

그래픽스 강의노트 09 - 다중 텍스처

강영민

동명대학교

2016년 2학기

다중 텍스처

- 지금까지는 하나의 물체에 하나의 텍스처만 적용
- 필요에 따라 다수의 텍스처를 하나의 물체에 적용해야 할 경우
- 텍스처 맵핑 유니트(texture mapping unit, TMU)에 대한 이해:
그리픽 처리 장치(GPU)의 부분 요소
- 각각의 TMU는 3차원 객체의 임의 평면에 이미지 비트맵(bitmap)을 회전하거나 크기조정하여 부착할 수 있는 기능을 가짐
- 현대의 그리픽 카드에서는 이 유니트들이 그리픽스 파이프라인 상에서 서로 다른 단계(stage)로 구현
- 하나의 물체에 여러 개의 텍스처를 입히는 작업은 하나의 객체를 그릴 때에 여러 개의 텍스처 맵핑 유니트를 거치는 일
- 텍스처 이미지를 읽고 텍스처를 준비하는 작업을 여러 개의 유니트, 혹은 단계(stage)에 따로 따로 이루어짐
- OpenGL에서는 텍스처 맵핑과 관련된 함수들이 어떠한 유니트에 적용되는지를 지정할 수 있는 API = `glActiveTexture`
- `glActiveTexture(GL_TEXTURE0)`가 호출되면, 이후의 모든 텍스처 작업은 텍스처 유니트 0번에서 이뤄지는 일

텍스처 준비



(a) 텍스처 A (surface.jpg)



(b) 텍스처 B (spheremap.jpg)

텍스처 유니트별로 텍스처 설정

```
\// filename: main.cpp

void init( int argc , char **argv ) {
    initWindow(&argc , argv );
    ...
    // 텍스처 유니트 0에 텍스처를 하나 할당
    glActiveTexture(GL_TEXTURE0);
    tex1 = setTexture("surface.jpg");

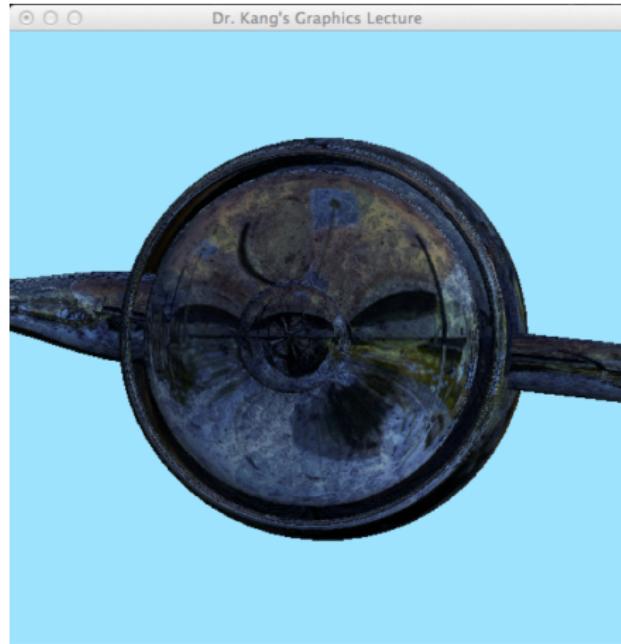
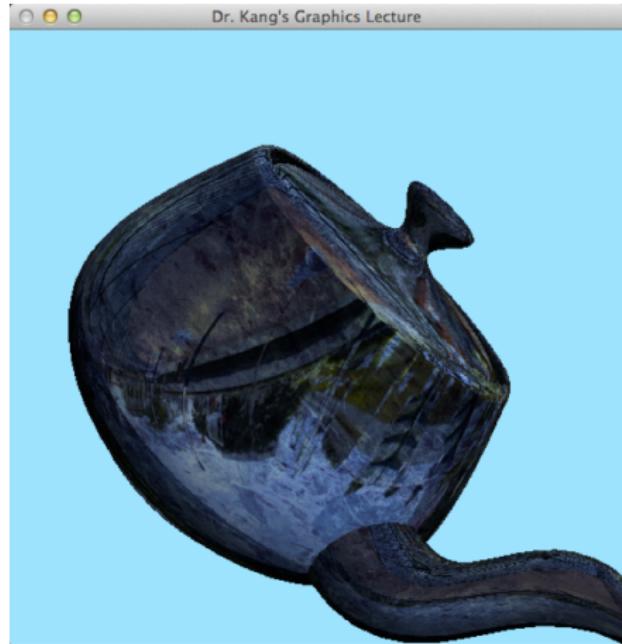
    // 텍스처 유니트 1에 텍스처를 하나 할당
    glActiveTexture(GL_TEXTURE1);
    tex2 = setTexture("spheremap.jpg");

    LightSet();
    glEnable(GL_DEPTH_TEST);
}
```

