지오메트리 변환 (4_2)

Try

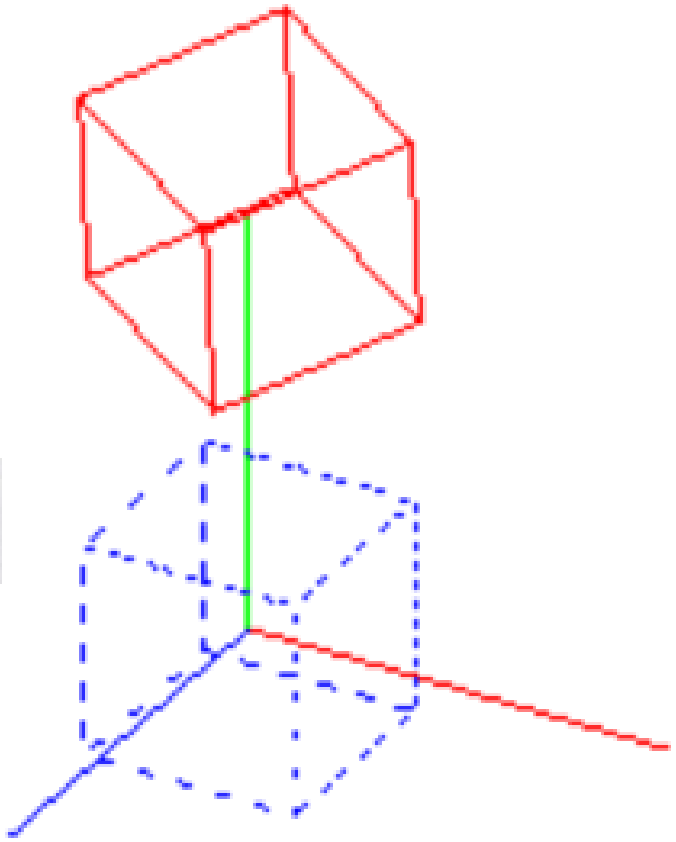
4_1 code를 기반으로 작성하시오

- 회전
  - 예) GLUT 라이브러리의 glutWireCube 함수를 사용하여 육면체를 그리고, 아래 출력을 만드시오

```
glutWireCube(10.0f);
```

✓ 하나의 버텍스(원점)를 중심으로 하여 10 크기만큼의 변 길이를 가지는 육면체

- **y축 양의 방향으로 10 만큼 이동한 후,**
- **(1,1,1)점과 원점을 연결한 벡터를 중심으로 육면체를 45도 회전하는 도형 그리기**



```
glRotatef(GLfloat 각도, GLfloat x, GLfloat y, GLfloat z);
```

