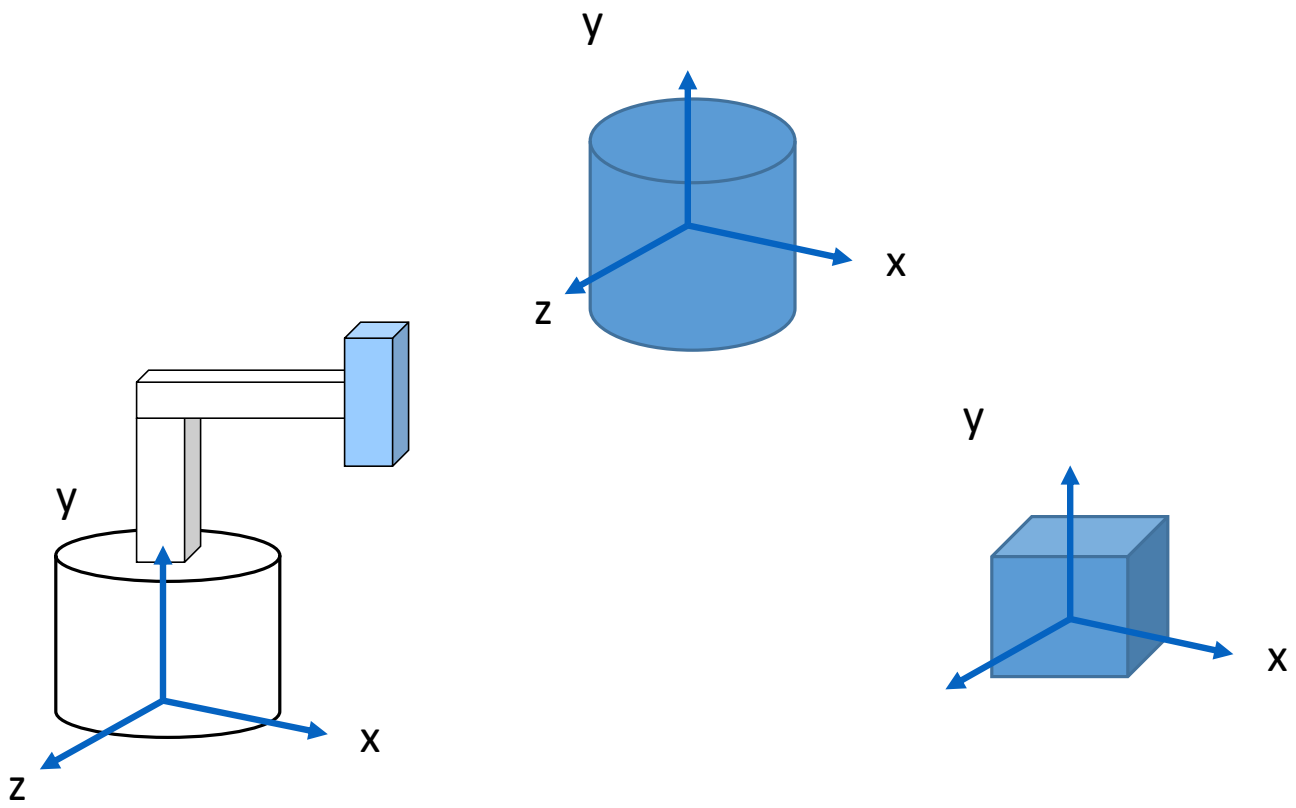
ลำดับชั้นของวัตถุ
Hierarchy of Objects

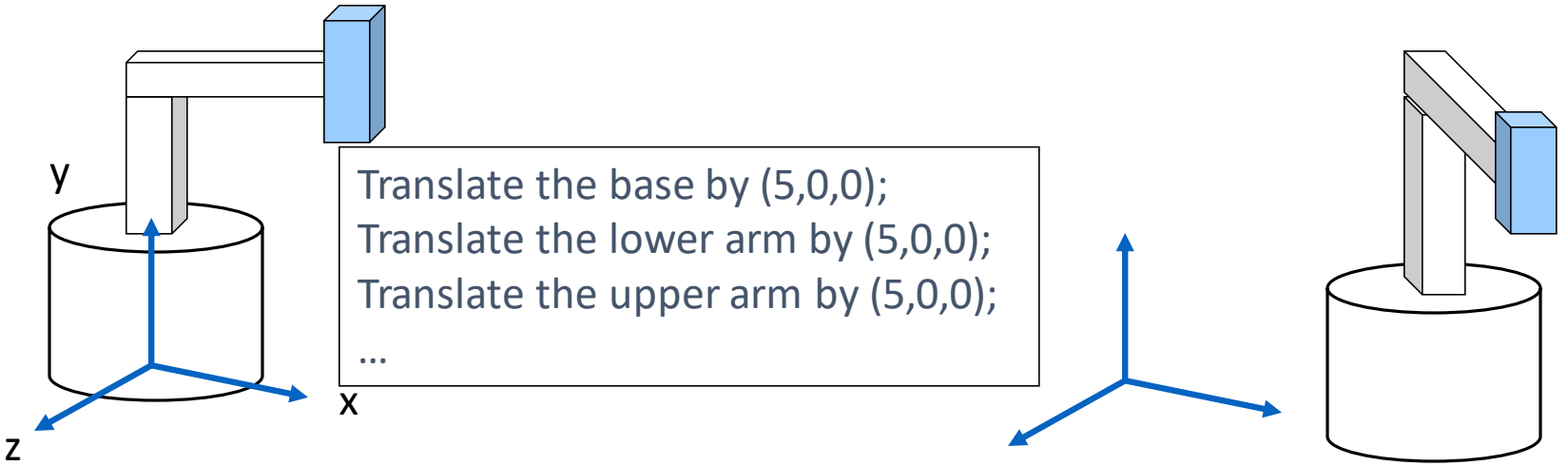
ฉาก (scene)



