CGL
COMPUTER GRAPHICS LABORATORY

컴퓨터 그래픽스
[05]

2023학년도 1학기

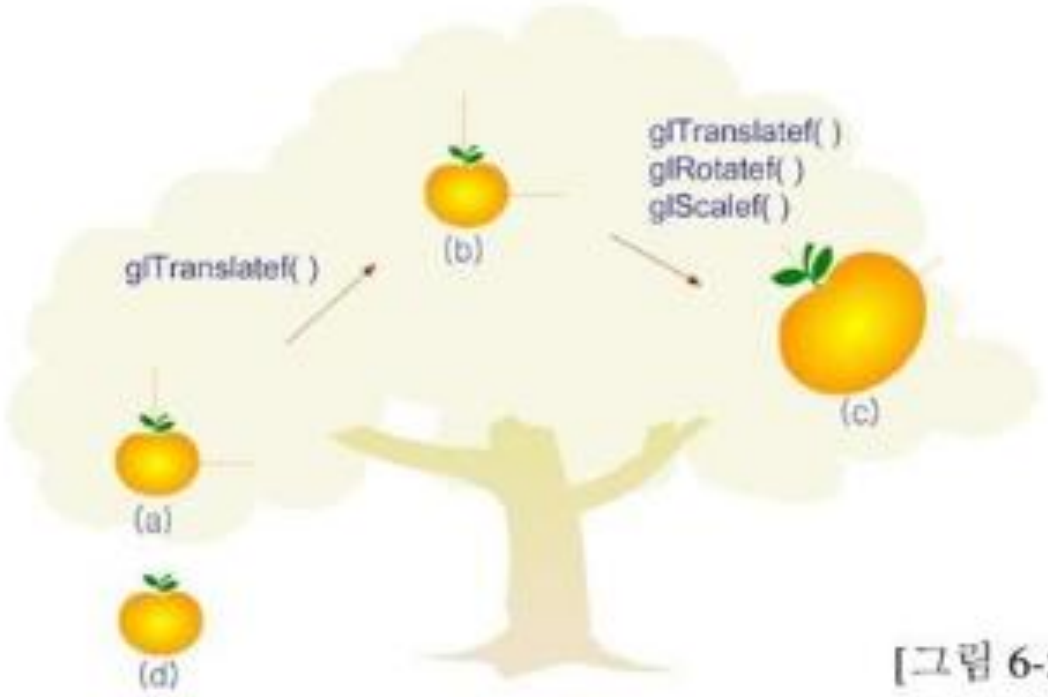
담당교수 : 마준

행렬 저장

- 복합 변환이 많거나 나누어 적용하고 싶은 경우 변환행렬을 저장해야함

- OpenGL은 각 타입의 행렬의 저장을 위해서 행렬 스택을 제공
  - GL_PROJECTION
  - GL_MODELVIEW

- glPushMatrix()
- glPopMatrix()

