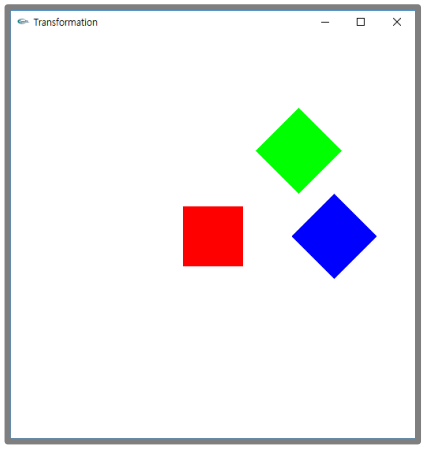


디지털 그래픽스 [6주차]

Geometric Transformation

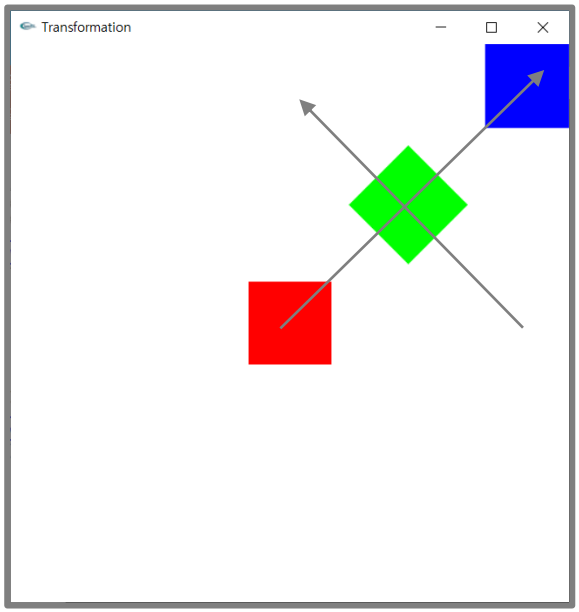
Goals

- **Matrix Stack**
- **glPushMatrix & glPopMatrix**
- **Scale Matrix**
- **Transformation by Keyboard**

Transformation의 누적 현상

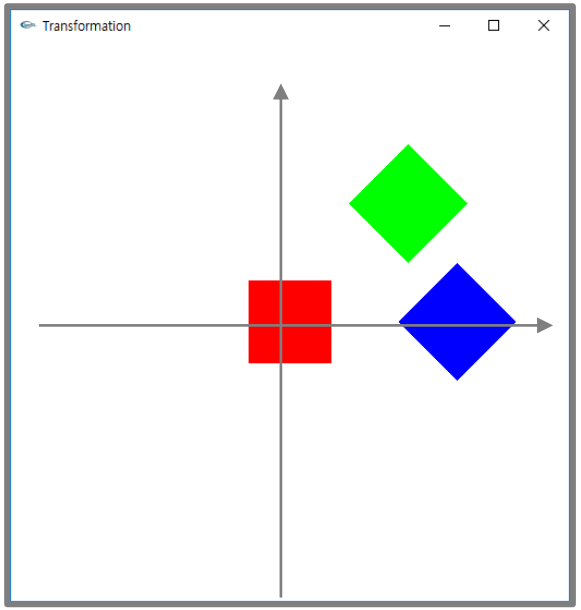
- Transformation은 누적된다.
  - Inverse Transformation 사용으로 해결

<Transformation 누적>



TR RT

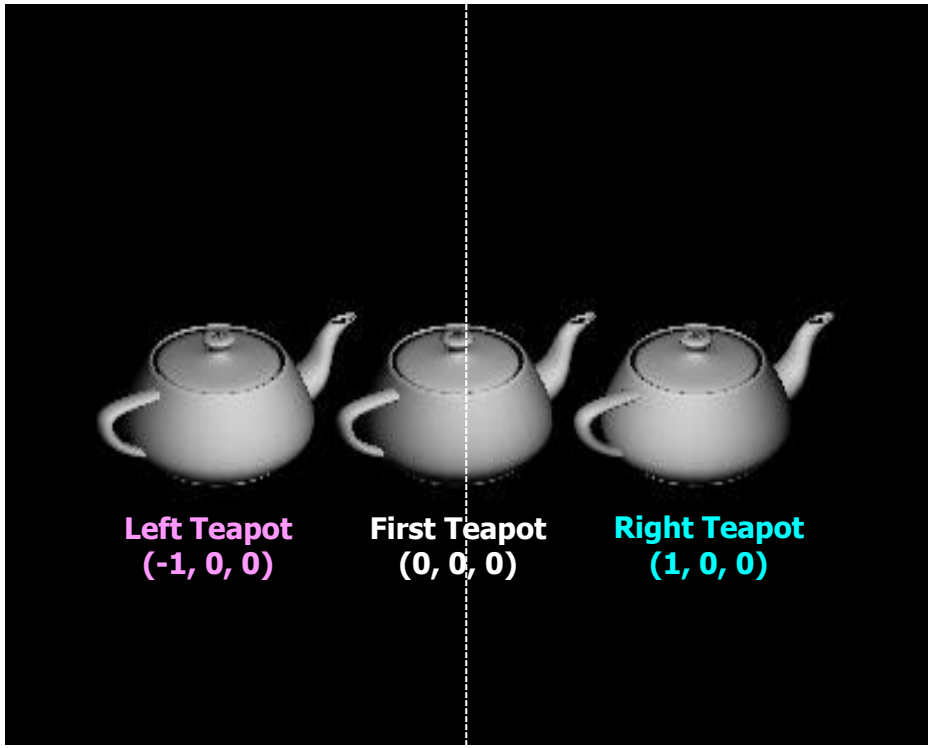
<Inverse Transformation>



TR $T^{-1}R^{-1}$ RT

Transformation의 누적 현상

- Three teapots



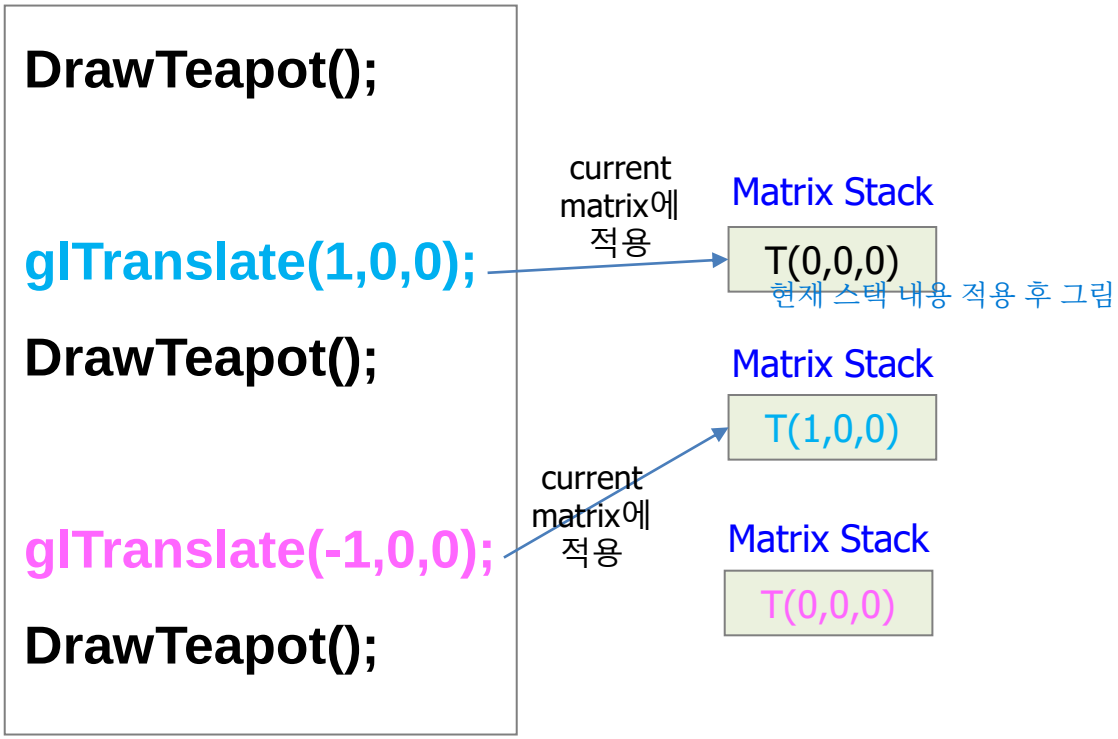

```
DrawTeapot();

glTranslate(1,0,0);
DrawTeapot();

glTranslate(-1,0,0);
DrawTeapot();
```

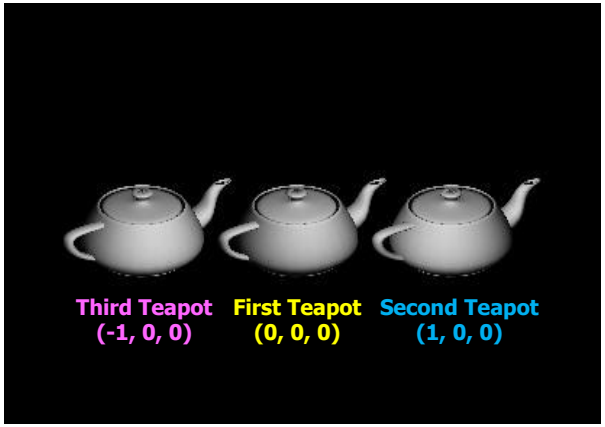
?

Matrix Multiplication 누적

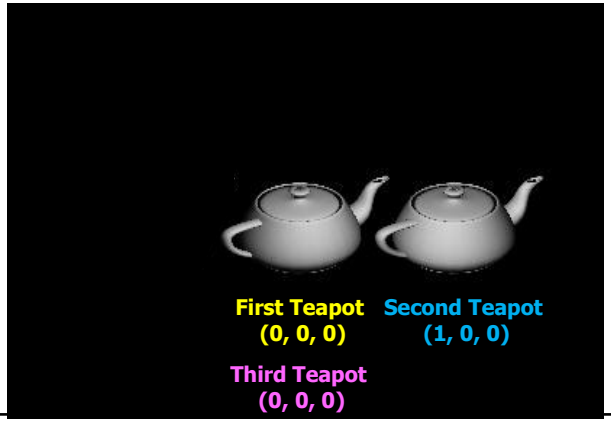
matrix stack 사용해서 누적해서 적용시킴.

