

Swimming Man

ID/ Name : 20174089/ 윤석원

1. Body

<pre>157 // body 158 modelMat = glm::scale(mat, glm::vec3(0.85, 0.5, 0.4)); 159 pvmMat = projectMat * viewMat * modelMat; 160 glUniformMatrix4fv(pvmMatrixID, 1, GL_FALSE, &pvmMat[0][0]); 161 glDrawArrays(GL_TRIANGLES, 0, NumVertices); 162</pre>	P*V*S*v(cube)
- body는 scale만 적용하고 바로 projectMat * viewMat을 곱해준다.	

2. Head

<pre>163 // head 164 modelMat = glm::translate(mat, glm::vec3(0.75, 0, 0)); 165 modelMat = glm::scale(modelMat, glm::vec3(0.3, 0.3, 0.3)); 166 pvmMat = projectMat * viewMat * modelMat; 167 glUniformMatrix4fv(pvmMatrixID, 1, GL_FALSE, &pvmMat[0][0]); 168 glDrawArrays(GL_TRIANGLES, 0, NumVertices);</pre>	P*V*S*T*v(cube)
- head는 먼저 머리 위치로 translate 하고 scale을 적용하여 projectMat * viewMat을 곱해준다.	

3. Arms

<pre>140 glm::vec3 arm[2]; 141 arm[0] = glm::vec3(0.2, 0.35, 0); 142 arm[1] = glm::vec3(-0.2, -0.35, 0);</pre>	왼쪽, 오른쪽 arm의 위치 정의
<pre>170 // arms 171 for (int i = 0; i < 2; i++) 172 { 173 modelMat = glm::translate(mat, glm::vec3(0.2, 0.0, 0.0)); 174 modelMat = glm::rotate(modelMat, rotAngle * 2.0f, glm::vec3(0, 1, 0)); 175 modelMat = glm::translate(modelMat, arm[i]); 176 modelMat = glm::scale(modelMat, glm::vec3(0.4, 0.1, 0.15)); 177 178 pvmMat = projectMat * viewMat * modelMat; 179 glUniformMatrix4fv(pvmMatrixID, 1, GL_FALSE, &pvmMat[0][0]); 180 glDrawArrays(GL_TRIANGLES, 0, NumVertices); 181 }</pre>	P*V*S*T*R*T*v
- arm을 먼저 회전축이 되는 어깨 관절 위치로 translate 시켜준다. - 이어서 회전축을 중심으로 rotate를 시켜준다. - 그 다음 원래 arm의 위치로 translate한 후에 scale로 크기를 설정한다	

4. Forearms

144 145 146	<pre> glm::vec3 forearm[2]; forearm[0] = glm::vec3(0.4, 0.35, 0); forearm[1] = glm::vec3(-0.4, -0.35, 0); //right </pre>	왼쪽, 오른쪽 forearm 위치 정의
183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205	<pre> // forearms for (int i = 0; i < 2; i++) { modelMat = glm::translate(mat, glm::vec3(0.2, 0.0, 0.0)); modelMat = glm::rotate(modelMat, rotAngle * 2.0f, glm::vec3(0, 1, 0)); if (i == 1) { modelMat = glm::translate(modelMat, forearm[i]); modelMat = glm::rotate(modelMat, rotAngleForearm * 4.0f, glm::vec3(0, 1, 0)); } else { modelMat = glm::translate(modelMat, forearm[i]); modelMat = glm::rotate(modelMat, rotAngleForearm * 4.0f, glm::vec3(0, 1, 0)); modelMat = glm::translate(modelMat, glm::vec3(0.4, 0.0, 0.0)); } modelMat = glm::translate(modelMat, glm::vec3(-0.2, 0.0, 0.0)); modelMat = glm::scale(modelMat, glm::vec3(0.4, 0.1, 0.15)); pvmMat = projectMat * viewMat * modelMat; glUniformMatrix4fv(pvmMatrixID, 1, GL_FALSE, &pvmMat[0][0]); glDrawArrays(GL_TRIANGLES, 0, NumVertices); } </pre>	<p>forearm[1] $P * V * S * T * R * T * R * T * v$</p> <p>forearm[0] $P * V * S * T * T * R * T * R * T * v$</p>

- 오른쪽과 왼쪽의 팔이 회전하기 시작하는 위치가 다르기 때문에 그에 맞춰 forearm도 왼쪽 오른쪽의 변형이 다르게 적용된다.
- 가장먼저 팔과 동일한 축으로 회전을 적용해야 하기 때문에 어깨관절 위치로 transpose한 후에 rotate을 적용한다.
- forearm[1]의 경우 팔 위치로 transpose한 후 rotate을 수행한다.
- forearm[0]의 경우 팔 위치로 transpose한 후 rotate을 수행하고 나서 회전축이 되는 팔꿈치 관절 위치로 transpose 해준다.