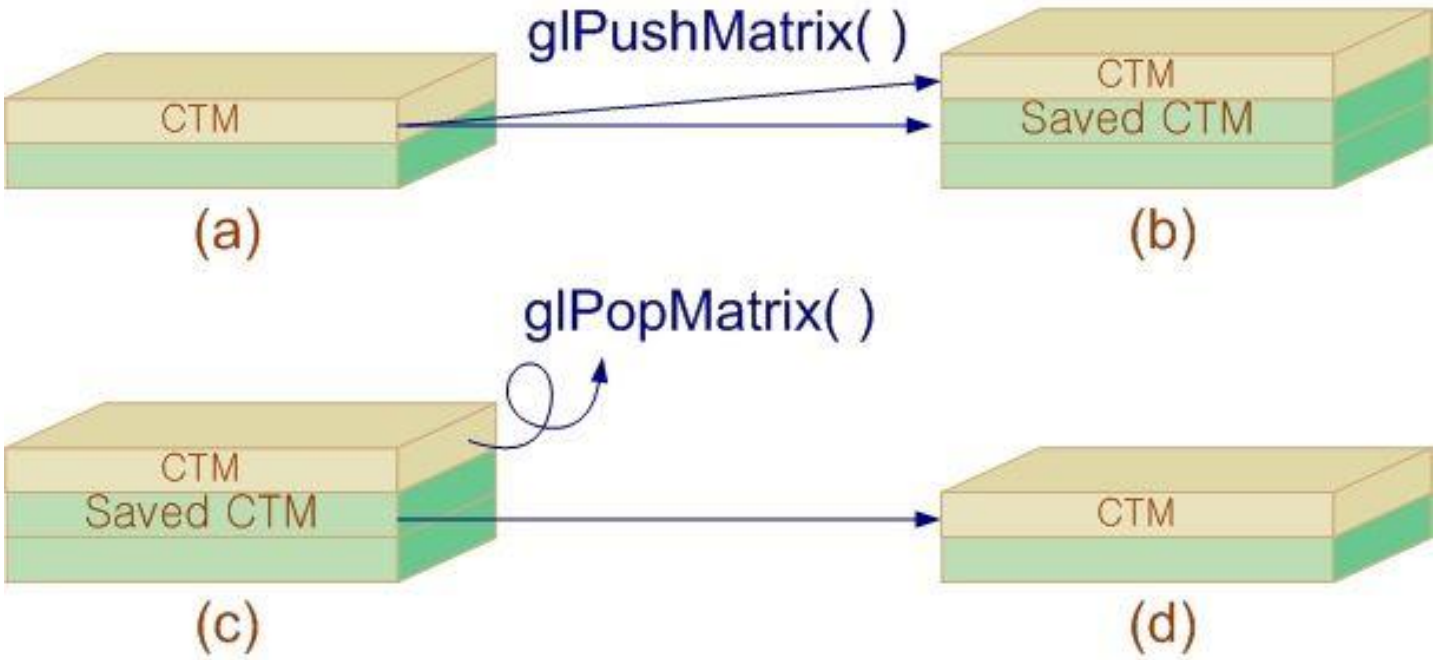


[그림 6-57] 오렌지 매달기

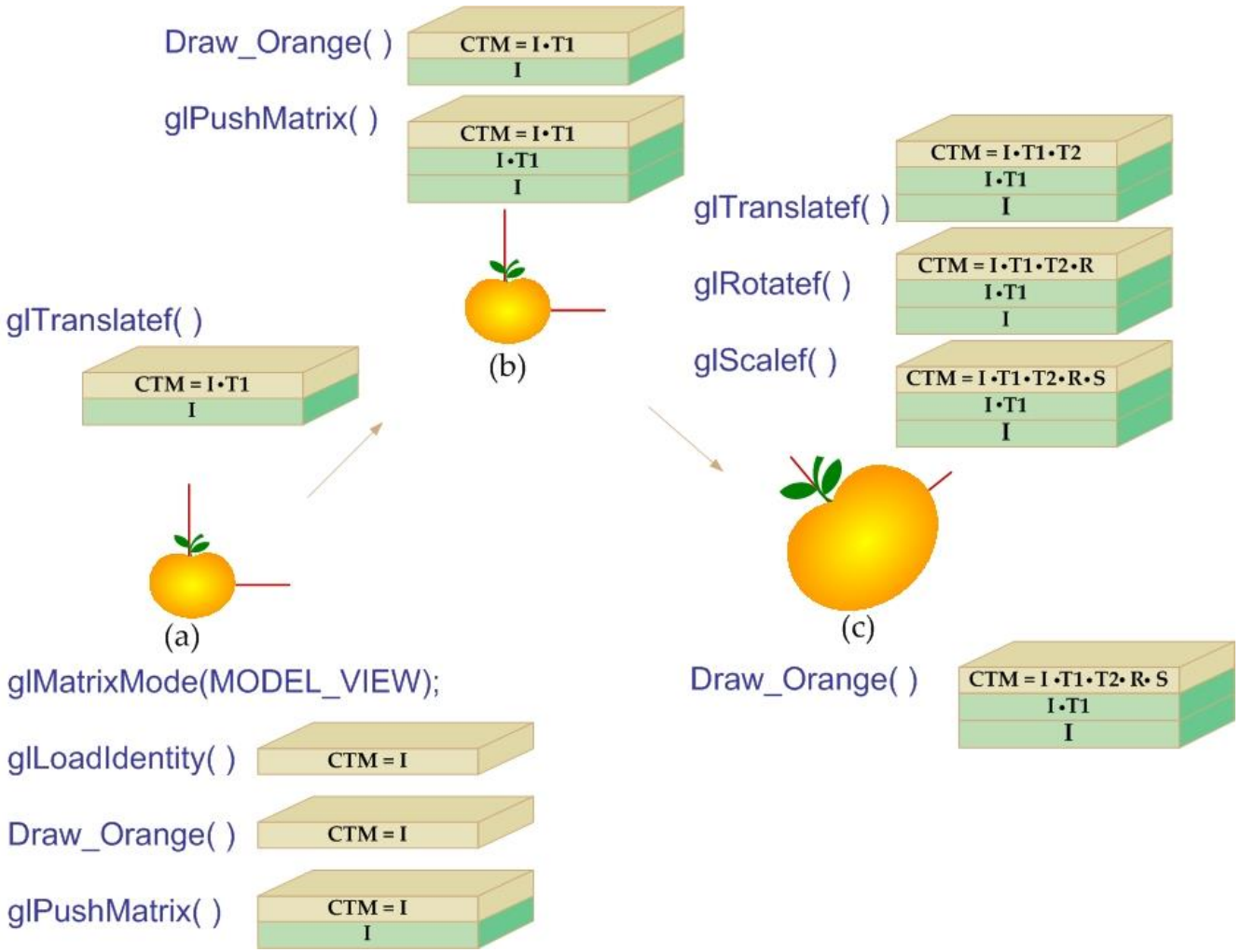
[표 6-2] 오렌지 매달기 프로그램

호출 함수	현 변환 행렬	작업
glMatrixMode(GL_MODELVIEW);	CTM ← ModelView CTM	모델 뷰 행렬 선택
❶ glLoadIdentity( );	CTM=I	초기화
❷ Draw_Orange( );	P'=CTM·P	(a)의 오렌지 그리기
❸ glTranslatef(4.0, 4.0, 0.0);	CTM=CTM·T1	좌표계 이동
❹ Draw_Orange( );	P'=CTM·P	(b)의 오렌지 그리기
