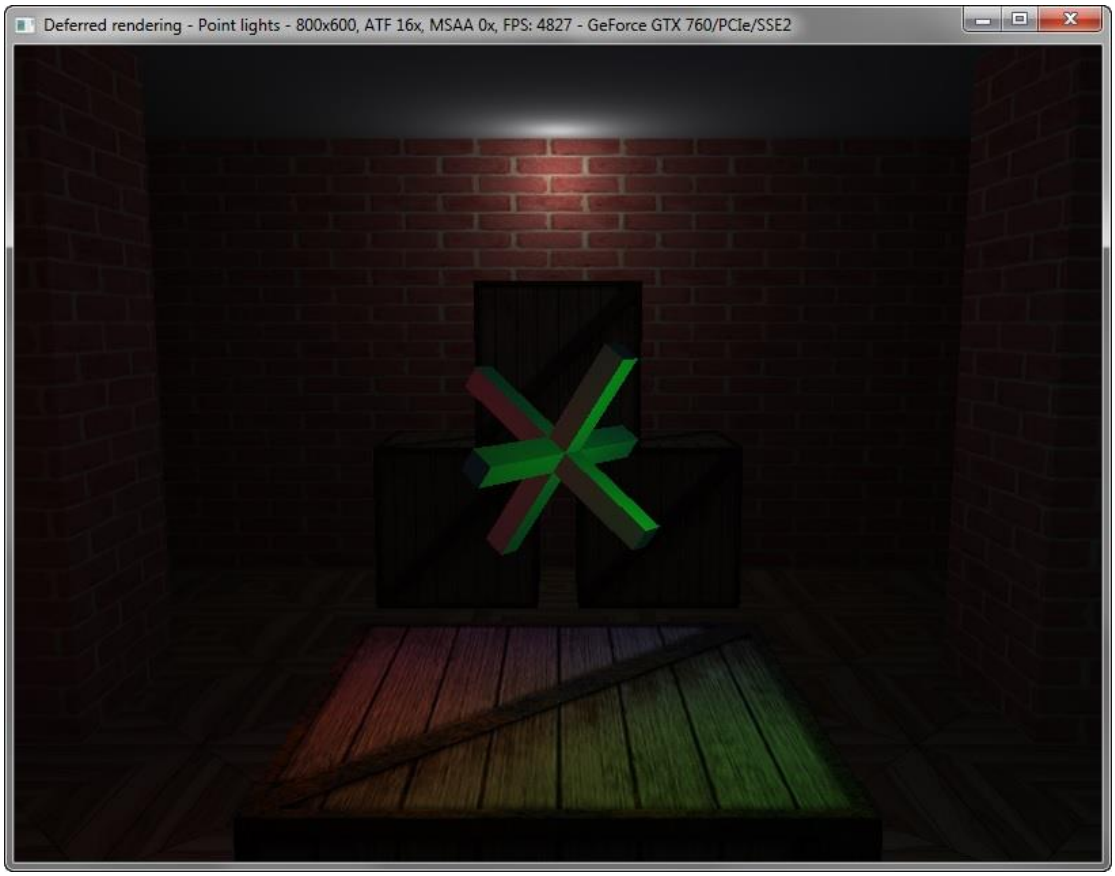
目录

第一章　延迟着色点光源



第一节 Shader Source

<1>defferedlighting.vs

```
#version 120
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

defferedlighting.fs

```
#version 120
uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer;
uniform mat4x4 ProjectionBiasMatrixInverse;
void main()
{
	gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;
	if(Depth < 1.0)
	{
		vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
1.0);
```

```
            vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
            Position /= Position.w;
            vec3 Light = vec3(0.0);
            for(int i = 0; i < 4; i++)
            {
                vec3 LightDirection = gl_LightSource[i].position.xyz - Position.xyz;
                float LightDistance2 = dot(LightDirection, LightDirection);
                float LightDistance = sqrt(LightDistance2);
                LightDirection /= LightDistance;
                float NdotLD = max(dot(Normal, LightDirection), 0.0);
                float Attenuation = gl_LightSource[i].constantAttenuation;
                Attenuation += gl_LightSource[i].linearAttenuation * LightDistance;
                Attenuation += gl_LightSource[i].quadraticAttenuation * LightDistance2;
                Light += (gl_LightSource[i].ambient.rgb + gl_LightSource[i].diffuse.rgb *
NdotLD) / Attenuation;
            }
            gl_FragColor.rgb *= Light;
        }
}
```

<2>preprocess.vs

```
#version 120

varying vec3 Normal;
void main()
{
    gl_FrontColor = gl_Color;
    gl_TexCoord[0] = gl_MultiTexCoord0;
    Normal = gl_NormalMatrix * gl_Normal;
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

preprocess.fs

```
#version 120
uniform sampler2D Texture;
uniform bool Texturing;
varying vec3 Normal;
void main()
{
    gl_FragData[0] = gl_Color;
    if(Texturing) gl_FragData[0] *= texture2D(Texture, gl_TexCoord[0].st);
    gl_FragData[1] = vec4(normalize(Normal) * 0.5 + 0.5, 1.0);
}
```

