ingeniería multimedia

Seminario 2

Árbol de la escena

Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

¿Qué es un árbol?

¿Qué es un árbol?

¿Qué es un árbol?

¿  ?

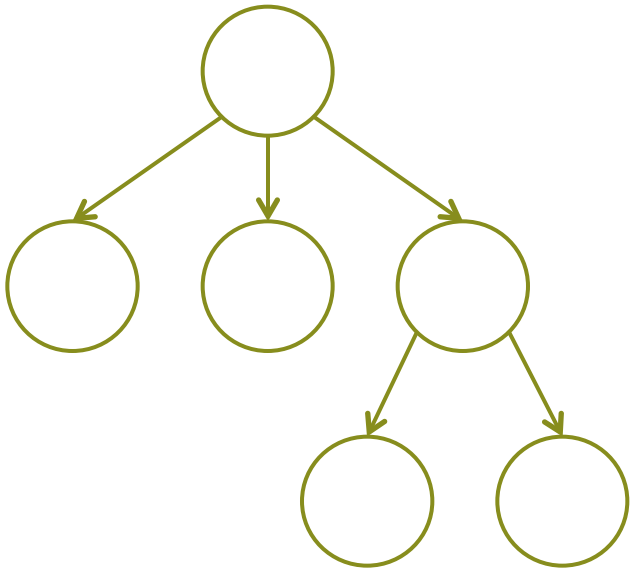
¿Qué es un árbol?

¿Qué es un árbol?

(Estructura de datos)

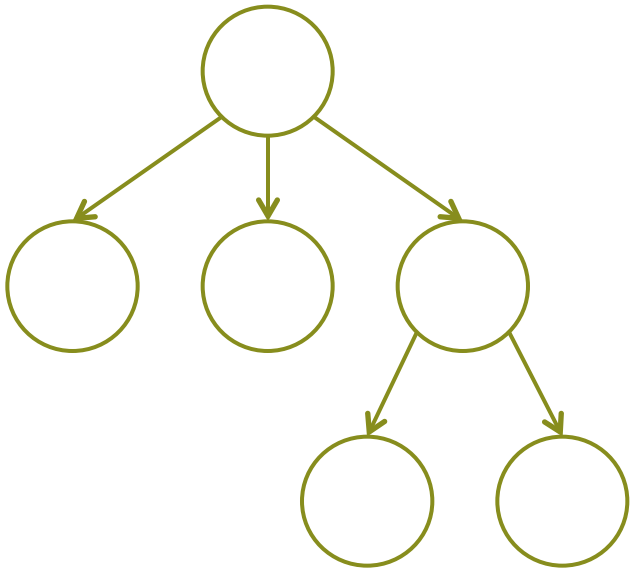
Árbol

- Estructura de datos no lineal formada por nodos
- Cada nodo puede apuntar a uno o varios nodos
- Cada nodo sólo puede ser apuntado por otro nodo



Árbol

- Tipos de nodos:
  - Raíz: Nodo que no tiene padre
  - Hoja: Nodo que no tiene hijo
  - Rama: Nodo que tiene padre e hijo
- Relaciones entre nodos:
  - A es hijo de B si A es apuntado por B
  - B es padre de A si B apunta a A
  - A y B son hermanos si tienen un mismo padre