- 동작 원리
  - **Matrix Stack**
  - Current Matrix: the top of matrix stack
  - 모든 matrix 연산은 current matrix에 누적





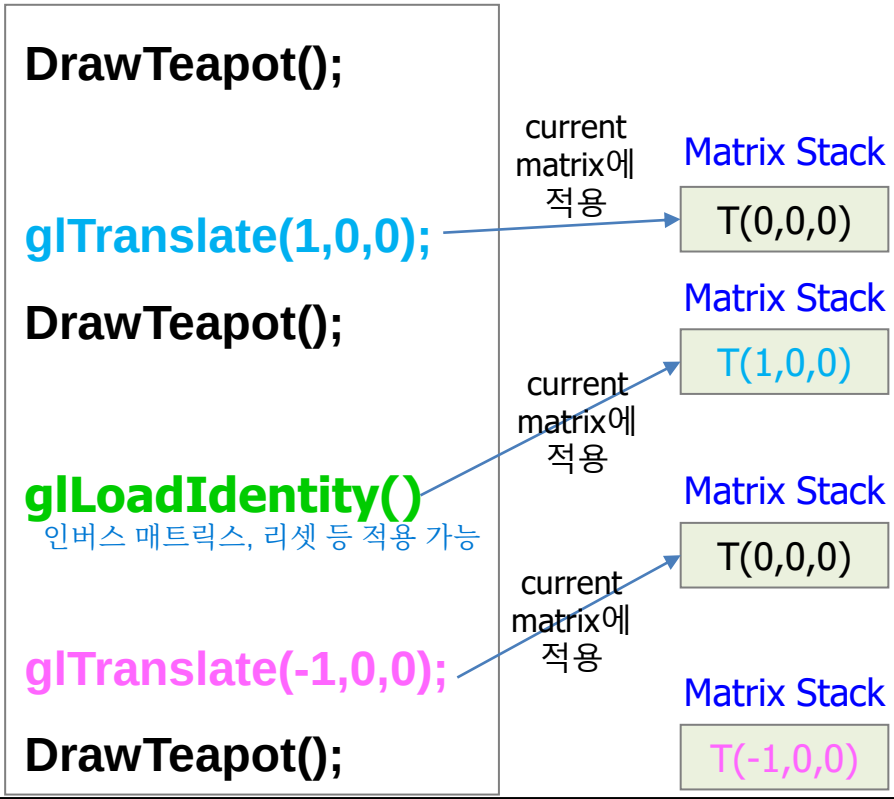
Expectation

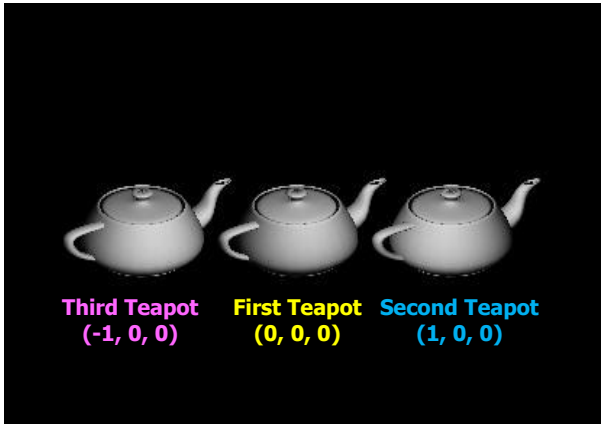


Result

glLoadIdentity()

- **OpenGL의 Transformation Matrix 초기화 필요**
  - **glLoadIdentity()** 함수는 Transformation Matrix 초기화 수행
  - **replace** the current matrix with the identity matrix

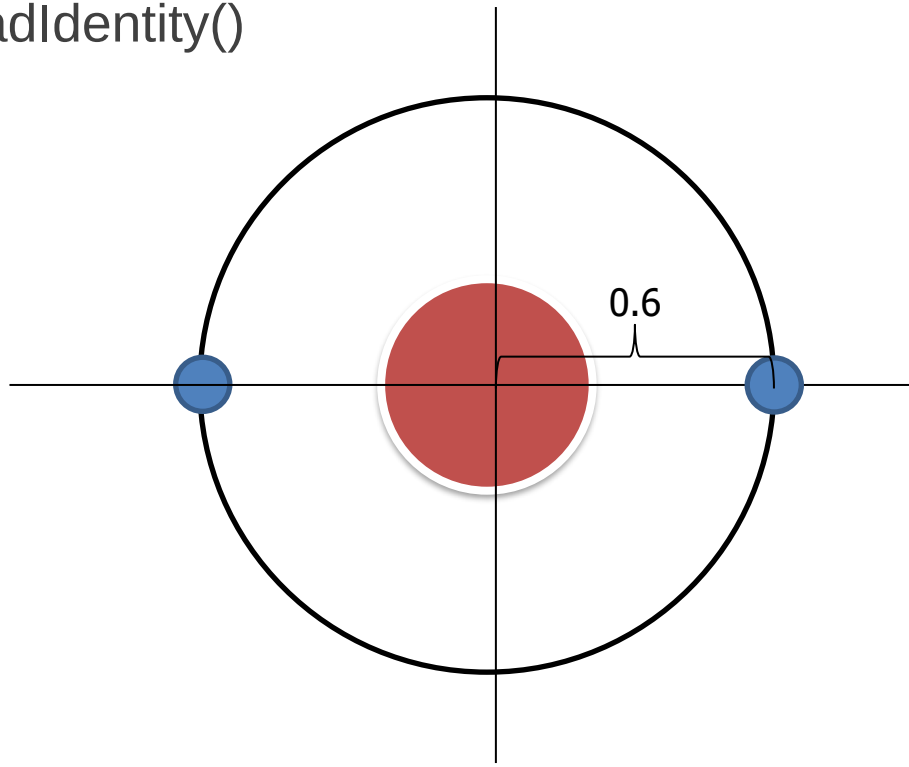




Solar system

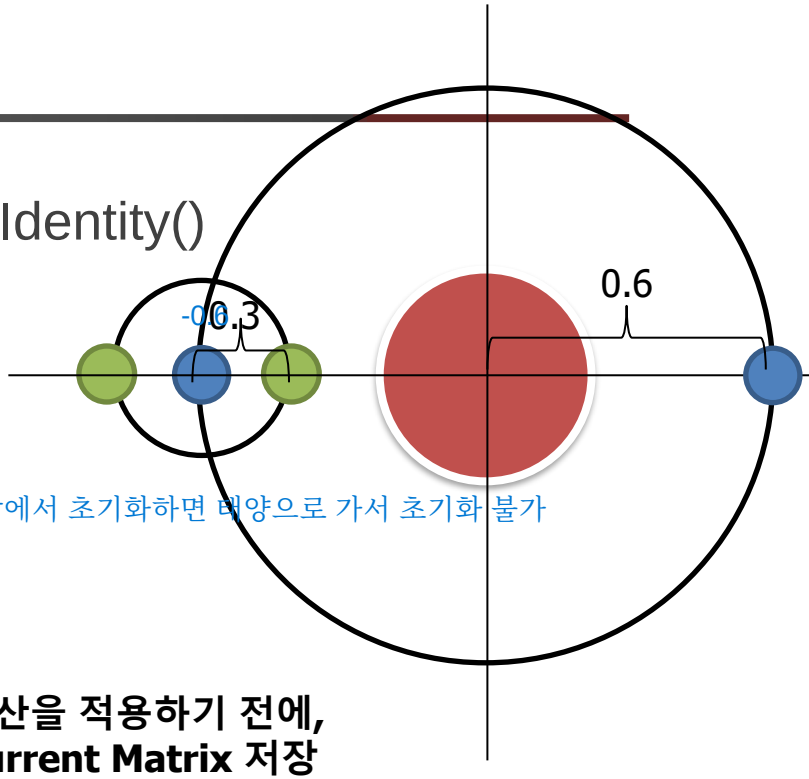
- We can draw **blue planets** with loadIdentity()

```
DrawCircle();

glTranslate(0.6,0,0);
DrawCircle();

glLoadIdentity()   매트릭스 스택 0으로 초기화

glTranslate(-0.6,0,0);
DrawCircle();
```



Solar system

- We can't draw **green planets** with loadIdentity()




```
DrawCircle();
glTranslate(0.6,0,0);
DrawCircle();
glLoadIdentity()
glTranslatef(-0.6,0,0);
DrawCircle();
glTranslate(-0.3,0,0);
DrawCircle();

How to do??
```

current matrix에 적용

Matrix Stack

T(0,0,0)

T(0.6,0,0)

T(0,0,0)

T(-0.6,0,0)

왼쪽지구 저장해둔 상태에서 아래 -0.3 적용시킴

다음 연산을 적용하기 전에, 현재 current Matrix 저장

T(-0.9,0,0)

왼쪽 green planet을 그린다

왼쪽지구 T(-0.6,0,0)

저장한 matrix를 Current matrix로 불러와서

T(0.3,0,0)

오른쪽 green planet을 그린다

Solar system

- We can't draw green planets with loadIdentity()

```
DrawCircle();
glTranslate(0.6,0,0);
DrawCircle();
glLoadIdentity()
glTranslatef(-0.6,0,0);
DrawCircle();


glTranslate(-0.3,0,0);
DrawCircle();


How to do??
```

current matrix에 적용

Matrix Stack

T(0,0,0)

T(0.6,0,0)

T(0,0,0)

T(-0.6,0,0)

다음 연산을 적용하기 전에, 현재 **current Matrix** 저장

T(-0.9,0,0) delete

T(-0.6,0,0)

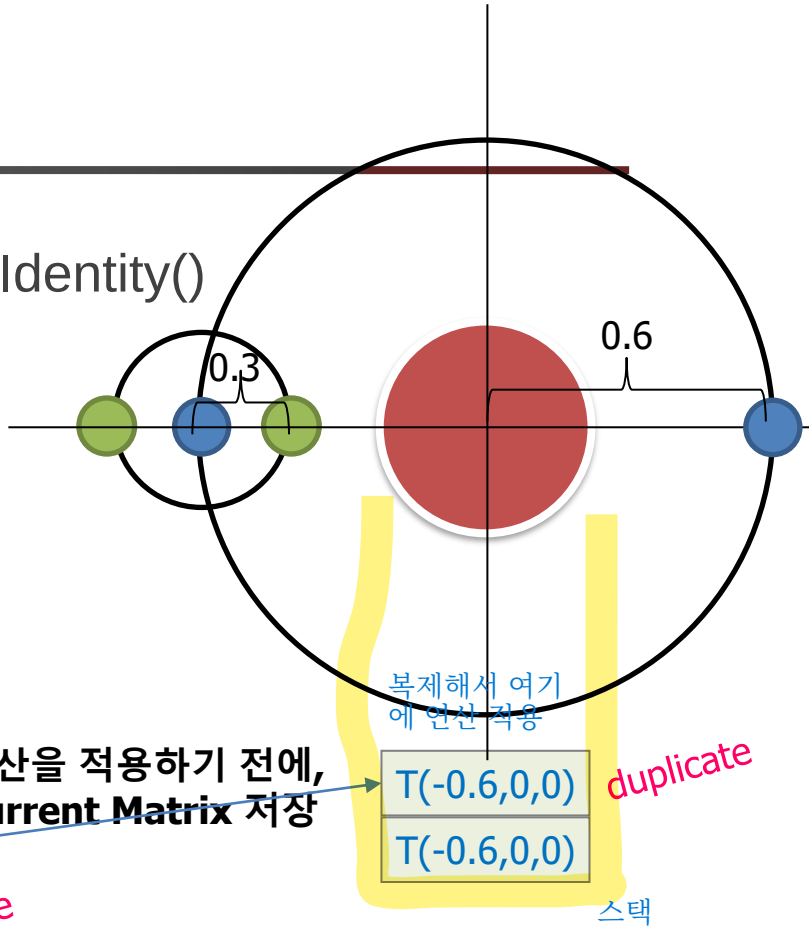
왼쪽 **green planet**을 그린다

T(-0.6,0,0)

