

기말고사

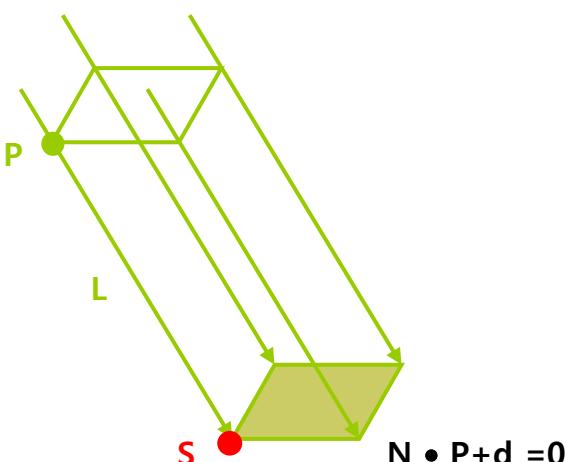
담당교수: 박경신

2019/12/20(금) 23:59까지 online.dankook.ac.kr 이러닝으로 학번_이름_final.zip 으로 끌어서 이러닝에 제출한다. 주의: 단순 번역이나 있는 그대로 복사/붙이기는 금지함. 반드시 본인이 참고한 문서의 인용을 넣어줌.

각 문제당 1장 이상씩 서술한다. 장수 제한 없음.

1. 다음 질문에 답하라. (30점)

- 1) 다음 그림에서와 같이 3차원 공간의 점 $P(x, y, z)$ 가 벡터 L 방향으로 가는 광선 ($P(t) = P + t L$)은 평면에 한 점 $S(S_x, S_y, S_z)$ 를 만난다. 평면의 공식 ($N \bullet P + d = 0$)을 이용하여 t 를 계산하라. 그리고 S 를 계산하는 공식을 유도하라. (10점)



$$S = P + tL \quad \& \quad N \bullet S + d = 0$$

- 2) `glm::ortho(5, 10, 5, 10, 5, 10)`와 `glm::frustum(5, 10, 5, 10, 5, 10)`이 생성하는 Projection 행렬(4×4)을 유도하라. 전체 풀이 과정을 보일 것. (10점)
3) 다음에서 R1 행렬과 R2 행렬 결과의 차이를 자세히 설명하라. 코드와 실행결과를 첨부하라. (10점)

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    spMain.useProgram();
    View = glm::lookAt(glm::vec3(0, 0, 10), glm::eye(0, 0, 0), glm::vec3(0, 1, 0));
    spMain.setUniform("gView", View);
    World = glm::mat4(1.0f);
    drawTeapot();
    glm::mat4 Ry = glm::rotate(glm::mat4(1.0f), M_PI/2.0, glm::vec3(0, 1, 0));
    glm::mat4 Rx = glm::rotate(glm::mat4(1.0f), M_PI/2.0, glm::vec3(1, 0, 0));
    glm::mat4 Rz = glm::rotate(glm::mat4(1.0f), M_PI/2.0, glm::vec3(0, 0, 1));
    glm::mat4 R1 = Rz * Rx * Ry;
    World = glm::translate(glm::mat4(1.0f), glm::vec3(-2.5f, 0, 0)) * R;
```

```

spMain.setUniform("gModel", World);
drawTeapot();
glm::mat4 R2 = glm::yawPitchRoll(M_PI/2, M_PI/2, M_PI/2);
World = glm::translate(glm::mat4(1.0f), glm::vec3(2.5f, 0, 0)) * R;
spMain.setUniform("gModel", World);
drawTeapot();
glutSwapBuffers();
}
    
```

3. 다음 조명에 관한 아래의 질문에 답하시오. (30점)

$$I = K_a I_a + \sum_{i=0}^{m-1} f_{att}(d_i) \{ K_d I_d (N \cdot L_i) + K_s I_s (R_i \cdot V)^n \}$$

$$I = K_a I_a + \sum_{i=0}^{m-1} f_{att}(d_i) \{ K_d I_d (N \cdot L_i) + K_s I_s (N \cdot H_i)^{n'} \}$$

- 1) 두 가지 형태의 직접 조명 모델 공식에서 램버트 법칙 (Lambert's Law)을 표현해주는 부분을 정확히 서술하라. (5점)
- 2) 위의 직접 조명 모델 공식에서 카메라에서 바라보는 방향에 직접적으로 영향을 받는 변수를 모두 설명하라. (5점)
- 3) 위의 직접 조명 모델 공식에서 간접적으로 들어오는 빛과 관련 있는 변수를 모두 설명하라. (5점)
- 4) 정반사 (reflection) 벡터, 즉 정반사 물질이 입사 광선을 반사시키는 방향을 수식으로 나타내라. (5점)
- 5) Sphere의 slice와 stack의 개수를 16개를 썼을 때와 256개를 썼을 때 Gouraud Shading과 Phong Shading를 사용했을 때 차이점을 설명하라. 4가지 각 실행결과 화면(wireframe mode와 solid drawing mode 모두)과 코드의 차이점을 보여라. (10점)

4. 다음 OpenGL 텍스쳐 매핑(Texture Mapping)과 블렌딩(Blending) 질문에 답하라. (40점)

- 1) Texture Mapping의 Two-part mapping을 자세히 서술하라. (5점)
- 2) 아래 그림과 같이 나타나도록 Quad에 texture coordinate (텍스쳐 좌표)와 texture wrapping 방식을 지정하라. 코드와 실행 결과를 첨부하라. (10점)



- 3) 다음 텍스쳐 매핑 코드에 주석을 달아라. 아래의 코드에서 확대, 축소, mipmapping 필터와 관련된 부분을 모두 찾아서 적어라. 확대, 축소, mipmapping 필터에 최근점 필터링(NEAREST)나 선형 필터링 LINEAR)를 사용시 차이점을 자세히 서술하라. (5점)