การแปลงของวัตถุหลายๆ ชิ้น

- วิธีการระบุการแปลงของวัตถุ
  - (1) แต่ละวัตถุมีการแปลงของตัวเองอ้างอิงกับจุดกำเนิด

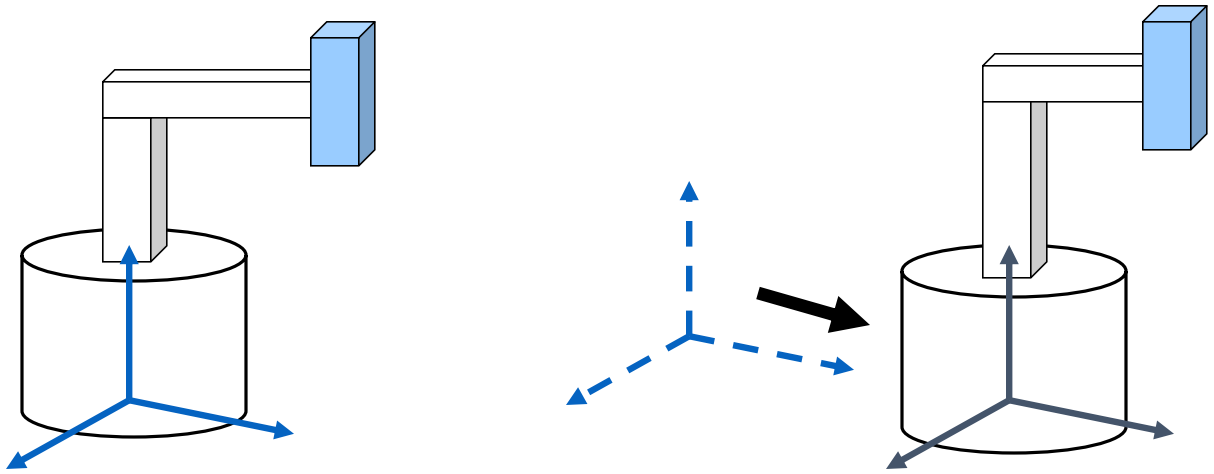
    ไม่เหมาะกับ OpenGL



การแปลงแบบสัมพัทธ์

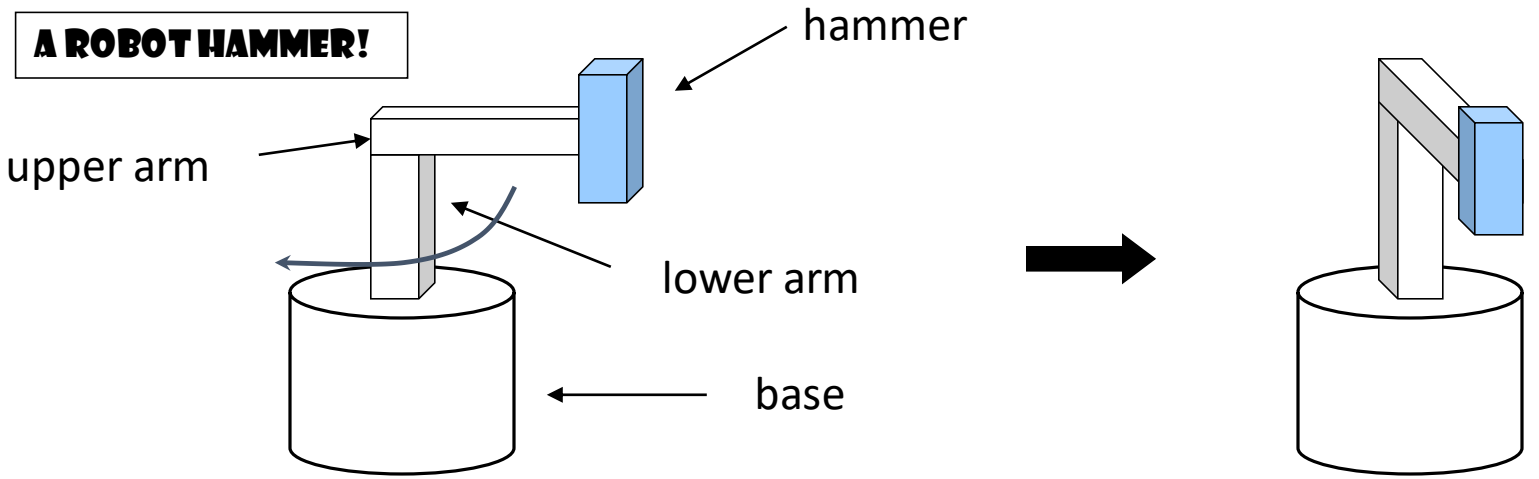
ดีและง่ายกว่า:

