

Вычислительная геометрия и алгоритмы компьютерной графики

Практика №3

Рябинин Константин Валентинович

e-mail: kostya.ryabinin@gmail.com

Пермь, 2015

● Вершинный шейдер:

```
attribute vec4 a_position;  
attribute vec2 a_texCoord;  
varying vec2 v_texCoord;  
void main()  
{  
    v_texCoord = a_texCoord;  
    gl_Position = a_position;  
}
```

● Фрагментный шейдер:

```
uniform sampler2D u_map;  
varying vec2 v_texCoord;  
void main()  
{  
    gl_FragColor = texture2D(u_map, v_texCoord);  
}
```

● Основная программа:

```
. . .
GLsizei texW, texH;
GLvoid *image = loadMyTexture(&texW, &texH);
GLuint texID;
glGenTextures(1, &texID);
glActiveTexture(GL_TEXTURE0 + texUnit);
glBindTexture(GL_TEXTURE_2D, texID);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, texW, texH, 0, GL_RGBA,
             GL_UNSIGNED_BYTE, image);
delete[] image;
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);

GLint mapLocation = glGetUniformLocation(program, "u_map");
glUniform1i(mapLocation, texUnit);
. . .
glDeleteTextures(&texID);
. . .
```