저장한 **matrix**를 **Current matrix**로 불러와서

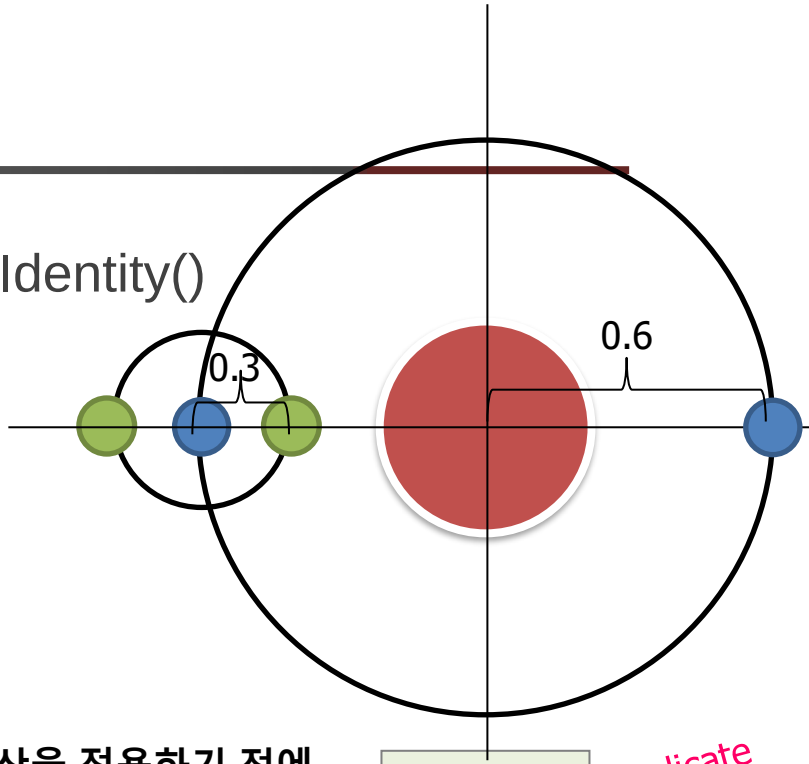
T(0.3,0,0)

오른쪽 **green planet**을 그린다



Solar system

- We can't draw green planets with loadIdentity()




```
DrawCircle();  sun
glTranslate(0.6,0,0);
DrawCircle();
glLoadIdentity()
glTranslatef(-0.6,0,0);
DrawCircle();  왼쪽 지구
Duplicate()  돌아오기 위해 복제
glTranslate(-0.3,0,0);
DrawCircle();  왼쪽딜 그리고
Delete()  복제한거 지움
glTranslate(0.3,0,0);  원래 지구에서 +0.3 해서 오른쪽 달 그림
DrawCircle();
```

current matrix에 적용

Matrix Stack

T(0,0,0)

T(0.6,0,0)

T(0,0,0)

T(-0.6,0,0)

다음 연산을 적용하기 전에, 현재 current Matrix 저장

T(-0.6,0,0) duplicate

T(-0.6,0,0)

T(-0.9,0,0) delete

T(-0.6,0,0)

왼쪽 green planet을 그린다

T(-0.6,0,0)

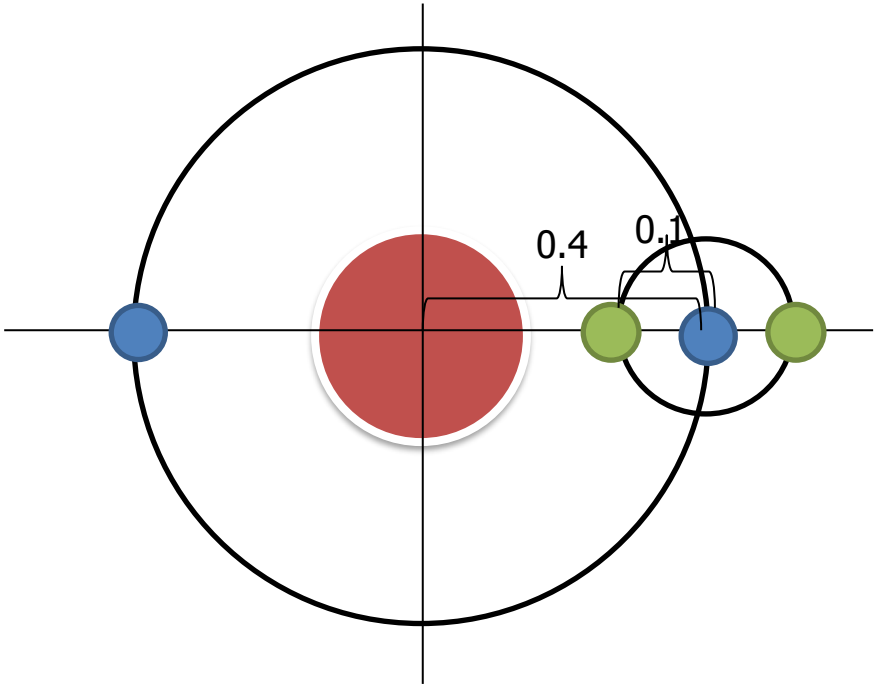
저장한 matrix를 Current matrix로 불러와서

T(0.3,0,0)

오른쪽 green planet을 그린다

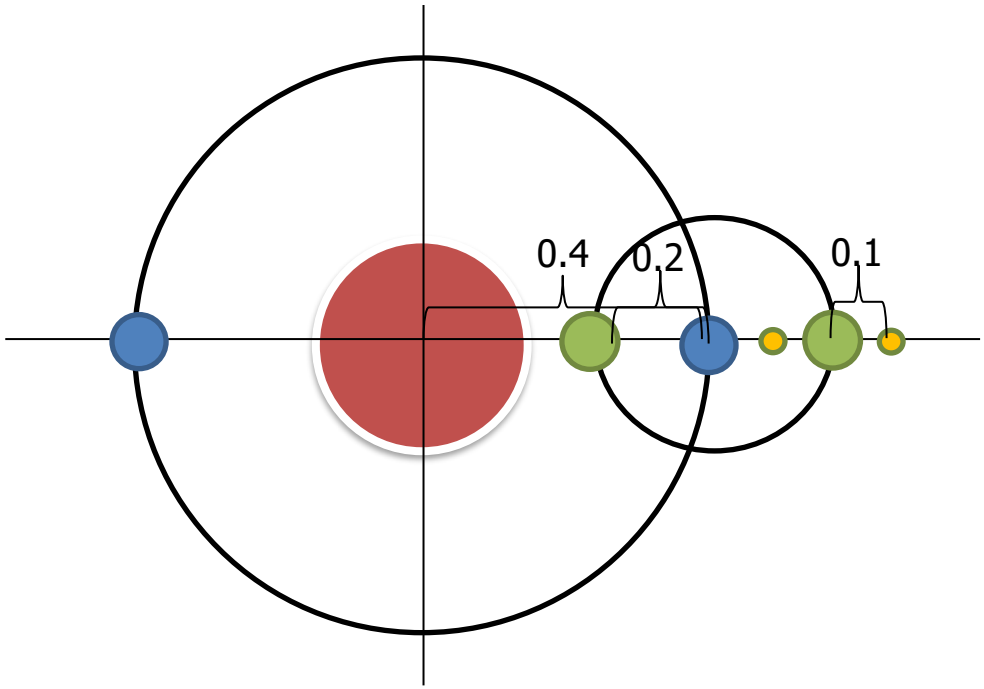
Quiz. duplicate/delete (제출x)

- Red, blue, green planets을 그리는 코드를 작성하세요.



Quiz1. duplicate/delete (제출o)

- Red, blue, green and **yellow** planets을 그리는 코드를 작성하세요.

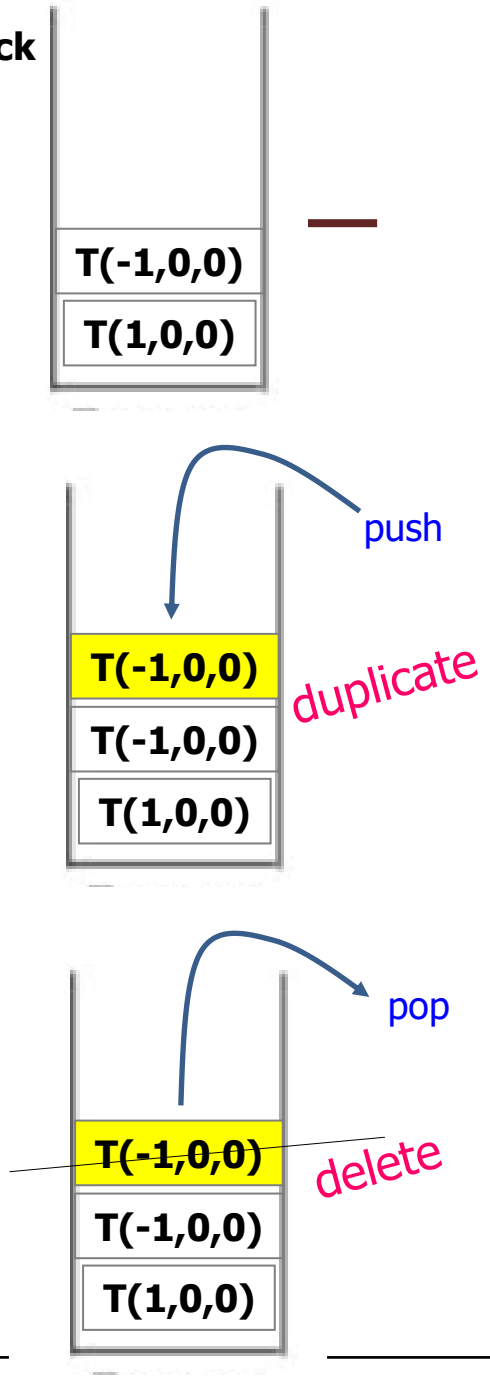


Goals

- Matrix Stack
- glPushMatrix & glPopMatrix
- Scale Matrix
- Transformation by Keyboard

glPushMatrix() / glPopMatrix()