(2) การแปลงแบบสัมพัทธ์: ระบุการแปลงของแต่ละวัตถุอ้างอิงกับ parent ของมัน



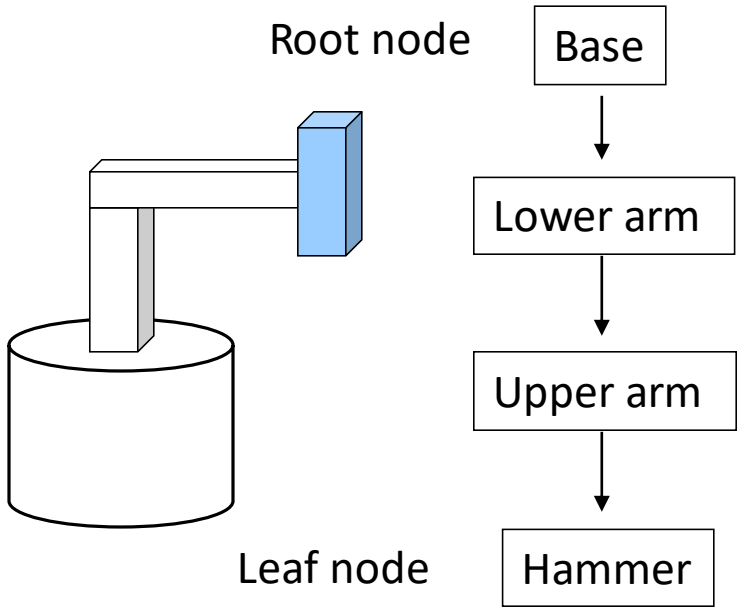
ความสัมพันธ์ระหว่างวัตถุ

- ในฉากๆ หนึ่งมักจะประกอบด้วยวัตถุหลายๆ ชิ้น
- คุณสมบัติของวัตถุชิ้นหนึ่งอาจจะอ้างอิงกับวัตถุชิ้นอื่นได้ เช่น ตำแหน่ง การหมุน ขนาด



การแทนแบบลำดับชั้น – กราฟของฉาก (scene graph)

- อธิบายความเกี่ยวข้องของวัตถุด้วยโครงสร้างข้อมูลต้นไม้

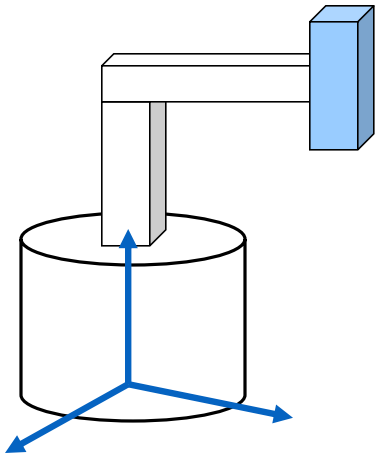


ตำแหน่งและการหมุนของวัตถุขึ้นกับปม (node) parent, grand parent , grand grand parent, …