Árbol de la escena

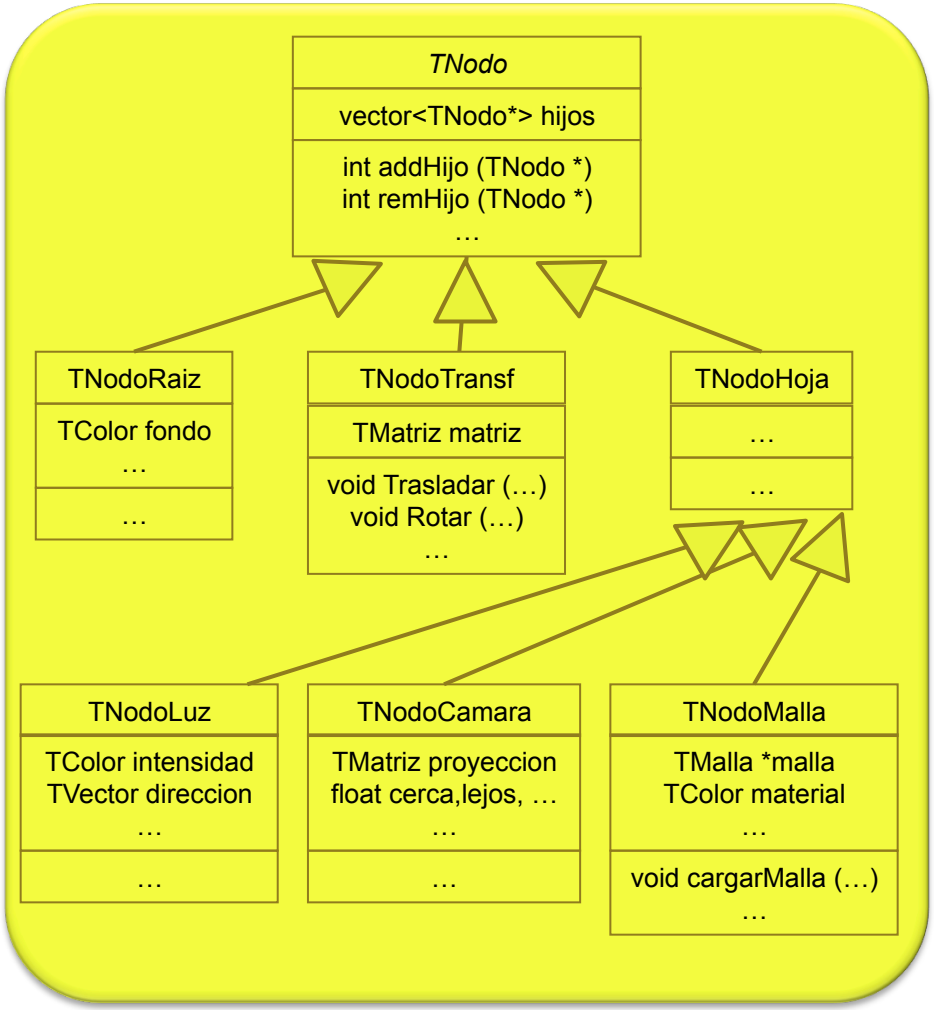
- Árbol que contiene los datos que definen un escenario virtual y controlan su proceso de dibujado
- El árbol debe contener todos los datos necesarios para dibujar la escena:
  - Geometría
  - Texturas
  - Animaciones
  - Luces
  - Cámaras
  - Transformaciones…

Modelos de árbol de la escena

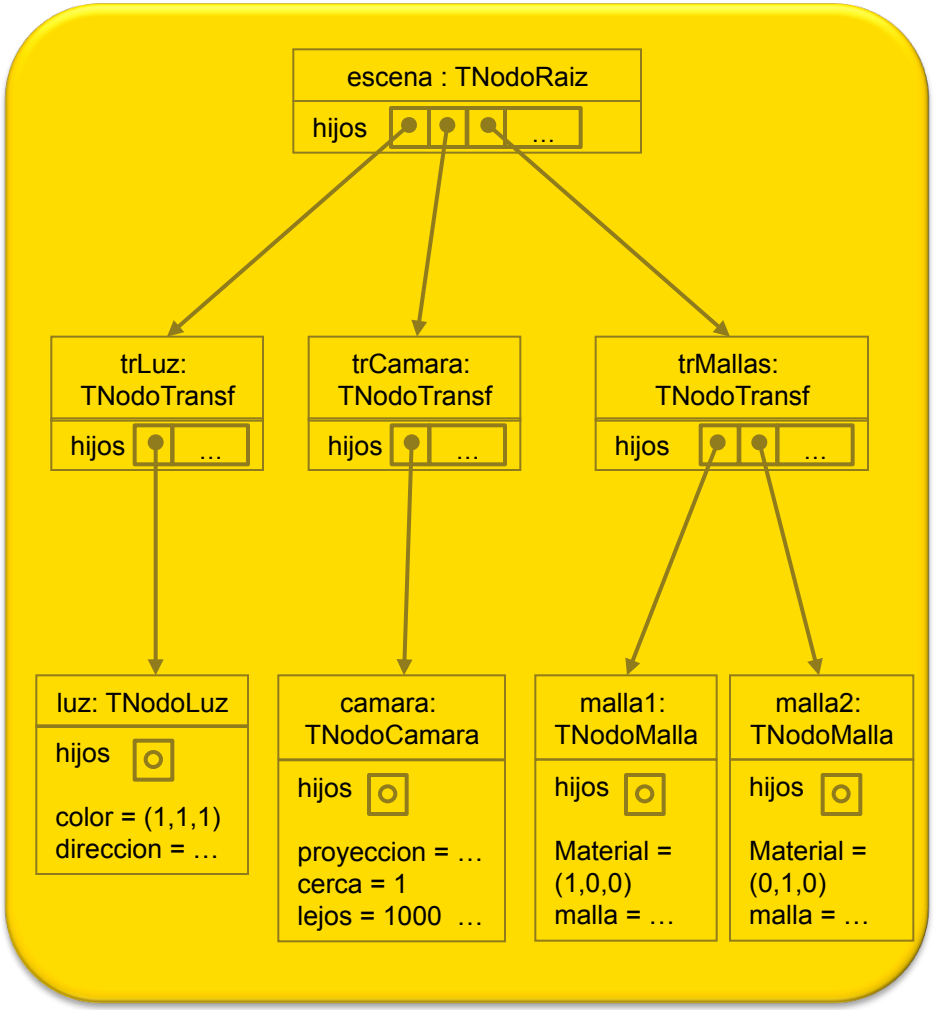
- Árbol y contenidos unidos
  - Las clases de contenido son tipos de nodos del árbol, integrados en la misma jerarquía de herencia
- Árbol y contenido separados
  - Las clases de los nodos del árbol y las clases de los contenidos están en diferentes jerarquías