❺ glTranslatef(6.0, −2.0, 0.0);	CTM=CTM·T2	좌표계 이동
❻ glRotatef(45, 0.0, 0.0, 1.0);	CTM=CTM·R	좌표계 회전
❼ glScalef(2.0, 2.0, 2.0);	CTM=CTM·S	좌표계 눈금 크기 조절
❽ Draw_Orange( );	P'=CTM·P	(c)의 오렌지 그리기

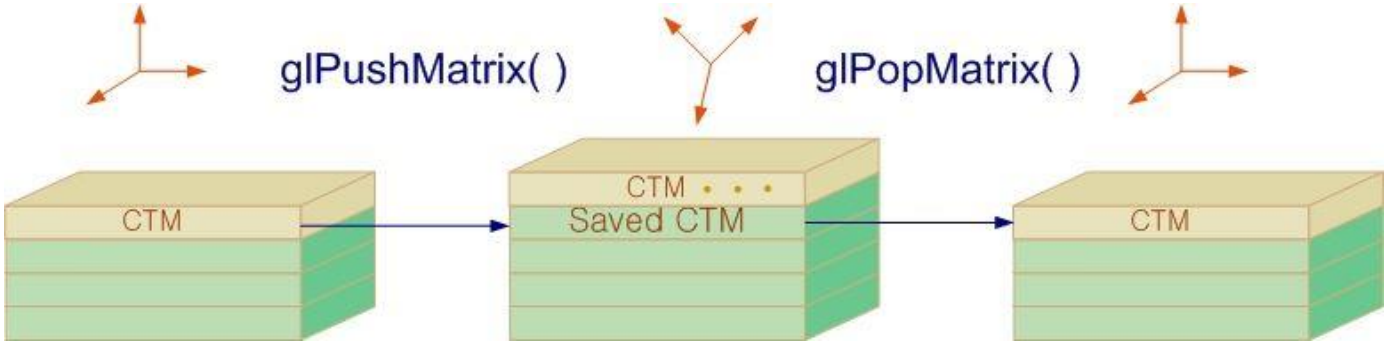
행렬 스택의 사용


glPushMatrix( )
CTM
CTM
Saved CTM
(a)
(b)
glPopMatrix( )
CTM
Saved CTM
CTM
(c)
(d)