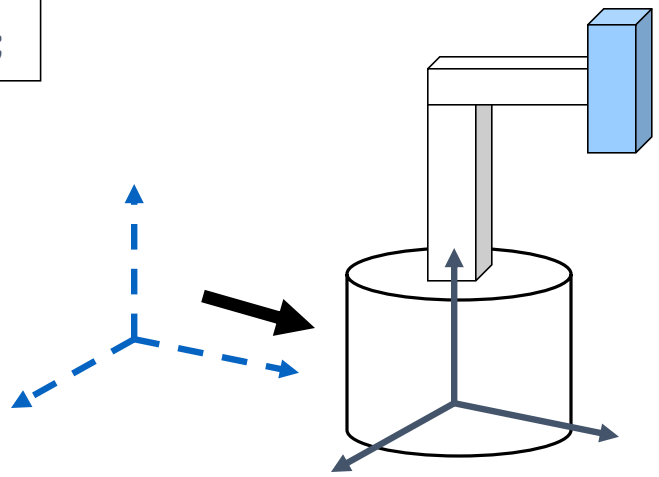
การแทนวัตถุแบบนี้เรียกว่า Scene Graph

การแปลงแบบสัมพัทธ์

ระบุการแปลงของวัตถุด้วยอ้างอิงกับวัตถุ parent

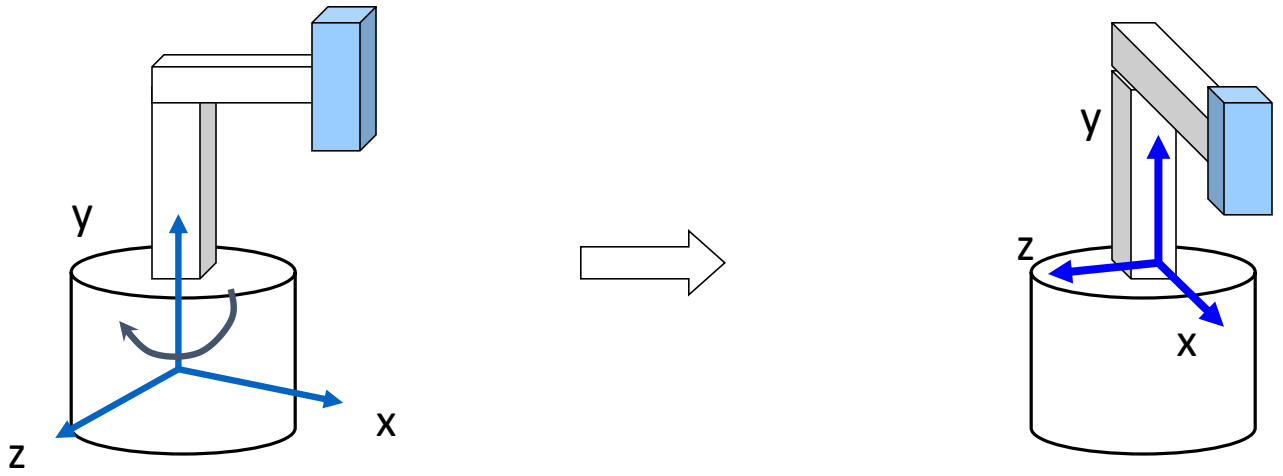


Step 1: Translate base and its descendants by (5,0,0);



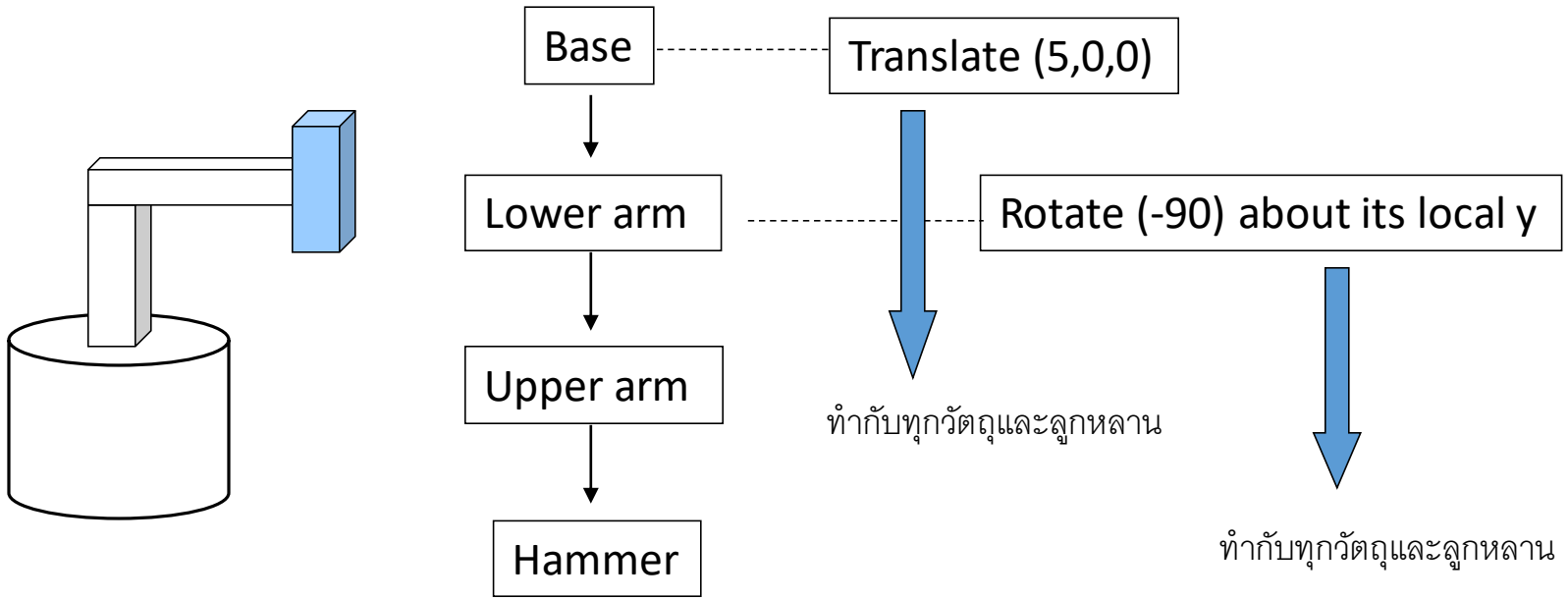
Relative Transformation (2)

Step 2: Rotate the lower arm and all its descendants relative to its local y axis by -90 degree



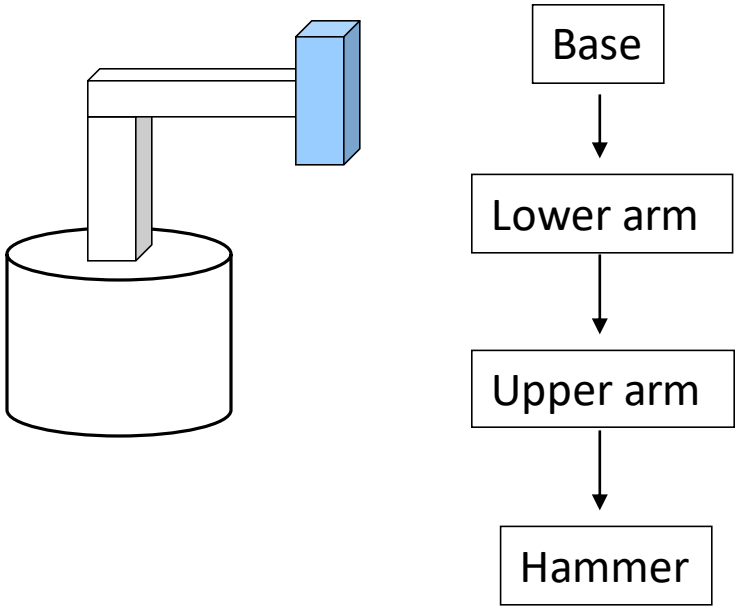
Relative Transformation (3)

