Lab #08 – CSC/CPE 474 – Forward Kinematics / Hierarchical Modeling

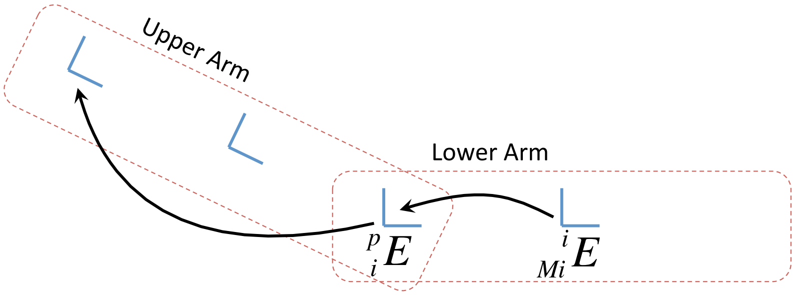
Today we're going to create a simple hierarchical robot.

Please download the source code from PolyLearn. When you run the code, you should see a textured object and a background grid. You'll be using this object as the mesh for the robot links. When you drag the mouse with the ALT key pressed, you should see some console output. You'll be using these values to control the joint angles of the robot.

Create a class called `Link`. It should have the following member variables:

- A pointer to the parent link
- An array (or a std::vector) of child links
- A 4x4 matrix, ${}^{p}_{i}E$, for the transform of this link with respect to the parent link
- A 4x4 matrix, ${}^{i}_{Mi}E$, for the transform of the mesh with respect to this link
- A float for the current angle.

Since you're going to create a 2D planar mechanism, you only need a single float for the rotation. For convenience, the origin of each link is going to be where the joint is. For example, for the "upper arm" link, the origin will be at the shoulder, and for the "lower arm" link the origin is going to be at the elbow.

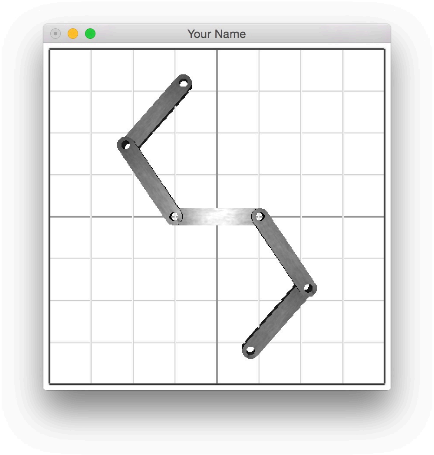


The arrows indicate what the two transformation matrices represent. ${}^{p}_{i}E$, describes where the current link is with respect to the parent link. In the figure above, it represents where the elbow joint is with respect to the shoulder joint. ${}^{i}_{Mi}E$, describes where the mesh's origin is with respect to the current link.

The first four of these member variables should be set in the scene loading function. The last variable, angle, should be set using the mouse in the mouseMotionGL() function.

The pseudocode for the recursive drawing function is

```
void Link::draw(MatrixStack &M)
{
        M.push();
        M.mult(${}^{p}_{i}E$);
        M.rotate(angle);
        M.push();
        M.mult(${}^{i}_{Mi}E$);
        Send M's top matrix to the GPU
        drawMesh();
        M.pop();
        for all child in children
                child.draw(M);
        M.pop();
}
```



Start with a simple robot first. Your final task is to build a two-arm robot composed of 5 links. The root should be draw at the world origin and should be fixed. Moving the mouse in one direction (x or y) should change the shoulder angles, and moving the mouse in the other direction should change the elbow angles.