**[CSE4170/AIE4012] 기초 컴퓨터 그래픽스**

**HW3: OpenGL API 함수를 사용한3차원뷰잉연습**

컴퓨터공학과 20231508 강유영

1. **기능 구현**

**(a) 예제 프로그램에서의 기하 물체 배치 외에 추가적으로 3차원 기하 물체를 배치하라.**

**i. Static Object 5개 구현**

구현 여부 : O

채점 점수 : 50점

구현 내용 : 정적인 물체 Table, Light, Teapot, New\_Chair, Frame을 구현하였다.

* + Table : (157.0f, 76.5f, 0.0f)와 (0.8f, 0.6f, 0.6f)의 위치에 2개 생성하였다. Scale 비율을 각각 0.5f와 0.6f로 하여 다른 크기를 가지도록 구현하였다.
  + Light : 5개를 구현하였는데, 3개는 rotate 각도 없이 building의 천장 높이에 붙어있고 2개는 90.0f rotate를 하여 벽면에 옆으로 붙어있는 것처럼 구현하였다.
  + Teapot : (155.0f, 80.0f, 10.0f)의 위치에 생성하였다. 두 번째 테이블 위에 있는 것처럼 표현하였다.
  + New\_Chair : (140.0f, 135.0f, 0.0f)의 위치에 생성하였다.
  + Frame : (115.0f, 116.0f, 30.0f)의 위치에 생성하였고, 벽면에 붙어있는 것처럼 구현하였다.

**ii. Dynamic Object 2개 구현**

구현 여부 : O

채점 점수 : 100점

구현 내용 : 동적으로 움직이는 Tiger\_D와 Cow\_D를 구현하였다.

* Tiger\_D : 지정된 7개 지점 (190,10 -> 190,70 -> 210,70 -> 210,100 -> 80,100 -> 80,70 -> 20,70)을 순서대로 선형이동하며, 마지막 지점에 도달하면 같은 경로를 역순으로 되돌아간다. 이동 중 항상 진행 방향을 바라보도록 heading변수를 이용해 회전시켜 자연스러운 애니메이션을 구현하였다.
* Cow\_D : (170,150)에서 (170,100)으로 천천히 걸어간 뒤, (80,100)으로 빠르게 이동한다. 이후 그 자리에서 2바퀴 회전 후 (80,20)으로 걸어간 다음 역순으로 되돌아 가는 경로를 반복한다. 호랑이와 마찬가지로 heading 변수를 이용해 회전시켜 이동 방향을 바라보도록 구현하였다.

**(b) 수업 시간 및 튜토리얼 시간에 설명한 가상 카메라를 설계하고 구현하라.**

**i. 주 카메라가 카메라의 세 기본축 방향으로 이동**

구현 여부 : O

채점 점수 : 30점

구현 내용 : Camera.cpp의 Perspective\_Camera 클래스에서 move\_forward(), move\_backward(), move\_left(), move\_right(), move\_up(), move\_down()을 정의하였고, 키보드 콜백에서 다음과 같이 정의를 하였다.

* w / s : naxis를 기준으로 앞뒤로 이동한다.
* a / d : uaxis를 기준으로 좌우로 이동한다.
* q / e : vaxis를 기준으로 위아래로 이동한다.

**ii. 주 카메라가 카메라의 세 기본축 둘레로 회전**

구현 여부 : O

채점 점수 : 30점

구현 내용 : Perspective\_Camera 클래스 내에 yaw\_left, yaw\_right(v축 회전), pitch\_up(), pitch\_down()(u축 회전), roll\_left, roll\_right(n축 회전) 매서드를 구현하였다. 그리고 키보드 콜백에서 다음과 같이 정의하여 카메라가 각 로컬 축을 기준으로 회전하도록 하였다.

* j / l : v축을 기준으로 왼쪽, 오른쪽 회전한다.
* i / k : u축을 기준으로 위쪽, 아래쪽으로 회전한다.
* u / o : n축을 기준으로 왼쪽, 오른쪽으로 회전한다.

**iii. 주 카메라가 화면의 내용을 확대 또는 축소**

구현 여부 : O

채점 점수 : 20점

구현 내용 : Perspective\_Camera 클래스에 zoom\_in(), zoom\_out() 매서드를 구현하여 cam\_proj.params.pers.fovy 값을 조절하게 하였다. 마우스 콜백에서 마우스 휠 이벤트 button == 3, button == 4을 감지하여 마우스 휠을 올리면 확대를 하고, 휠을 내리면 축소를 한다. (b)-v 항목에서 구현할 동적인 카메라와 분리를 하기 위해 currentCam != 4 인 경우에만 zoom\_in(), zoom\_out()을 호출한다.

**iv. 주 카메라 외에 정적인 CCTV 3개 구현**

구현 여부 : O

채점 점수 : 45점

구현 내용 : display() 함수 내에서 currentCam = 1,2,3 일 때 각각 다른 eye[], center[]값을 사용해 glm::lookAt(eye, center, up)로 뷰 행렬을 설정하였다.