행렬 스택의 사용


Draw_Orange( )
CTM = I•T1
I
glPushMatrix( )
CTM = I•T1
I•T1
I
glTranslatef( )
CTM = I•T1
I
(b)
glTranslatef( )
CTM = I•T1•T2
I•T1
I
glRotatef( )
CTM = I•T1•T2•R
I•T1
I
glScalef( )
CTM = I•T1•T2•R•S
I•T1
I
(a)
(c)
glMatrixMode(MODEL_VIEW);
glLoadIdentity( )
CTM = I
Draw_Orange( )
CTM = I
glPushMatrix( )
CTM = I
I
Draw_Orange( )
CTM = I•T1•T2•R•S
I•T1
I

- 일반적 형태
  - glPushMatrix();
    glTranslatef(x, y, z);
    glRotatef(angle, 0, 1, 0);
    glScalef(1.0,2.0,1.0);
    …
    Draw_TransformedObject();
    glPopMatrix();



05_1 Push Pop Matrix

CGL
COMPUTER GRAPHICS LABORATORY

2

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glColor3f(1.0, 0.0, 0.0);   //빨간색
    draw_box();

    glPushMatrix();
    glColor3f(0.0, 1.0, 0.0);   //녹색
    glTranslatef(-1.2, 0.6, 0.0);
    draw_box();
    glPopMatrix();

    glPushMatrix();
    glColor3f(0.0, 0.0, 1.0);   //파란색
    glTranslatef(0.6, 0.6, 0.0);
    glRotatef(45, 0.0, 0.0, 1.0);
    draw_box();
    glPopMatrix();

    glPushMatrix();
    glColor3f(1.0, 0.0, 1.0);   //분홍색
    glScalef(1.2, 1.2, 1.0);
    glRotatef(15, 0.0, 0.0, 1.0);
    glTranslatef(1.2, -1.2, 0.0);
    draw_box();
    glPopMatrix();

    glPushMatrix();
    glColor3f(1.0, 1.0, 0.0);   //노랑색
    glTranslatef(-0.6, -0.6, 0.0);
    glScalef(1.2, 1.2, 1.0);
    draw_box();
    glPopMatrix();

    glutSwapBuffers();
}
```

1

```
void draw_box()
{
    glBegin(GL_POLYGON);
    glVertex3f(-0.25, -0.25, 0.0);
    glVertex3f(0.25, -0.25, 0.0);
    glVertex3f(0.25, 0.25, 0.0);
    glVertex3f(-0.25, 0.25, 0.0);
    glEnd();
}
```

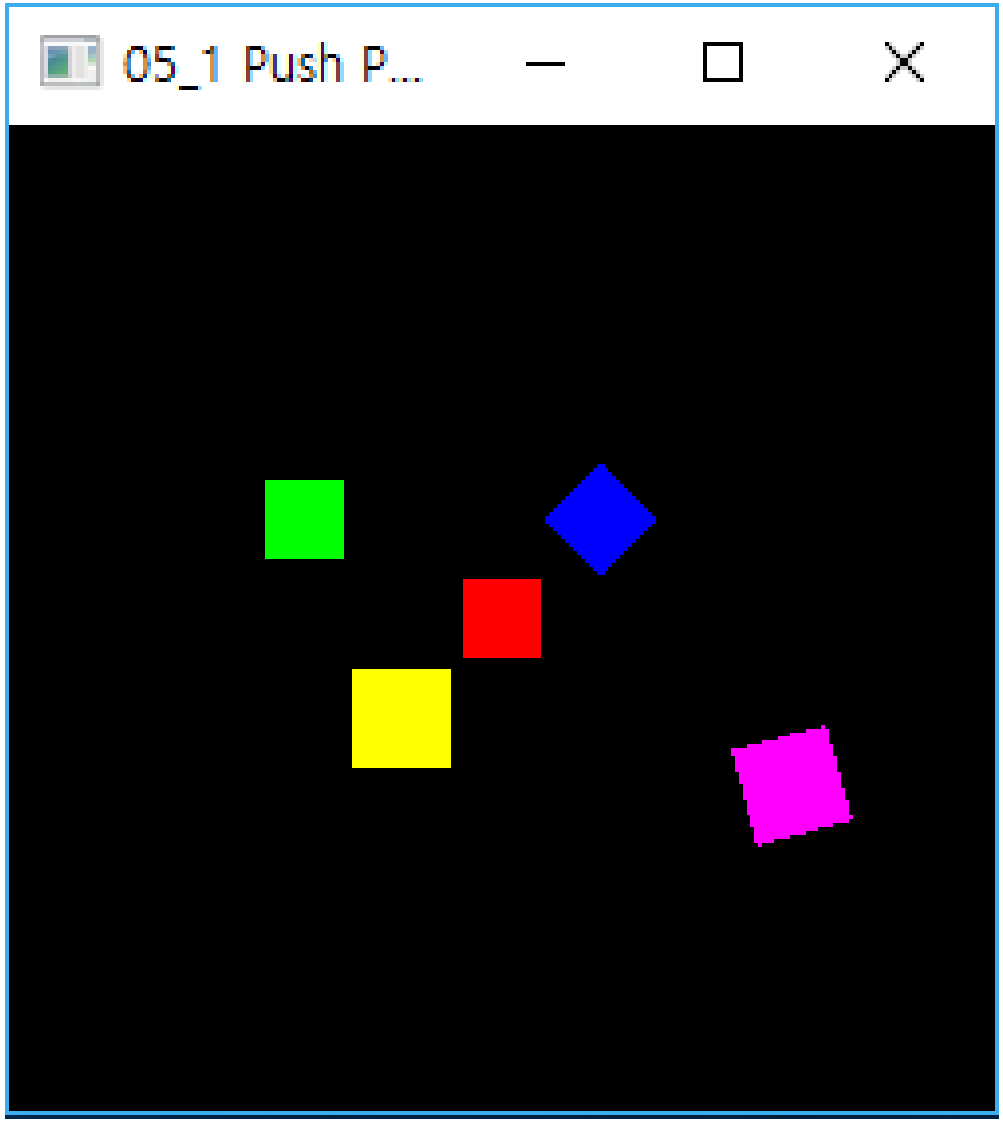
4

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

    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

    glutInitWindowSize(250, 250);
    glutInitWindowPosition(100, 100);

    glutCreateWindow("05_1 Push Pop Matrix");
    init();

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();

    return 0;
}
```

