<컴퓨터 그래픽스 Project 1, Cubeman Running 보고서>

소프트웨어학부 20214184 김재오

**0. variable**

**<Global variable>**

a) arm\_xlen, arm\_ylen, …, lowerleg\_zlen: arm, forearm, leg, lowerleg에 대한 길이 설정

b) left\_arm\_angle, …, rightlowerleg\_angle : 각 부위에 대한 각도 설정

**<DrawMan function variable>**

a) vec3 body\_locate : 모델의 body의 좌표에 대한 vector

b) mat4 pvmMat : glUniformMatrix4fu function의 parameter로 들어갈 최종 matrix

c) mat4 modelMat, leftarm, leftforearm, …, rightlowerleg : 각 부위에 대한 matrix

**1. 사람 모델의 각 부분 구현**

a) body

A close-up of a computer code

Description automatically generated

주어진 parameter manMat에 대하여 Scaling을 실시하고 translation을 적용시킨다. 여기서 translation은 뛰어다니며 몸이 위아래로 움직일 것이기 때문에 body\_locate를 이용하여 평행이동한다.

b) head

A drawing of a cube and a cube with a diagram

Description automatically generatedA computer code with text

Description automatically generated

Body와 마찬가지로 scaling 후 translation을 적용한다. 여기서 head는 body의 위치에 따라 정확하게 0.3만큼 위에 존재해야 하기 때문에 head의 길이를 고려하여 body\_locate + 0.4(y축 방향)으로 translation한다.

c) left arm

A diagram of a cube

Description automatically generatedA computer code with blue text

Description automatically generated

위쪽 팔은 육면체의 중심을 기준으로 회전하는 것이 아니라, 육면체의 윗면 중점을 기준으로 도형이 회전해야 팔이 회전하는 것처럼 보일 것이다.

즉 arm\_ylen / 2만큼 아래로 평행이동한 후, left\_arm\_angle만큼 x축 기준으로 회전한다.

A drawing of a cube

Description automatically generated그 후, arm또한 body에 의한 상대적 좌표이기 때문에 body\_locate에서 (0.25, 0.3, 0)만큼 더한 곳으로 평행이동 시킨다.

d) left fore arm

A computer code with blue and white text

Description automatically generated

Left fore arm은 arm의 끝부분에 붙어 있는데, 마찬가지로 cube의 중간지점이 아니라 끝부분에서 회전해야 한다. 그러므로 맨 처음 cube 끝면의 중심을 (0, 0, 0)으로 translate한 다음, 회전을 하고 팔의 끝부분으로 translation한다.

회전하는 상황까지는 위의 left arm과 같은 규칙이며(단, 달리는 모습에 근사하기 위하여 forearm은 x축, y축 두 축에 대해 회전한다.), translation에서 팔의 끝부분으로 이동하기 위해 위의 left arm에서의 끝부분 좌표를 구한다.



우선 leftarm mat4는 4번째 row에 좌표값이 저장되어 있기 때문에, (0, 0, 0, 1) vector와 연산을 하여 homogeneous coordinate를 가져온다

그 후에 left arm이 회전한 각도에 따른 좌표를 가져온다. 회전한 각도 left\_arm\_angle와 팔 길이 arm\_ylen에 대하여 cube의 중심으로부터

y값은 arm\_ylen / 2 \* cos(left\_arm\_angle),

z값은 arm\_ylen / 2 \* sin(left\_arm\_angle)

만큼 감소한다.

A drawing of a cube with lines and numbers

Description automatically generated

e) 남은 부분들

right\_arm, right\_fore\_arm을 한 쌍, left\_leg, left\_lower\_leg을 한 쌍, right\_leg, right\_lower\_leg를 한 쌍으로 앞의 c), d)과정을 논리적으로 반복한다.

단, lower\_leg는 fore\_arm과 다르게 일자로 회전하기 위하여 x축에 대해서만 회전하는 것으로 한다.

**3. idle (회전 각도 밑 body 위치 결정)**

초기 회전 각도와 이에 대한 그림은 다음과 같다.

A screenshot of a computer program

Description automatically generated A colorful gradient on a black background

Description automatically generated

초기에는 level 1로 시작하여, 특정한 자세에 레벨을 바꾸면서 body, arm, leg 등의 위치와 각도를 조절한다. Level은 1부터 6까지 존재하며, 특정 조건이 만족되면 level n에서 level (n+1)으로 변경한다. (단 level 6 -> level 1)

위의사진은 level 1의 초기 사진이며, 2, 3, 4, 5, 6에서의 초기 사진은 다음과 같다.

<level 2> <level 3> <level 4>

A colorful rectangles on a black background

Description automatically generated A number one with colorful squares

Description automatically generated A colorful rectangles on a black background

Description automatically generated

<level 5> <level 6>

A colorful rectangles on a black background

Description automatically generated A rainbow colored rectangles on a black background

Description automatically generated

자세를 살펴보았을 때, level 2의 시작점에서 사람이 발을 딛고, level 3의 시작점에서 땅을 밀며 다음 발을 내딛는다. 그래서 level 1, level 2에서는 body의 위치를 낮추고, level 3에서는 body의 위치를 높여준다. 마찬가지로 level 4, level 5에서는 위치를 낮추고 level 6에서는 높여준다.

Level 1, 2, 3, 4, 5, 6의 소요 시간을 측정했을 때 print되는 횟수가 6~7번으로 모두 동일하기 때문에 level 3, 6에서는 body의 위치를 2번 높여주어 균형을 맞춘다.

또한 극단적 사례에서는 한 level에서만 횟수가 7로 되는 상황이 충분히 발생할 수 있기 때문에 level 6에서 1로 바뀔 때에는 body\_locate를 (0, 0, 0)으로 초기화 한다.

A screen shot of a black background

Description automatically generated A black text on a white background

Description automatically generated

각도를 변환할 때에는 currTime와 prevTime을 이용하여 20ms마다 변형을 주도록 하고, d\_t라는 변수를 두어 man이 한 cycle 동안 걸릴 시간을 결정한다. 그리고 d\_body vector로 body가 움직이는 양을 (0, 0.01, 0)으로 정의한다.

각 level에서 arm이 앞, 뒤로 움직이고, forearm이 body 앞으로 향하고 돌아가고, leg가 앞 뒤로 흔들리는 과정을 생각하면서 각 변수에 radian을 더하고 뺀다. 그리고 각 frame의 자세에서 특정되는 각도를 기준으로 삼아 다음 level로 보낸다.

Level 1 -> level 2: right lowerleg가 90도가 되는 경우

Level 2 -> level 3: left leg가 0도가 되는 경우

Level 3 -> level 4: left leg가 45도가 되는 경우

Level 4 -> level 5: left lowerleg가 90도가 되는 경우

Level 5 -> level 6: right leg가 0도가 되는 경우

Level 6 -> level 1: right leg가 45도가 되는 경우