Árbol y contenidos unidos

Jerarquía de clases

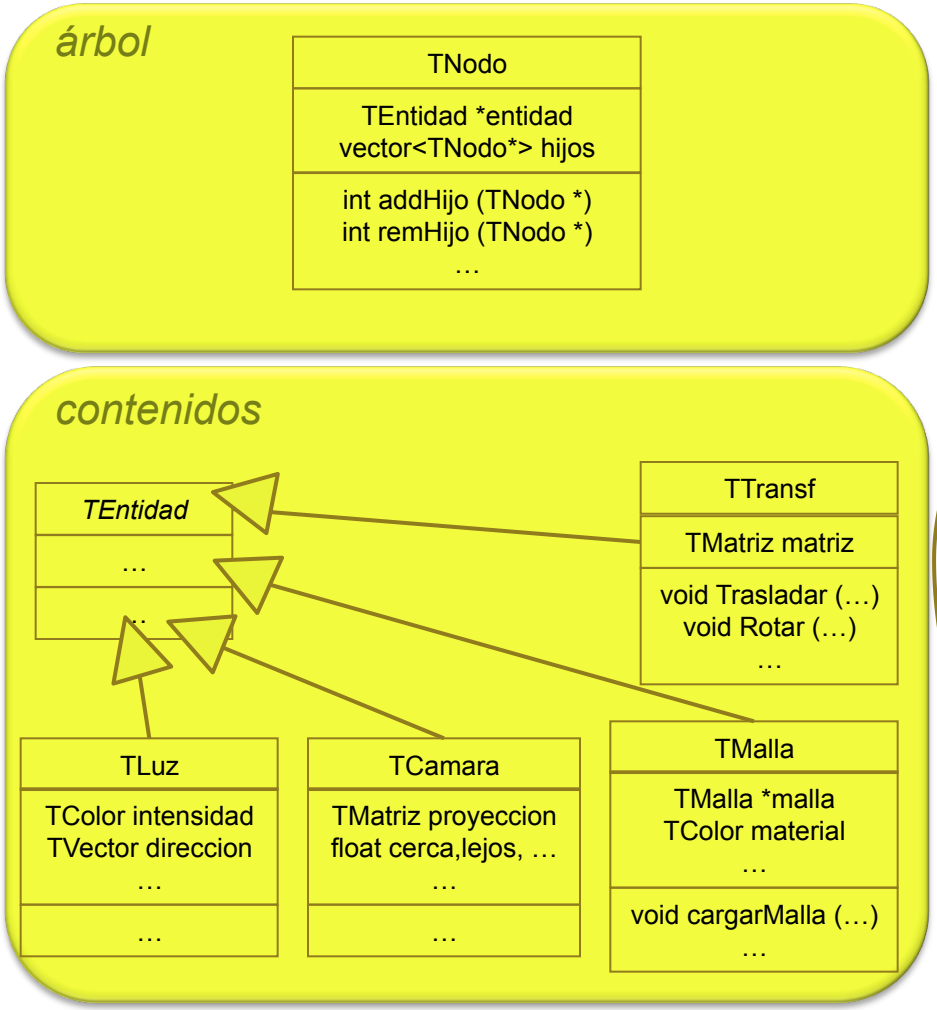


Árbol de la escena

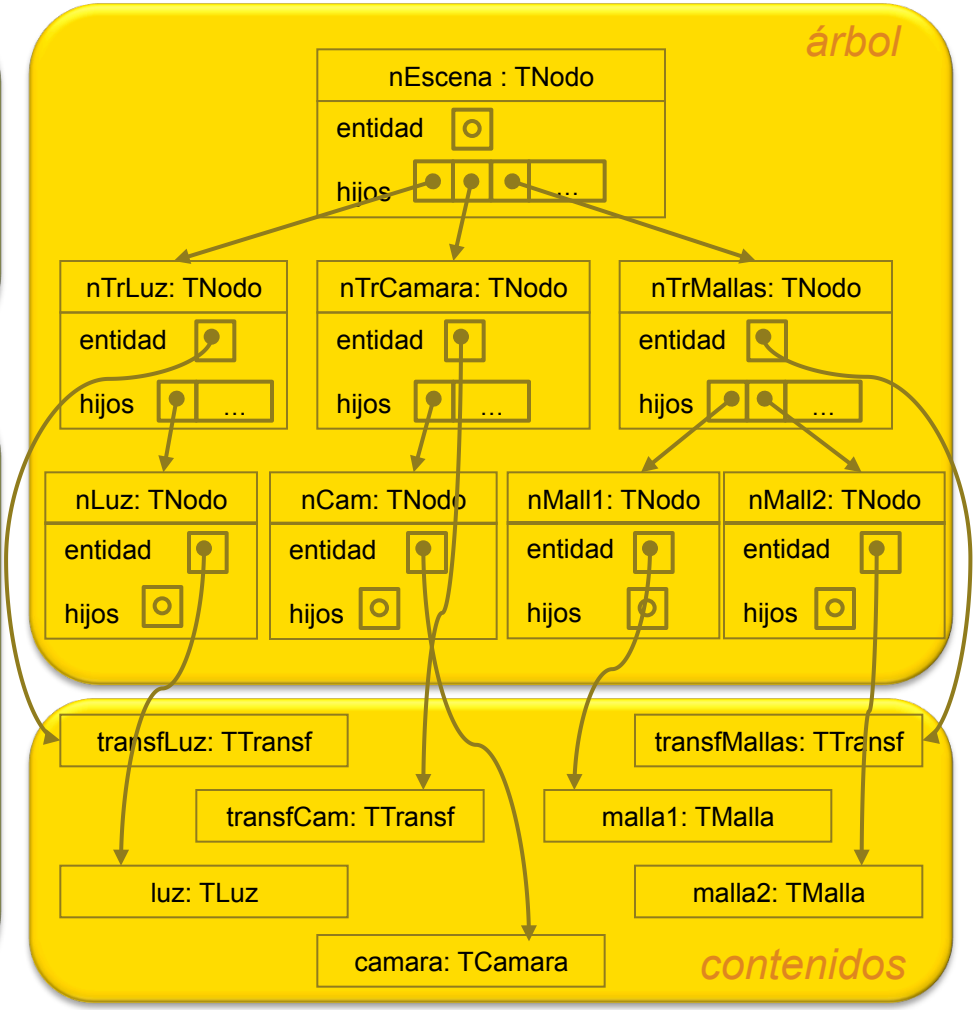


Árbol y contenidos separados

Jerarquía de clases



Árbol de la escena



¿Qué modelo?

¿Qué modelo elegirías tú?

¿Qué modelo?

- Árbol y contenidos unidos
  - Difícil de extender, debemos diseñar muy bien las clases desde el principio
- Árbol y contenido separados
  - Más fácil de extender, podemos empezar por un motor sencillo e añadirle nuevas características

¿?

¿Qué modelo?