✓ 원점과 x, y, z 좌표의 점을 연결하는 선을 기준으로 회전

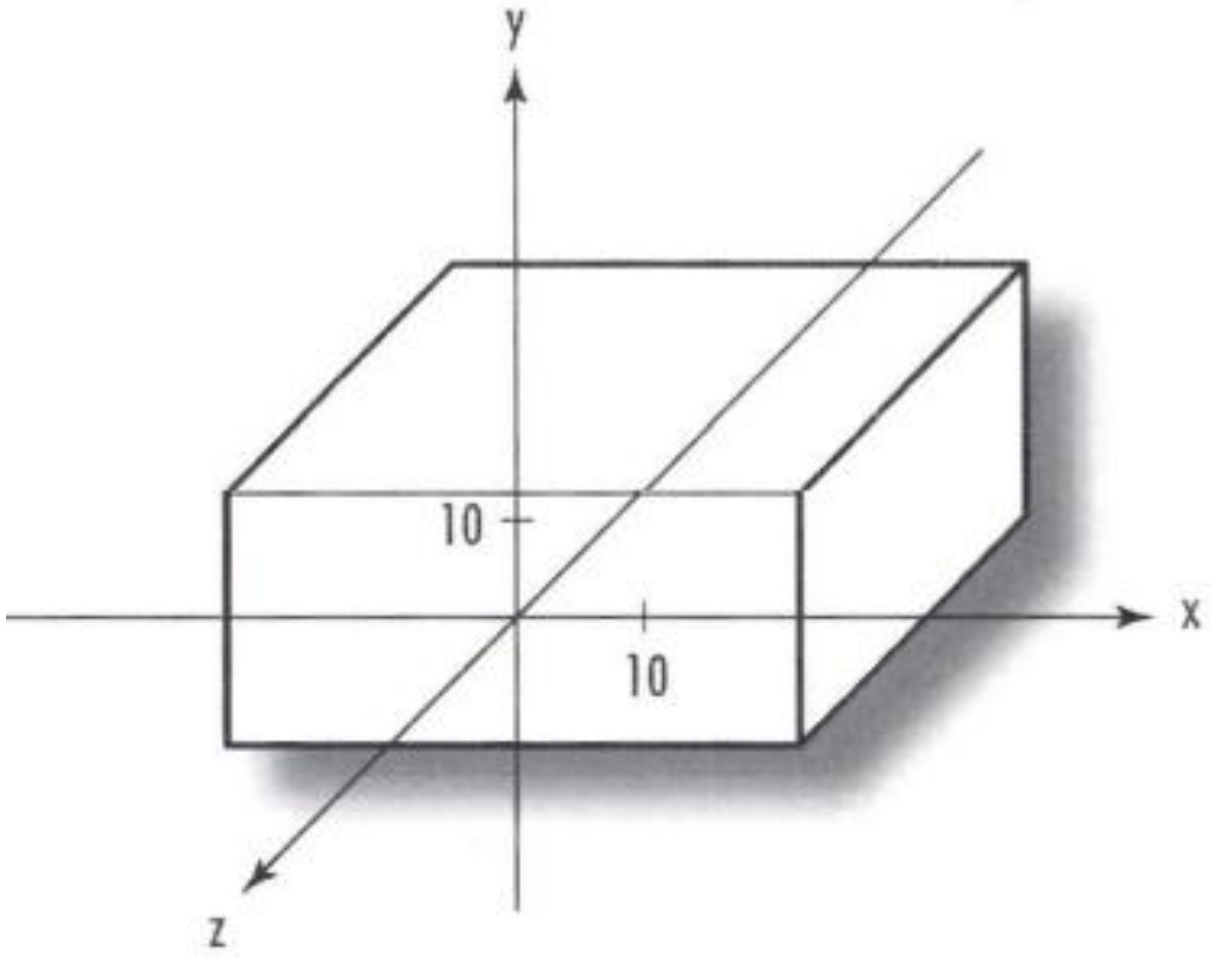
✓ 회전 각도는 반시계(CCW) 방향의 도 단위

지오메트리 변환

- 크기

```
glScalef(GLfloat x, GLfloat y, GLfloat z);
```

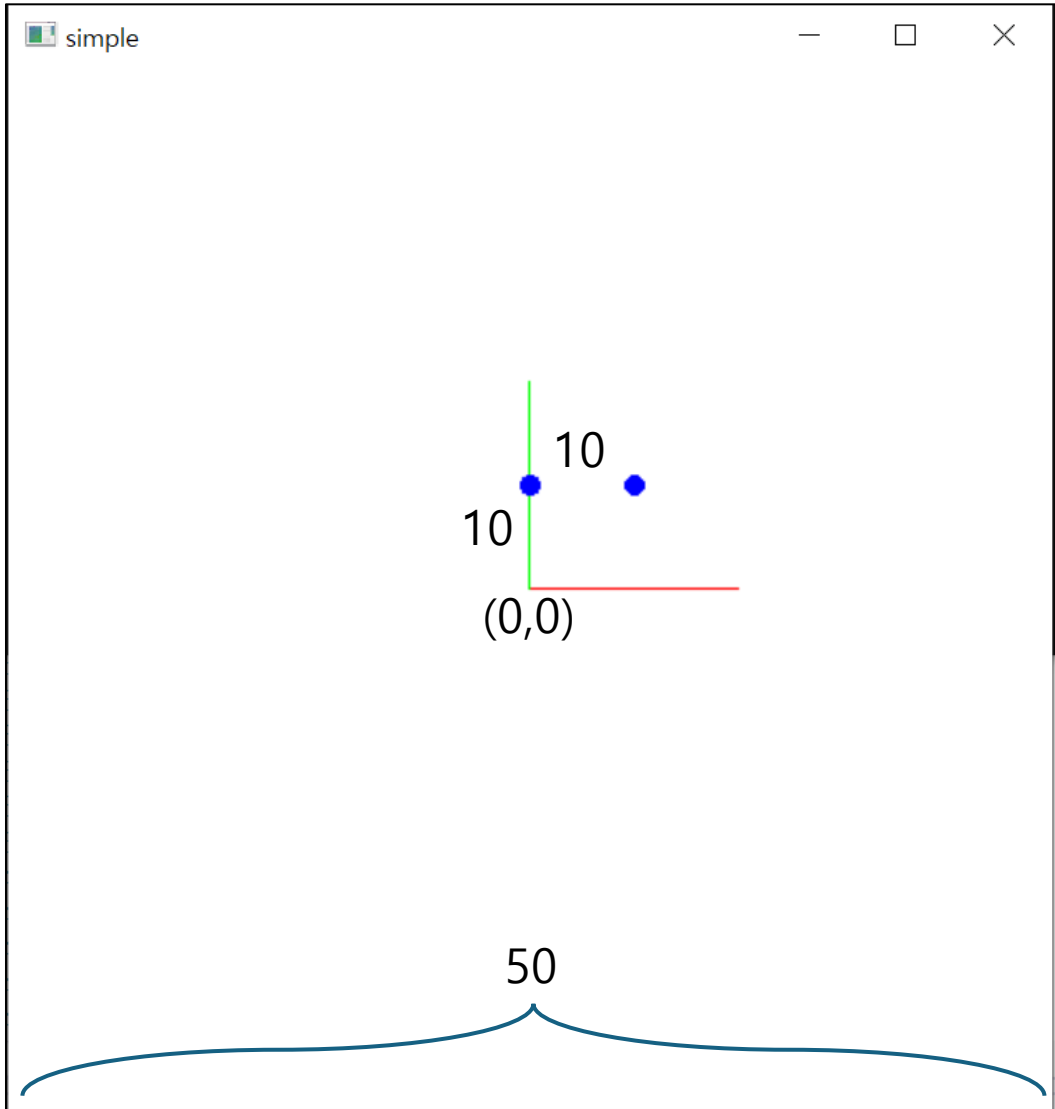
- 예
  - glScalef(2.0f, 1.0f, 2.0f) ;
    - ✓ x축으로 두 배 늘림
    - ✓ y축으로 크기는 그대로
    - ✓ z축으로 2배 늘림