```
bool Texture2D::load(const char * filename, bool mipmaps) {
    GLsizei xdim2,ydim2;
    GLenum type;
    unsigned char *imgPtr;
    //
    imageData = simage_read_image(filename, &width, &height, &numComponents);
    if (!imageData) {
        printf("ERROR: %s is NOT loaded\n", filename);
        return false;
    }
    //
    xdim2 = 1;
    while (xdim2 <= width) xdim2 *= 2;
    xdim2 /= 2;
    ydim2 = 1;
    while (ydim2 <= height) ydim2 *= 2;
    ydim2 /= 2;
    //
    if ((width != xdim2) || (height != ydim2)) {
        rescaledImageData = simage_resize(imageData, width, height,
                                         numComponents, xdim2, ydim2);
        imgPtr = rescaledImageData;
    }
    else    imgPtr = imageData;
    //
    if (numComponents == 4) type = GL_RGBA;
    else if (numComponents == 3) type = GL_RGB;
    else if (numComponents == 2) type = GL_LUMINANCE_ALPHA;
    else if (numComponents == 1) type = GL_LUMINANCE;
    //
    if (uiTexture == 0) glGenTextures(1, &uiTexture);
    glBindTexture(GL_TEXTURE_2D, uiTexture);
    //
    glTexImage2D(GL_TEXTURE_2D, 0, (GLint) type, xdim2, ydim2, 0, type,
                 GL_UNSIGNED_BYTE, imgPtr);
    if (mipmaps) glGenerateMipmap(GL_TEXTURE_2D);
    //
}
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, magFilter);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, minFilter);
if (minFilter == GL_LINEAR_MIPMAP_NEAREST || minFilter ==
    GL_LINEAR_MIPMAP_LINEAR) {
    glGenerateMipmap(GL_TEXTURE_2D); // generate mipmap
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_BASE_LEVEL, 0);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAX_LEVEL, 4);
}
//  

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, wrapS);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, wrapT);
if (wrapS == GL_CLAMP_TO_BORDER || wrapT == GL_CLAMP_TO_BORDER) {
    const GLfloat borderColor[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    glTexParameterfv(GL_TEXTURE_2D, GL_TEXTURE_BORDER_COLOR,
borderColor);
}
glBindTexture(GL_TEXTURE_2D, 0);
//  

if (uiSampler == 0)
    glGenSamplers(1, &uiSampler);
//  

glSamplerParameteri(uiSampler, GL_TEXTURE_MAG_FILTER, magFilter);
glSamplerParameteri(uiSampler, GL_TEXTURE_MIN_FILTER, minFilter);
glSamplerParameterf(uiSampler, GL_TEXTURE_MAX_ANISOTROPY_EXT, 16.0f);
//  

glSamplerParameteri(uiSampler, GL_TEXTURE_WRAP_S, wrapS);
glSamplerParameteri(uiSampler, GL_TEXTURE_WRAP_T, wrapT);
if (wrapS == GL_CLAMP_TO_BORDER || wrapT == GL_CLAMP_TO_BORDER) {
    const GLfloat borderColor[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    glSamplerParameterfv(uiSampler, GL_TEXTURE_BORDER_COLOR,
borderColor);
}
return true;
}
```

- 4) 검정색 배경화면에서 두 개의 텍스쳐 이미지(벽돌 이미지와 라이트맵 이미지)를 겹쳐서 multi-texturing 할 때, glBlendFunc(GL_ZERO, GL_ONE), glBlendFunc(GL_ONE, GL_ONE), glBlendFunc(GL_ZERO, GL_SRC_COLOR), glBlendFunc(GL_SRC_COLOR, GL_DST_COLOR), glBlendFunc(GL_ZERO, GL_ONE_MINUS_SRC_COLOR) 블렌딩 함수는 각각 무엇인지 공식을 적고 자세히 설명하라. 코드와 실행 결과를 첨부하라. (10점)

단국대학교 응용컴퓨터공학 컴퓨터그래픽스 기말고사 (2019년 가을학기) 2019년 12월 9일 월요일
학과 _____ 학번 _____ 이름 _____

- 5) Cylinder(slice=16)를 사용하여 geometryPositionColor, geometryPositionNormal, geometryPositionNormalTexture 방식의 차이점을 설명하라. Cylinder를 그림을 그리고 그 위에 정확한 position (정점)과 normal (법선벡터)과 texture coordinate (텍스쳐 좌표) 값을 표시하라. (10점)

5. 본인의 이름(모빌 예제처럼 계층적 구조를 가지고, 움직이는)을 각각 geometryPositionColor, geometryPositionNormal, geometryPositionNormalTexture을 사용하여 프로그램을 만들어서 실행결과 화면을 보여라. source code와 실행파일을 포함한 프로젝트 파일 전체를 같이 제출하라. (20점)

- 끝 -