第二节 Source Code Header

```
class COpenGLRenderer
{
protected:
	int Width, Height;
	mat3x3 NormalMatrix;
	mat4x4 ModelMatrix, ViewMatrix, ProjectionMatrix, ProjectionBiasMatrixInverse;

protected:
	CTexture Texture[3];
	CShaderProgram Preprocess, DeferredLighting;
	GLuint ColorBuffer, NormalBuffer, DepthBuffer;
	GLuint VBO, FBO;

public:
	bool Pause;
	vec3 LightColors[4], LightPositions[4];

public:
	CString Text;

public:
	COpenGLRenderer();
	~COpenGLRenderer();

	bool Init();
	void Render(float FrameTime);
	void Resize(int Width, int Height);
	void Destroy();

protected:
	void InitArrayBuffers();
};
```

第三节 Source Code Cpp

```
COpenGLRenderer::COpenGLRenderer()
{
	Pause = false;

	Camera.SetViewMatrixPointer(&ViewMatrix);
}

COpenGLRenderer::~COpenGLRenderer()
{
}
```

```
bool COpenGLRenderer::Init()
{
	// ------------------------------------------------------------------------
------------------------------------------

	bool Error = false;

	// ------------------------------------------------------------------------
------------------------------------------

	if(!GLEW_ARB_texture_non_power_of_two)
	{
		ErrorLog.Append("GL_ARB_texture_non_power_of_two not supported!\r\n");
		Error = true;
	}

	if(!GLEW_ARB_depth_texture)
	{
		ErrorLog.Append("GLEW_ARB_depth_texture not supported!\r\n");
		Error = true;
	}

	if(!GLEW_EXT_framebuffer_object)
	{
		ErrorLog.Append("GL_EXT_framebuffer_object not supported!\r\n");
		Error = true;
	}

	// ------------------------------------------------------------------------
------------------------------------------

	char *TextureFileName[] = {"cube.jpg", "floor.jpg", "wall.jpg"};

	for(int i = 0; i < 3; i++)
	{
		Error |= !Texture[i].LoadTexture2D(TextureFileName[i]);
	}

	// ------------------------------------------------------------------------
------------------------------------------

	Error |= !Preprocess.Load("preprocess.vs", "preprocess.fs");
	Error |= !DeferredLighting.Load("deferredlighting.vs", "deferredlighting.fs");
```

```
	// --------------------------------------------------------------------------------
--------------------------------------------

	if(Error)
	{
		return false;
	}

	// --------------------------------------------------------------------------------
--------------------------------------------

	Preprocess.UniformLocations = new GLuint[1];
	Preprocess.UniformLocations[0] = glGetUniformLocation(Preprocess, "Texturing");

	DeferredLighting.UniformLocations = new GLuint[1];
	DeferredLighting.UniformLocations[0] = glGetUniformLocation(DeferredLighting,
"ProjectionBiasMatrixInverse");

	// --------------------------------------------------------------------------------
--------------------------------------------

	glUseProgram(DeferredLighting);
	glUniform1i(glGetUniformLocation(DeferredLighting, "ColorBuffer"), 0);
	glUniform1i(glGetUniformLocation(DeferredLighting, "NormalBuffer"), 1);
	glUniform1i(glGetUniformLocation(DeferredLighting, "DepthBuffer"), 2);
	glUseProgram(0);

	// --------------------------------------------------------------------------------
--------------------------------------------

	glGenTextures(1, &ColorBuffer);
	glGenTextures(1, &NormalBuffer);
	glGenTextures(1, &DepthBuffer);

	// --------------------------------------------------------------------------------
--------------------------------------------

	glGenBuffers(1, &VBO);

	InitArrayBuffers();

	// --------------------------------------------------------------------------------
--------------------------------------------
```

```
	glGenFramebuffersEXT(1, &FBO);

	// -------------------------------------------------------------------------
-----------------------------------------------

	LightColors[0] = vec3(1.0f, 0.0f, 0.0f);
	LightPositions[0] = vec3(0.0f, 1.5f, 0.33f);
	LightColors[1] = vec3(0.0f, 1.0f, 0.0f);
	LightPositions[1] = rotate(LightPositions[0], 120.0f, vec3(0.0f, 1.0f, 0.0f));
	LightColors[2] = vec3(0.0f, 0.0f, 1.0f);
	LightPositions[2] = rotate(LightPositions[1], 120.0f, vec3(0.0f, 1.0f, 0.0f));
	LightColors[3] = vec3(1.0f, 1.0f, 1.0f);
	LightPositions[3] = vec3(0.0f, 2.75f, -4.75f);

	for(int i = 0; i < 3; i++)
	{
		glLightfv(GL_LIGHT0 + i, GL_AMBIENT, &vec4(LightColors[i] * 0.125f, 1.0f));
		glLightfv(GL_LIGHT0 + i, GL_DIFFUSE, &vec4(LightColors[i] * 0.875f, 1.0f));
		glLightf(GL_LIGHT0 + i, GL_LINEAR_ATTENUATION, 1.0f);
		glLightf(GL_LIGHT0 + i, GL_QUADRATIC_ATTENUATION, 1.0f);
	}

	glLightfv(GL_LIGHT3, GL_AMBIENT, &vec4(LightColors[3] * 0.25f, 1.0f));
	glLightfv(GL_LIGHT3, GL_DIFFUSE, &vec4(LightColors[3] * 0.75f, 1.0f));
	glLightf(GL_LIGHT3, GL_LINEAR_ATTENUATION, 1.0f);
	glLightf(GL_LIGHT3, GL_QUADRATIC_ATTENUATION, 1.0f);

	// -------------------------------------------------------------------------
-----------------------------------------------

	Camera.Look(vec3(0.0f, 1.75f, 1.875f), vec3(0.0f, 1.5f, 0.0f));

	// -------------------------------------------------------------------------
-----------------------------------------------

	return true;

	// -------------------------------------------------------------------------
-----------------------------------------------
}

void COpenGLRenderer::Render(float FrameTime)
{
```

```
    // ------------------------------------------------------------------------
------------------------------------------

    GLenum Buffers[] = {GL_COLOR_ATTACHMENT0_EXT,
GL_COLOR_ATTACHMENT1_EXT};

    // render scene to textures ---------------------------------------------
------------------------------------------

    glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
    glDrawBuffers(2, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
    glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT0_EXT,
GL_TEXTURE_2D, ColorBuffer, 0);
    glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT1_EXT,
GL_TEXTURE_2D, NormalBuffer, 0);
    glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_DEPTH_ATTACHMENT_EXT,