지오메트리 변환

Try

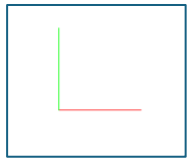
- 구(sphere) 를 그리는 함수를 이용하여 다음을 2개의 구를 출력하시오
  - sphere *radius = 1* ➔ `glutSolidSphere(1, 30, 30);`
  - glutSolidSphere(1, 30, 30)와 **동일한 좌표를 2개 이용**
  - glTranslatef()를 **2번 이용**



- **glutSolidSphere**(GLdouble *radius*, GLint *slices*, GLint *stacks*)
  - 좌표의 중심을 기준으로 그려짐

인자	의미
radius	반지름
slices	구의 세로 방향 분할 수 (경도)
stacks	구의 가로 방향 분할 수 (위도)

- view 방향은 기본
  - R,T 없음



```
void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 0.0f);

    //code
```

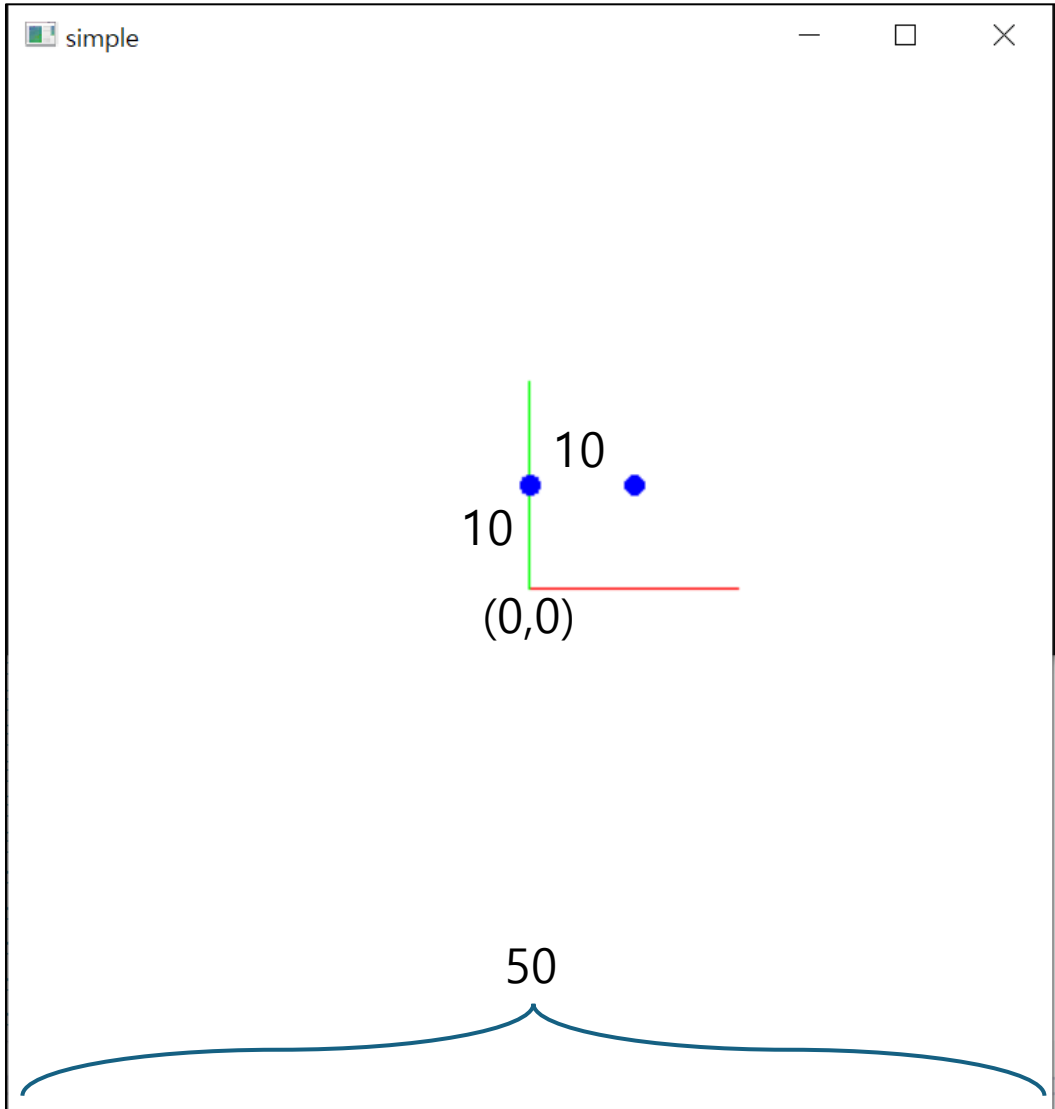
지오메트리 변환

- 구(sphere) 를 그리는 함