- **glPushMatrix()** duplicate
  - pushes the current matrix stack down by one, duplicating the current matrix.
  - after a glPushMatrix call, the matrix on top of the stack is identical to the one below it.
- **glPopMatrix()** delete
  - pops the current matrix stack, replacing the current matrix with the one below it on the stack.



glPushMatrix() / glPopMatrix()

- glLoadIdentity()로 대체 가능

T(0,0,0)

DrawSquare();

T(0,0,0)

glPushMatrix(); T(0,0,0)

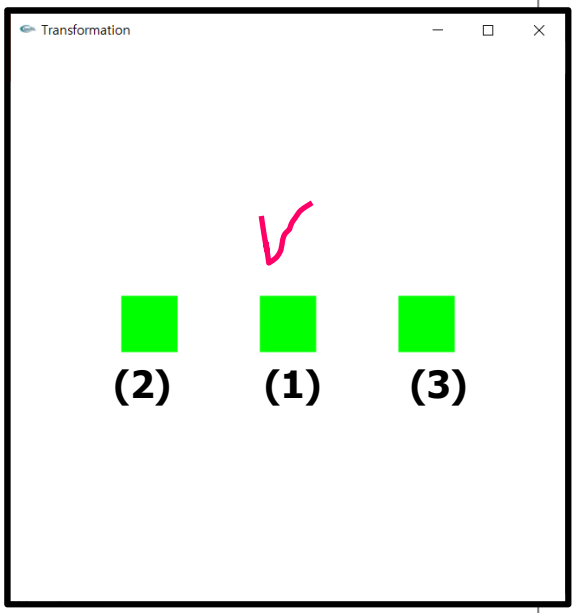
glTranslatef(-0.6f, 0.0f, 0.0f); ~~T(-0.6,0,0)~~ T(0,0,0)

DrawSquare();

glPopMatrix(); T(0,0,0)

glTranslatef(0.6f, 0.0f, 0.0f); T(0.6,0,0)

DrawSquare();



glPushMatrix() / glPopMatrix()

- **glLoadIdentity()로 해결이 안되는 경우**

glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

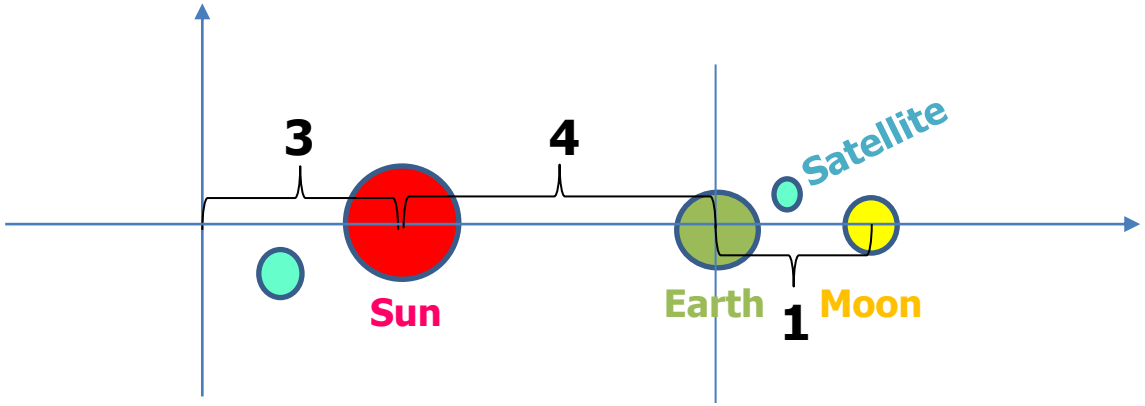
DrawSquare(); — T(0,0,0)

glTranslatef(-0.6f, 0.0f, 0.0f); — T(-0.6,0,0)
DrawSquare();

glPushMatrix(); — T(-0.6,0,0)
glTranslatef(-0.3f, 0.0f, 0.0f); — T(-0.6,0,0) → T(-0.9,0,0)
DrawSquareSmall(); — T(-0.6,0,0)
glPopMatrix(); — T(-0.6,0,0)

glTranslatef(0.3f, 0.0f, 0.0f); — T(-0.3,0,0)
DrawSquareSmall();

Coding Example. Solar System




```
glTranslatef(3.0, 0.0, 0.0);
   drawSun();
   glPushMatrix();
      glTranslatef(4.0, 0.0, 0.0);
      drawEarth();
      glPushMatrix();
         glTranslatef(1.0, 0.0, 0.0);
      drawMoon();
      glPopMatrix();
      glTranslatef(0.5, 0.5, 0.0);
      drawSatelite();
   glPopMatrix();
```

T(3,0,0)

T(3,0,0)

T(3,0,0)

T(7,0,0)

T(3,0,0)

T(7,0,0)

T(7,0,0)

T(3,0,0)

~~T(8,0,0)~~

T(7,0,0)

T(3,0,0)

T(7,0,0)

T(3,0,0)

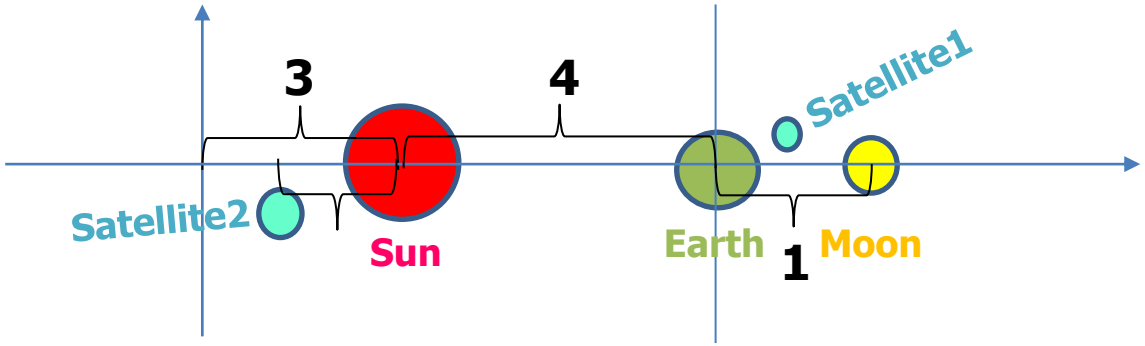
~~T(7.5,0.5,0)~~

T(3,0,0)

T(3,0,0)

Quiz2. Satellite2를 그리는 코드 추가

- **Satellite2 는 Sun 에서 왼쪽으로 0.8 아래쪽으로 0.8 떨어져 있다.**




```
glTranslatef(3.0, 0.0, 0.0);
  drawSun();
  glPushMatrix();
    glTranslatef(4.0, 0.0, 0.0);
    drawEarth();
    glPushMatrix();
      glTranslatef(1.0, 0.0, 0.0);
    drawMoon();
    glPopMatrix();
    glTranslatef(0.5, 0.5, 0.0);
    drawSatelite();
  glPopMatrix();
```

T(3,0,0)

T(3,0,0)
T(3,0,0)

T(7,0,0)
T(3,0,0)

T(7,0,0)
T(7,0,0)
T(3,0,0)

~~T(8,0,0)~~
T(7,0,0)
T(3,0,0)

T(7,0,0)
T(3,0,0)

~~T(7.5,0.5,0)~~
T(3,0,0)

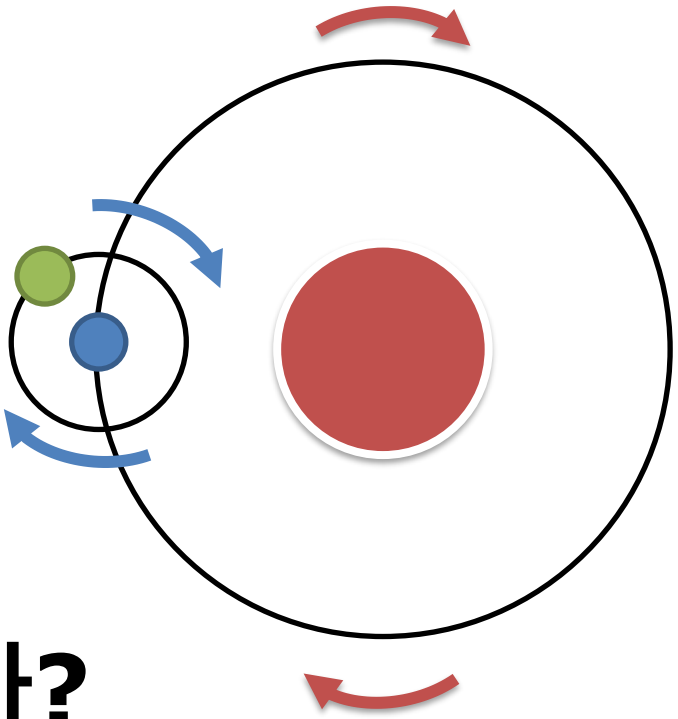
T(3,0,0)

Coding Example. Solar System

각 **Sphere**는 행성을 의미

- 각 행성은 공전과 자전을 수행
- 검정색 라인은 비행선이 움직인 궤도

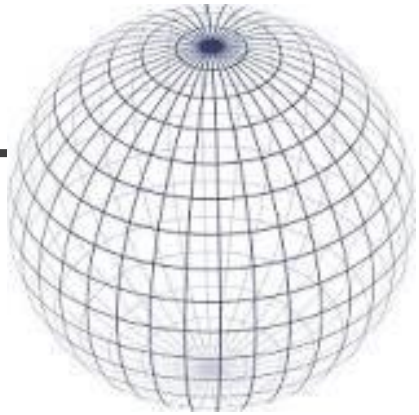
- Red Sun
- Red Sun을 회전하는 Blue Earth
- Blue Earth를 회전하는 green moon



어떻게 구현할 수 있을까?

- glPushMatrix() / glPopMatrix() 이용
- Animation 구현

sphere



Usage 특정색상으로 채워진 구 반지름, 가로세로 몇개로 슬라이스 할건지(클수록 정교)

- void glutSolidSphere(GLdouble radius, GLint slices, GLint stacks);
- void glutWireSphere(GLdouble radius, GLint slices, GLint stacks);

오른쪽 사진처럼 생김

radius

- The radius of the sphere.

slices

- The number of subdivisions around the Z axis (similar to lines of **longitude**).

stacks

- The number of subdivisions along the Z axis (similar to lines of **latitude**).