GL_TEXTURE_2D, DepthBuffer, 0);

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadMatrixf(&ViewMatrix);

    glEnable(GL_DEPTH_TEST);
    glEnable(GL_CULL_FACE);

    glBindBuffer(GL_ARRAY_BUFFER, VBO);

    glEnableClientState(GL_TEXTURE_COORD_ARRAY);
    glTexCoordPointer(2, GL_FLOAT, 32, (void*)0);

    glEnableClientState(GL_NORMAL_ARRAY);
    glNormalPointer(GL_FLOAT, 32, (void*)8);

    glEnableClientState(GL_VERTEX_ARRAY);
    glVertexPointer(3, GL_FLOAT, 32, (void*)20);

    glUseProgram(Preprocess);

    glUniform1i(Preprocess.UniformLocations[0], true);

    glColor3f(1.0f, 1.0f, 1.0f);

    glBindTexture(GL_TEXTURE_2D, Texture[0]);
```

```
        glDrawArrays(GL_QUADS, 0, 96);

        glBindTexture(GL_TEXTURE_2D, Texture[1]);
        glDrawArrays(GL_QUADS, 96, 4);

        glBindTexture(GL_TEXTURE_2D, Texture[2]);
        glDrawArrays(GL_QUADS, 100, 80);

        glBindTexture(GL_TEXTURE_2D, 0);

        glUniform1i(Preprocess.UniformLocations[0], false);

        glDrawArrays(GL_QUADS, 180, 4);

        glMultMatrixf(&ModelMatrix);
        glColor3f(0.33f, 0.66f, 1.0f);
        glDrawArrays(GL_QUADS, 184, 72);

        glUseProgram(0);

        glDisableClientState(GL_VERTEX_ARRAY);
        glDisableClientState(GL_NORMAL_ARRAY);
        glDisableClientState(GL_TEXTURE_COORD_ARRAY);

        glBindBuffer(GL_ARRAY_BUFFER, 0);

        glDisable(GL_CULL_FACE);
        glDisable(GL_DEPTH_TEST);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

        // set lights positions ------------------------------------------------------
------------------------------------------

        glMatrixMode(GL_MODELVIEW);
        glLoadMatrixf(&ViewMatrix);

        for(int i = 0; i < 4; i++)
        {
                glLightfv(GL_LIGHT0 + i, GL_POSITION, &vec4(LightPositions[i], 1.0f));
        }

        // calculate lighting --------------------------------------------------------
-----------------------------------------
```

```
	glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffer);
	glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
	glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
	glUseProgram(DeferredLighting);
	glBegin(GL_QUADS);
		glVertex2f(0.0f, 0.0f);
		glVertex2f(1.0f, 0.0f);
		glVertex2f(1.0f, 1.0f);
		glVertex2f(0.0f, 1.0f);
	glEnd();
	glUseProgram(0);
	glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
	glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
	glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

	// rotate object and lights ----------------------------------------------------
----------------------------------------

	if(!Pause)
	{
		static float a = 0.0f;

		ModelMatrix = translate(0.0f, 1.5f, 0.0f) * rotate(a, vec3(0.0f, 1.0f, 0.0f)) * rotate(a,
vec3(1.0f, 0.0f, 0.0f));

		a += 22.5f * FrameTime;

		for(int i = 0; i < 3; i++)
		{
			LightPositions[i] = rotate(LightPositions[i], -180.0f * FrameTime, vec3(0.0f, 1.0f,
0.0f));
		}
	}

	// ----------------------------------------------------------------------------
----------------------------------------------
}

void COpenGLRenderer::Resize(int Width, int Height)
{
	this->Width = Width;
	this->Height = Height;
```

```
	glViewport(0, 0, Width, Height);

	ProjectionMatrix = perspective(45.0f, (float)Width / (float)Height, 0.125f, 512.0f);

	glMatrixMode(GL_PROJECTION);
	glLoadMatrixf(&ProjectionMatrix);

	ProjectionBiasMatrixInverse = inverse(ProjectionMatrix) * BiasMatrixInverse;

	glBindTexture(GL_TEXTURE_2D, ColorBuffer);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
	glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
	glBindTexture(GL_TEXTURE_2D, 0);

	glBindTexture(GL_TEXTURE_2D, NormalBuffer);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
	glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
	glBindTexture(GL_TEXTURE_2D, 0);

	glBindTexture(GL_TEXTURE_2D, DepthBuffer);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
	glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT24, Width, Height, 0,
GL_DEPTH_COMPONENT, GL_FLOAT, NULL);
	glBindTexture(GL_TEXTURE_2D, 0);

	glUseProgram(DeferredLighting);
	glUniformMatrix4fv(DeferredLighting.UniformLocations[0], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
	glUseProgram(0);
}

void COpenGLRenderer::Destroy()
{
```

```
	for(int i = 0; i < 3; i++)
	{
		Texture[i].Destroy();
	}

	Preprocess.Destroy();
	DeferredLighting.Destroy();

	glDeleteBuffers(1, &VBO);

	glDeleteTextures(1, &ColorBuffer);
	glDeleteTextures(1, &NormalBuffer);
	glDeleteTextures(1, &DepthBuffer);

	if(GLEW_EXT_framebuffer_object)
	{
		glDeleteFramebuffersEXT(1, &FBO);
	}
}

void COpenGLRenderer::InitArrayBuffers()
{
	CBuffer buffer;

	vec3 m;

	// cubes

	m = vec3( 0.0f, 0.5f, 0.0f);

	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.0f, 1.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    m = vec3(-0.625f, 0.5f,-4.5f);
```

```
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.625f, 0.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    // floor

    buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,    5.0f), 12);
    buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,    5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, -5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, -5.0f), 12);

    // walls

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
```

```
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);

    // pillars

    m = vec3(-2.5f, 0.0f,-2.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f,-2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f, 2.5f);
```

```
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3(-2.5f, 0.0f, 2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    // ceiling

    buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);

    // rotating object

    m = vec3(0.0f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
```