- แทนการแปลงแบบสัมพัทธ์ด้วย **scene graph**



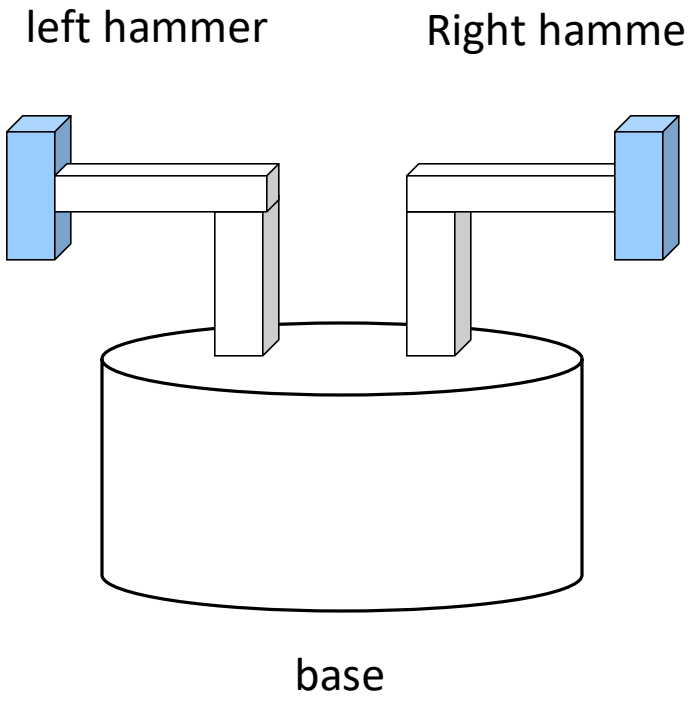
เขียนใน OpenGL

- Translate base and all its descendants by (5,0,0)
- Rotate the lower arm and its descendants by -90 degree about the local y

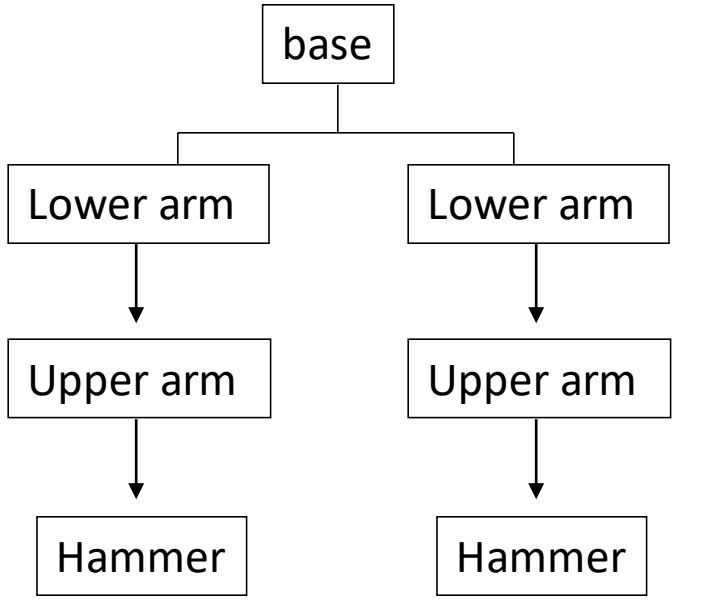



```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

… // setup your camera

glTranslatef(5,0,0);

Draw_base();

glRotatef(-90, 0, 1, 0);

Draw_lower _arm();
Draw_upper_arm();
Draw_hammer();
```

วัตถุซับซ้อนขึ้น



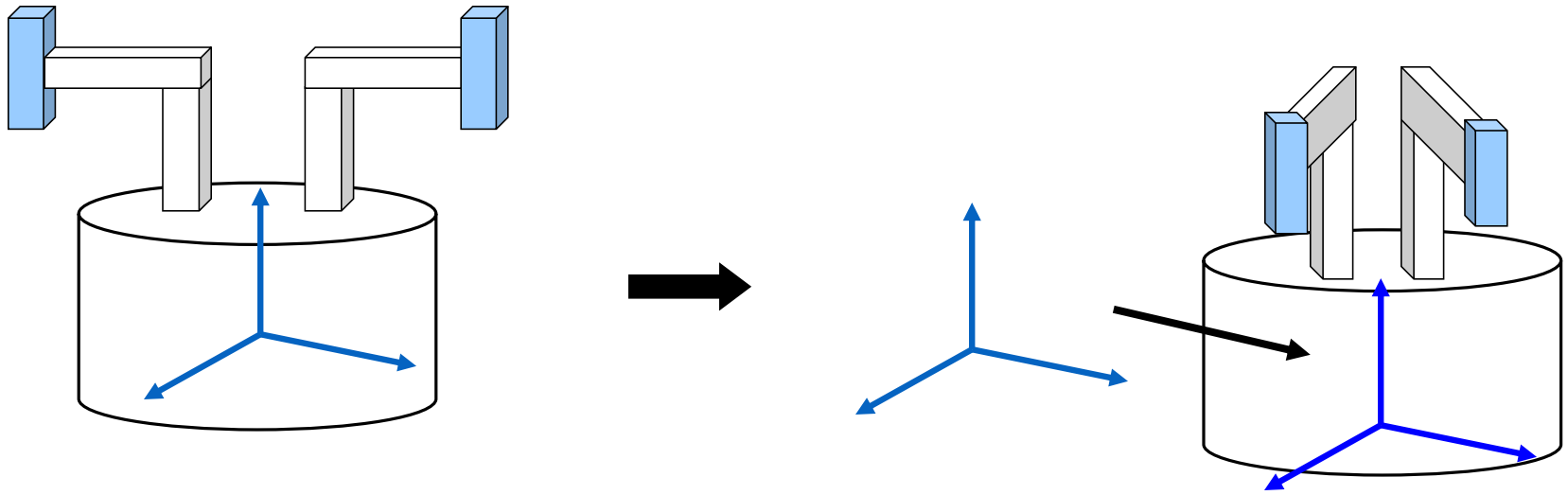
Scene Graph?