Description

- Renders a sphere centered at the modeling coordinates origin of the specified radius. The sphere is subdivided around the Z axis into slices and along the Z axis into stacks.

Solar System

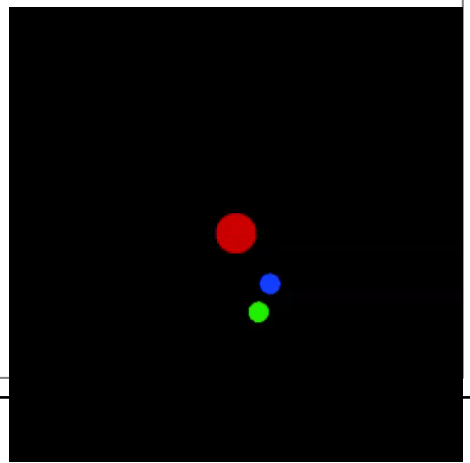
- Red Sun
- Red Sun을 회전하는 blue Earth
- Blue Earth를 회전하는 green moon

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

GLfloat blueAngle = 0.0f;

int main(int argc, char** argv)
{
   glutInit(&argc, argv);

   glutInitDisplayMode(GLUT_RGBA |
            GLUT_DEPTH | GLUT_SINGLE);
   glutCreateWindow("Transformation");

   glutDisplayFunc(display);
   glutIdleFunc(display);

   glutMainLoop();

   return 0;
}
```

이벤트: 화면 크기 조정 등

디스플레이 이벤트가 발생할때 내가 만든 display함수가 호출됨

idle:cpu가 한가하면 발생.



```
void display(void)
{
   glClear(GL_COLOR_BUFFER_BIT |
   GL_DEPTH_BUFFER_BIT);
   glLoadIdentity();

   glColor3f(1.0f, 0.0f, 0.0f);     // red
   glutSolidSphere(0.1f, 100, 100);

   glPushMatrix();
   glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
   glTranslatef(0.6f, 0.0f, 0.0f);
   glColor3f(0.0f, 0.0f, 1.0f);     // blue
   glutSolidSphere(0.05f, 100, 100);

   glPushMatrix();
   glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
   glTranslatef(0.3f, 0.0f, 0.0f);
   glColor3f(0.0f, 1.0f, 0.0f);     // green
   glutSolidSphere(0.05f, 100, 100);
   glPopMatrix();          달 그림
   glPopMatrix();

   blueAngle += 1.0f;
   glFlush();
}
```

Solar System

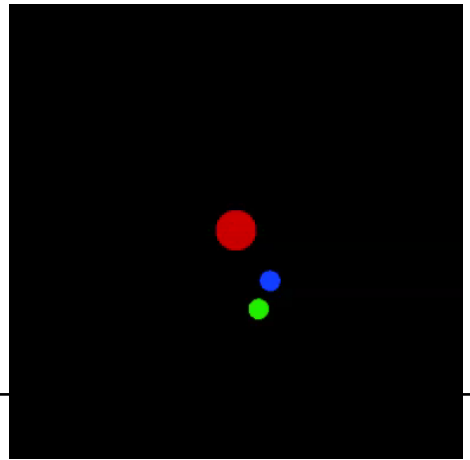
- 이 예제에서 glPushMatrix(), glPopMatrix() 함수를 반드시 사용해야 할까요?

NO!

loadidentity, push, pop 모두 없애면 검정 윈도우만 발생
>> 이동이 축적되니까 초기화를 하지 않으면 화면에서 벗어나서

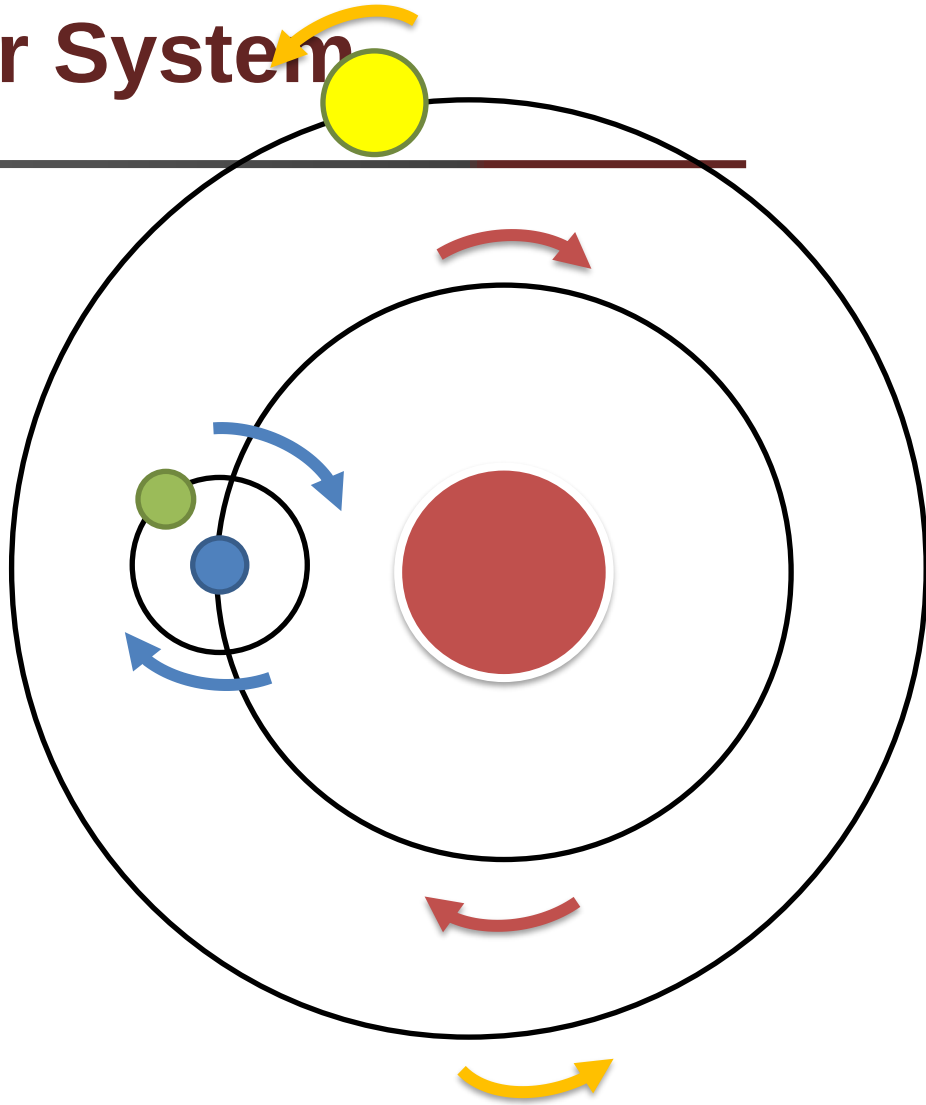
load만 두고 push pop 없애도 나옴
>> push, pop은 없어도 기능을 함. 아무때나 푸시팝을 하는 건 아님.

load만 없애고 push pop 해도 나옴
>> 초기화 안해도 transformation은 해서 화면에 행성 뜸.



```
void display(void)
{
   glClear(GL_COLOR_BUFFER_BIT |
   GL_DEPTH_BUFFER_BIT);
   glLoadIdentity();

   glColor3f(1.0f, 0.0f, 0.0f);     // red
   glutSolidSphere(0.1f, 100, 100);

   glPushMatrix();
   glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
   glTranslatef(0.6f, 0.0f, 0.0f);
   glColor3f(0.0f, 0.0f, 1.0f);     // blue
   glutSolidSphere(0.05f, 100, 100);

   glPushMatrix();
   glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
   glTranslatef(0.3f, 0.0f, 0.0f);
   glColor3f(0.0f, 1.0f, 0.0f);     // green
   glutSolidSphere(0.05f, 100, 100);
   glPopMatrix();
   glPopMatrix();