3

```
void init()
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glClear(GL_COLOR_BUFFER_BIT);
}

void reshape(int new_w, int new_h)
{
    float WidthFactor = (float)new_w / 250.0;
    float HeightFactor = (float)new_h / 250.0;

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    gluOrtho2D(-3.0 * WidthFactor, 3.0 * WidthFactor, -3.0 * HeightFactor, 3.0 * HeightFactor);
}
```

05_1 Push Pop Matrix



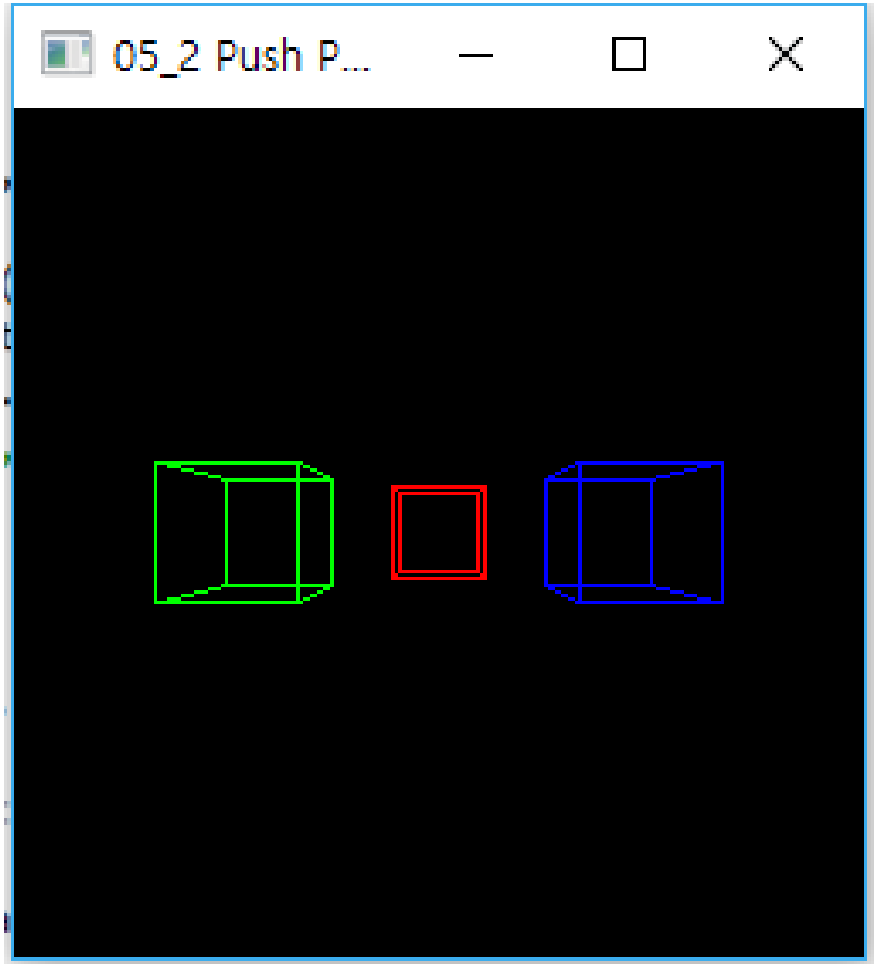
05_2 투영 변경

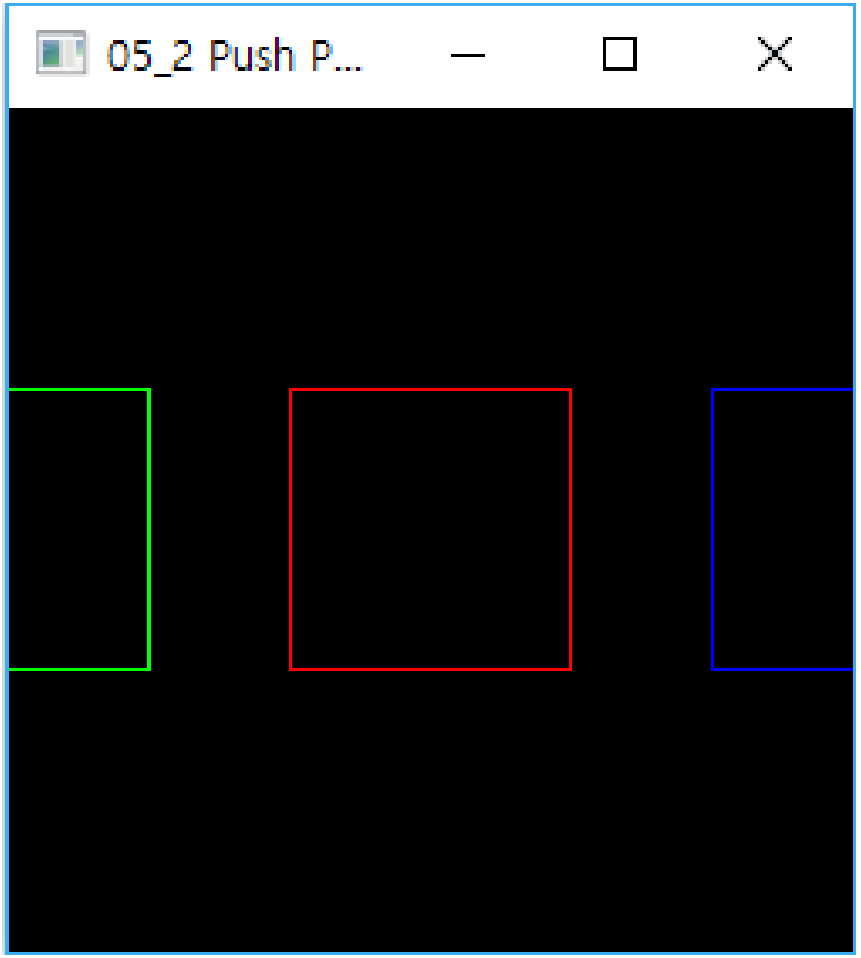
CGL
COMPUTER GRAPHICS LABORATORY

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(1.0, 1.0, 1.0);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

    glColor3f(1.0, 0.0, 0.0);
    glutWireCube(1.0);

    glPushMatrix();
    glColor3f(0.0, 1.0, 0.0);
    glTranslatef(-1.5, 0.0, 1.5);
    glutWireCube(1.0);
    glPopMatrix();

    glPushMatrix();
    glColor3f(0.0, 0.0, 1.0);
    glTranslatef(1.5, 0.0, 1.5);
    glutWireCube(1.0);
    glPopMatrix();

    glutSwapBuffers();
}
```

```
void init()
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glShadeModel(GL_SMOOTH);
}

void reshape(int new_w, int new_h)
{
    glViewport(0, 0, new_w, new_h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
//  glFrustum(-1.5, 1.5, -1.5, 1.5, 1.5, 20.0);
    glOrtho(-1.5, 1.5, -1.5, 1.5, 1.5, 20.0);

