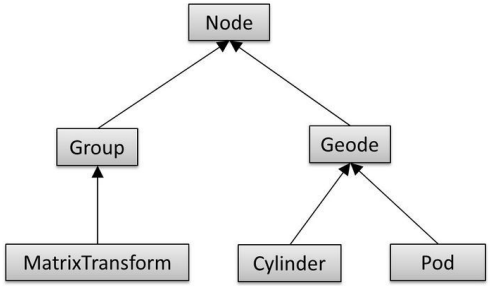
# CSE 167 Discussion #6

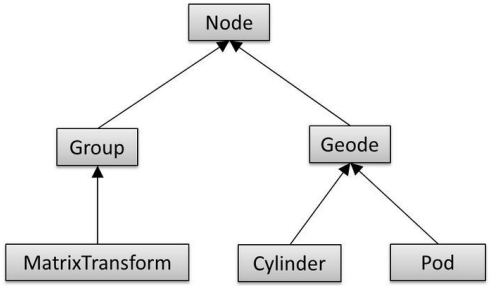
Never scene that before

# Scene Graph Engine



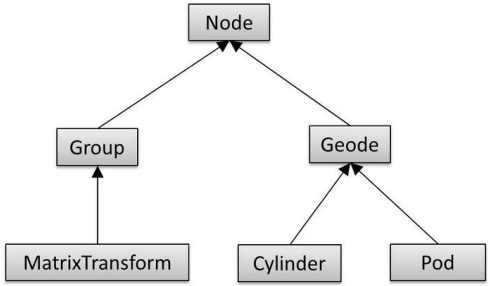

- Node
  - virtual void update(glm::mat4 C) = 0;
  - virtual void draw(...) = 0;

# Scene Graph Engine



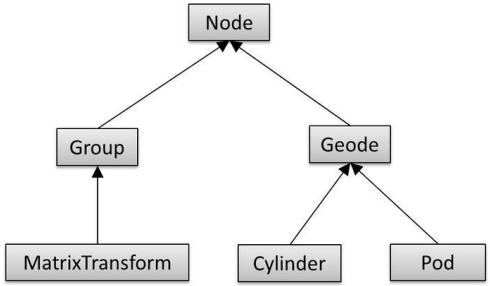

- Group
  - std::list<Node*> children;  //Can we do better?
  - void addChild(...);
  - void removeChild(...);
  - virtual void update(glm::mat4 C);
    - Loop through all Node* in children and call update(C) passing in glm::mat4 C as an argument
  - virtual void draw();  //Loop over children

# Scene Graph Engine



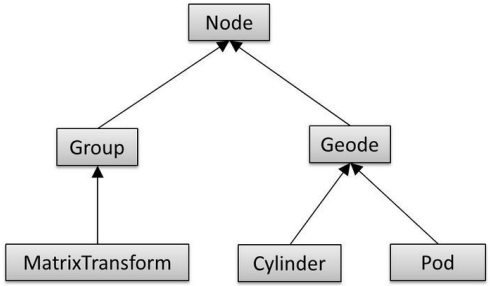

- MatrixTransform
  - glm::mat4 M;
  - void update(glm::mat4 C);
    - Multiplies glm::mat4 M with glm::mat4 C
      - In what order??
    - Passes resulting glm::mat4 down to children
      - How?