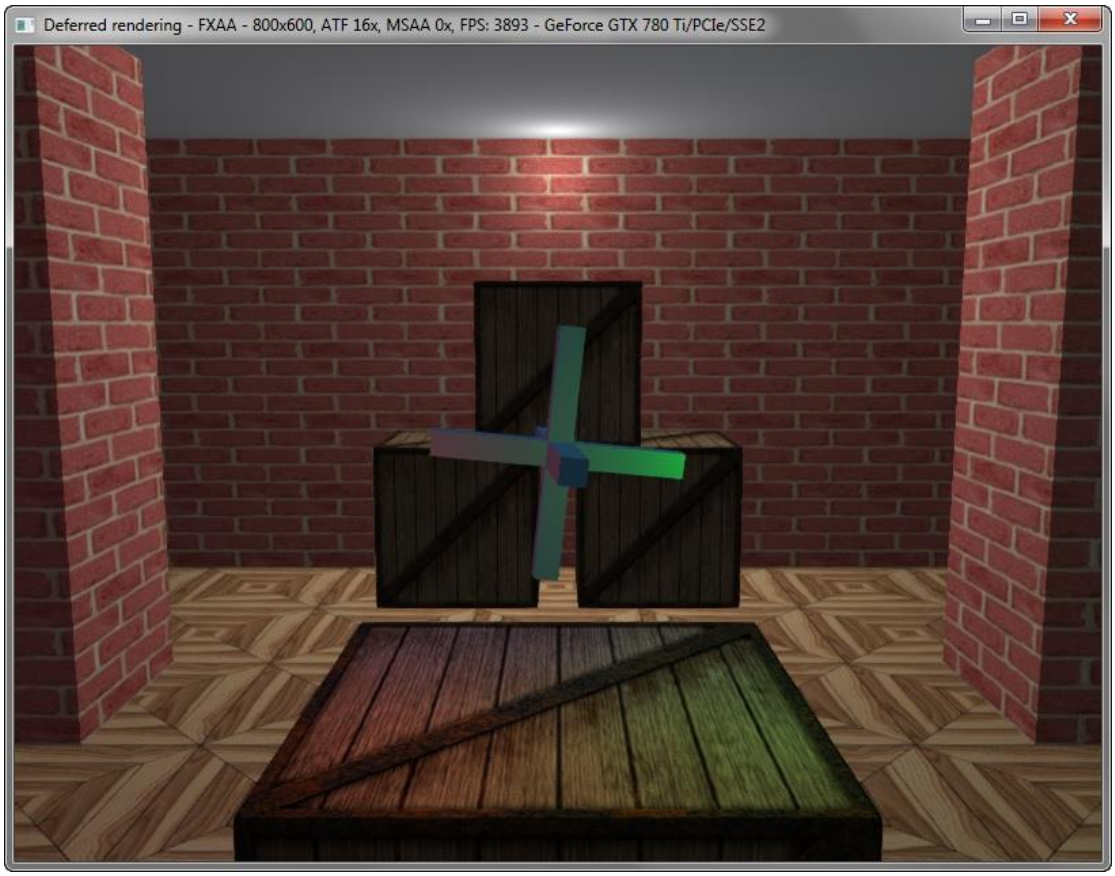
```
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
```

```
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
```

```
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);

	glBindBuffer(GL_ARRAY_BUFFER, VBO);
	glBufferData(GL_ARRAY_BUFFER, buffer.GetDataSize(), buffer.GetData(),
GL_STATIC_DRAW);
	glBindBuffer(GL_ARRAY_BUFFER, 0);

	buffer.Empty();
}
```