(left hammer) (right hammer)

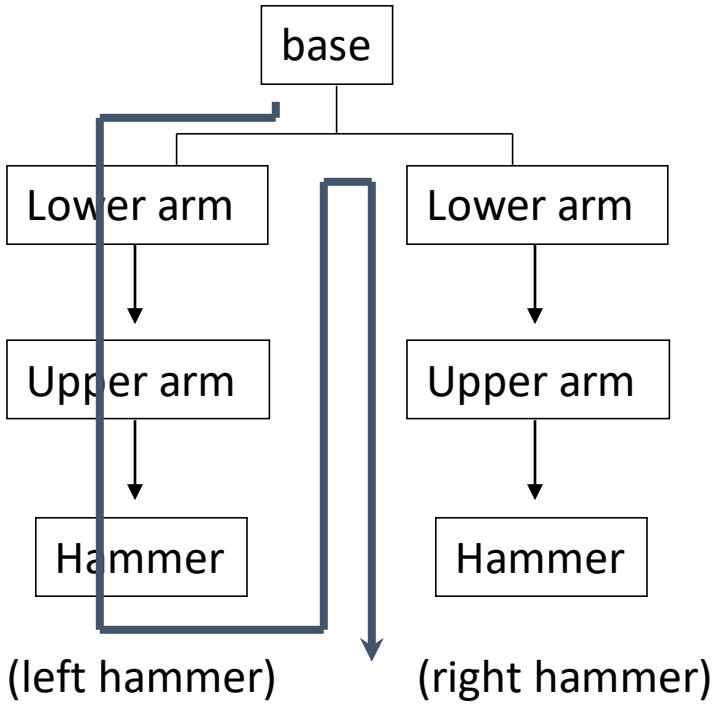
ตัวอย่าง

- Base and everything – translate (5,0,0)
- Left hammer – rotate 75 degree about the local y
- Right hammer – rotate -75 degree about the local y

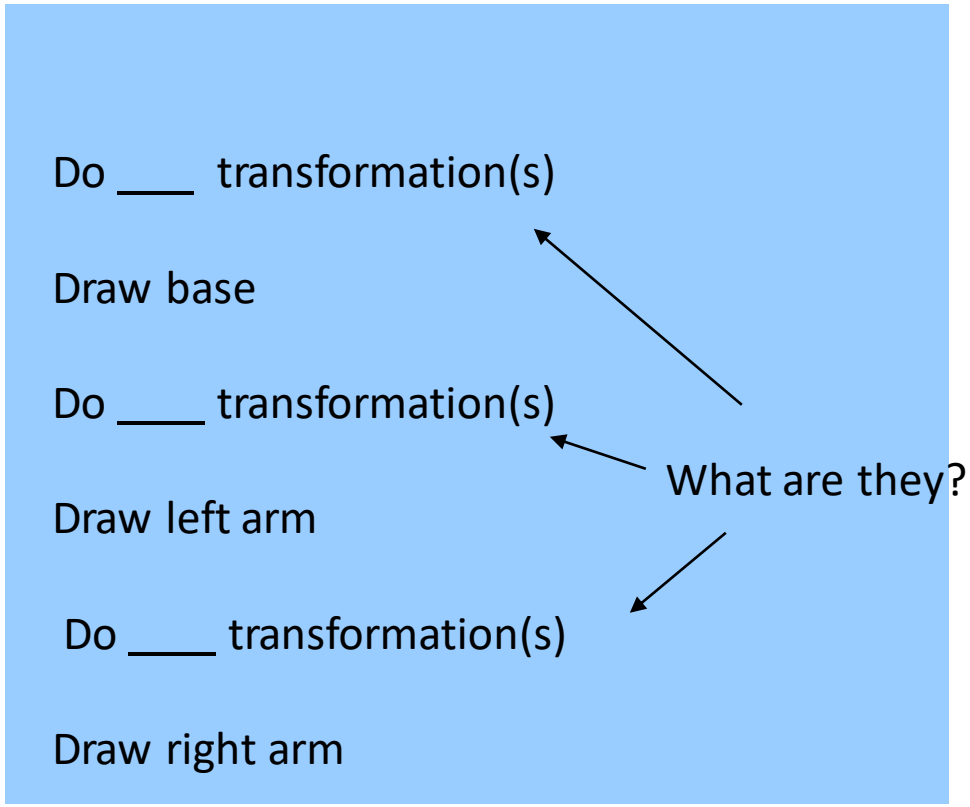


เดินทางเชิงลึกก่อน (depth first traversal)