  - sphere *radius = 1* ➔
  - glTranslatef()를 **2번**

- **glutSolidSphere**(GLdo
  - 좌표의 중심을 기준으

인자	
radius	
slices	구의
stacks	구의

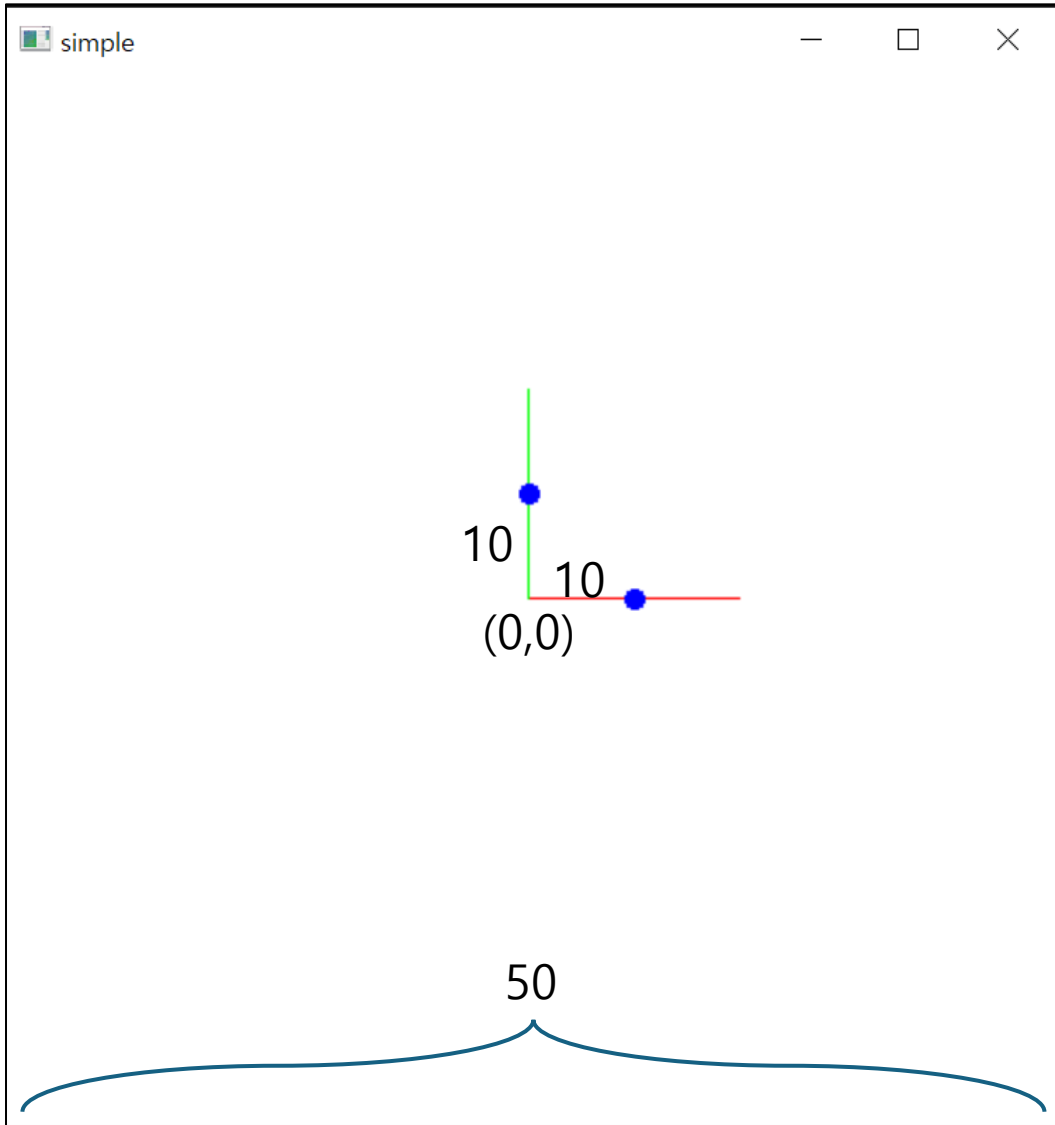
```
//code
glColor3f(1.0f, 0.0f, 0.0f);
glBegin(GL_LINES);
glVertex3f(0, 0, 0);
glVertex3f(20, 0, 0);
glEnd();
glColor3f(0.0f, 1.0f, 0.0f);
glBegin(GL_LINES);
glVertex3f(0, 0, 0);
glVertex3f(0, 20, 0);
glEnd();
glColor3f(0.0f, 0.0f, 1.0f);
glBegin(GL_LINES);
glVertex3f(0, 0, 0);
glVertex3f(0, 0, 20);
glEnd();

glTranslatef(0, 10, 0);
glutSolidSphere(1, 30, 30);
glTranslatef(10, 0, 0);
glutSolidSphere(1, 30, 30);
```