   blueAngle += 1.0f;
   glFlush();
}
```

Coding Example. Solar System



각 Sphere는 행성을 의미

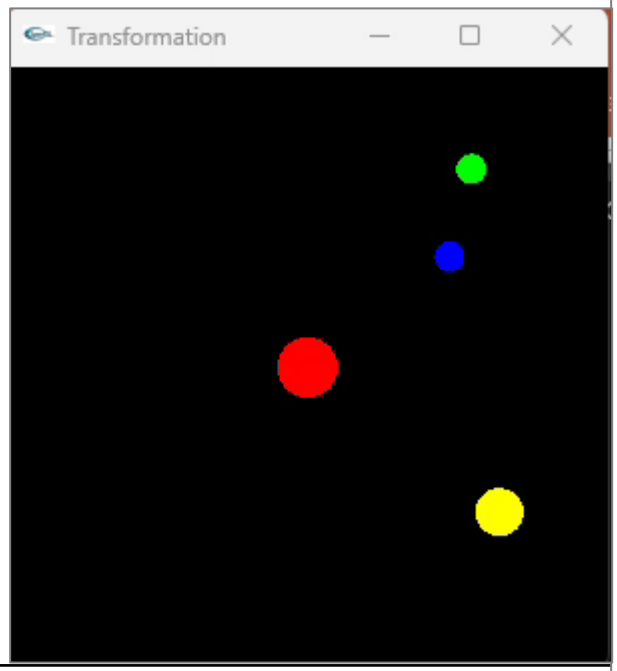
- **각 행성은 공전과 자전을 수행**
- **검정색 라인은 비행선이 움직인 궤도**

- Red Sun
- Red Sun을 회전하는 Blue Earth
- Blue Earth를 회전하는 green moon
- Red Sun을 반시계 방향으로 회전하는 Yellow Satellite

- 이 예제에서 glPushMatrix(), glPopMatrix() 함수를 반드시 사용해야 할까요?

YES!

은하계를 여행하는 비행선



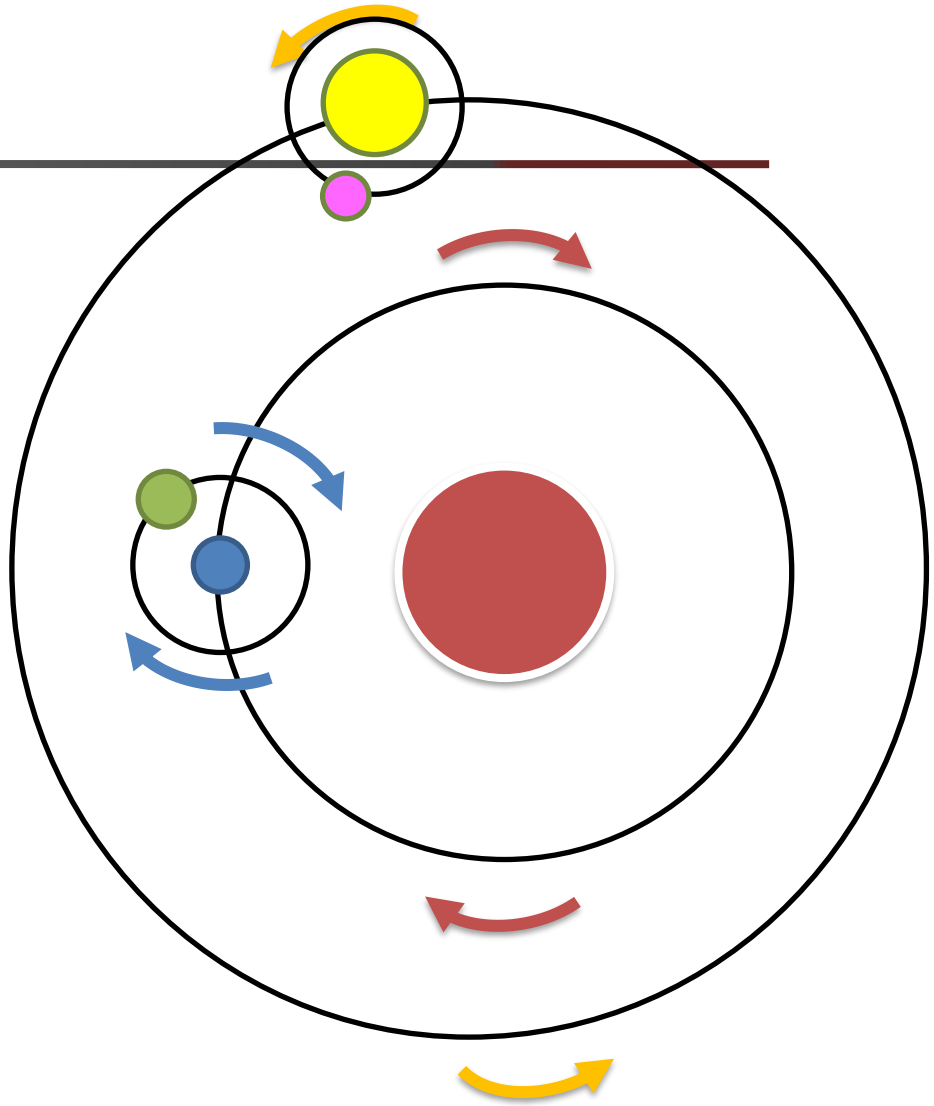

```
void display(void)
{
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();

        glColor3f(1.0f, 0.0f, 0.0f);            // red
        glutSolidSphere(0.1f, 100, 100);

        glPushMatrix();
        glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
        glTranslatef(0.6f, 0.0f, 0.0f);
        glColor3f(0.0f, 0.0f, 1.0f);            // blue
        glutSolidSphere(0.05f, 100, 100);

        glPushMatrix();
        glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
        glTranslatef(0.3f, 0.0f, 0.0f);
        glColor3f(0.0f, 1.0f, 0.0f);            // green
        glutSolidSphere(0.05f, 100, 100);

        glPopMatrix();
        glPopMatrix();
        glRotatef(1-blueAngle, 0.0f, 0.0f, 1.0f);
        glTranslatef(0.8f, 0.0f, 0.0f);
        glColor3f(1.0f, 1.0f, 0.0f);            // yellow
        glutSolidSphere(0.08f, 100, 100);

        blueAngle += 1.0f;
        if (blueAngle == 360) blueAngle = 0;
        glFlush();
}
```

Quiz2. Solar System

이전 슬라이드에 있는 코드에 pink satellite의 회전을 추가하세요.

각 Sphere는 행성을 의미

- **각 행성은 공전과 자전을 수행**
- **검정색 라인은 비행선이 움직인 궤도**



- Red Sun
- Red Sun을 회전하는 Blue Earth
- Blue Earth를 회전하는 green moon
- Red Sun을 반시계 방향으로 회전하는 Yellow Satellite
- Yellow satellite를 회전하는 pink satellite

* 방향을 표기하지 않으면, default 시계 방향 회전

Quiz3. Solar System

이전 슬라이드에 있는 코드에 pink satellite를 추가하세요.

회전 없이
태양, 지구, 달, 위성을 고정 위치에 그리세요

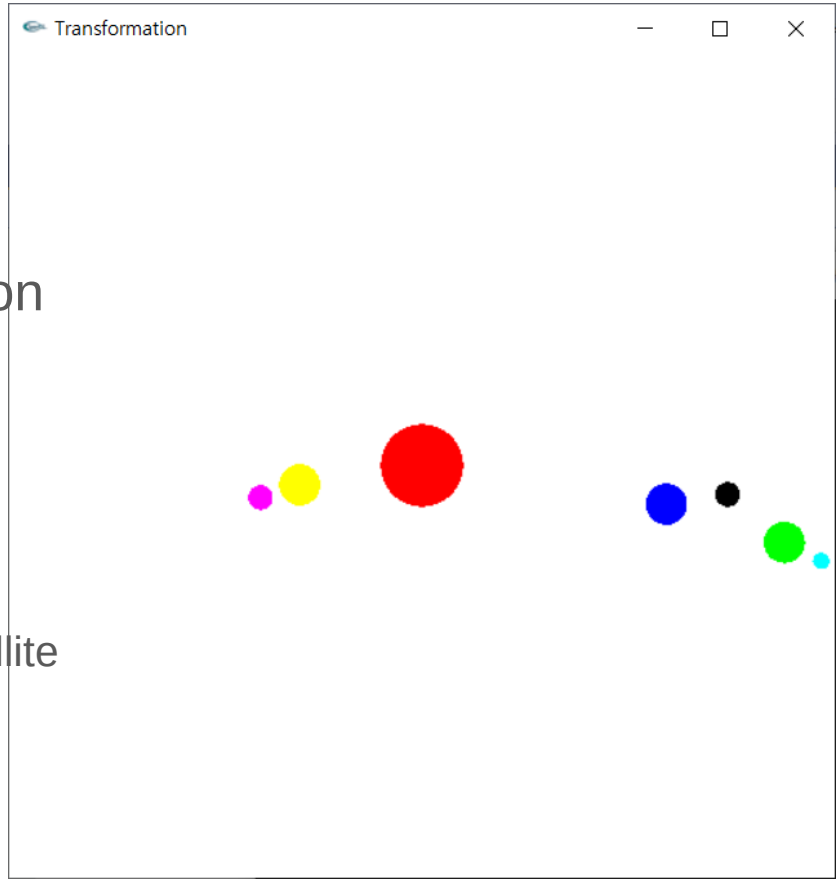
```
void display(void)
{
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();

        glColor3f(1.0f, 0.0f, 0.0f);    // red
        glutSolidSphere(0.1f, 100, 100);

        glPushMatrix();
        glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
        glTranslatef(0.6f, 0.0f, 0.0f);
        glColor3f(0.0f, 0.0f, 1.0f);    // blue
        glutSolidSphere(0.05f, 100, 100);

        glPushMatrix();
        glRotatef(blueAngle, 0.0f, 0.0f, 1.0f);
        glTranslatef(0.3f, 0.0f, 0.0f);
        glColor3f(0.0f, 1.0f, 0.0f);    // green
        glutSolidSphere(0.05f, 100, 100);

        glPopMatrix();
        glPopMatrix();
        glRotatef(1-blueAngle, 0.0f, 0.0f, 1.0f);
        glTranslatef(0.8f, 0.0f, 0.0f);
        glColor3f(1.0f, 1.0f, 0.0f);    // yellow
        glutSolidSphere(0.08f, 100, 100);