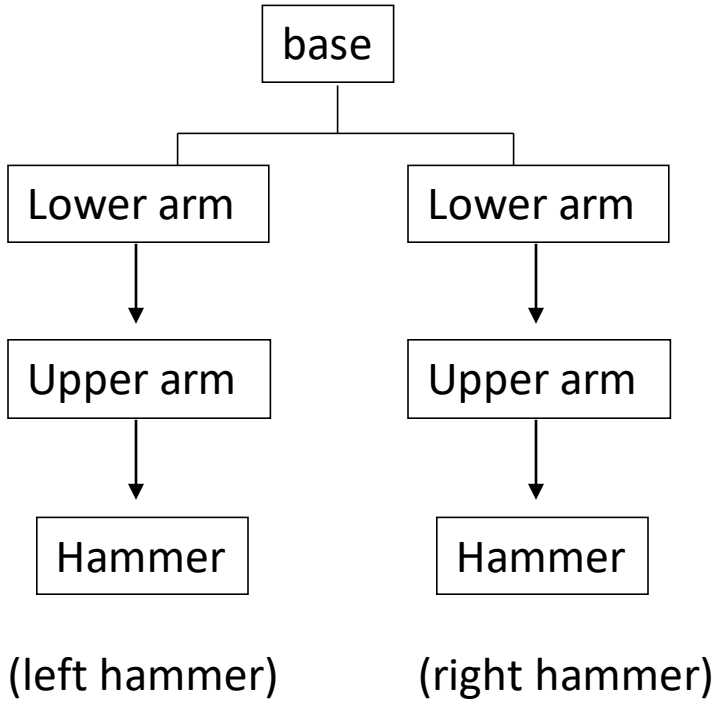
- เวลาเขียนโปรแกรมให้เขียนแบบเดินทางเชิงลึกก่อน



Depth First Traversal



แล้วแบบนี้ล่ะ?



Translate(5,0,0)

Draw base

Rotate(75, 0, 1, 0)

Draw left hammer

Rotate(-75, 0, 1, 0)

Draw right hammer

What's wrong?!

มีบางอย่างไม่ชอบมาพากล ...

- เราต้องการหมุน right hammer อ้างอิงกับ base ไม่ใช่อิงกับ left hammer

How about this?

Do **Translate(5,0,0)**

Draw base

Do **Rotate(75, 0, 1, 0)**

Draw left hammer

What's wrong?!