# Scene Graph Engine



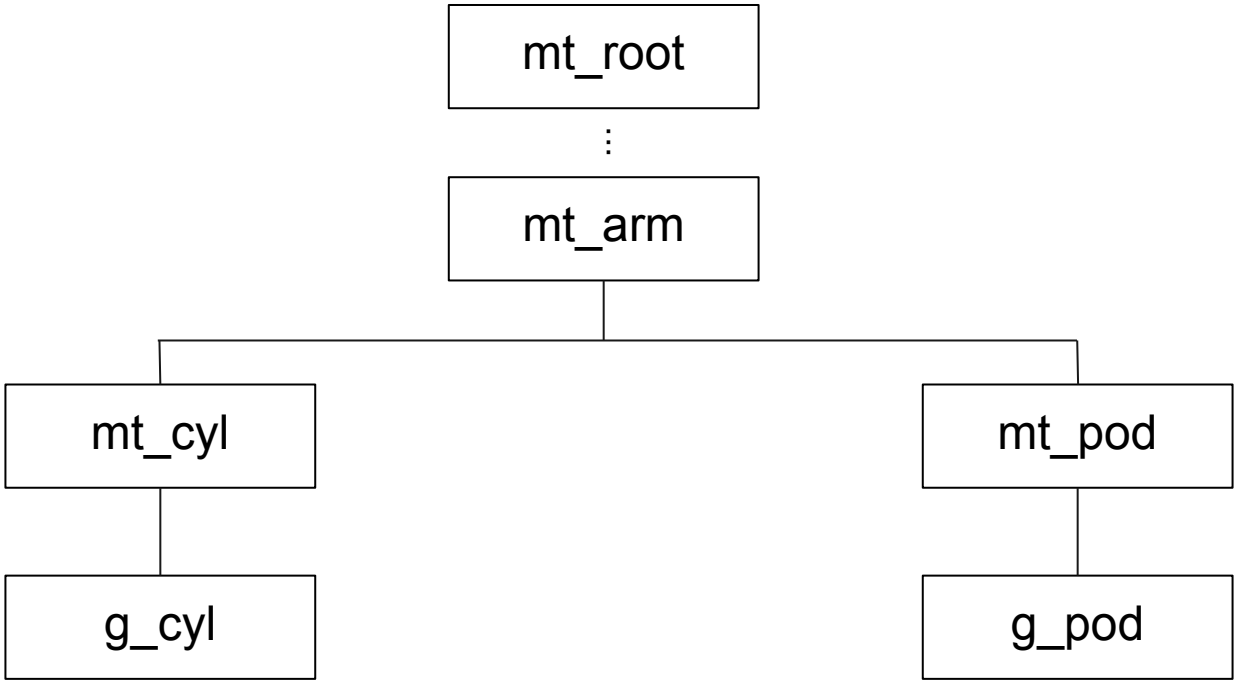

- MatrixTransform
  - glm::mat4 M;
  - void update(glm::mat4 C);
    - Multiplies glm::mat4 M with glm::mat4 C
      - C*M
    - Passes resulting glm::mat4 down to children
      - Group::update(...);

