**기초 컴퓨터 그래픽스**

HW3

제출일 : 5/20(월) 20:00

**20171690 정유석**

**명세서의 모든 기능 구현했습니다.**

**목차**

1. . 물체의 배치 및 움직임 구현
   1. . 움직이는 호랑이
   2. . 움직이는 자동차
   3. . 5개까지의 서로 다른 물체
      1. . COW
      2. . IRON MAN
      3. . 사람
      4. . WOLF
      5. . SPIDER (’**b**’)
   4. . 물체의 움직임, 멈춤 조절
      1. . ‘**a’** (모든 움직이는 물체)
      2. . ‘**i**’ (iron man)
      3. . ‘**m**’ ‘**n**’ (자동차)
      4. . ‘**z**’ ‘**x**’ (wolf)
2. . 카메라의 배치 및 움직임 구현
   1. . 주 카메라
      1. . 주 카메라 모드 전환 (’**r**’)
      2. . ZOOM IN/ OUT (**SHIFT + MOUSE left button**)
      3. . 카메라 이동 (**방향키**)
   2. . 부 카메라
      1. . 부 카메라 모드 전환 (’**l**’)
      2. . 앞뒤, 좌우, 위아래 이동 (**방향키**, **CTRL + 위아래 방향키**)
      3. . 좌우 회전 (**CTRL + 왼오른 방향키**)
      4. **.** ZOOM IN/ OUT (**SHIFT + MOUSE left button**)
   3. . 움직이는 자동차 운전석에 고정된 카메라 (’**d**’)
   4. . 움직이는 호랑이 눈에 고정된 카메라 (’**t**’)
3. . 추가 기능
   1. . 이동하는 자동차를 바라보는 카메라 (’**c**’)
   2. . 초록색 field에 나타나는 글씨 (’**h**’)
   3. . brick의 polygon mode (’**p**’)
4. . 물체의 배치 및 움직임 구현
   1. . **움직이는 호랑이**

호랑이는 주어진 공간의 초록색 field 위를 움직인다. 이 때 호랑이가 걸어가는 경로는 random한 직선이다. 호랑이가 경로를 따라 이동 중, field의 경계에 부딪히면 반사되는 방향의 random한 각도로 경로가 다시 계산된다.

* 1. . **움직이는 자동차**

자동차는 평면 위의 흰색 선을 따라 움직인다. 이 때 자동차의 방향이 바뀌는 지점에서 계층적 모델링 방식을 적용해 자동차의 wheel과 nut이 적절한 방향의 적절한 각도로 회전한다. 바퀴의 회전을 자연스럽게 하기 위해 앞바퀴와 뒷바퀴의 회전각도는 상이하다.

‘m’과 ‘n’ 버튼을 활용해 자동차의 속도를 조절할 수 있다. ‘m’ 버튼을 누르면 자동차의 속도가 빨라지고, ‘n’ 버튼을 누르면 속도가 느려진다. 이 때 바퀴가 회전하는 속도도 함께 변화한다.

‘d’ 버튼을 이용해 움직이는 자동차의 운전석에서 바라보는 세상을 나타낼 수 있으며, ‘c’ 버튼을 이용해 움직이는 자동차를 일정한 거리에서 관찰할 수 있다. 이 때 SHIFT + MOUSE left button을 이용해 ZOOM IN/ OUT 기능을 적용할 수 있다. ‘r’ 버튼 또는 ‘w’ 버튼을 눌러 다시 해당 모드를 해제할 수 있다.

* 1. . 최대 5개까지의 서로 다른 물체

(’l’ 버튼을 누른 후, 다시 ‘w’ 버튼을 눌러 KEY-UP 버튼으로 카메라 위치를 조절하면 field 영역을 관찰하기 용이합니다. ‘r’ 버튼을 누르면 원점을 바라보는 위치로 카메라가 이동합니다.)

* + 1. . **COW**

소는 평면의 초록색 field 위의 임의의 한 점에 위치한다. 소는 기본적으로 움직임이 없으며 고정된 위치에 자리한다.

한편 field를 돌아다니는 호랑이가 소와 충돌할 경우, 특별한 효과와 함께 소는 임의의 다른 위치로 이동한다. field의 영역을 y 축 기준으로 3등분 하며 3 구역으로 나눈다면, 소가 이동하는 위치는 3개의 구역 중 기존의 구역이 아닌 다른 2개의 구역 중 한 곳이 된다.