第二章　延迟着色 FXAA



第一节 Shader Source

<1>defferedlighting.vs

```
#version 120
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

defferedlighting.fs

```
#version 120
uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer;
uniform mat4x4 ProjectionBiasMatrixInverse;
void main()
{
	gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;
	if(Depth < 1.0)
	{
		vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
```

```
1.0);
            vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
            Position /= Position.w;
            vec3 Light = vec3(0.0);
            for(int i = 0; i < 4; i++)
            {
                vec3 LightDirection = gl_LightSource[i].position.xyz - Position.xyz;
                float LightDistance2 = dot(LightDirection, LightDirection);
                float LightDistance = sqrt(LightDistance2);
                LightDirection /= LightDistance;
                float NdotLD = max(dot(Normal, LightDirection), 0.0);
                float Attenuation = gl_LightSource[i].constantAttenuation;
                Attenuation += gl_LightSource[i].linearAttenuation * LightDistance;
                Attenuation += gl_LightSource[i].quadraticAttenuation * LightDistance2;
                Light += (gl_LightSource[i].ambient.rgb + gl_LightSource[i].diffuse.rgb *
NdotLD) / Attenuation;
            }
            gl_FragColor.rgb *= Light;
        }
}
<2>preprocess.vs
#version 120
varying vec3 Normal;
void main()
{
    gl_FrontColor = gl_Color;
    gl_TexCoord[0] = gl_MultiTexCoord0;
    Normal = gl_NormalMatrix * gl_Normal;
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
Preprocess.fs
#version 120
uniform sampler2D Texture;
uniform bool Texturing;
varying vec3 Normal;
void main()
{
    gl_FragData[0] = gl_Color;
    if(Texturing) gl_FragData[0] *= texture2D(Texture, gl_TexCoord[0].st);
    gl_FragData[1] = vec4(normalize(Normal) * 0.5 + 0.5, 1.0);
}
```

第二节 Source Code Header

lass COpenGLRenderer

```
{
protected:
	int Width, Height;
	mat3x3 NormalMatrix;
	mat4x4 ModelMatrix, ViewMatrix, ProjectionMatrix, ProjectionBiasMatrixInverse;

protected:
	CTexture Texture[3];
	CShaderProgram Preprocess, DeferredLighting, FXAA;
	GLuint ColorBuffers[2], NormalBuffer, DepthBuffer;
	GLuint VBO, FBO;

public:
	bool CalculateFXAA, Pause;
	vec3 LightColors[4], LightPositions[4];

public:
	CString Text;

public:
	COpenGLRenderer();
	~COpenGLRenderer();

	bool Init();
	void Render(float FrameTime);
	void Resize(int Width, int Height);
	void Destroy();

protected:
	void InitArrayBuffers();
};
```

第三节 Source Code Cpp

```
COpenGLRenderer::COpenGLRenderer()
{
	CalculateFXAA = true;

	Pause = false;

	Camera.SetViewMatrixPointer(&ViewMatrix);
}

COpenGLRenderer::~COpenGLRenderer()
{
```

```
}

bool COpenGLRenderer::Init()
{
    // ------------------------------------------------------------------------
-----------------------------------------------

    bool Error = false;

    // ------------------------------------------------------------------------
-----------------------------------------------

    if(!GLEW_ARB_texture_non_power_of_two)
    {
        ErrorLog.Append("GL_ARB_texture_non_power_of_two not supported!\r\n");
        Error = true;
    }

    if(!GLEW_ARB_depth_texture)
    {
        ErrorLog.Append("GLEW_ARB_depth_texture not supported!\r\n");
        Error = true;
    }

    if(!GLEW_EXT_framebuffer_object)
    {
        ErrorLog.Append("GL_EXT_framebuffer_object not supported!\r\n");
        Error = true;
    }