# Scene Graph Engine

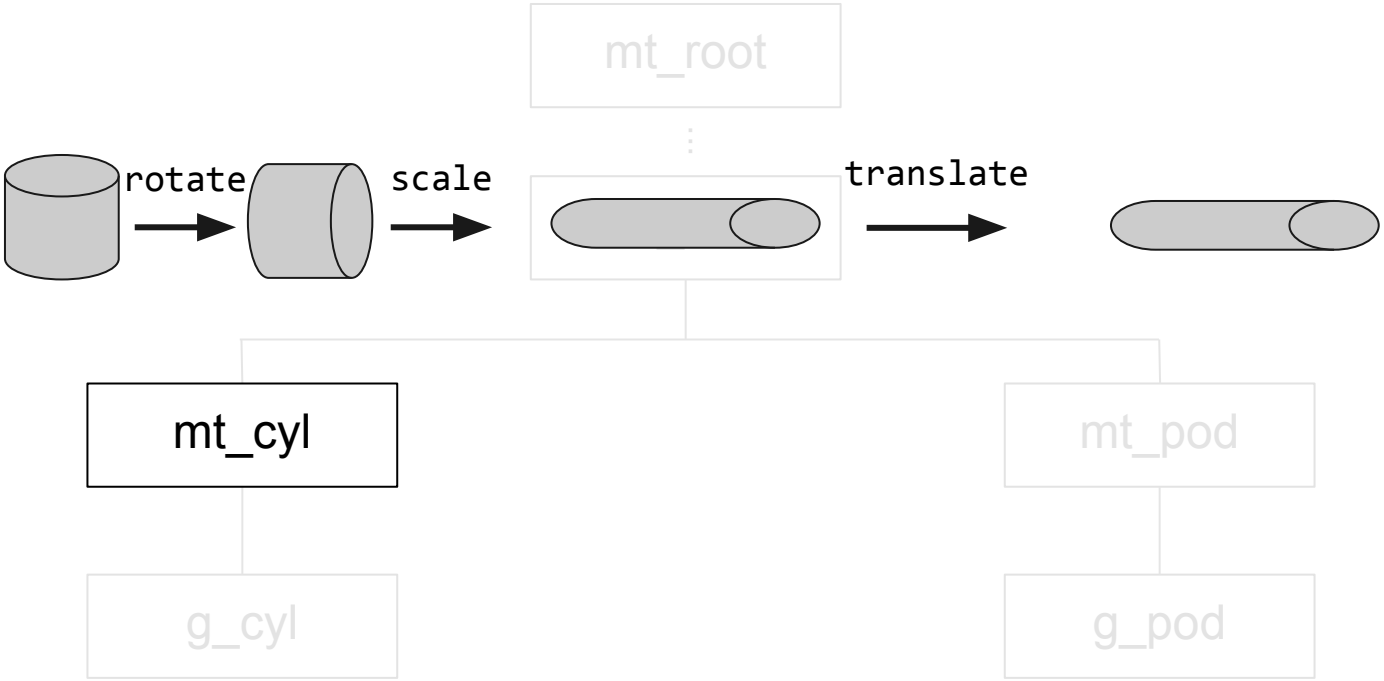



- Geode
  - OBJObject *toDraw;
  - glm::mat4 M;
  - void update(glm::mat4 C);
    - Our Geode now has to hold the matrix passed down to it (M = C)
  - void draw();
    - Call draw on the toDraw pointer