* + 1. . **IRON MAN**

ironman은 (-182.5, -70, 30)의 위치에 떠서 자신을 중심으로 360도 회전한다. 이 때 ‘i’ 버튼을 누르면 ironman이 회전하는 속도가 빨라지며 z 좌표가 증가한다. 다시 ‘i’ 버튼을 누르면 회전하는 속도가 느려지며 기존의 위치로 돌아온다.

* + 1. . **사람**

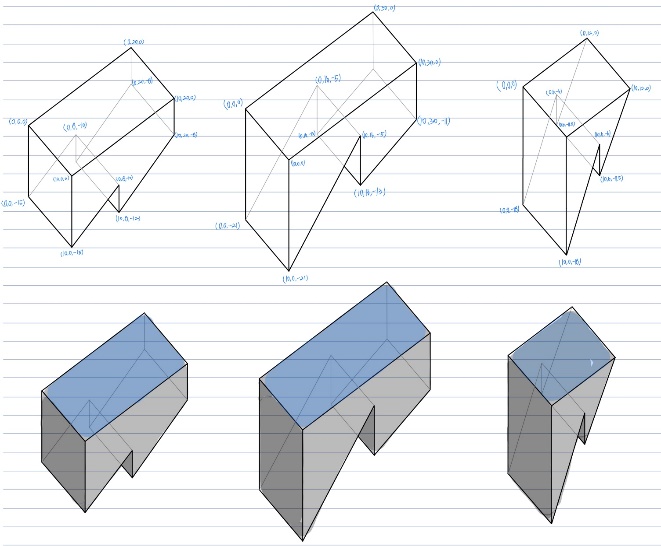
Field 주위의 노란 경로 안으로 사람이 달려간다.

* + 1. . **WOLF**

200의 높이에서 특정한 크기의 반지름을 갖는 원의 둘레를 따라 wolf가 뛰어간다. 이 때 ‘z’와 ‘x’ 버튼을 이용해 wolf가 회전하는 반지름의 크기를 조절할 수 있다. ‘z’ 버튼을 누르면 반지름의 크기가 작아지며, ‘x’ 버튼을 누르면 반지름의 크기가 커진다.

* + 1. . **SPIDER** (’**b**’)

하늘색의 영역 위에 3개의 물체가 떠있다. 이 때 세 물체(brick)는 아래의 도형을 모델링한 것이다.



SPIDER는 세 물체 중 가장 큰 물체의 위를 앞뒤로 이동한다. 세 물체는 공간 상에서 서로 떨어져 있으므로 SPIDER는 세 물체 사이를 이동할 수 없다. 하지만 ‘b’ 버튼을 눌렀을 경우, 세 물체가 같은 높이에 떠있어 발생하는 착시 현상으로 SPIDER는 세 물체를 옮겨다닐 수 있다.

SPIDER가 각 물체 사이를 이동할 때 x 좌표를 점진적으로 변경하여 원근감에 의해 크기가 자연스레 변한다. 카메라를 이동시키는 동작이 입력되었을 경우에, SPIDER는 자신이 속한 brick 위만을 앞뒤로 이동한다.

한편 ‘p’ 버튼을 눌러 brick의 polygon mode를 토글할 수 있다. 누를 때마다 LINE, FILL 모드가 토글된다.

* 1. . **물체의 움직임, 멈춤 조절**
     1. . ‘**a’** (모든 움직이는 물체)

‘a’ 버튼을 누르면 움직이는 모든 물체가 정지한다. ‘a’ 버튼을 한 번 더 누르면 다시 모든 물체가 움직인다.

* + 1. . ‘**i**’ (iron man)

‘I’ 버튼을 누르면 iron man이 위로 (z축 양의 방향으로) 이동한다. 이 때 회전 속도는 빨라진다. ‘I’ 버튼을 다시 누르면 기존 위치로 돌아온다.

* + 1. . ‘**m**’ ‘**n**’ (자동차)

‘m’ 버튼과 ‘n’ 버튼을 이용해 이동하는 자동차의 속도를 조절한다. ‘m’을 누르면 빨라지고 ‘n’을 누르면 느려진다.

* + 1. . ‘**z**’ ‘**x**’ (wolf)

‘z’ 버튼과 ‘x’ 버튼을 이용해 이동하는 wolf의 반지름 크기를 조절한다. ‘z’를 누르면 반지름이 감소하고, ‘x’를 누르면 반지름이 증가한다.

1. . 카메라의 배치 및 움직임 구현
   1. . 주 카메라
      1. . **주 카메라 모드 전환** (’**r**’)

‘r’ 버튼을 눌러 주 카메라 모드로 전환한다. 이 때 PRP는 (500, 400, 400)으로, VRP는 원점으로 초기화된다.