5. Upper Leg

149 150 151	<pre> glm::vec3 uleg[2]; uleg[0] = glm::vec3(-0.8, -0.13, 0); uleg[1] = glm::vec3(-0.8, 0.13, 0); </pre>	왼쪽, 오른쪽 upper Leg 위치 정의
207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223	<pre> // upper legs for (int i = 0; i < 2; i++) { modelMat = mat; if (i == 1) { modelMat = glm::rotate(modelMat, rotAngleLeg * -0.5f, glm::vec3(0, 1, 0)); } else { modelMat = glm::rotate(modelMat, rotAngleLeg * 0.5f, glm::vec3(0, 1, 0)); } modelMat = glm::translate(modelMat, uleg[i]); modelMat = glm::scale(modelMat, glm::vec3(0.6, 0.15, 0.2)); pvmMat = projectMat * viewMat * modelMat; glUniformMatrix4fv(pvmMatrixID, 1, GL_FALSE, &pvmMat[0][0]); glDrawArrays(GL_TRIANGLES, 0, NumVertices); } </pre>	<p>uleg[1] $P * V * S * T * R * v$</p> <p>uleg[0] $P * V * S * T * R * v$</p>

- rotate을 적용하고 uleg(허벅지)위치로 translate하여 scale로 크기를 설정한다.
- 오른쪽, 왼쪽 uleg(허벅지)가 서로 반대방향으로 이동해야 하기 때문에 분기하여 처리한다.

6. Lower Leg

153	glm::vec3 lleg[2];	왼쪽, 오른쪽 lower Leg 위치 정의
154	lleg[0] = glm::vec3(-1.0, -0.13, 0);	
155	lleg[1] = glm::vec3(-1.0, 0.13, 0);	
225	<pre>// lower legs for (int i = 0; i < 2; i++) { modelMat = mat; // left or right if (i == 1) { modelMat = glm::rotate(modelMat, rotAngleLeg * -0.5f, glm::vec3(0, 1, 0)); modelMat = glm::translate(modelMat, lleg[i]); modelMat = glm::rotate(modelMat, rotAngleLowerLeg * 0.5f, glm::vec3(0, 1, 0)); } else { modelMat = glm::rotate(modelMat, rotAngleLeg * 0.5f, glm::vec3(0, 1, 0)); modelMat = glm::translate(modelMat, lleg[i]); modelMat = glm::rotate(modelMat, rotAngleLowerLeg * 0.5f, glm::vec3(0, 1, 0)); } modelMat = glm::translate(modelMat, glm::vec3(-0.5, 0, 0)); modelMat = glm::scale(modelMat, glm::vec3(0.6, 0.15, 0.2)); pvmMat = projectMat * viewMat * modelMat; glUniformMatrix4fv(pvmMatrixID, 1, GL_FALSE, &pvmMat[0][0]); glDrawArrays(GL_TRIANGLES, 0, NumVertices); }</pre>	
lleg[1] $P * V * S * T * R * T * R * v$ lleg[0] $P * V * S * T * R * T * R * v$		
<ul style="list-style-type: none">- 오른쪽, 왼쪽 uleg(허벅지)가 서로 반대로 회전하는 것처럼 lleg(종아리)도 반대로 움직여야 하기 때문에 분기하여 처리한다.- 골반축을 기준으로 회전하면서 무릎관절 축으로도 회전해야 하기 때문에 rotate이 두 번 적용된다.- 가장 먼저 골반축 회전을 적용해주고 무릎관절 위치로 transpose한 후에 회전을 또 적용한다.		

7. Rotation Angle

```

18 float rotAngle = 0.0f;
19 float rotAngleRForearm = 0.0f;
20 float rotAngleLForearm = 0.0f;
21 float rotAngleLeg = 0.0f;
22 float rotAngleLowerLeg = 0.0f;

```

18 : arm의 회전각
 19 : 오른쪽 forearm의 회전각
 20 : 왼쪽 forearm의 회전각
 21 : upper Leg의 회전각
 22 : lower Leg의 회전각

```

266 static int prevTime = glutGet(GLUT_ELAPSED_TIME);
267 int currTime = glutGet(GLUT_ELAPSED_TIME);
268
269 if (abs(currTime - prevTime) >= 20)
270 {
271     float t = abs(currTime - prevTime);
272
273     rotAngle += glm::radians(t*360.0f / 10000.0f);
274     if (rotAngle >= glm::radians(2* 900000.0f / 10000.0f)) {
275         rotAngle -= glm::radians(2* 900000.0f / 10000.0f);
276         //std::cout << "한바퀴" << std::endl;
277     }
278
279     if (rotAngle <= glm::radians(900000.0f / 10000.0f) ) { // half circle
280         rotAngleLForearm = 0.0f;
281         rotAngleRForearm -= glm::radians(t * 360.0f / 10000.0f);
282     }
283     else
284     {
285         rotAngleRForearm = 0.0f;
286         rotAngleLForearm -= glm::radians(t * 360.0f / 10000.0f);
287     }
288 }