지오메트리 변환

Try

- 구(sphere) 를 그리는 함수를 이용하여 다음을 출력하시오.
  - sphere *radius = 1* ➔ `glutSolidSphere(1, 30, 30);`
  - glutSolidSphere(1, 30, 30)와 **동일한 좌표를 2개 이용**
  - glTranslatef()를 **2번 이용**



- **glutSolidSphere**(GLdouble *radius*, GLint *slices*, GLint *stacks*)
  - 좌표의 중심을 기준으로 그려짐

인자	의미
radius	반지름
slices	구의 세로 방향 분할 수 (경도)
stacks	구의 가로 방향 분할 수 (위도)

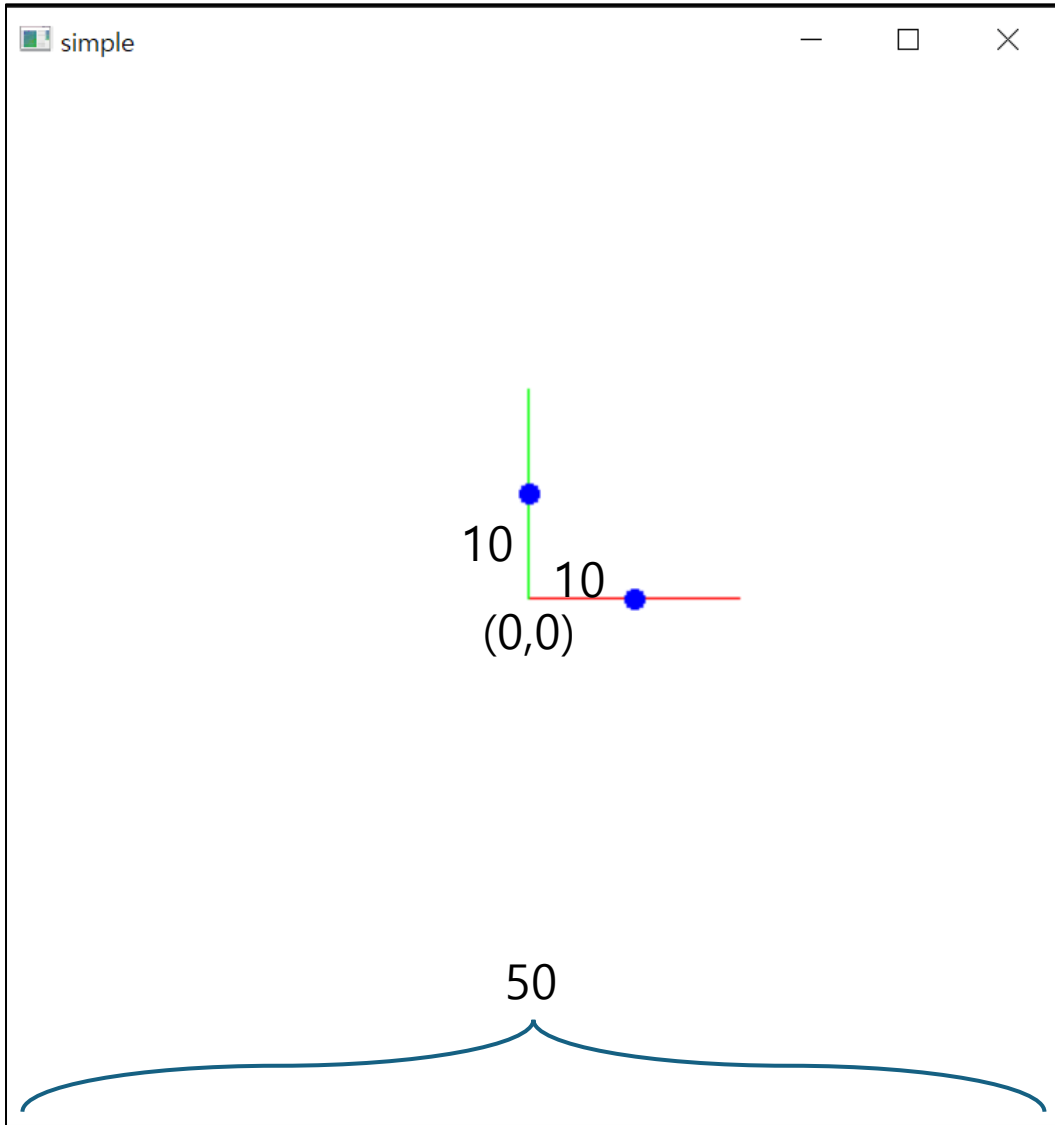
지오메트리 변환

- 구(sphere) 를 그리는 함수를 이용하여 다음을 출력하시오.
  - sphere *radius = 1* ➔ `glutSolidSphere(1, 30, 30);`
  - glTranslatef()를 **2번 이용**

