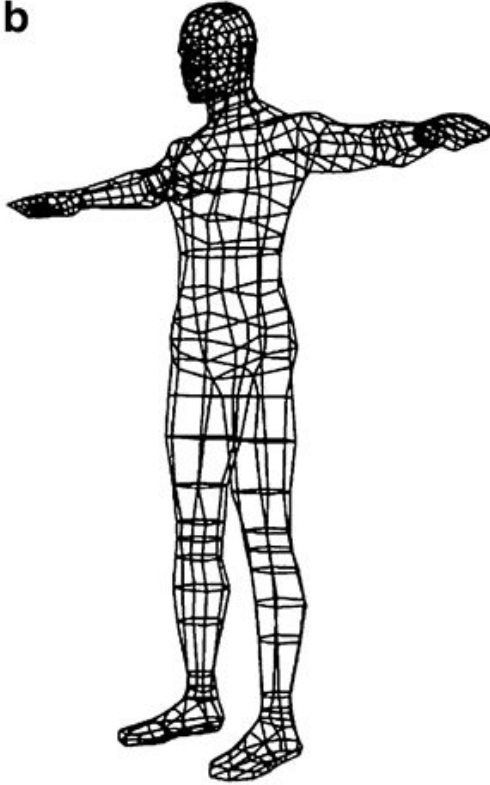
El orden en que se recorre la red es importante. Para componer matrices, el mejor es pre-order. **¿Cómo implementar la multiplicación de matrices?**



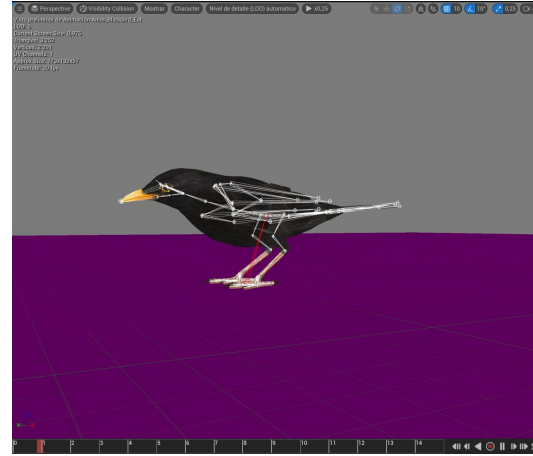
Otro ejemplo: ser de cajas.

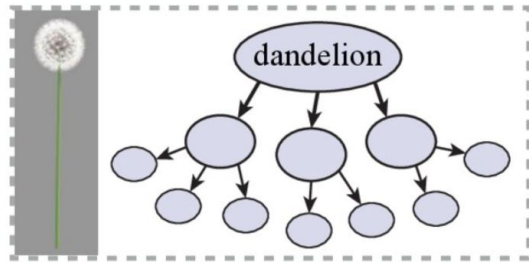


¿Es cada caja un objeto diferente? Notamos que en realidad pueden ser todas la misma caja pero con una transformación distinta.