}


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

    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

    glutInitWindowSize(250, 250);
    glutInitWindowPosition(100, 100);

    glutCreateWindow("05_2 Push Pop + Projection");
    init();

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();

    return 0;
}
```

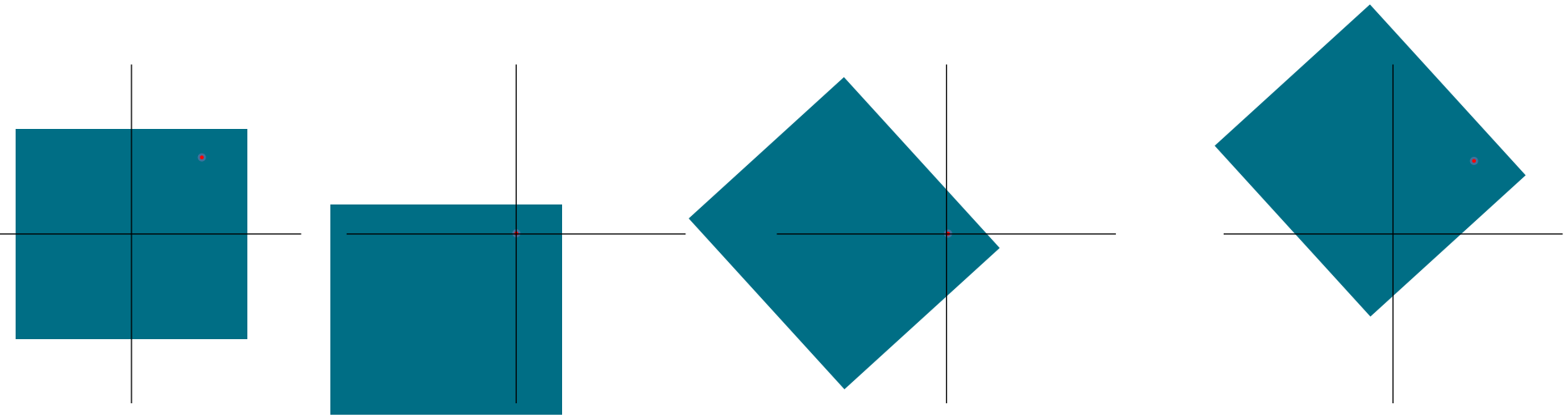
05_2 투영 변경





임의의 점에 대한 회전

- 어떠한 물체를 임의의 점(x, y, z)을 중심으로 회전 시킬 경우
  - 1. 임의의 점이 좌표축의 원점과 일치하도록 물체를 이동시킴
  - 2. 좌표축의 원점을 중심으로 회전 변환 실시
  - 3. 임의의 점이 원래의 위치가 되도록 물체를 이동시킴



1단계 2단계 3단계

05_3 임의의 점 회전

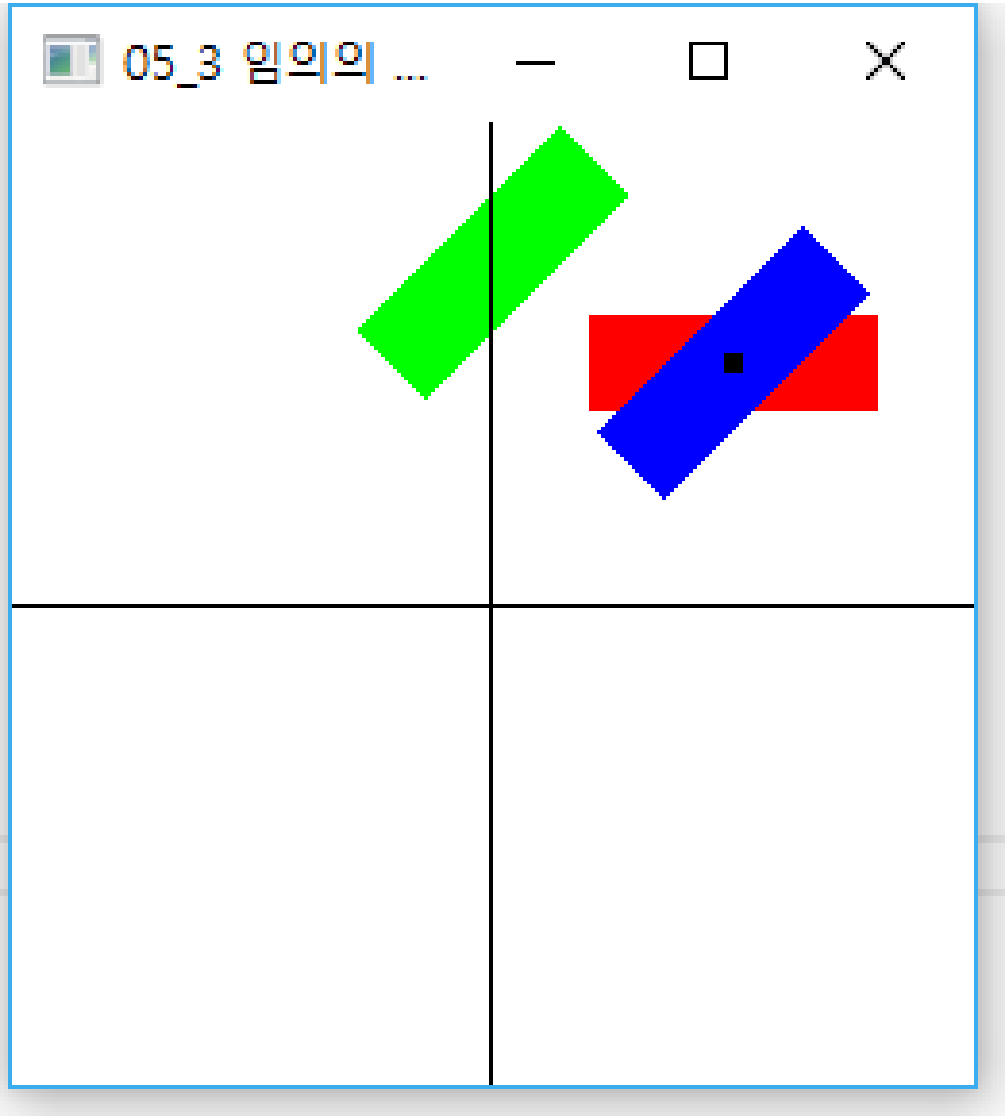
CGL
COMPUTER GRAPHICS LABORATORY