        blueAngle += 1.0f;
        if (blueAngle == 360) blueAngle = 0;
        glFlush();
}
```

과제 1. Matrix Stack

- **아래 애니메이션을 구현하세요.** (포탈 <강의자료>에 동영상 파일 첨부)

- 기존 코드
  - Red Sun
  - Red Sun을 회전하는 Blue Earth
  - Blue Earth를 회전하는 green moon

- 신규 코드 작성
  - Red Sun을 회전하는 Yellow satellite
  - Yellow satellite를 회전하는 pink satellite
  - Blue Earth를 회전하는 또 다른 black satellite
  - Green Moon을 회전하는 sky satellite



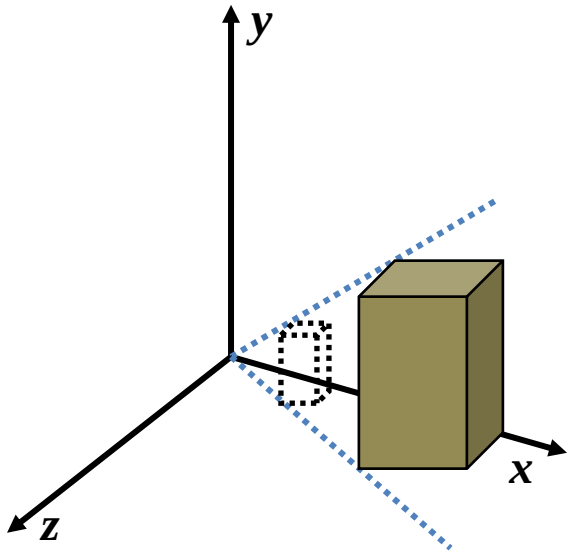
Goals

- Matrix Stack
- glPushMatrix & glPopMatrix
- Scale Matrix
- Transformation by Keyboard

3D Scale Matrix

- **glScalef(x, y, z)**
- **Not just for a size**
  - Also for a position, which is a distance from the origin

$$x' = x \cdot s_x, \quad y' = y \cdot s_y, \quad z' = z \cdot s_z$$



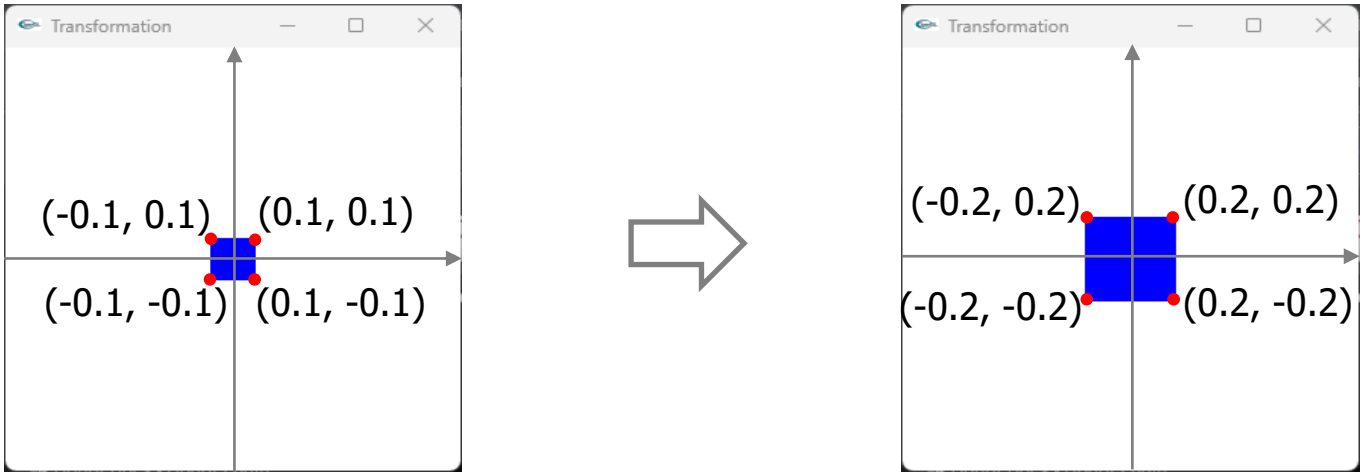
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$x' = S_x*x + 0*y + 0*z + 0*1$
$x' = S_x*x$

Scaling

물체의 중심이 원점에 있을 때 scaling

- glScalef(2.0, 2.0, 0.0)
- glutSolidCube(0.2)

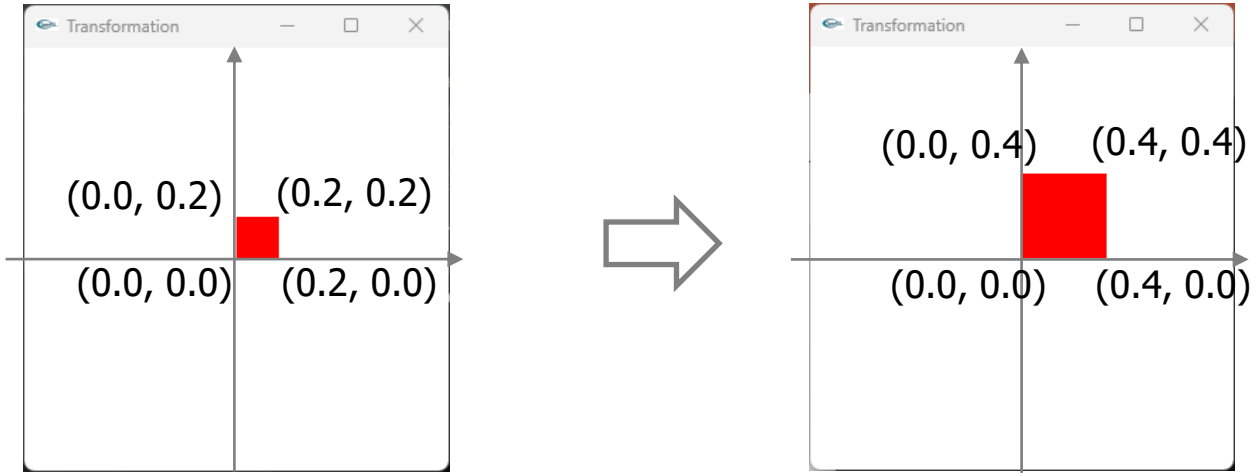


Scaling

물체의 vertex가 원점에 있을 때, scaling

glScalef(2.0, 2.0, 0.0)

- glTranslatef(0.1, 0.1, 0.0)
- glutSolidCube(0.2)

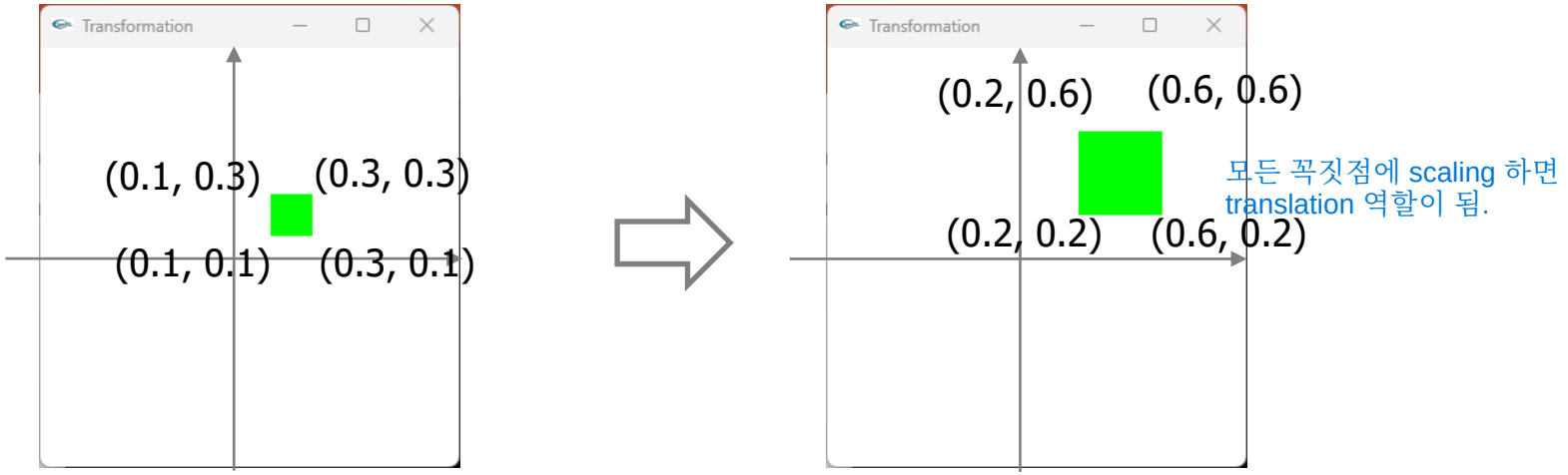


Scaling

물체가 원점에 있지 않을 때, scaling

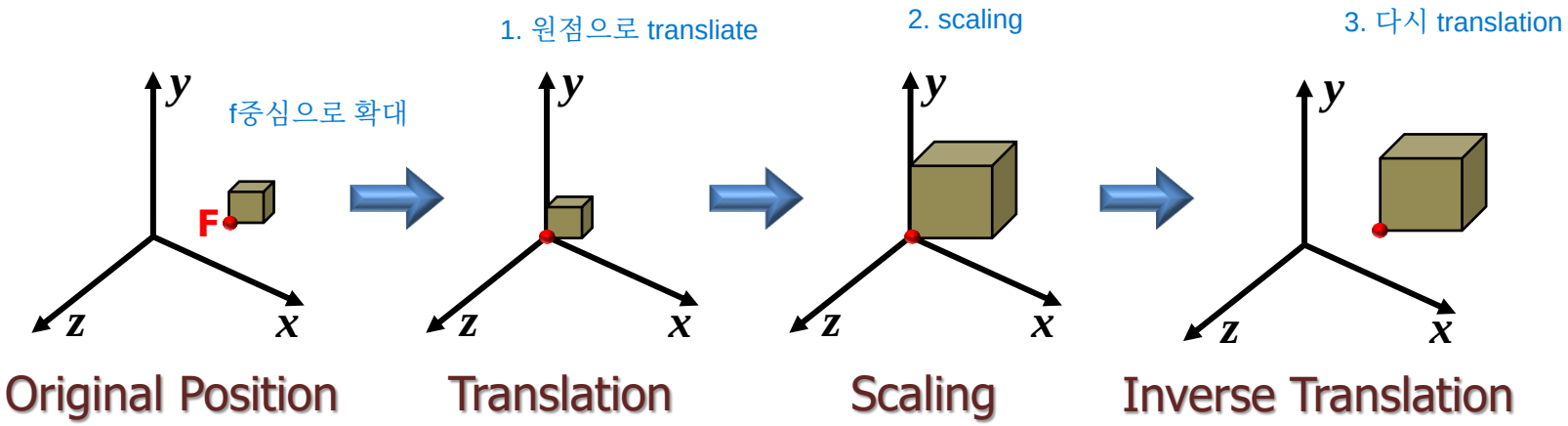
glScalef(2.0, 2.0, 0.0)

- glTranslatef(0.2, 0.2, 0.0)
- glutSolidCube(0.2)



Fixed-point Scaling

중간과제에 포함된 기능이니 꼭 활용하자.



$$T(x_f, y_f, z_f)\,S(s_x, s_y, s_z)\,T(-x_f, -y_f, -z_f) = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_f \\ 0 & 1 & 0 & -y_f \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

translate　　scaling　　minus translate

Goals

- Matrix Stack
- glPushMatrix & glPopMatrix
- Scale Matrix
- Transformation by Keyboard

키보드 제어

누를때마다 특정 기능하도록 !

event driven 함수임.

- **glutKeyboardFunc(..)**
  - sets the keyboard callback for the current window.
- **Usage**

<키보드 콜백 함수>를 등록하는 함수

  - **void glutKeyboardFunc(void (*func)(unsigned char key,**
    int x,
    int y));

키보드 콜백함수명

<키보드 콜백 함수>의 매개변수 3개

  - func : The new keyboard callback function.
- **Coding example**

콜백함수 제작 필요

자료형은 꼭 void

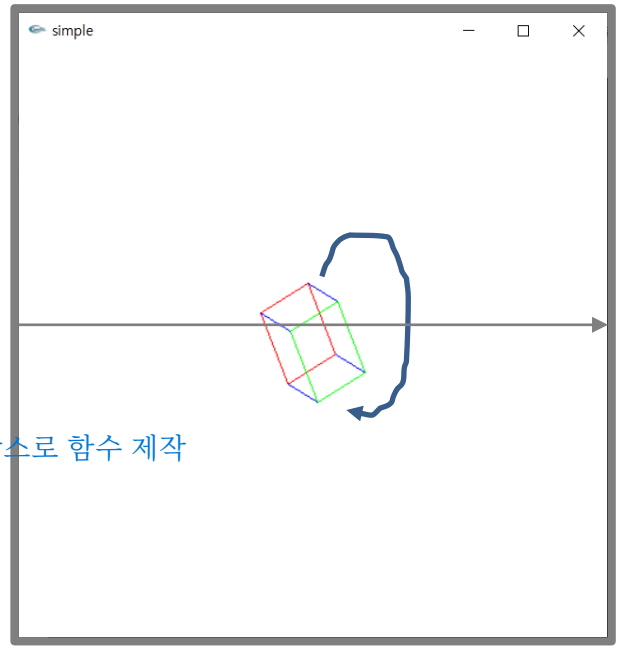
```
void keyboard(unsigned char key, int x, int y)
{