    // ------------------------------------------------------------------------
-----------------------------------------------

    char *TextureFileName[] = {"cube.jpg", "floor.jpg", "wall.jpg"};

    for(int i = 0; i < 3; i++)
    {
        Error |= !Texture[i].LoadTexture2D(TextureFileName[i]);
    }

    // ------------------------------------------------------------------------
-----------------------------------------------

    Error |= !Preprocess.Load("preprocess.vs", "preprocess.fs");
```

```
	Error |= !DeferredLighting.Load("deferredlighting.vs", "deferredlighting.fs");
	Error |= !FXAA.Load("FXAA.vert", "FXAA_Extreme_Quality.frag");

	// ------------------------------------------------------------------------
-------------------------------------------

	if(Error)
	{
		return false;
	}

	// ------------------------------------------------------------------------
-------------------------------------------

	Preprocess.UniformLocations = new GLuint[1];
	Preprocess.UniformLocations[0] = glGetUniformLocation(Preprocess, "Texturing");

	DeferredLighting.UniformLocations = new GLuint[1];
	DeferredLighting.UniformLocations[0] = glGetUniformLocation(DeferredLighting,
"ProjectionBiasMatrixInverse");

	FXAA.UniformLocations = new GLuint[1];
	FXAA.UniformLocations[0] = glGetUniformLocation(FXAA, "RCPFrame");

	// ------------------------------------------------------------------------
-------------------------------------------

	glUseProgram(DeferredLighting);
	glUniform1i(glGetUniformLocation(DeferredLighting, "ColorBuffer"), 0);
	glUniform1i(glGetUniformLocation(DeferredLighting, "NormalBuffer"), 1);
	glUniform1i(glGetUniformLocation(DeferredLighting, "DepthBuffer"), 2);
	glUseProgram(0);

	// ------------------------------------------------------------------------
-------------------------------------------

	glGenTextures(2, ColorBuffers);
	glGenTextures(1, &NormalBuffer);
	glGenTextures(1, &DepthBuffer);

	// ------------------------------------------------------------------------
-------------------------------------------

	glGenBuffers(1, &VBO);
```

```
    InitArrayBuffers();

    // ------------------------------------------------------------------------
------------------------------------------

    glGenFramebuffersEXT(1, &FBO);

    // ------------------------------------------------------------------------
------------------------------------------

    LightColors[0] = vec3(1.0f, 0.0f, 0.0f);
    LightPositions[0] = vec3(0.0f, 1.5f, 0.33f);
    LightColors[1] = vec3(0.0f, 1.0f, 0.0f);
    LightPositions[1] = rotate(LightPositions[0], 120.0f, vec3(0.0f, 1.0f, 0.0f));
    LightColors[2] = vec3(0.0f, 0.0f, 1.0f);
    LightPositions[2] = rotate(LightPositions[1], 120.0f, vec3(0.0f, 1.0f, 0.0f));
    LightColors[3] = vec3(1.0f, 1.0f, 1.0f);
    LightPositions[3] = vec3(0.0f, 2.75f, -4.75f);

    for(int i = 0; i < 3; i++)
    {
        glLightfv(GL_LIGHT0 + i, GL_AMBIENT, &vec4(LightColors[i] * 0.125f, 1.0f));
        glLightfv(GL_LIGHT0 + i, GL_DIFFUSE, &vec4(LightColors[i] * 0.875f, 1.0f));
        glLightf(GL_LIGHT0 + i, GL_LINEAR_ATTENUATION, 1.0f);
        glLightf(GL_LIGHT0 + i, GL_QUADRATIC_ATTENUATION, 1.0f);
    }

    glLightfv(GL_LIGHT3, GL_AMBIENT, &vec4(LightColors[3] * 0.25f, 1.0f));
    glLightfv(GL_LIGHT3, GL_DIFFUSE, &vec4(LightColors[3] * 0.75f, 1.0f));
    glLightf(GL_LIGHT3, GL_LINEAR_ATTENUATION, 1.0f / 32.0f);
    glLightf(GL_LIGHT3, GL_QUADRATIC_ATTENUATION, 1.0f / 64.0f);

    // ------------------------------------------------------------------------
------------------------------------------

    Camera.Look(vec3(0.0f, 1.75f, 1.875f), vec3(0.0f, 1.5f, 0.0f));

    // ------------------------------------------------------------------------
------------------------------------------

    return true;

    // ------------------------------------------------------------------------
```

```
-----------------------------------------------
}

void COpenGLRenderer::Render(float FrameTime)
{
	// ------------------------------------------------------------------------
-----------------------------------------------

	GLenum Buffers[] = {GL_COLOR_ATTACHMENT0_EXT, GL_COLOR_ATTACHMENT1_EXT};