- **glu** *cks)*
  -

```
glTranslatef(0, 10, 0);
glutSolidSphere(1, 30, 30);
glLoadIdentity();
glTranslatef(10, 0, 0);
glutSolidSphere(1, 30, 30);
```

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
```



지오메트리 변환

- 모델뷰 행렬을 초기화 하면 됨
- 방법은 그 내 용을 단위 행렬 (identity matrix)로 대체하는 것
  - 아무런 변환이 일어나지 않은 초기 상태로 되돌리는 것
  - 단위 행렬이란 대각선 행의 값들만 1이고 나머지는 모두 0인 행렬

$$\begin{bmatrix} 8.0 & 4.5 & -2.0 & 1.0 \end{bmatrix} \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} = \begin{bmatrix} 8.0 & 4.5 & -2.0 & 1.0 \end{bmatrix}$$

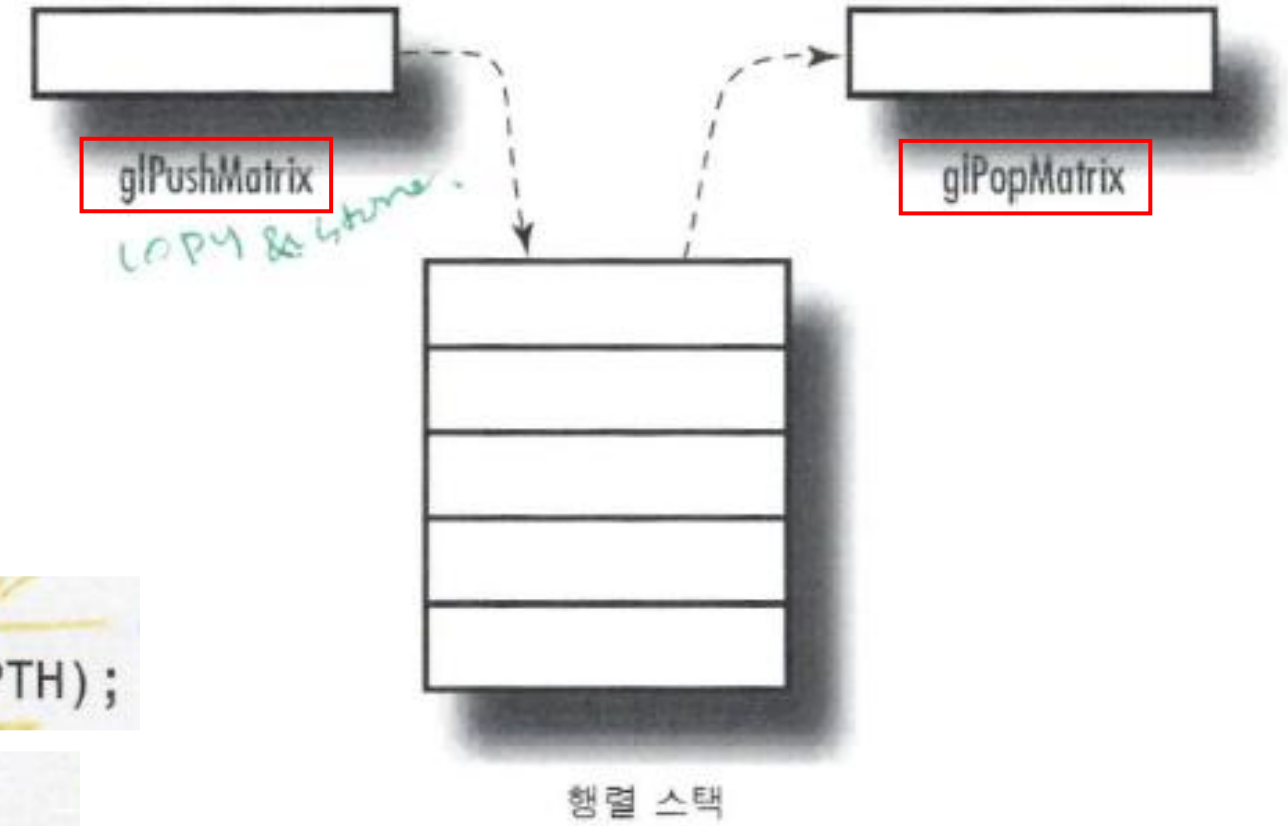
- 현재 다루는 행렬이 모델뷰 행렬임 을 지정하고
- 단위 행렬을 현재 행렬의 내용으로 로딩