* CCTV 1 : (145,90,50) 에서 (140, 105, 10) 바라본다.
* CCTV 2 : (225,75,50) 에서 (210,75,10) 바라본다.
* CCTV 3 : (40,110,50) 에서 (50,80,10) 바라본다.

키보드 콜백에 키보드 1번을 누르면 CCTV 1, 2번을 누르면 CCTV 2, 3번을 누르면 CCTV 3번으로 전환되도록 하였다. 키보드 0번을 누르면 주 카메라로 돌아온다.

**v. 한 개의 동적인 CCTV구현 (field of view로 원격 조절)**

구현 여부 : O

채점 점수 : 30점

구현 내용 : 키보드 콜백에서 4번을 누르면 동적 CCTV 모드로 전환되게 하였다.

* cam\_eye는 고정된 (100,100,40)이다.
* 카메라 방향은 dynYaw, dynPitch 변수를 사용해 dir = (cos\*cos, cos\*sin, sin) 계산 후 glm::lookAt(dynEye, dynEye + dir, (0,0,1)) 으로 뷰 행렬을 구성하였다.
* 키보드의 화살표 ←/→로 dynYaw를 조절하고, ↑/↓로 dynPitch를 조절하여 카메라가 보는 방향을 실시간으로 변경할 수 있게 하였다.
* 마우스 휠 이벤트에서는 dynFov의 값을 +1, -1씩 glm::clamp한 뒤 glm::perspective(glm::radians(dynFov), aspect, near, far)로 원근 투영 매트릭스를 갱신한다. 이를 통해 동적 CCTV를 고정된 위치에서 FOV를 원격으로 조절할 수 있다.

**vi. 직교 투영을 사용하여 정면도-측면도-상면도 구현**

구현 여부 : O

채점 점수 : 45점

구현 내용 : 키보드 콜백에 5번은 상면도, 6번은 측면도, 7번은 정면도를 보여주도록 구현하였다.

* 상면도 : eye = (0,0,300), up = (0,1,0)
* 측면도 : eye = (300,0,0), up = (0,0,1)
* 정면도 : eye = (0,300,0), up = (0,0,1)

공통으로 center = (0,0,0)으로 설정하였다. glm::ortho(left, right, bottom, top, zNear, zFar)를 통해 원근 없이 평행 투영된 뷰를 구현하였다.

**vii. 주 카메라와 CCTV 카메라 프레임을 적절한 키로 토글**

구현 여부 : O

채점 점수 : 30점

구현 내용 : 전역변수로 static bool showCamFrame = false;를 선언하였다. 그리고 키보드 콜백에 f를 누르면 showCamFrame = !showCamFrame;을 하여 켜고 끌 수 있게 하였다.

display() 함수에서 showCamFrame이 true 일 때, 주 카메라의 경우 뷰 행렬을 역산해 카메라의 위치를 구하고 각 축을 RGB선으로 그렸다. 정적 CCTV의 경우 각기 정의된 eyeC[], cenC[]벡터를 사용해 forward, right, up을 계산한 뒤 RGB선으로 그렸다.

1. **기능별 사용법**

주 카메라 이동

* w / s : 앞, 뒤 이동 (n축)
* a / d : 좌, 우 이동 (u축)
* q / e : 위, 아래 이동 (v축)

주 카메라 회전

* j / l : v축을 중심으로 왼쪽, 오른쪽 회전
* i / k : u축을 중심으로 위쪽, 아래쪽 회전
* u / o : n축을 중심으로 시계 반대, 시계 방향으로 회전

주 카메라 확대 및 축소

* 마우스 휠 up : 주 카메라 확대
* 마우스 휠 down : 주 카메라 축소

카메라 전환

* 0 : 주 카메라 전환
* 1 : CCTV 1(정적 카메라) 전환
* 2 : CCTV 2(정적 카메라) 전환
* 3 : CCTV 3(정적 카메라) 전환
* 4 : 동적 카메라 전환
  + - 키보드 ←/→/↑/↓: 동적 카메라 이동 (4번 상태에서만 사용)
    - 마우스 휠 up : field of view 증가 (4번 상태에서만 사용)
    - 마우스 휠 down : field of view 감소 (4번 상태에서만 사용)
* 5 : 상면도 전환
* 6 : 측면도 전환
* 7 : 정면도 전환

기타

* f : 카메라 프레임 토글
* z : 면 제거 모드를 없음, 뒤면, 앞면 순으로 순환 (스켈레톤 코드에 있던 기능, 키 옮김)
* x : 폴리곤 채움 모드와 선 모드를 번갈아 그림 (스켈레톤 코드에 있던 기능, 키 옮김)
* c : 깊이 테스트를 켜거나 끔 (스켈레톤 코드에 있던 기능, 키 옮김)