	// render scene to textures ---------------------------------------------
----------------------------------------

	glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
	glDrawBuffers(2, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,  GL_COLOR_ATTACHMENT0_EXT,
GL_TEXTURE_2D, ColorBuffers[0], 0);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,  GL_COLOR_ATTACHMENT1_EXT,
GL_TEXTURE_2D, NormalBuffer, 0);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,  GL_DEPTH_ATTACHMENT_EXT,
GL_TEXTURE_2D, DepthBuffer, 0);

	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

	glMatrixMode(GL_MODELVIEW);
	glLoadMatrixf(&ViewMatrix);

	glEnable(GL_DEPTH_TEST);
	glEnable(GL_CULL_FACE);

	glBindBuffer(GL_ARRAY_BUFFER, VBO);

	glEnableClientState(GL_TEXTURE_COORD_ARRAY);
	glTexCoordPointer(2, GL_FLOAT, 32, (void*)0);

	glEnableClientState(GL_NORMAL_ARRAY);
	glNormalPointer(GL_FLOAT, 32, (void*)8);

	glEnableClientState(GL_VERTEX_ARRAY);
	glVertexPointer(3, GL_FLOAT, 32, (void*)20);

	glUseProgram(Preprocess);

	glUniform1i(Preprocess.UniformLocations[0], true);
```

```
        glColor3f(1.0f, 1.0f, 1.0f);

        glBindTexture(GL_TEXTURE_2D, Texture[0]);
        glDrawArrays(GL_QUADS, 0, 96);

        glBindTexture(GL_TEXTURE_2D, Texture[1]);
        glDrawArrays(GL_QUADS, 96, 4);

        glBindTexture(GL_TEXTURE_2D, Texture[2]);
        glDrawArrays(GL_QUADS, 100, 80);

        glBindTexture(GL_TEXTURE_2D, 0);

        glUniform1i(Preprocess.UniformLocations[0], false);

        glDrawArrays(GL_QUADS, 180, 4);

        glMultMatrixf(&ModelMatrix);
        glColor3f(0.33f, 0.66f, 1.0f);
        glDrawArrays(GL_QUADS, 184, 72);

        glUseProgram(0);

        glDisableClientState(GL_VERTEX_ARRAY);
        glDisableClientState(GL_NORMAL_ARRAY);
        glDisableClientState(GL_TEXTURE_COORD_ARRAY);

        glBindBuffer(GL_ARRAY_BUFFER, 0);

        glDisable(GL_CULL_FACE);
        glDisable(GL_DEPTH_TEST);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

        // set lights positions ------------------------------------------------------
------------------------------------------------------------------

        glMatrixMode(GL_MODELVIEW);
        glLoadMatrixf(&ViewMatrix);

        for(int i = 0; i < 4; i++)
        {
            glLightfv(GL_LIGHT0 + i, GL_POSITION, &vec4(LightPositions[i], 1.0f));
```

```
    }

    // calculate lighting ----------------------------------------------------
-----------------------------------------

    if(CalculateFXAA)
    {
        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
        glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, ColorBuffers[1], 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);
    }

    glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffers[0]);
    glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
    glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
    glUseProgram(DeferredLighting);
    glBegin(GL_QUADS);
        glVertex2f(0.0f, 0.0f);
        glVertex2f(1.0f, 0.0f);
        glVertex2f(1.0f, 1.0f);
        glVertex2f(0.0f, 1.0f);
    glEnd();
    glUseProgram(0);
    glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
    glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
    glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

    if(CalculateFXAA)
    {
        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);
    }

    // calculate antialiasing ------------------------------------------------
-----------------------------------------

    if(CalculateFXAA)
    {
        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffers[1]);
        glUseProgram(FXAA);
```

```
		glBegin(GL_QUADS);
			glVertex2f(0.0f, 0.0f);
			glVertex2f(1.0f, 0.0f);
			glVertex2f(1.0f, 1.0f);
			glVertex2f(0.0f, 1.0f);
		glEnd();
		glUseProgram(0);
		glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);
	}

	// rotate object and lights ----------------------------------------------------
----------------------------------------

	if(!Pause)
	{
		static float a = 0.0f;

		ModelMatrix = translate(0.0f, 1.5f, 0.0f) * rotate(a, vec3(0.0f, 1.0f, 0.0f)) * rotate(a,
vec3(1.0f, 0.0f, 0.0f));

		a += 22.5f * FrameTime;

		for(int i = 0; i < 3; i++)
		{
			LightPositions[i] = rotate(LightPositions[i], -180.0f * FrameTime, vec3(0.0f, 1.0f,
0.0f));
		}
	}

	// -----------------------------------------------------------------------------
----------------------------------------------
}

void COpenGLRenderer::Resize(int Width, int Height)
{
	this->Width = Width;
	this->Height = Height;

	glViewport(0, 0, Width, Height);