```
// 단위 행렬을 모델뷰 행렬로 설정
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

// y축으로 10만큼 이동
glTranslatef(0.0f, 10.0f, 0.0f);

// 첫 번째 구를 그림
glutSolidSphere(1.0f, 15, 15);

// 모델뷰 행렬을 다시 초기화
glLoadIdentity();
```

지오메트리 변환

- 행렬 스택
  - 각 물체의 위치를 지정할 때마다 단위 행렬을 사용하여 모델뷰 행렬을 초기화하는 방법은 번거로움
    - ✓ 현재의 변환 상태를 저장해 두었다가 이후에 다시 복구해야 할 경우도 종종 있음
    - ✓ 매번 원점으로 돌아가는 것 이 아니라 특정한 변환을 거친 상태로 돌아가야 하는 경우도 있음

- 행렬 스택은 모델뷰 행렬과 투영 행렬을 저장해두는 일종의 저장 공간
- 다른 프로그램 스택과 마찬가지로 현재 행렬을 스택에 저장했다가 필요할 때 꺼내 쓰는 구조
- glPushMatrix();
- glPopMatrix();



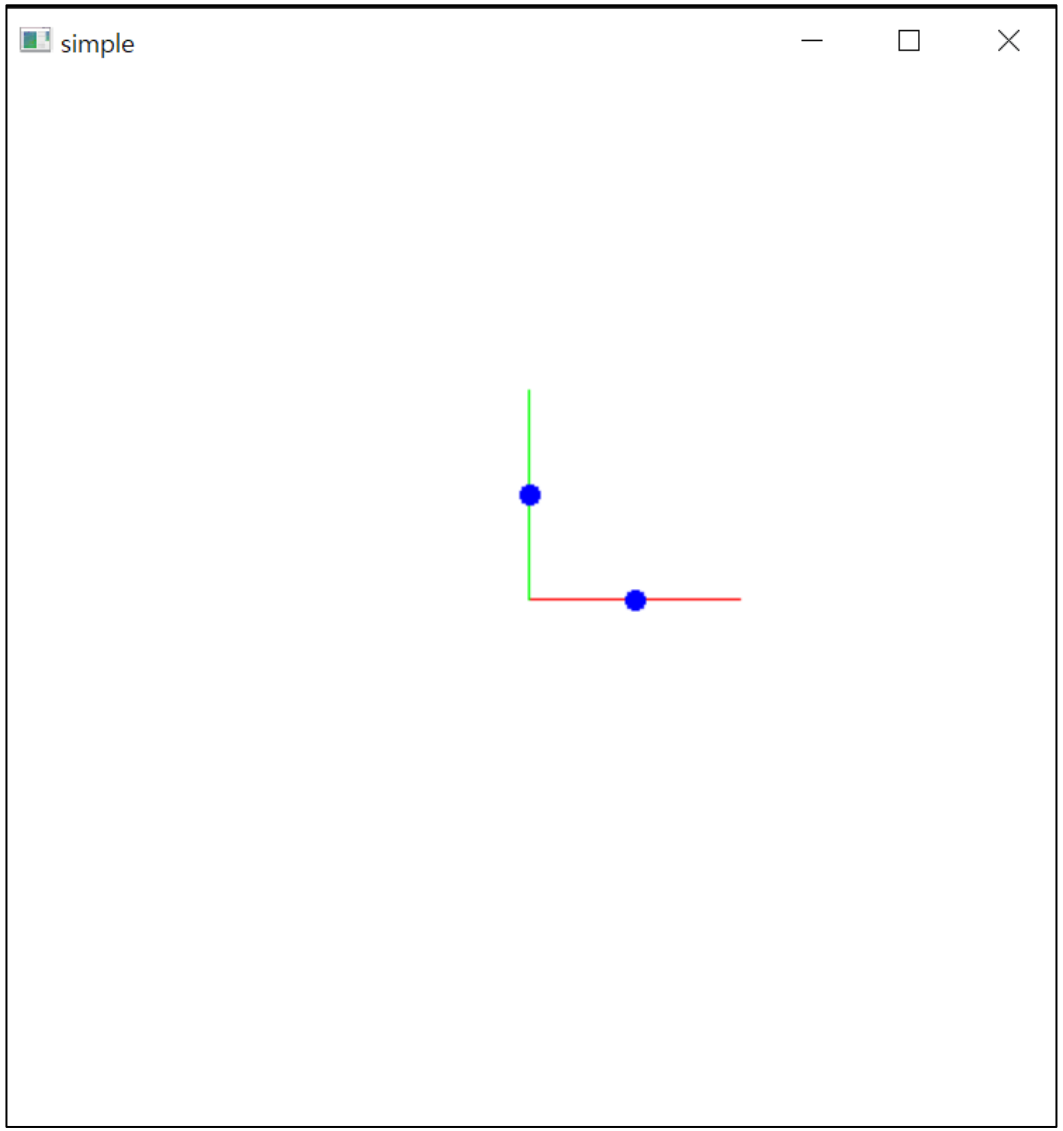