- Árbol y contenidos unidos
  - Difícil de extender, debemos diseñar muy bien las clases desde el principio
- Árbol y contenido separados
  - Más fácil de extender, podemos empezar por un motor sencillo e añadirle nuevas características

Nodos

- Puede ser una única clase
- Atributos
  - Puntero a un objeto de tipo entidad
  - Punteros a los nodos hijos (array, lista…)
  - Puntero al nodo padre (¿para qué?)
- Métodos
  - Gestionar la entidad (get/set)
  - Gestionar el padre (get)
  - Gestionar los hijos (add/remove)
  - Dibujar la entidad asociada en la pantalla (draw)
  - Otros

TNodo
TEntidad *entidad vector <TNodo*> hijos TNodo *padre
int addHijo (TNodo *) int remHijo (TNodo *) bool setEntidad (TEntidad *) TEntidad *getEntidad () TNodo *getPadre () void draw ()

Entidades

- Debe haber una única clase de la que deriven todas las entidades
- Esta clase debe ser virtual
- Es probable que no tenga atributos y sólo dos métodos virtuales para dibujar (beginDraw y endDraw)

TEntidad *(virtual)*
void beginDraw (); (virtual) *void endDraw (); (virtual)*

Entidad transformación

- Deriva de la entidad
- Atributos
  - Matriz de transformación (4x4 reales)
- Métodos
  - Gestionar la matriz: cargar identidad, cargar una matriz dada, multiplicar matriz por vector o por otra matriz, trasponer o invertir la matriz …
  - Operaciones de transformación básicas (acumulativas): trasladar, rotar, escalar …
  - Sobrecargar métodos de dibujado

TTransform : TEntidad

TMatriz4x4 matriz

void identidad ()
void cargar (TMatriz4x4)
void trasponer ()
…
void trasladar (float, float, float)
void rotar (float, float, float, float)
…
void beginDraw ();
void endDraw ();

Entidades luz y cámara

- Derivan de la entidad
- Atributos
  - Atributos de la luz o de la cámara
- Métodos
  - Gestionar los atributos de la luz y de la cámara
  - Sobrecargar métodos de dibujado

TLuz : TEntidad
TColor intensidad
void setIntensidad (TColor) TColor getIntensidad (); … void beginDraw (); void endDraw ();

TCamara: TEntidad
Bool esPerspectiva float cercano, lejano …
void setPerspectiva (float, float…) void setParalela (float, float…) … void beginDraw (); void endDraw ();

Entidades visualizables

- Derivan de la entidad
- Son objetos visualizables (mallas, etc)
- Atributos (mallas)
  - Puntero al recurso malla (obtenido del gestor de recursos), que incluye vértices, polígonos, materiales…
- Métodos
  - Gestionar la malla y sus atributos (cargar …)
  - Sobrecargar métodos de dibujado

TMalla: TEntidad

TRecursoMalla *malla

void cargarMalla (TFichero)

…

void beginDraw ();

void endDraw ();

¿Cómo se visualiza?

Los métodos de dibujado

TNodo :: draw

- Es el método que permite visualizar la escena
- Llama a los métodos de dibujado de las entidades asociadas
- Así independizamos la estructura de la escena de la librería con que se visualiza: para cambiar de librería basta con cambiar los métodos de dibujado de las entidades