텍스처 유니트별로 텍스처 좌표 설정

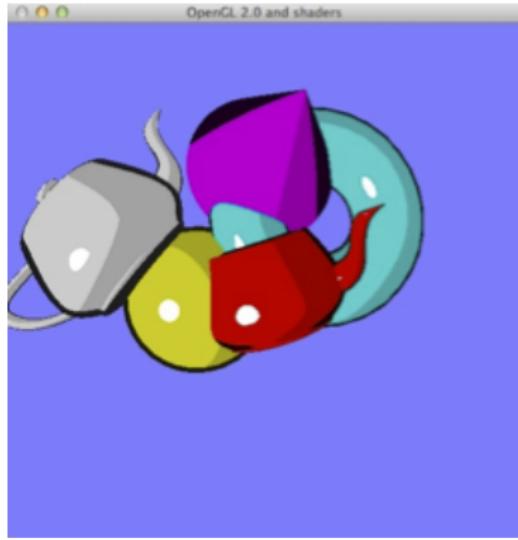
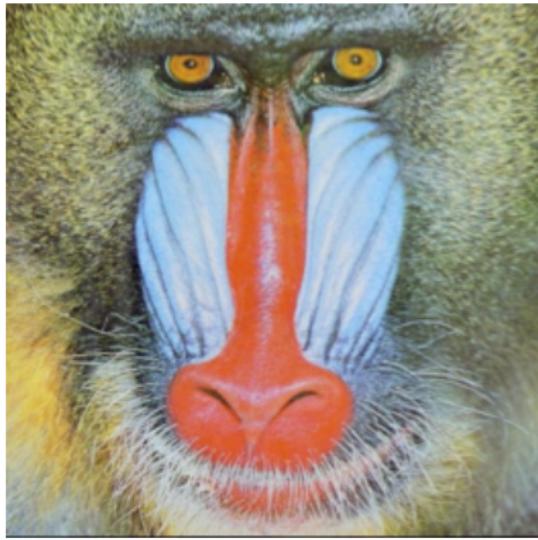
```
glEnable(GL_TEXTURE_2D);
glActiveTexture(GL_TEXTURE0);
 glBindTexture(GL_TEXTURE_2D, tex1);
glActiveTexture(GL_TEXTURE1);
 glBindTexture(GL_TEXTURE_2D, tex2);
 glEnable(GL_TEXTURE_GEN_S);    glEnable(GL_TEXTURE_GEN_T);
 glTexGenf(GL_S, GL_TEXTURE_GEN_MODE, GL_SPHERE_MAP);
 glTexGenf(GL_T, GL_TEXTURE_GEN_MODE, GL_SPHERE_MAP);
 glutSolidTeapot(1.2);
 glDisable(GL_TEXTURE_GEN_S);   glDisable(GL_TEXTURE_GEN_T);
```