```
// 모델뷰 행렬의 최대 깊이
glGet(GL_MAX_MODELVIEW_STACK_DEPTH);
```

```
// 투영 행렬의 최대 깊이
glGet(GL_MAX_PROJECTION_STACK_DEPTH);
```

지오메트리 변환

- glPushMatrix();
- glPopMatrix();
- glLoadIdentity() 이용 금지
- 이것을 이용해서 오른쪽 예제를 만드시오

```
glTranslatef(0, 10, 0);
glutSolidSphere(1, 30, 30);
glLoadIdentity();
glTranslatef(10, 0, 0);
glutSolidSphere(1, 30, 30);
```



지오메트리 변환 - 잠깐, glutTimerFunc()

- **일정 시간 간격으로 콜백 함수를 호출**할 수 있도록 하는 타이머 함수
- 애니메이션이나 주기적인 작업을 구현하기 위해 타이머가 필요

- void **glutTimerFunc**(unsigned int **millis**, void (***func**)(int value), int **value**);
  - 일정시간(millis) 뒤에 func함수 실행
  - **한번만** 실행

매개변수	설명
millis	타이머가 작동할 시간(밀리초 단위, 1000ms = 1초)
func	지정한 시간이 지난 후 호출될 콜백 함수의 포인터
value	콜백 함수로 전달될 정수 값 (보통 상태나 ID 식별에 사용)

- 예) `glutTimerFunc(1000, TimerFunction, 1);`
  - ✓ 1000ms마다 TimerFunction(int value) 함수를 호출함

지오메트리 변환 - 잠깐, glutTimerFunc() 4_3_0

- 기본 구성 code

```
#include <GL/glut.h>

void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);

    //code


    glutSwapBuffers();
}

void TimerFunction(int value)
{
    // 화면 다시 그리기 요청,
    //glutDisplayFunc에 등록된 함수 실행
    glutPostRedisplay();

    // 1초마다 다시 호출
    glutTimerFunc(1000, TimerFunction, 1);
}
```

```
void SetupRC()
{
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutCreateWindow("Timer Example");

    glutDisplayFunc(RenderScene);
    SetupRC();

    // 처음 타이머 호출
    glutTimerFunc(1000, TimerFunction, 1);

    glutMainLoop();
    return 0;
}
```

지오메트리 변환 - 잠깐, glutTimerFunc() 4_3

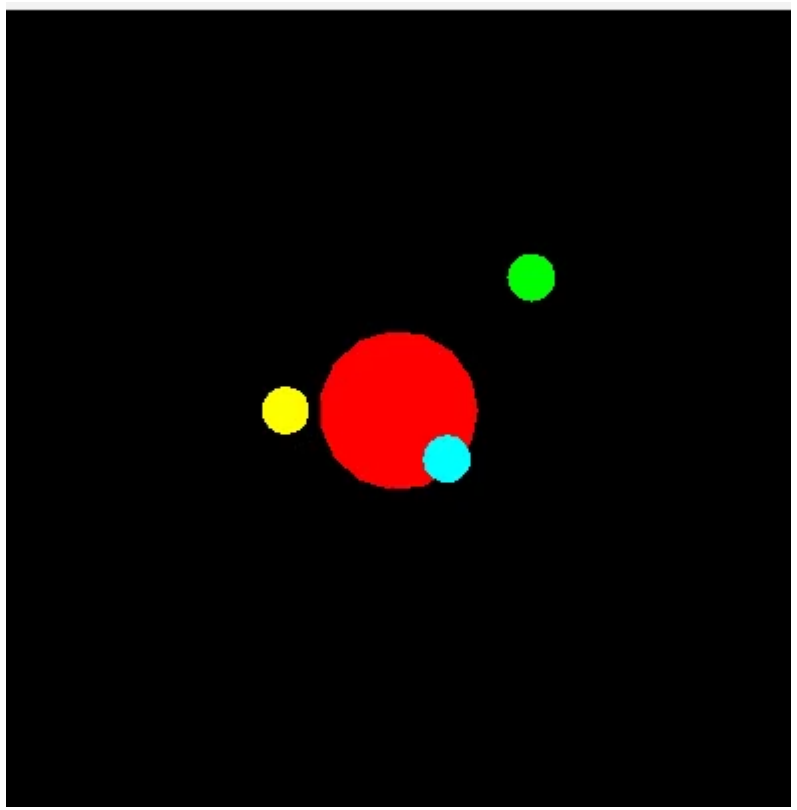
Try

- 1초마다 배경색이 R, B, G 순서대로 바뀌는 코드 짜보기
- 4_3_0 code 기반

구의 움직임 4_4_0 / 4_4

glOrtho

code



gluPerspective

code