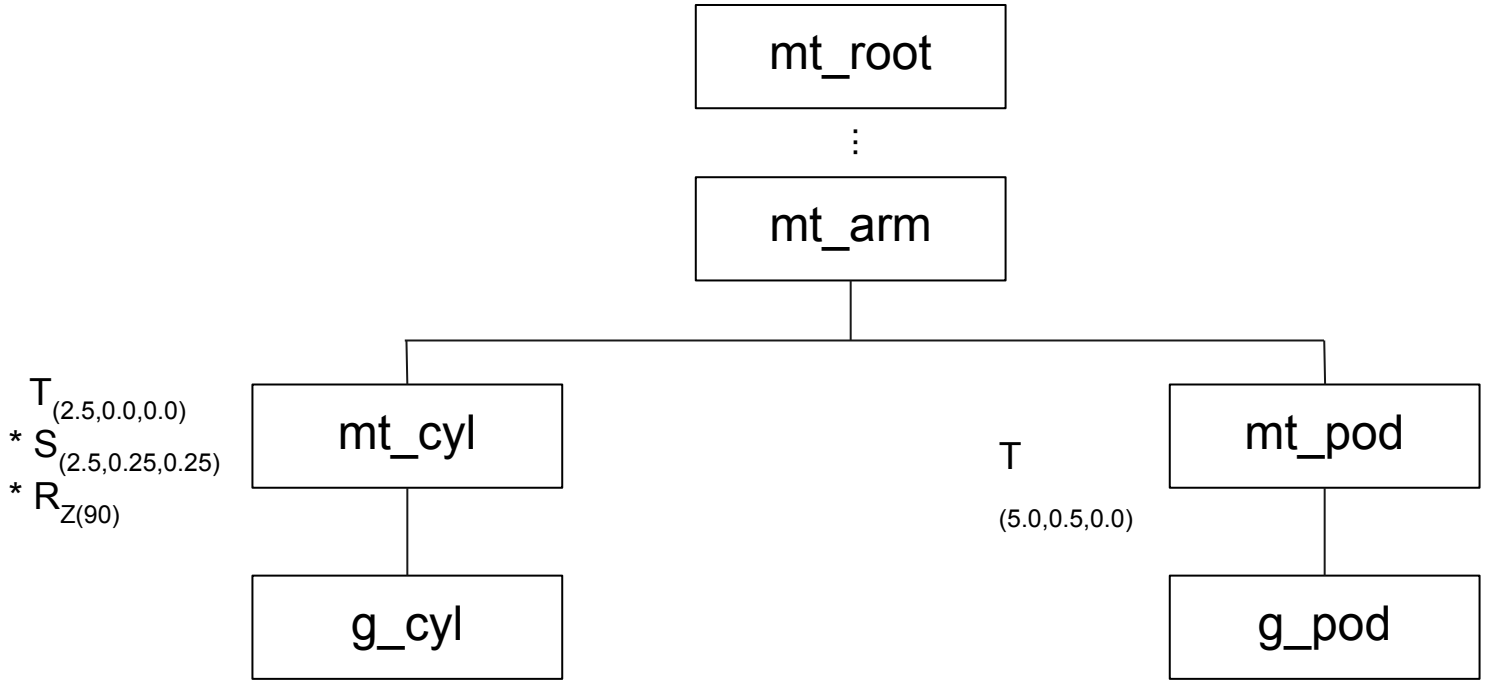
# A Single Arm



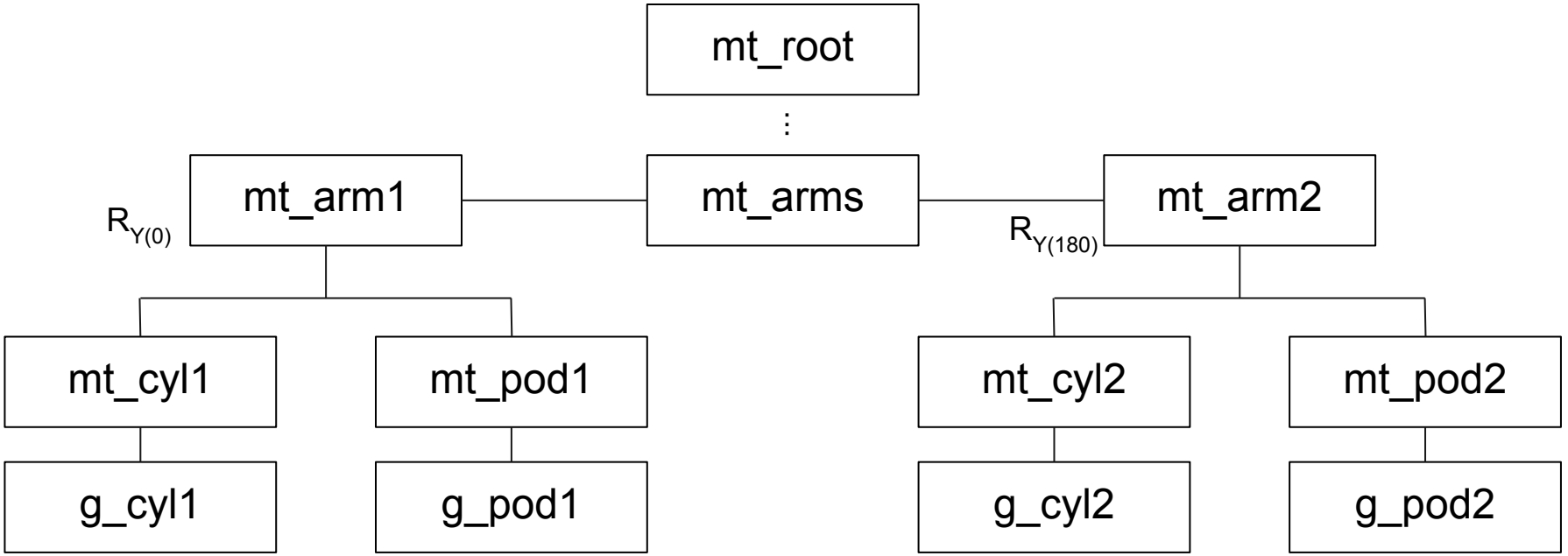
# Cylinder Matrix



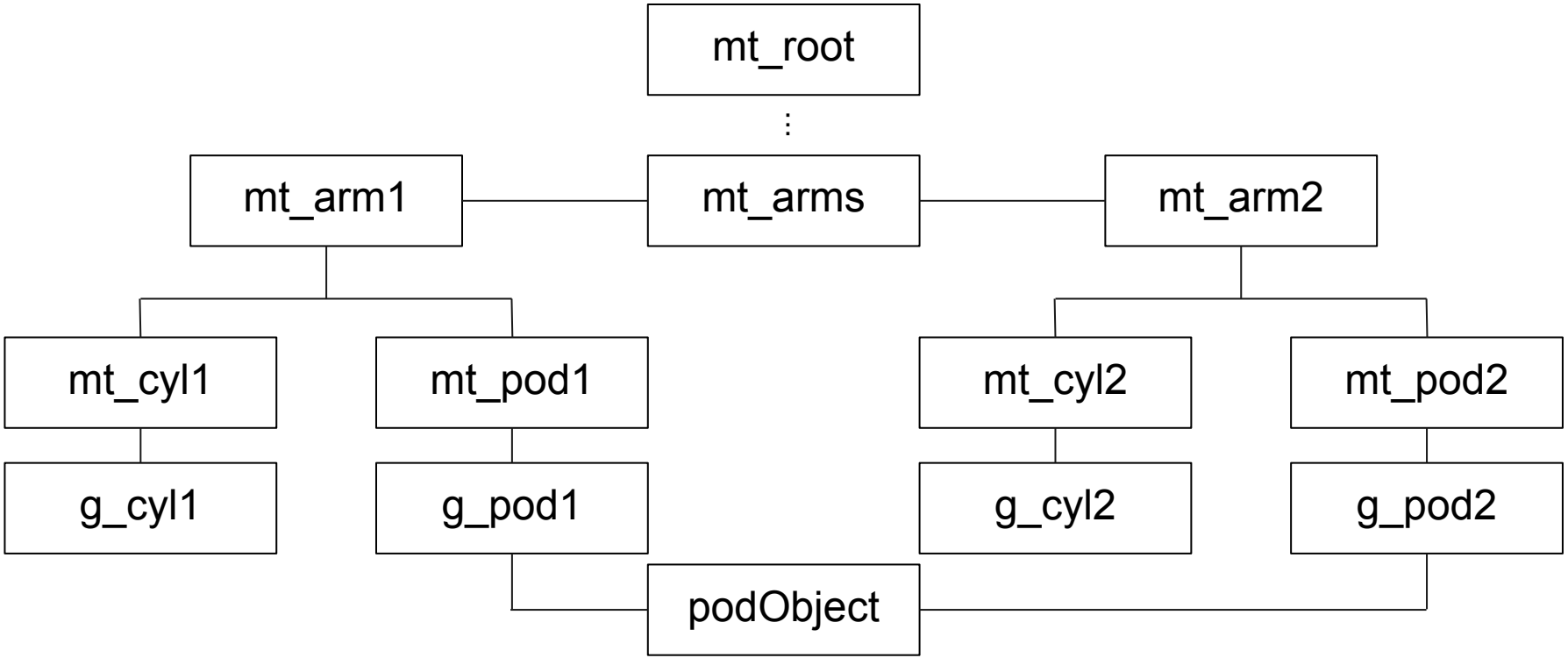
# A Single Arm


mt_root
mt_arm
T (2.5,0.0,0.0)
* S (2.5,0.25,0.25)
* R Z(90)
mt_cyl
g_cyl
T
(5.0,0.5,0.0)
mt_pod
g_pod

# Two Arms


mt_root
mt_arms
mt_arm1
mt_arm2
$R_{Y(0)}$
$R_{Y(180)}$
mt_cyl1
mt_pod1
mt_cyl2
mt_pod2
g_cyl1
g_pod1
g_cyl2
g_pod2

# Geometry Instancing



# Bear Grylls Cam

- We have to make the camera follow the bear
- How do we get the position of the bear?
- How do we set the position of the camera?

# Bear With Me

- We need to set `Window::V`
  - Window::V is actually $C^{-1}$
  - We did this inversion in the midterm!
  - $C^{-1}$ = `(g_bear.R)`$^T$ `* (-g_bear.T + offset)`