한편 VRP와 PRP가 달라진 상태의 다른 모드에서 ‘w’ 버튼을 누르면 주 카메라 모드로 변경되어 주 카메라 모드의 동작(2-1-2, 2-1-3에 기술된 동작)을 수행할 수 있다.

* + 1. . **ZOOM IN/ OUT** (**SHIFT + MOUSE left button**)

주 카메라 모드에서 SHIFT 버튼을 누른 채 왼쪽 마우스 버튼을 누르고 마우스를 좌-우로 움직이면 각각 화면이 ZOON OUT, ZOOM IN 된다.

* + 1. . **카메라 이동** (**방향키**)

주 카메라 모드에서 방향키 버튼을 이용해 카메라의 위치를 이동할 수 있다. 이 때 VRP는 원점으로 일정하다.

* 1. . 부 카메라
     1. . **부 카메라 모드 전환** (’**l**’)

‘l’ 버튼을 누르면 vrp = (-142.5, 0, 15), prp = (-400, 400, 15)로 하여 field를 바라보는 카메라로 이동한다. 이 때 부 카메라의 v 벡터 방향은 항상 (0, 0, 1)로 일정하다.

한편 ‘w’ 버튼을 이용해 다시 주 카메라 모드로 변경가능하다. 이 때 vrp는 원점이 아니며 prp와 vrp가 부 카메라 모드에서 이동한 위치 그대로이므로 과제에서 요구하는 주 카메라를 나타내진 않는다. 그러나 기존의 주 카메라 모드에서 사용하던 방향키를 이용해 고정된 VRP 둘레로 카메라를 이동시키며 세상 좌표를 볼 수 있다.

* + 1. . **앞뒤, 좌우, 위아래 이동** (**방향키**, **CTRL + 위아래 방향키**)

부 카메라 모드에서 방향키 버튼을 이용해 앞뒤, 좌우, 위아래로 이동할 수 있다.

KEY-UP 버튼은 앞으로, KEY-DOWN은 뒤로, KEY-LEFT 버튼은 왼쪽으로, KEY-RIGHT 버튼은 오른쪽으로, CTRL + KEY-UP 버튼은 위로, CTRL + KEY-DOWN 버튼은 아래로 이동한다. 각각 -n, u, v 벡터 방향을 따라 카메라가 이동한다.

* + 1. . **좌우 회전** (**CTRL + 왼오른 방향키**)

부 카메라 모드에서 방향키를 이용해 카메라 위치(PRP)를 중심으로 좌우로 회전할 수 있다.

CTRL + KEY-LEFT 버튼을 이용해 왼쪽으로 회전하고, CTRL + KEY-RIGHT 버튼을 이용해 오른쪽으로 회전한다.

* + 1. **. ZOOM IN/ OUT** (**SHIFT + MOUSE left button**)

주 카메라 모드에서와 마찬가지로 SHIFT키를 누르고 마우스 왼쪽 버튼을 누른 상태에서 마우스를 좌-우로 움직이면 ZOOM OUT, ZOOM IN 된다.

* 1. . **움직이는 자동차 운전석에 고정된 카메라** (’**d**’)

‘d’ 버튼을 누르면 자동차의 운전석에서 자동차가 이동하는 방향을 바라보는 카메라가 설정된다. 이 때 SHIFT + 마우스 왼쪽 버튼을 이용해 ZOOM IN과 ZOOM OUT 기능을 사용할 수 있다.

* 1. . **움직이는 호랑이 눈에 고정된 카메라** (’**t**’)

‘t’ 버튼을 누르면 움직이는 호랑이가 세상을 바라본 세상이 나타난다. 운전석에 고정된 카메라와 마찬가지로 SHIFT + 마우스 왼쪽 버튼을 이용해 ZOOM IN/ OUT 기능을 사용 가능하다.

1. . 추가 기능
   1. . **이동하는 자동차를 바라보는 카메라** (’**c**’)

‘c’ 버튼을 누르면 이동하는 자동차를 적당히 떨어진 거리에서 관찰할 수 있다. 이 때 SHIFT + 마우스 왼쪽 버튼을 이용해 ZOOM IN/ OUT을 조절하여 바라보는 크기를 조절할 수 있다.

* 1. . **초록색 field에 나타나는 글씨** (’**h**’)

‘h’ 버튼을 이용해 field에 글씨를 toggle할 수 있다. 이 때 나타나는 글은 ‘RUSH’ 이다.

* 1. . **brick의 polygon mode** (’**p**’)

‘p’ 버튼을 이용해 모델링한 3개의 물체의 polygon 모드를 토글할 수 있다.