```

rotAngle(arm)은 360도 회전한다.

arm이 물속으로 들어갔을 때, forearm은 rotAngle(arm)과 동일한 축으로 동일한 각도만큼 움직여야 한다. 따라서 rotAngle이 반바퀴(180도)를 도는 시점마다 rotAngleForearm을 0으로 바꿔준다.



arm이 물 밖으로 나올 때는 forearm이 arm과 함께 회전하는 동시에 팔꿈치 축을 기준으로 회전해야 하므로 rotAngle이 반바퀴(180도)를 도는 시점 마다에 rotAngleForearm을 glm::radians(t * 360.0f / 10000.0f)로 바꿔준다.



```

32 bool up = true; // flag for angle of leg movement
33 bool lowerLeg_up = true;

```

uleg와 lleg가 올라갈 때,
내려갈 때를 구분하기 위한
flag 선언

```

290 if (up) {
291     rotAngleLeg += glm::radians(t * 360.0f / 10000.0f);
292     if (rotAngleLeg >= glm::radians(8.0f)) {
293         up = false;
294     }
295 }
296 else
297 {
298     rotAngleLeg -= glm::radians(t * 360.0f / 10000.0f);
299     if (rotAngleLeg <= glm::radians(-8.0f)) {
300         up = true;
301     }
302 }

```

uleg(허벅지)가 올라갈 때, 내려갈 때를 나누어서 rotAngleLeg을 변화시킨다.
허벅지의 회전각의 범위는 -8도 ~ 8도이다.

```

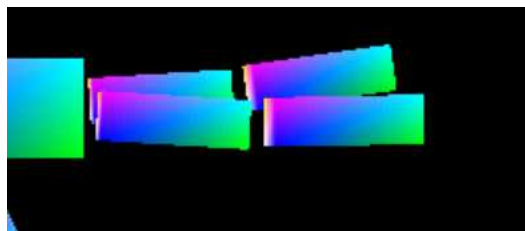
304 if (lowerLeg_up) {
305     rotAngleLowerLeg += glm::radians(t * 360.0f / 10000.0f * 1.2f);
306     if (rotAngleLowerLeg >= glm::radians(30.0f)) {
307         lowerLeg_up = false;
308     }
309 }
310 else
311 {
312     rotAngleLowerLeg -= glm::radians(t * 360.0f / 10000.0f * 1.2f);
313     if (rotAngleLowerLeg <= glm::radians(-3.0f)) {
314         lowerLeg_up = true;
315     }
316 }
317 prevTime = currTime;
318 glutPostRedisplay();
319 }
320 }

```

lleg는 골반 관절을 축으로 uleg(허벅지)와 함께 동일한 각도를 회전하면서도 무릎관절을 축으로도 회전해야한다.

lleg(종아리)도 마찬가지로 올라갈 때, 내려갈 때를 나누어 rotAngleLowerLeg를 변화시킨다.

lleg(종아리)의 회전각 범위는 -3도 ~ 30도이다. 종아리가 허벅지보다 앞으로 꺾어질 수 없기때문에 -3도부터 시작한다.



허벅지가 물 밖으로 나갈 때는 조금씩 종아리가 접히고.
허벅지가 물 속으로 들어갈 때는 조금씩 다리가 퍼진다.

8. View

```
324 void
325 keyboard(unsigned char key, int x, int y)
326 {
327     switch (key) {
328     case 'c': case 'C':
329         viewMat = glm::lookAt(glm::vec3(0, 0, 4), glm::vec3(0, 0, 0), glm::vec3(0, -1, 0));
330         break;
331     case 'v': case 'V':
332         viewMat = glm::lookAt(glm::vec3(0, 4, 0), glm::vec3(0, 0, 0), glm::vec3(0, 0, 1));
333         break;
334     case 033: // Escape key
335     case 'q': case 'Q':
336         exit(EXIT_SUCCESS);
337         break;
338     }
339 }
```

keyboard input으로 C가 들어오면 swimming man의 등을 볼 수 있고 V가 들어오면 측면을 관찰한다. 카메라 눈의 각도를 바꿔가며 올바르게 modeling이 되었는지 확인한다.