}
```

키보드 이벤트 발생 시 keyboard 함수 호출

```
glutKeyboardFunc( keyboard );
```

Transformation by Keyboard

- Coding example
  - drawBox()
    - cube 그리기
  - display()
    - 기본 변환
    - drawBox() 선 그리듯이 박스로 함수 제작
  - xyzRotate()
    - X축 기준 회전
    - glRotatef()
  - keyboard()
    - 키보드 처리 누르면 rotate 함수 호출. 한번 더 함수로 만들어서 호출해야함
    - glutPostRedisplay()
  - main()
    - display 콜백함수 등록
    - keyboard 콜백함수 등록



키보드 x 누르면 x축 기준 회전

키보드 y 누르면 y축 기준 회전

키보드 z 누르면 z축 기준 회전

키보드 s 누르면 확대

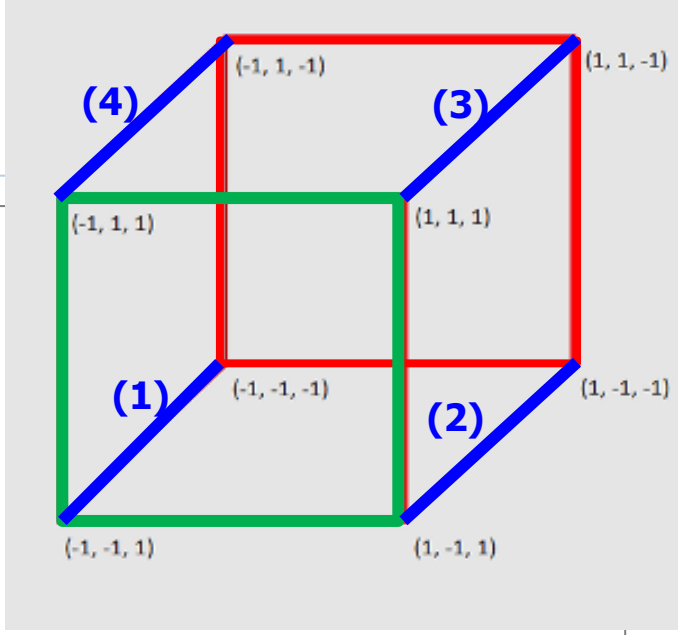
키보드 a 누르면 축소

키보드 t 누르면 오른쪽으로 이동

키보드 r 누르면 왼쪽으로 이동

구현해야할 코드

Transformation by Keyboard




```
#include <iostream>
#include <gl/glut.h>
#include <math.h> //sin(),cos()
using namespace std;

#define PI 3.141592

void drawBox()
{
        // Red rectangle
        glColor3f(1.0, 0.0, 0.0);
        glBegin(GL_LINE_LOOP);
        glVertex3f(-1.0, -1.0, -1.0);
        glVertex3f( 1.0, -1.0, -1.0);
        glVertex3f( 1.0,  1.0, -1.0);
        glVertex3f(-1.0,  1.0, -1.0);
        glEnd();

        // Green rectangle
        glColor3f(0.0, 1.0, 0.0);
        glBegin(GL_LINE_LOOP);
        glVertex3f(-1.0, -1.0, 1.0);
        glVertex3f( 1.0, -1.0, 1.0);
        glVertex3f( 1.0,  1.0, 1.0);
        glVertex3f(-1.0,  1.0, 1.0);
        glEnd();

        // Blue lines
        glColor3f(0.0, 0.0, 1.0);
        glBegin(GL_LINES);
        glVertex3f(-1.0, -1.0, -1.0);
        glVertex3f(-1.0, -1.0,  1.0);
        glVertex3f( 1.0, -1.0, -1.0);
        glVertex3f( 1.0, -1.0,  1.0);
        glVertex3f( 1.0,  1.0, -1.0);
        glVertex3f( 1.0,  1.0,  1.0);
        glVertex3f(-1.0,  1.0, -1.0);
        glVertex3f(-1.0,  1.0,  1.0);
        glEnd();
}

void display(void)
{
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        //glLoadIdentity();
        glPushMatrix();                 // initial setup: no accumulation
        glRotatef(45.0, 1.0, 1.0, 1.0); // rotation w.r.t v=(1,1,1)
        glScalef(0.1, 0.2, 0.1);
        drawBox();
        glPopMatrix();
        glFlush();
}
```

push, pop은 오로지 저 박스만을 위해서 넣은 함수.
다른 기능에 영향주지 않음.

1. y쪽 크기를 키워서

위 사진 박스 그리는거 ㅇㅇ

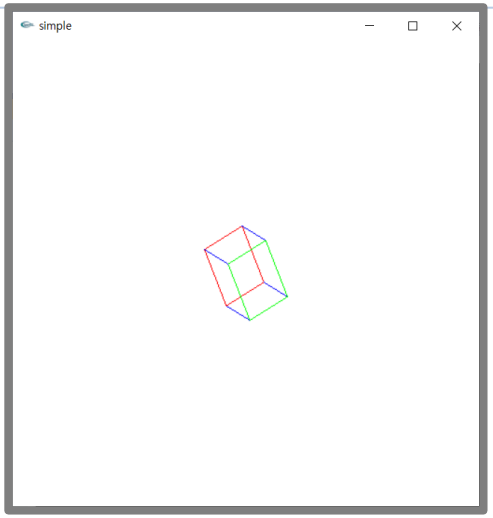
2. 45도 기울여라 / 1,1,1 축을 중심으로

1,2번 없으면 앞면밖에 보이지 않음.

Transformation by Keyboard

```
void xyzRotate(char mode)
{
        switch (mode) {
                case 'x': glRotatef(PI, 1, 0, 0); break;
                case 'y': glRotatef(PI, 0, 1, 0); break;
                case 'z': glRotatef(PI, 0, 0, 1); break;
                default: break;
        }
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'x': xyzRotate('x'); glutPostRedisplay(); break;
        case 'y': xyzRotate('y'); glutPostRedisplay(); break;
        case 'z': xyzRotate('z'); glutPostRedisplay(); break;
        default: break;
    }
}
```

x면 rotate 함수를 호출함. x를 기준으로 pi만큼 회전한다.



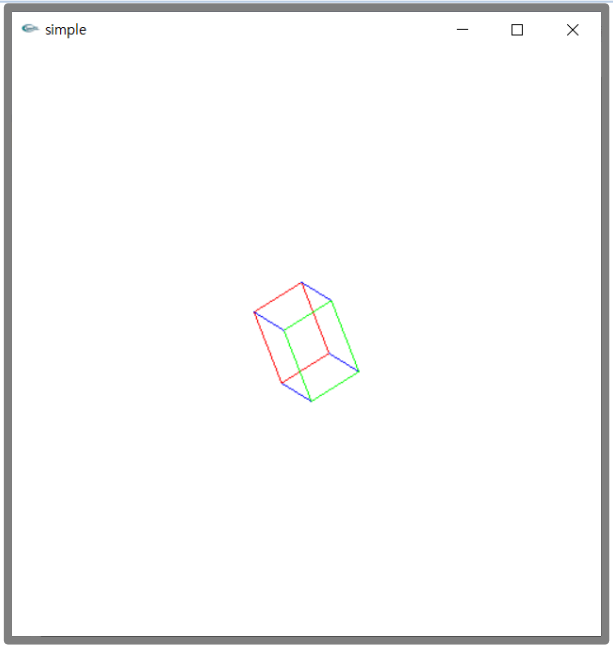
x누를때마다 다시 그려지는가?
>> ㄴㄴ display event가 있을 때만 그려짐. (처음실행)
호출은 되지만 그려지지 않음. >> glutPostRedispl

glutPostRedisplay()
- display event 발생
- display 콜백함수를 의도적으로 호출

Transformation by Keyboard

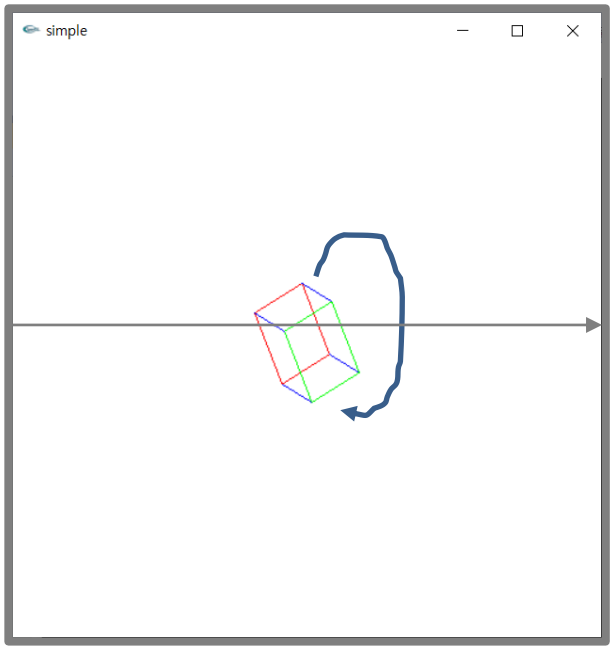
```
int main(int argc, char** argv)
{
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_RGBA | GLUT_DEPTH | GLUT_SINGLE);
	glutCreateWindow("Transformation");

	// Set Background Color
	glClearColor(1.0, 1.0, 1.0, 0.0);   배경 흰색
	glutDisplayFunc(display);   << 디스플레이, 키보드 함수(아래) 등록
	glutKeyboardFunc(keyboard);
	glutMainLoop();

	return 0;
}
```

빙빙 돌지 않아도 되니까 idlefunc 등록할 필요가 없음.
>> idle 있으면 cpu여유있을 때 display 계속 발생시킴



과제 2. Transformation by Keyboard

- 주어진 코드에 **그린색 기능**을 추가하세요.
  - drawBox()
    - cube 그리기
  - display()
    - 기본 변환
    - drawBox()
  - xyzRotate()
    - X축 기준 회전
    - glRotatef()
  - keyboard()
    - 키보드 처리
    - glutPostRedisplay()
  - main()
    - display 콜백함수 등록
    - keyboard 콜백함수 등록



키보드 x 누르면 x축 기준 회전

키보드 y 누르면 y축 기준 회전

키보드 z 누르면 z축 기준 회전

키보드 s 누르면 확대

키보드 a 누르면 축소

키보드 t 누르면 오른쪽으로 이동

키보드 r 누르면 왼쪽으로 이동

Thank you !