

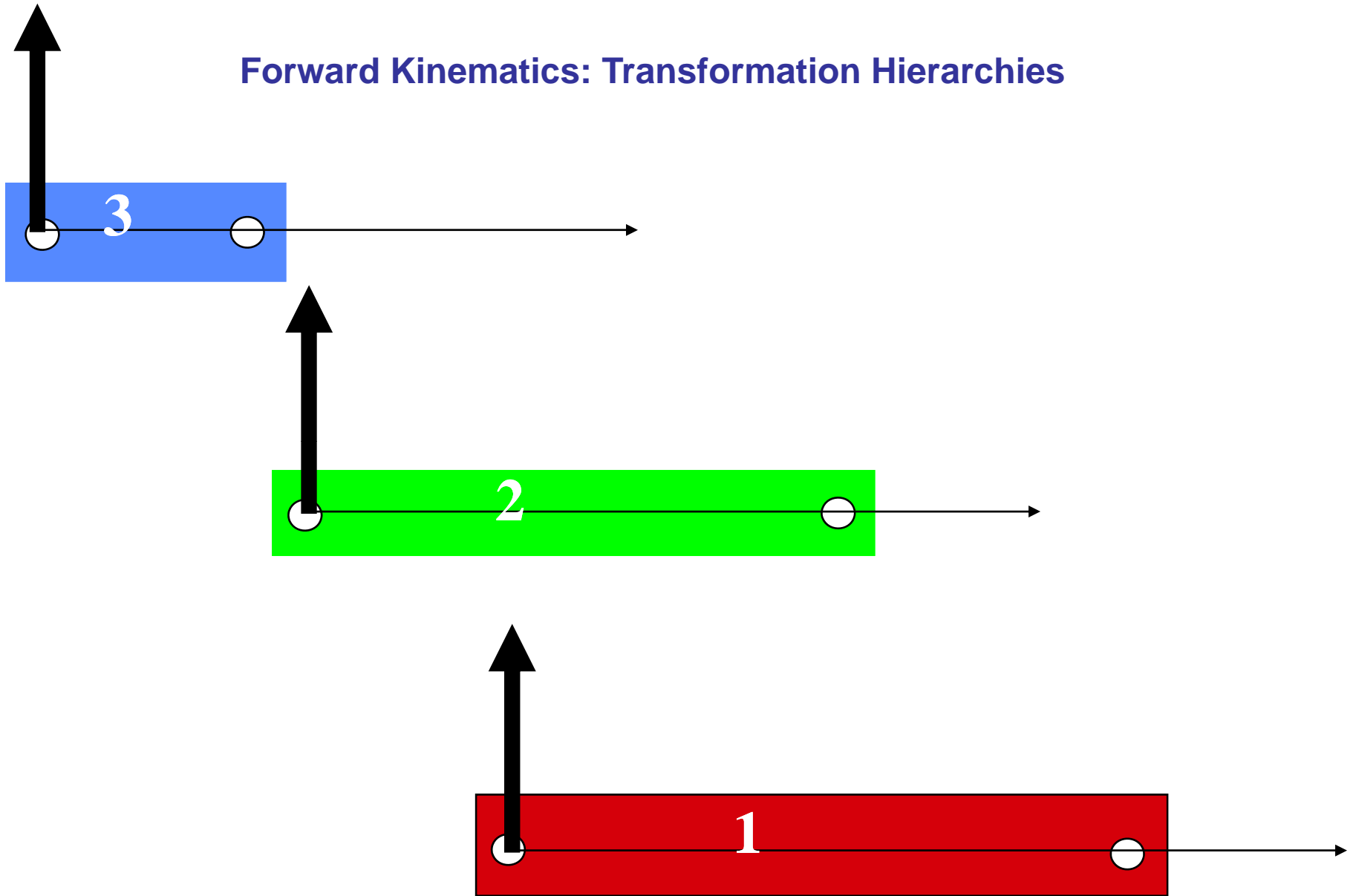
Forward Kinematics

Mike Bailey

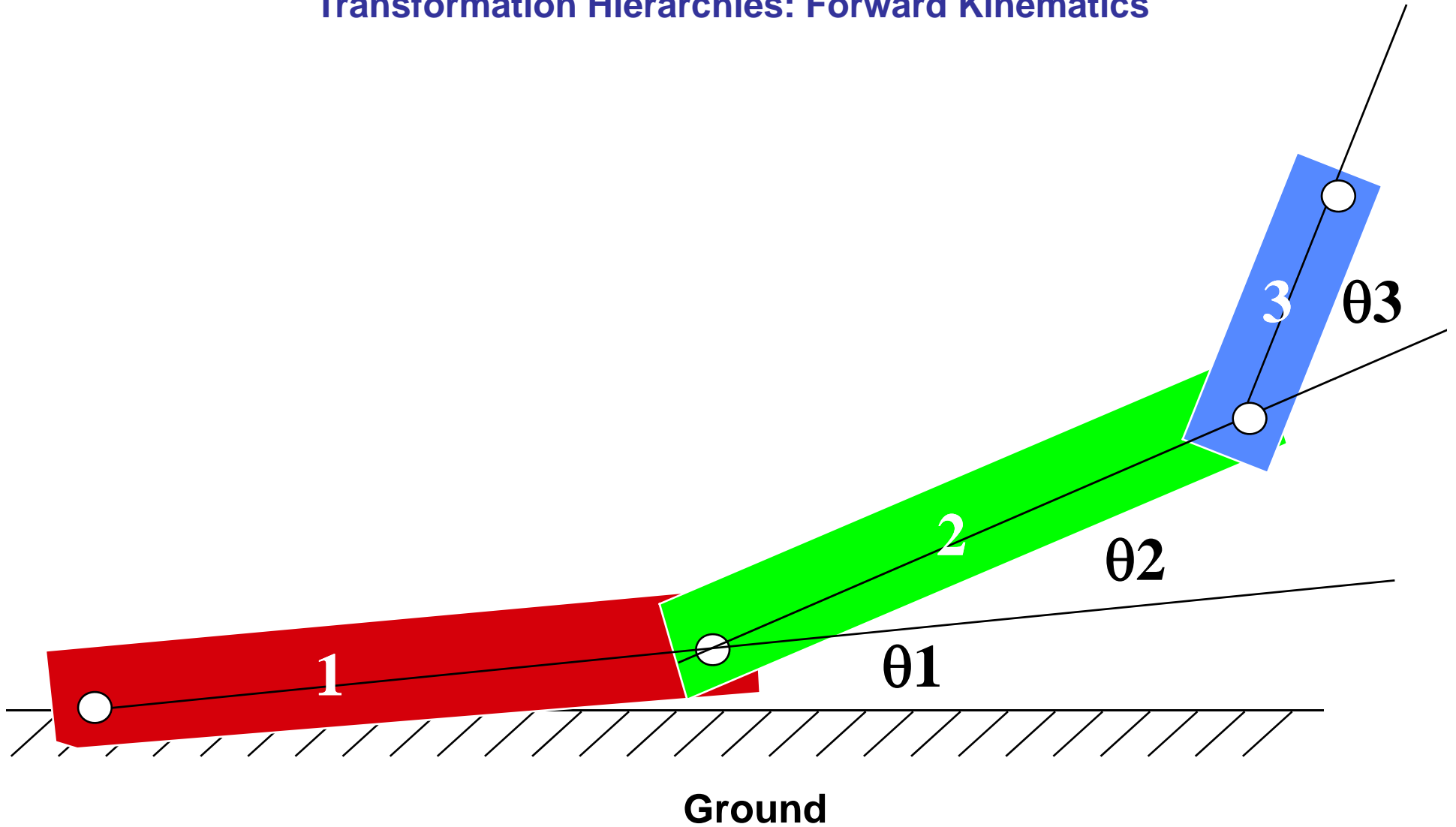
Oregon State University



Forward Kinematics: Transformation Hierarchies



Transformation Hierarchies: Forward Kinematics



Positioning Part #1 With Respect to Ground

1. Rotate by θ_1
2. Translate by $\Delta_{1/G}$

Write it

$$[\mathbf{M}_{1/G}] = [\mathbf{T}_{1/G}] * [\mathbf{R}_{\theta_1}]$$

Say it

Positioning Part #2 With Respect to Ground

1. Rotate by θ_2
2. Translate the length of part 1
3. Rotate by θ_1
4. Translate by $\Delta_{1/G}$

Write it

$$\begin{aligned} [M_{2/G}] &= [T_{1/G}] * [R_{\theta_1}] * [T_{2/1}] * [R_{\theta_2}] \\ &= [M_{1/G}] * [M_{2/1}] \end{aligned}$$