```
void draw_cube()
{
    glBegin(GL_POLYGON);
        glVertex2f(0.2, 0.4);
        glVertex2f(0.8, 0.4);
        glVertex2f(0.8, 0.6);
        glVertex2f(0.2, 0.6);
    glEnd();
}

void draw_point()
{
    glPointSize(5.0);
    glBegin(GL_POINTS);
        glVertex2f(0.5, 0.5);
    glEnd();
}

void draw_lines()
{
    glBegin(GL_LINES);
        glVertex2f(-1.0, 0.0);
        glVertex2f(1.0, 0.0);
        glVertex2f(0.0, -1.0);
        glVertex2f(0.0, 1.0);
    glEnd();
}
```

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glColor3f(1.0, 0.0, 0.0);
    draw_cube();

    glPushMatrix();
    glColor3f(0.0, 1.0, 0.0);
    glRotatef(45, 0.0, 0.0, 1.0);
    draw_cube();
    glPopMatrix();

    glColor3f(0.0, 0.0, 1.0);
    glTranslatef(0.5, 0.5, 0.0);
    glRotatef(45, 0.0, 0.0, 1.0);
    glTranslatef(-0.5, -0.5, 0.0);
    draw_cube();

    glColor3f(0.0, 0.0, 0.0);
    draw_point();

    glLoadIdentity();

    glColor3f(0.0, 0.0, 0.0);
    draw_lines();

    glutSwapBuffers();
}
```

```
void init()
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glColor3f(1.0, 1.0, 1.0);
}

void reshape(int new_w, int new_h)
{
    glViewport(0, 0, new_w, new_h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    gluOrtho2D(-1.0,1.0,-1.0,1.0);
}

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

    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

    glutInitWindowSize(250, 250);
    glutInitWindowPosition(100, 100);

    glutCreateWindow("05_3 임의의 점 회전");
    init();

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();

    return 0;
}
```

05_3 임의의 점 회전



CGL
COMPUTER GRAPHICS LABORATORY

THANK YOU

마준 | majun@sch.ac.kr