

컴퓨터 애니메이션

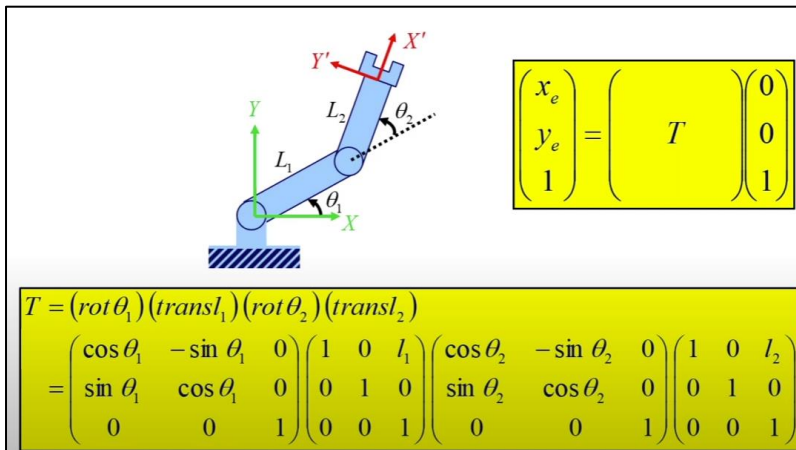
실습 보고서



Self-Scoring Table

	P1	P2	E1
Score	1	1	1

P1 - Draw a simple kinematic model using OpenGL transformation



Local coordinate 관점에서 보았을 때, joint 회전을 먼저 한 후, Link transform을 해야한다.

```

328 // Joints and links
329 for (int i = 0; i < nLinks; i++)
330 {
331     // Joint transform
332     // z축을 중심으로 회전시키구
333     glRotatef(radian2degree(jointAngle[i]), 0, 0, 1);
334
335     // 그려라
336     // Geometry of the joint and link
337     drawJoint(radiusJoint, Vector3f(0.95f, 0, 0)); // Draw a joint
338     drawLink(radiusLink, linkLength[i], Vector3f(0, 0.95f, 0.95f)); // Draw a link

```

line 333 : joint 회전을 한다

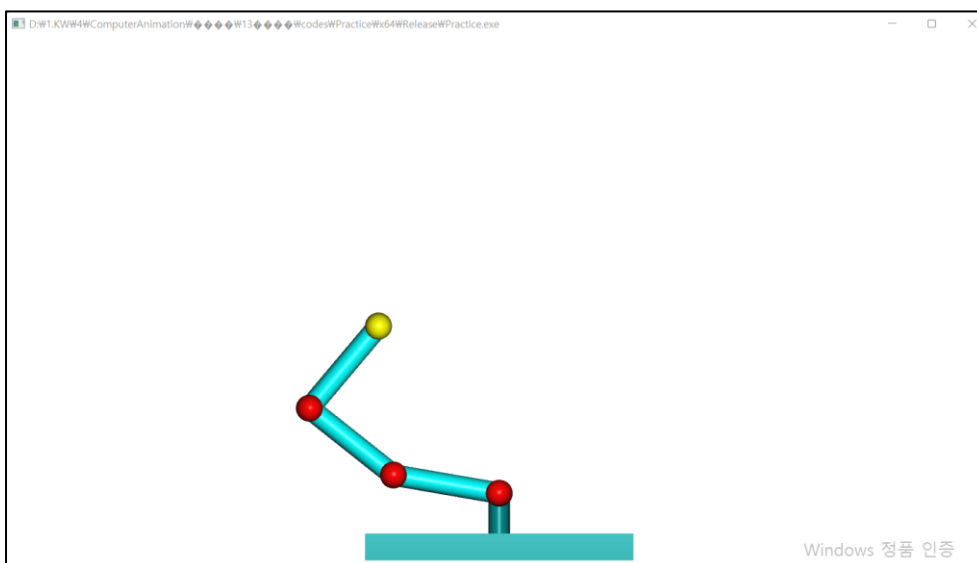
line 337~338 : joint와 link를 그린다.

```

341
342 // Link transform
343 // y축으로 이동
344 glTranslatef(0, linkLength[i], 0);

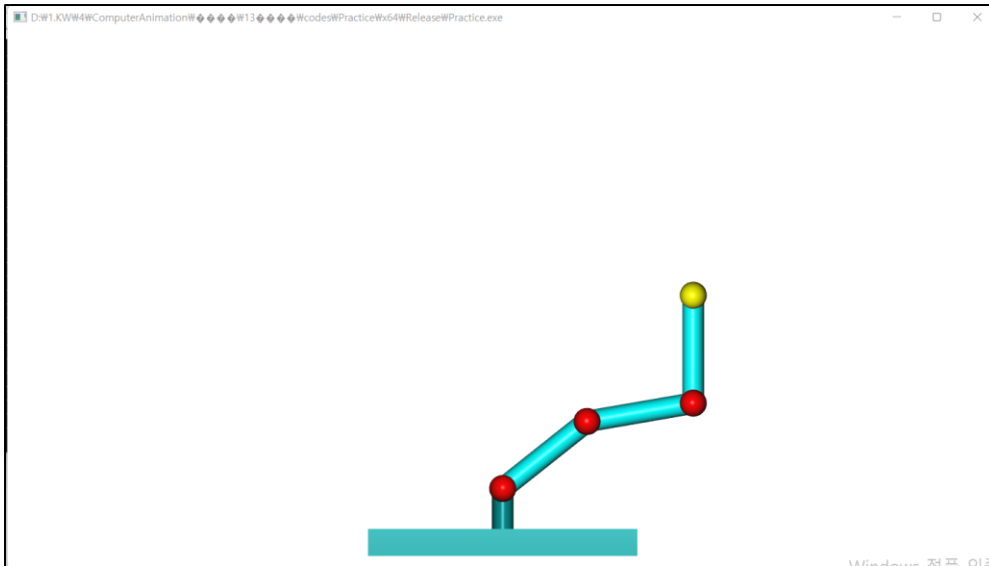
```