Say it

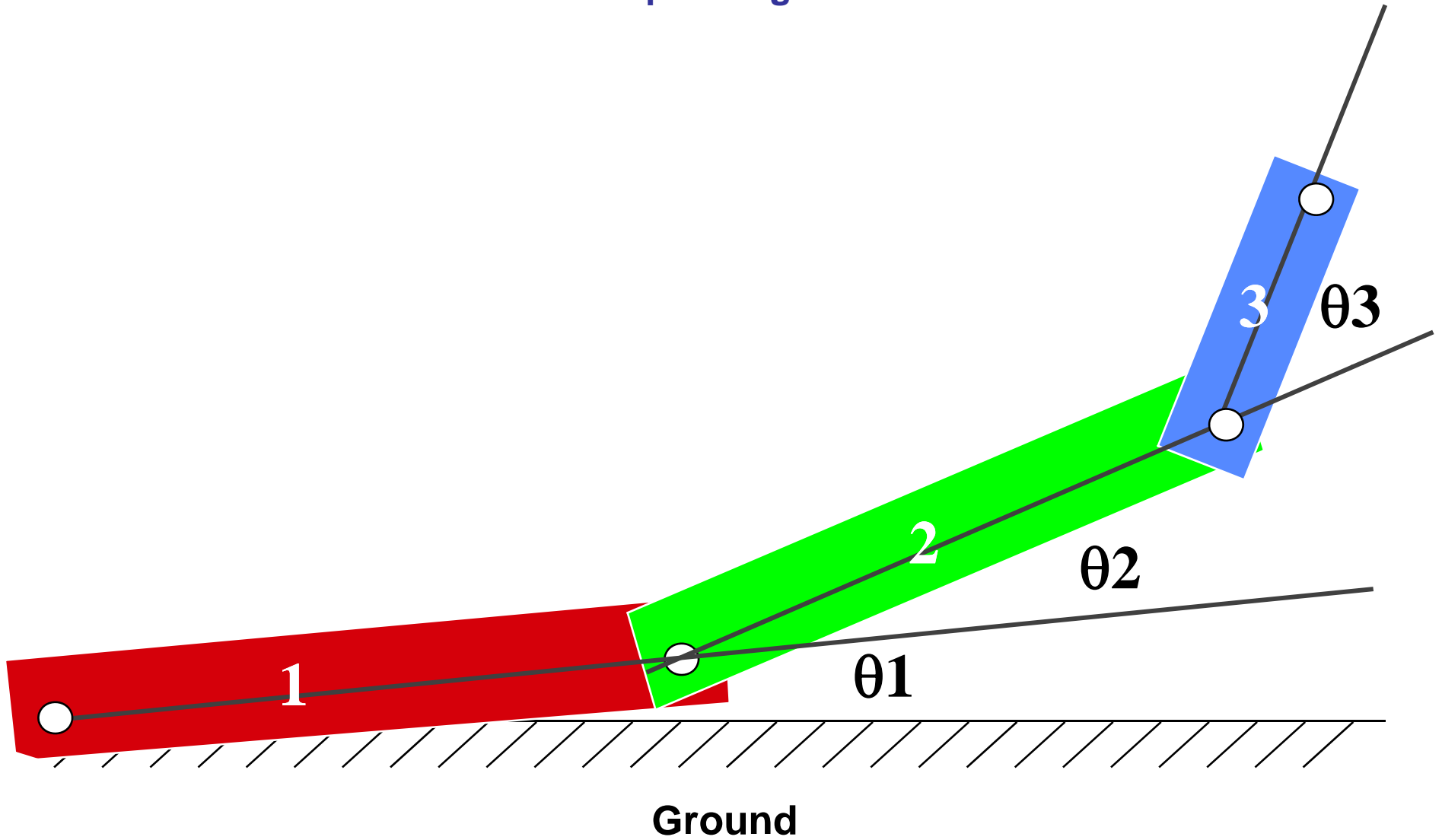
Positioning Part #3 With Respect to Ground

1. Rotate by θ_3
2. Translate the length of part 2
3. Rotate by θ_2
4. Translate the length of part 1
5. Rotate by θ_1
6. Translate by $\Delta_{1/G}$