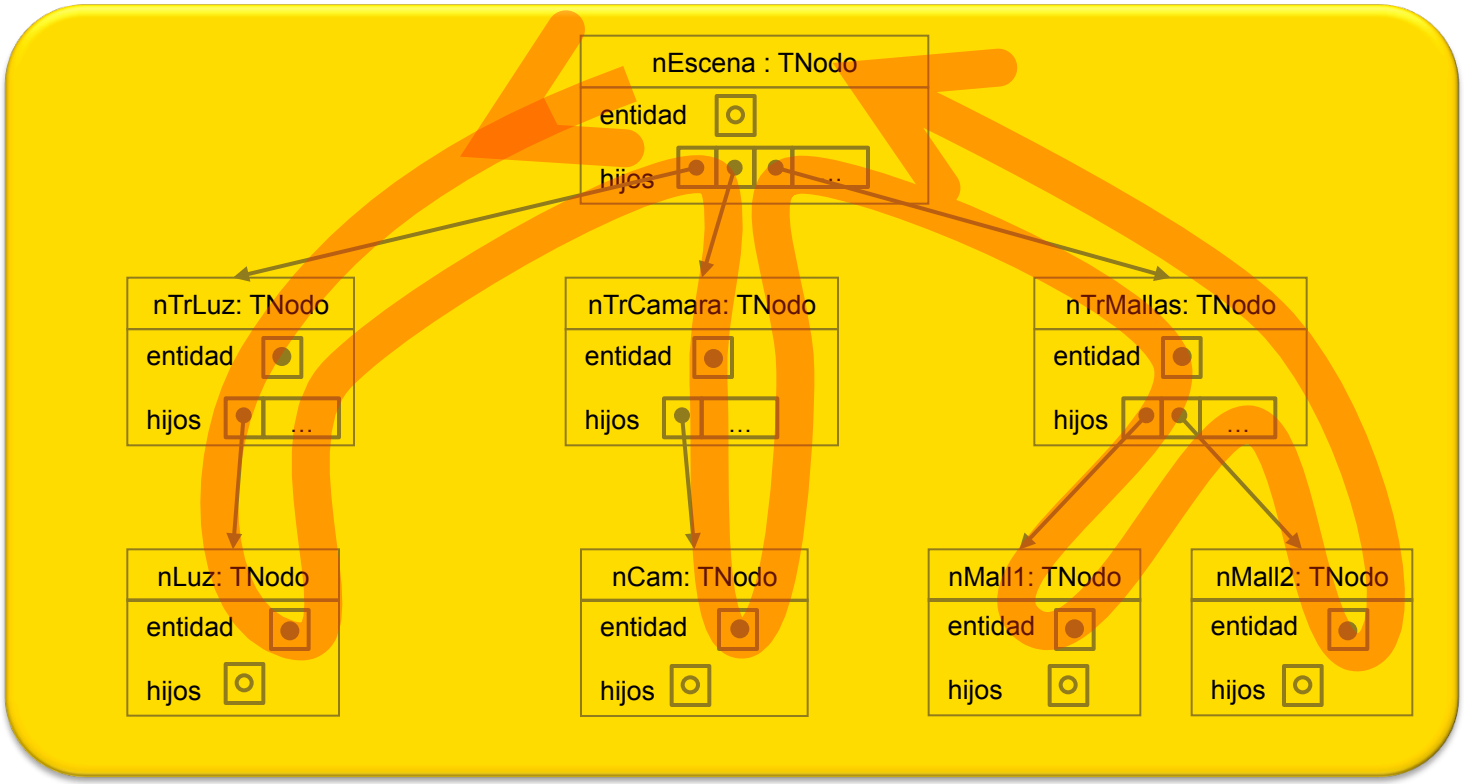
TNodo :: draw

- El dibujado de la escena parte del método draw del nodo escena (raíz del árbol)
- El método draw de cada nodo llama al método beginDraw de la entidad asociada
- A continuación, el dibujado se desencadena en cascada, llamando al dibujado de los nodos hijos de cada nodo padre
- Por último se termina el dibujado llamando al método endDraw de la entidad

```
TNodo :: draw  (…)
{
        entidad -> beginDraw (…);
        para cada nodo hijo i
                hijo[i] -> draw (…);
        entidad -> endDraw (…);
}
```

TNodo :: draw

- De esta forma, los nodos del árbol se recorren en preorden.



Visualización de las entidades

- Cada entidad tiene dos métodos para el dibujado: beginDraw y endDraw
- beginDraw se ejecuta cuando se baja un nivel en el árbol de la escena
- endDraw se ejecuta cuando se sube un nivel en el árbol de la escena
- Normalmente sólo las transformaciones necesitan hacer algo en el método endDraw cuando se sube un nivel del árbol
- Las demás entidades, normalmente tendrán un método endDraw vacío

TTransform :: beginDraw / endDraw

- Manejan las matrices de transformación utilizando una pila (en nuestro caso, la pila de matrices de OpenGL)
  - Apilar al bajar un nivel del árbol
  - Desapilar al subir un nivel del árbol

```
TTransform :: beginDraw  (…)
{
        Apilar matriz actual
        Multiplicar la matriz de la transformación a la matriz actual
}
```

```
TTransform :: endDraw  (…)
{
        Desapilar matriz y ponerla como  actual
}
```

beginDraw de entidades visualizables

- Dependerá del tipo de entidad
- Por ejemplo, si es una malla, habrá que llamar al método de dibujado del recurso que se encargará del dibujado en OpenGL de los polígonos a partir de los vértices, las normales y las coordenadas de textura (lo veremos con el Gestor)

```
TMalla :: beginDraw  (…)
{
        malla->draw (…)
}
```

Visualización de TCamara y TLuz

- La posición y orientación de la cámara y de las luces se maneja a través de las transformaciones de nodo padre, como para el resto de entidades.
- Sin embargo, las cámaras y las luces se manejan de forma especial, puesto que se pueden encontrar en cualquier nodo del árbol y, sin embargo, al dibujar la escena deben definirse antes de dibujar el resto de entidades.
- Por eso, los métodos beginDraw y endDraw de las cámaras y las luces suelen estar vacíos.