Do **Rotate(-75, 0, 1, 0)**

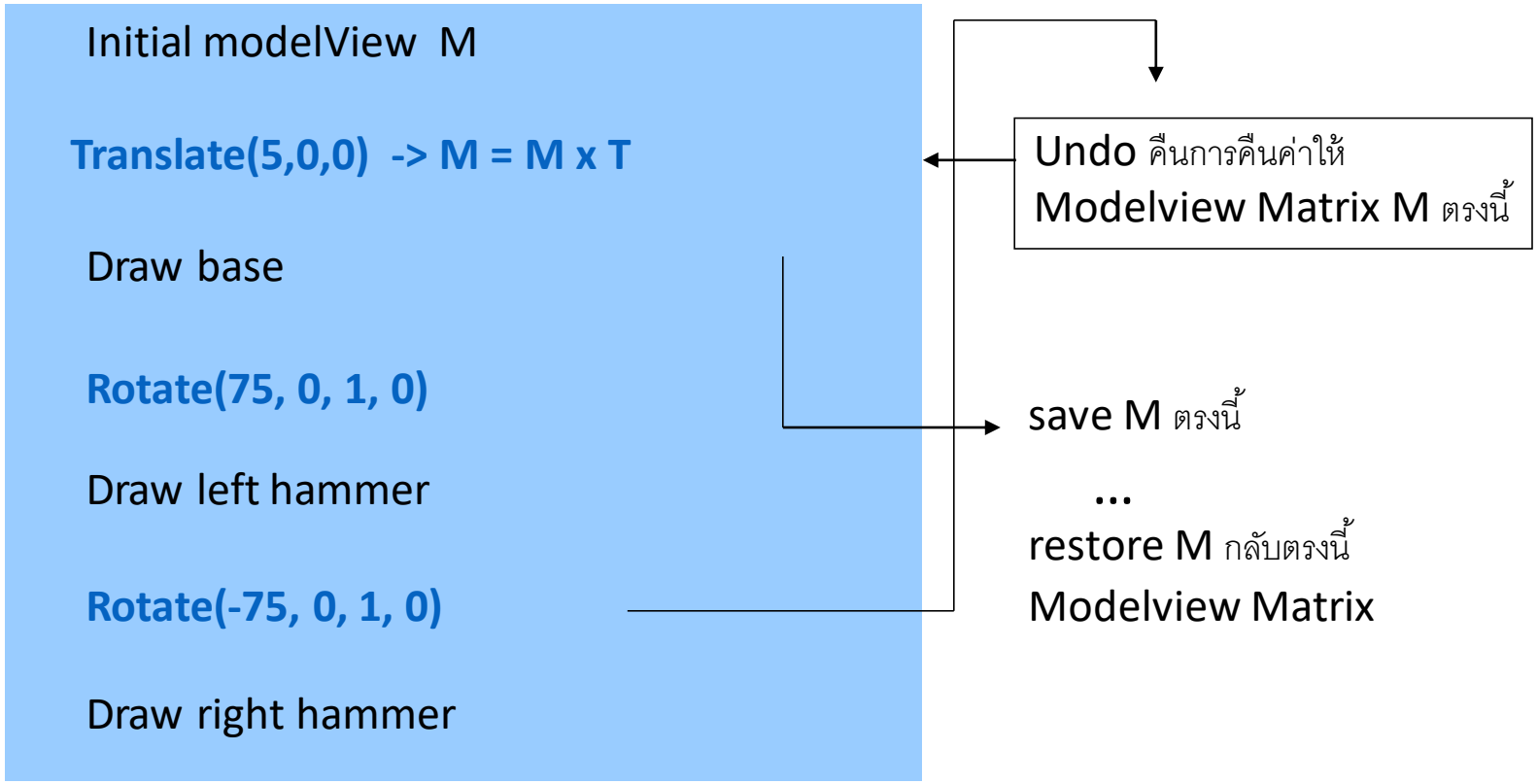
Draw right hammer

ต้อง **undo** การแปลงที่ **left hammer** ก่อนที่จะแปลง right hammer

undo ก่อน

การ Undo การแปลง

- ต้องเก็บ modelview matrix ทันทีหลัง draw base



OpenGL Matrix Stack

- สามารถใช้ OpenGL Matrix Stack เพื่อ save และ restore

modelView M เริ่มต้น

Do **Translate(5,0,0) -> M = M x T**

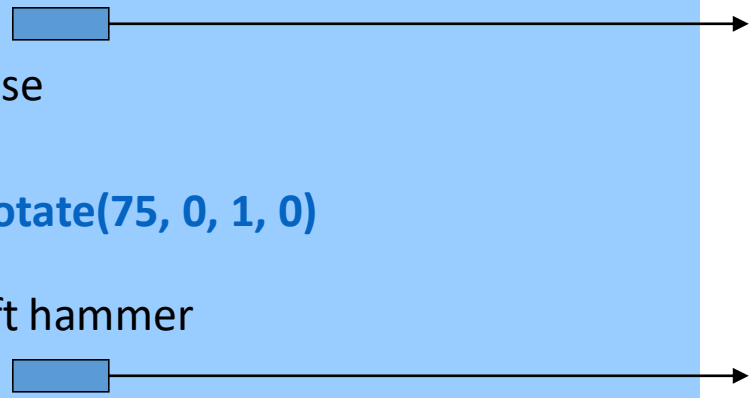
Draw base

Do **Rotate(75, 0, 1, 0)**

Draw left hammer

Do **Rotate(-75, 0, 1, 0)**

Draw right hammer



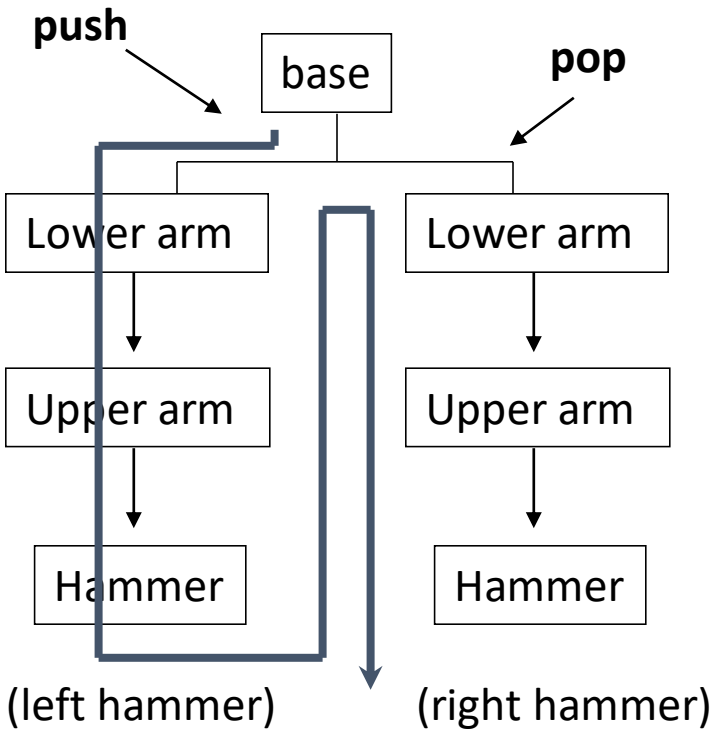
* Store modelview matrix ปัจจุบัน
- สร้าง copy ของ matrix ปัจจุบัน
แล้ว **push** เข้าใน OpenGL Matrix Stack:
ด้วย glPushMatrix()

- ทำต่อกับ matrix ปัจจุบัน

* Restore modelview matrix
- Pop Matrix ออกจาก OpenGL Matrix Stack แล้ว
copy กลับเข้า modelview Matrix:
ด้วย glPopMatrix()

Push และ Pop Matrix Stack

- routine ทั่วไปของ OpenGL :



Depth First Traversal

```
glTranslate(5,0,0)
Draw_base();
glPushMatrix();

glRotate(75, 0,1,0);
Draw_left_hammer();

glPopMatrix();
glRotate(-75, 0,1,0);
Draw_right_hammer();
```

Push and Pop Matrix Stack

- Nested push และ pop