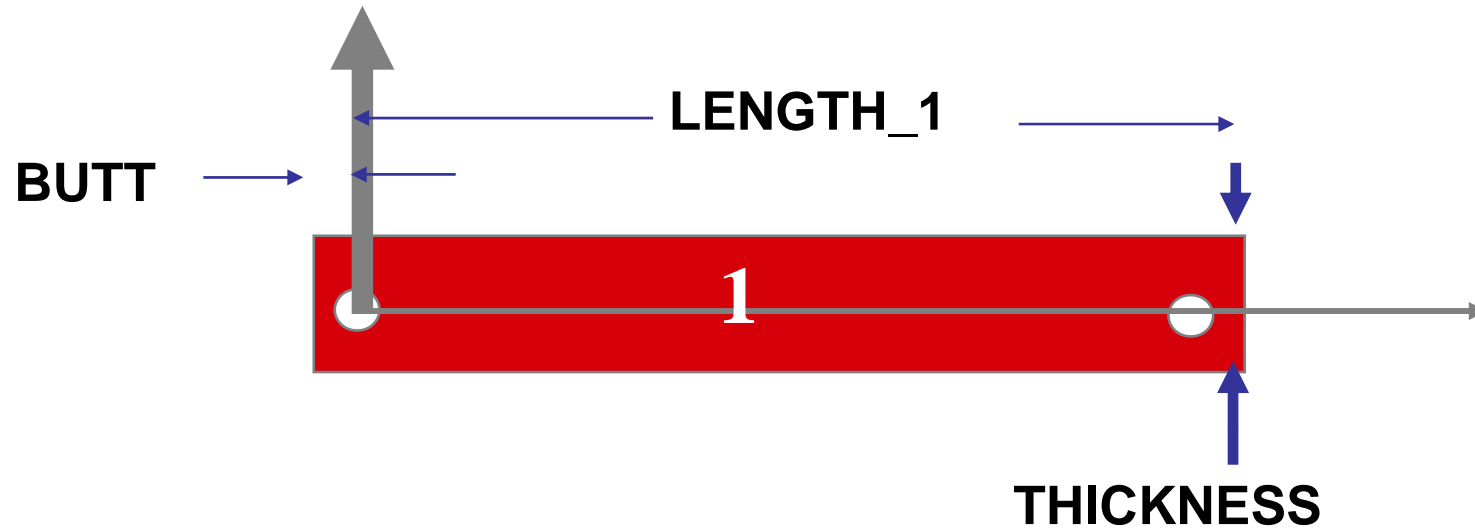
Write it

$$\begin{aligned} [M_{3/G}] &= [T_{1/G}] * [R_{\theta_1}] * [T_{2/1}] * [R_{\theta_2}] * [T_{3/2}] * [R_{\theta_3}] \\ &= [M_{1/G}] * [M_{2/1}] * [M_{3/2}] \end{aligned}$$

Sample Program



Sample Program



```
DrawLinkOne( )
{
    glBegin( GL_QUADS );
    glVertex2f( -BUTT, -THICKNESS/2 );
    glVertex2f( LENGTH_1, -THICKNESS/2 );
    glVertex2f( LENGTH_1, THICKNESS/2 );
    glVertex2f( -BUTT, THICKNESS/2 );
    glEnd( );
}
```

Sample Program

```
DrawMechanism( float  $\theta_1$ , float  $\theta_2$ , float  $\theta_3$  )
{
    glPushMatrix( );
        glRotatef(  $\theta_1$ , 0., 0., 1. );
        glColor3f( 1., 0., 0. );
        DrawLinkOne( );

        glTranslatef( LENGTH_1, 0., 0. );
        glRotatef(  $\theta_2$ , 0., 0., 1. );
        glColor3f( 0., 1., 0. );
        DrawLinkTwo( );

        glTranslatef( LENGTH_2, 0., 0. );
        glRotatef(  $\theta_3$ , 0., 0., 1. );
        glColor3f( 0., 0., 1. );
        DrawLinkThree( );
    glPopMatrix( );
}
```

Sample Program

Where in the window to display (pixels)

Viewing Info: angle, x:y aspect ratio, near, far

Whatever interaction is being used to get the eye position

Set the eye position

```
glViewport( 100, 100, 500, 500 );

glMatrixMode( GL_PROJECTION );
glLoadIdentity( );
gluPerspective( 90., 1.0, 1., 10. );

glMatrixMode( GL_MODELVIEW );
glLoadIdentity( );

done = FALSE;
while( ! done )
{
    << Determine  $\theta_1, \theta_2, \theta_3$  >>
    glPushMatrix();
    gluLookAt( eyex, eyey, eyez,
              centerx, centery, centerz,
              upx, upy, upz );
    DrawMechanism(  $\theta_1, \theta_2, \theta_3$  );
    glPopMatrix();
}
```