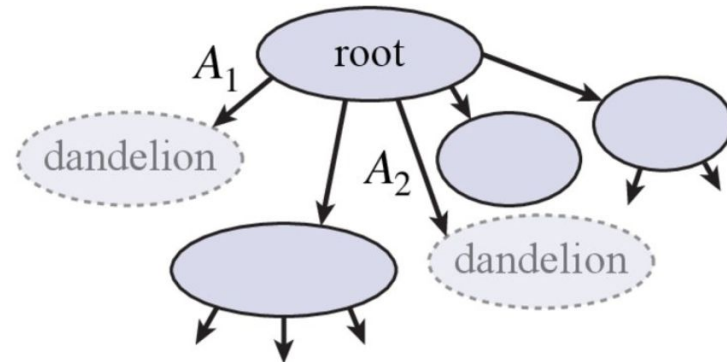
a**b**

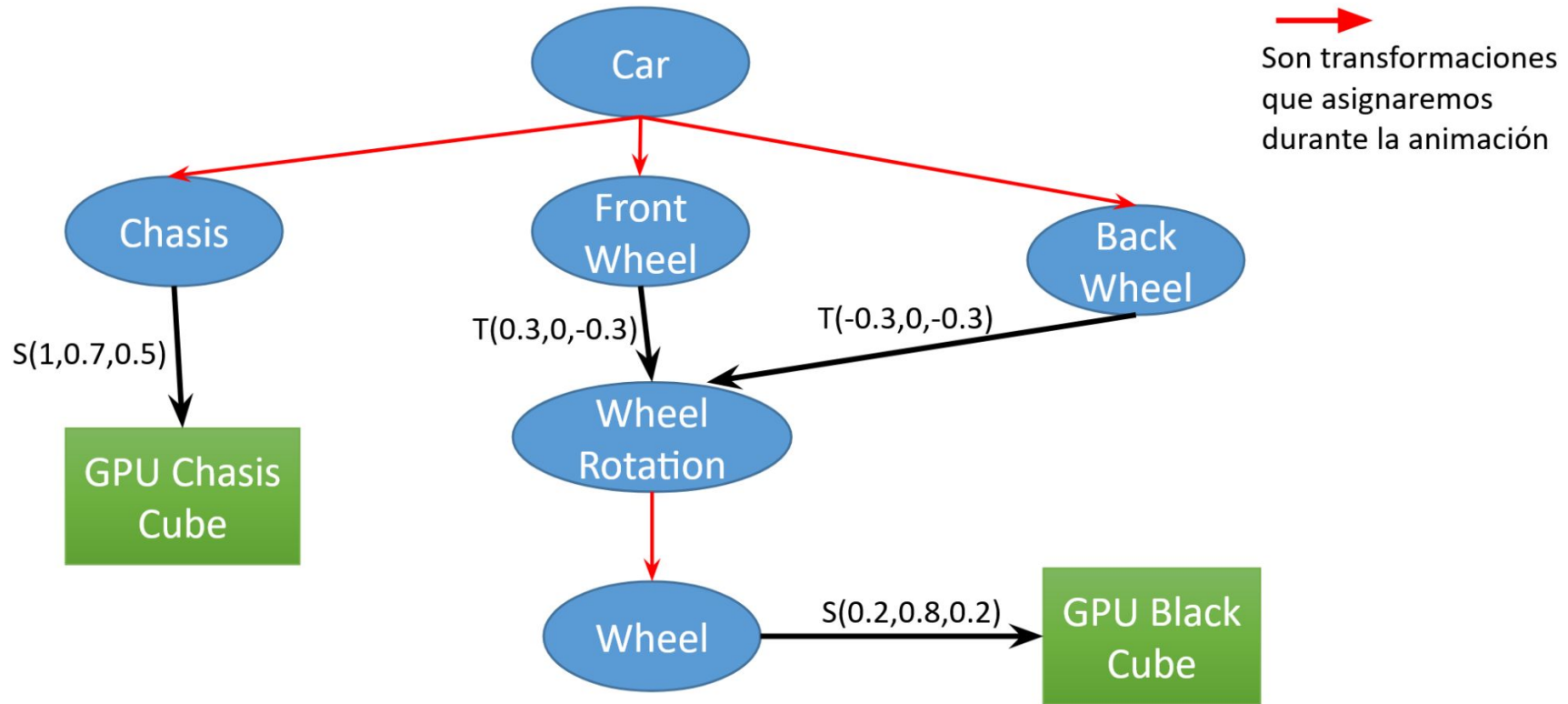
Estos esqueletos son la base de las animaciones en la actualidad. La técnica se llama **skinning**.





Ahora podemos retomar el concepto de **instanciamiento** (instanting). Un objeto en el grafo es un **puntero** o **referencia** a otro objeto (¡que puede ser un grafo!)





Show me the code!
`examples_for_lessons/scene_graphs/ex_solar3d.py`

¿Preguntas?