line 344: 이동한다.



P2 - Draw a simple kinematic model using Eigen transformation

Eigen을 통해 그릴 때에도 P1과 마찬가지로 적용된다.



E1 - Draw body attached coordinate systems

[구현 코드]

```

164 void drawAxes(float l, float w, int z)
165 {
166     glDisable(GL_LIGHTING);
167     glDisable(GL_LIGHT0);
168     glLineWidth(w * dpiScaling);
169
170     glBegin(GL_LINES);
171     glColor3f(1, 0, 0); glVertex3f(0, 0, z); glVertex3f(l, 0, z); //x-axis
172     glColor3f(0, 1, 0); glVertex3f(0, 0, z); glVertex3f(0, l, z); //y-axis
173     glColor3f(1, 0, 1); glVertex3f(0, 0, z); glVertex3f(0, 0, l+z); //z-axis
174     glEnd();
175     glEnable(GL_LIGHTING);
176     glEnable(GL_LIGHT0);
177 }

```

glSetup.cpp에 있는 drawAxes 함수를 수정하였다. 매개변수로 추가된 z값은 축이 앞에 그려지기 위해 넣었다. z=1인 평면에 축을 그리도록 하여 카메라로 볼 때, 축이 joint나 link에 가려지지 않는다.

[OpenGL]

```

328 // Joints and links
329 for (int i = 0; i < nLinks; i++)
330 {
331     // Joint transform
332     // z축을 중심으로 회전시키구
333     glRotatef(radian2degree(jointAngle[i]), 0, 0, 1);
334
335     // 그려라
336     // Geometry of the joint and link
337     drawJoint(radiusJoint, Vector3f(0.95f, 0, 0)); // Draw a joint
338     drawLink(radiusLink, linkLength[i], Vector3f(0, 0.95f, 0.95f)); // Draw a link
339
340     drawAxes(0.2, 2, 1);
341
342     // Link transform
343     // y축으로 이동
344     glTranslatef(0, linkLength[i], 0);
345 }
346

```

joint를 회전시키고, joint와 link를 그린다. 그 다음 link transform을 하기 전, joint 자리에 축을 그려준다.

[Eigen]

```
366     for (int i = 0; i < nLinks; i++)
367     {
368         // Joint transform using an Eigen 4x4 matrix. 3x3 rotation matrix
369         T_joint.setIdentity();
370         // 3,3 Mat, 시작은 0,0부터
371         T_joint.block<3, 3>(0, 0) = Matrix3f(AngleAxisf(jointAngle[i], Vector3f::UnitZ()));
372         T = T * T_joint;
373
374         // Geometry of the joint and link
375         glLoadMatrixf(T.data());
376         drawJoint(radiusJoint, Vector3f(0.95f, 0, 0)); //Draw a joint
377         drawLink(radiusLink, linkLength[i], Vector3f(0, 0.95f, 0.95f)); // Draw a link
378
379         drawAxes(0.2, 2, 1);
380
381         // Link transform using an Eigen 4x4 matrix : 3x1 translation vector
382         T_link.setIdentity();
383         T_link.block<3, 1>(0, 3) = Vector3f(0.0, linkLength[i], 0.0);
384         T = T * T_link;
385     }
```

Eigen을 사용하여 그릴 때에도 마찬가지이다.