텍스처 유니트별로 텍스처 좌표 설정 결과



텍스처 좌표를 명시적으로 달리 설정하는 법

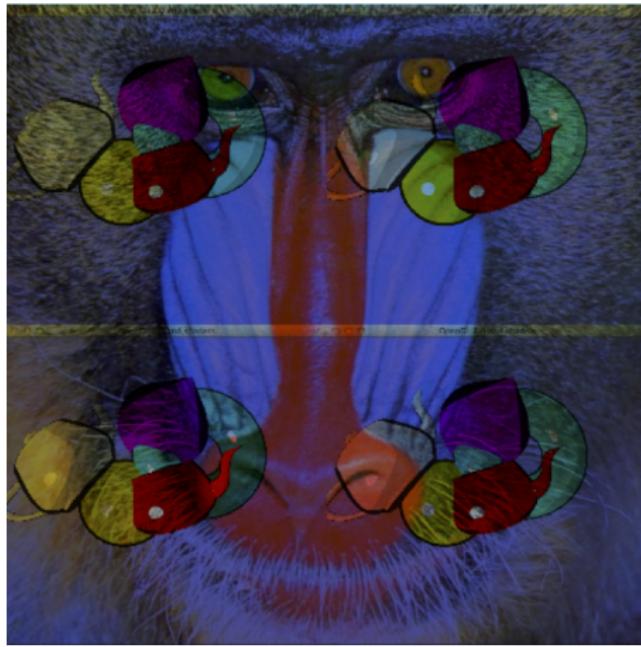
사용될 텍스처



텍스처 좌표를 명시적으로 달리 설정하는 법

```
glBegin(GL_QUADS);
glNormal3f(0.0, 0.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE0, 0.0, 0.0);
glMultiTexCoord2f(GL_TEXTURE1, 0.0, 0.0);
 glVertex3f(-0.5, 0.5, 0.0);
glMultiTexCoord2f(GL_TEXTURE0, 0.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE1, 0.0, 2.0);
 glVertex3f(-0.5, -0.5, 0.0);
glMultiTexCoord2f(GL_TEXTURE0, 1.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE1, 2.0, 2.0);
 glVertex3f( 0.5, -0.5, 0.0);
glMultiTexCoord2f(GL_TEXTURE0, 1.0, 0.0);
glMultiTexCoord2f(GL_TEXTURE1, 2.0, 0.0);
 glVertex3f( 0.5, 0.5, 0.0);
glEnd();
```

텍스처 좌표를 명시적으로 달리 설정하는 법

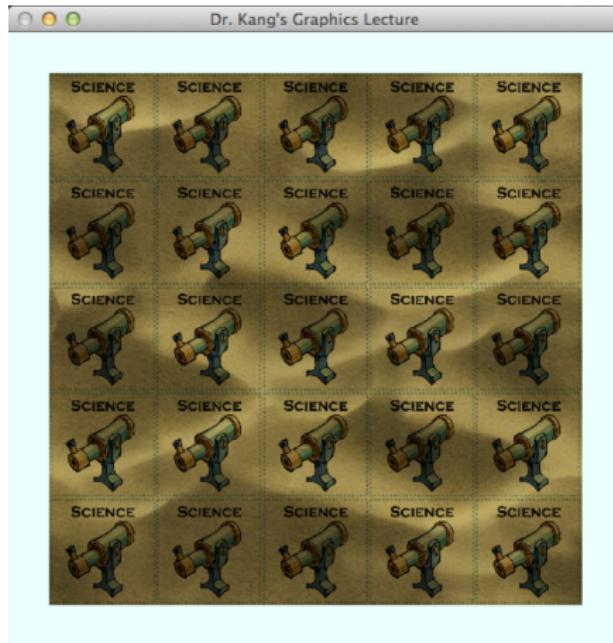


텍스처 좌표의 변환

```
glBegin(GL_QUADS);
glMultiTexCoord2f(GL_TEXTURE0, 0.0, 0.0);
glMultiTexCoord2f(GL_TEXTURE1, 0.0, 0.0);
 glVertex3f(-0.5, 0.5, 0.0);
glMultiTexCoord2f(GL_TEXTURE0, 0.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE1, 0.0, 1.0);
 glVertex3f(-0.5,-0.5, 0.0);
glMultiTexCoord2f(GL_TEXTURE0, 1.0, 1.0);
glMultiTexCoord2f(GL_TEXTURE1, 1.0, 1.0);
 glVertex3f( 0.5,-0.5, 0.0);
glMultiTexCoord2f(GL_TEXTURE0, 1.0, 0.0);
glMultiTexCoord2f(GL_TEXTURE1, 1.0, 0.0);
 glVertex3f( 0.5, 0.5, 0.0);
glEnd();
```

```
glActiveTexture(GL_TEXTURE0);
 glBindTexture(GL_TEXTURE_2D, tex1);
 glActiveTexture(GL_TEXTURE1);
 glBindTexture(GL_TEXTURE_2D, tex2);
 glMatrixMode(GL_TEXTURE); // 행렬 모드를 텍스처 모드로 바꾼다
 glLoadIdentity();
 glScalef(5.0, 5.0, 1.0);
 glBegin(GL_QUADS);
 앞의 코드와 같은 내용으로 작성
 glEnd();
```

텍스처 좌표의 변환



텍스처 애니메이션

```
glActiveTexture(GL_TEXTURE0);  
glMatrixMode(GL_TEXTURE);  
glLoadIdentity();  
glScalef(1.0+sin(t)*0.1, 1.0+sin(t)*0.1, 1.0);  
  
glActiveTexture(GL_TEXTURE1);  
glMatrixMode(GL_TEXTURE);  
glLoadIdentity();  
glTranslatef(t,0,0);  
glScalef(5.0, 5.0, 1.0);  
glEnd();
```

