

HW13 3D objects

8조

남궁호민

오지훈

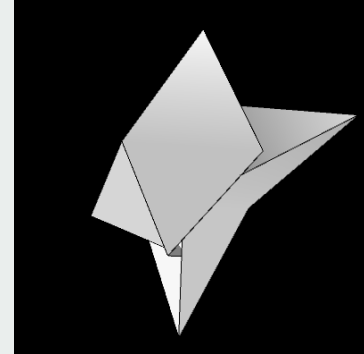
현시온



8조 내용설명

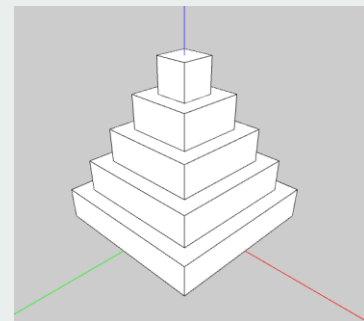
남궁호민

화면상 3D 오브젝트인 뾰족한 육면체가 돌아가는 프로그램입니다. 6개의 면으로 나누어서 vertex를 사용하였습니다. 원래 다른 것을 만들려고 하였으나, 마치 별모양 같은 모습이 마음에 들어 제출하기로 마음먹었습니다.



오지훈

PPT의 사각뿔 피라미드를 보고 참고하여 직육면체를 각 층으로 쌓아 피라미드를 만들었습니다.



남궁호민

2D프로그램만 만들다가 3D프로그램을 만들다 보니 어렵고 생각해야 할 부분이 많았습니다. 하지만 결과물만 보면 3D프로그램이 좀더 만족도가 높고 퀄리티가 높은 편인 것 같습니다. 조금만 더 배우면 재미있을 것 같습니다.

오지훈

TRIANGLE_STRIP이 아니라 QUAD_STRIP을 사용해봤는데 처음에는 몰라서 많은 실패를 해보다가 사용법을 깨달을 수 있었습니다. camera 역시 교수님의 PPT 원통 코드를 참고하여 실험적으로 사용법을 알 수 있게 되었습니다. 2D에 비해서 3D가 더 사용하기 까다롭지만 이를 잘 사용하면 더 몰입감 있는 그래픽을 만들 수 있을 것 같습니다.

8조 소스코드

남궁호민

```
void setup() {
  size(400, 400, P3D);
}

void draw() {
  background(0);
  lights();
  translate(width/2, height/2, 0);
  rotateX(frameCount * 0.05);
  rotateY(frameCount * 0.05);
  beginShape(QUADS);
  vertex(-50, -50, 50);
  vertex(100, -100, 100);
  vertex(50, 50, 50);
  vertex(-100, 100, 100);
  vertex(-50, -50, -50);
  vertex(100, -100, -100);
  vertex(50, 50, -50);
  vertex(-100, 100, -100);
  vertex(-50, -50, 50);
  vertex(-100, -100, -100);
  vertex(-50, 50, -50);
  vertex(-100, 100, 100);

  vertex(50, -50, 50);
  vertex(100, -100, -100);
  vertex(50, 50, -50);
  vertex(100, 100, 100);
  vertex(-50, -50, 50);
  vertex(-100, -100, -100);
  vertex(50, -50, -50);
  vertex(100, -100, 100);
  vertex(-50, 50, 50);
  vertex(-100, 100, -100);
  vertex(50, 50, -50);
  vertex(100, 100, 100);
  endShape();
}
```

오지훈

```
void setup() {
  size(800, 800, P3D);
  camera(150, 150, 150, 0, 0, 0, 1, 1, 0);
  stroke(255,0,0);
  line(0,0,0, 800,0,0);
  stroke(0,255,0);
  line(0,0,0, 0,800,0);
  stroke(0,0,255);
  line(0,0,0, 0,0,800);
  stroke(0);
  for (int i = 0; i<5; i++) {
    floor(0, 0, i*20, 100-i*20);
  }
}

void floor(int x, int y, int z, int d) {
  // (x,y,z) 는 바닥면 중앙의 좌표
  beginShape(QUAD_STRIP);
  vertex(x-d/2,y-d/2,z+20);
  vertex(x+d/2,y-d/2,z+20);
  vertex(x-d/2,y+d/2,z+20);
  vertex(x+d/2,y+d/2,z+20);
  vertex(x-d/2,y+d/2,z);
  vertex(x+d/2,y+d/2,z);
  vertex(x-d/2,y-d/2,z);
  vertex(x+d/2,y-d/2,z);
  endShape(QUAD_STRIP);
  beginShape(QUAD_STRIP);
  vertex(x-d/2,y+d/2,z+20);
  vertex(x-d/2,y+d/2,z);
  vertex(x-d/2,y-d/2,z);
  vertex(x+d/2,y-d/2,z);
  vertex(x+d/2,y-d/2,z);
  vertex(x+d/2,y+d/2,z);
  vertex(x+d/2,y+d/2,z);
  endShape(QUAD_STRIP);
}
```