	ProjectionMatrix = perspective(45.0f, (float)Width / (float)Height, 0.125f, 512.0f);

	glMatrixMode(GL_PROJECTION);
	glLoadMatrixf(&ProjectionMatrix);
```

```
    ProjectionBiasMatrixInverse = inverse(ProjectionMatrix) * BiasMatrixInverse;

    glBindTexture(GL_TEXTURE_2D, ColorBuffers[0]);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
    glBindTexture(GL_TEXTURE_2D, 0);

    glBindTexture(GL_TEXTURE_2D, ColorBuffers[1]);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
    glBindTexture(GL_TEXTURE_2D, 0);

    glBindTexture(GL_TEXTURE_2D, NormalBuffer);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
    glBindTexture(GL_TEXTURE_2D, 0);

    glBindTexture(GL_TEXTURE_2D, DepthBuffer);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT24, Width, Height, 0,
GL_DEPTH_COMPONENT, GL_FLOAT, NULL);
    glBindTexture(GL_TEXTURE_2D, 0);

    glUseProgram(DeferredLighting);
    glUniformMatrix4fv(DeferredLighting.UniformLocations[0], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
    glUseProgram(0);
```

```
	glUseProgram(FXAA);
	glUniform2f(FXAA.UniformLocations[0], 1.0f / (float)Width, 1.0f / (float)Height);
	glUseProgram(0);
}

void COpenGLRenderer::Destroy()
{
	for(int i = 0; i < 3; i++)
	{
		Texture[i].Destroy();
	}

	Preprocess.Destroy();
	DeferredLighting.Destroy();
	FXAA.Destroy();

	glDeleteBuffers(1, &VBO);

	glDeleteTextures(2, ColorBuffers);
	glDeleteTextures(1, &NormalBuffer);
	glDeleteTextures(1, &DepthBuffer);

	if(GLEW_EXT_framebuffer_object)
	{
		glDeleteFramebuffersEXT(1, &FBO);
	}
}

void COpenGLRenderer::InitArrayBuffers()
{
	CBuffer buffer;

	vec3 m;

	// cubes

	m = vec3( 0.0f, 0.5f, 0.0f);

	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.0f, 1.5f,-4.5f);
```

```
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    m = vec3(-0.625f, 0.5f,-4.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
```

```
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

m = vec3( 0.625f, 0.5f,-4.5f);

buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    // floor

    buffer.AddData(&vec2( 0.0f,  0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,  5.0f), 12);
    buffer.AddData(&vec2(10.0f,  0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,  5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, -5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, -5.0f), 12);

    // walls

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,  0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,  0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,  0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,  0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,  0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,  0.0f), 12);
```

```
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,  3.0f),  8);  buffer.AddData(&vec3(  1.0f,  0.0f,    0.0f),  12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(  0.0f,  3.0f),  8);  buffer.AddData(&vec3(  1.0f,  0.0f,    0.0f),  12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2(  0.0f,  0.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,  -1.0f),  12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f,  0.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,  -1.0f),  12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f,  3.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,  -1.0f),  12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2(  0.0f,  3.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,  -1.0f),  12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2(  0.0f,  0.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,    1.0f),  12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,  0.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,    1.0f),  12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,  3.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,    1.0f),  12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(  0.0f,  3.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,    1.0f),  12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);

    // pillars

    m = vec3(-2.5f, 0.0f,-2.5f);

    buffer.AddData(&vec2(  0.0f,  0.0f),  8);  buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2(  1.0f,  0.0f),  8);  buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2(  1.0f,  3.0f),  8);  buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2(  0.0f,  3.0f),  8);  buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2(  0.0f,  0.0f),  8);  buffer.AddData(&vec3(  1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2(  1.0f,  0.0f),  8);  buffer.AddData(&vec3(  1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2(  1.0f,  3.0f),  8);  buffer.AddData(&vec3(  1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2(  0.0f,  3.0f),  8);  buffer.AddData(&vec3(  1.0f,  0.0f,  0.0f),  12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2(  0.0f,  0.0f),  8);  buffer.AddData(&vec3(  0.0f,  0.0f,-1.0f),  12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f,-2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    m = vec3( 2.5f, 0.0f, 2.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    m = vec3(-2.5f, 0.0f, 2.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  0.0f), 8); buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  3.0f), 8); buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  3.0f), 8); buffer.AddData(&vec3(-1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  0.0f), 8); buffer.AddData(&vec3( 1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  0.0f), 8); buffer.AddData(&vec3( 1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  3.0f), 8); buffer.AddData(&vec3( 1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  3.0f), 8); buffer.AddData(&vec3( 1.0f,  0.0f,  0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  0.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  0.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  3.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  3.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  0.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,  1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  0.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,  1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f,  3.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,  1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f,  3.0f), 8); buffer.AddData(&vec3( 0.0f,  0.0f,  1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    // ceiling

    buffer.AddData(&vec2( 0.0f,   0.0f), 8); buffer.AddData(&vec3(0.0f,  -1.0f,  0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,   0.0f), 8); buffer.AddData(&vec3(0.0f,  -1.0f,  0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,  10.0f), 8); buffer.AddData(&vec3(0.0f,  -1.0f,  0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f,  10.0f), 8); buffer.AddData(&vec3(0.0f,  -1.0f,  0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);

    // rotating object
```

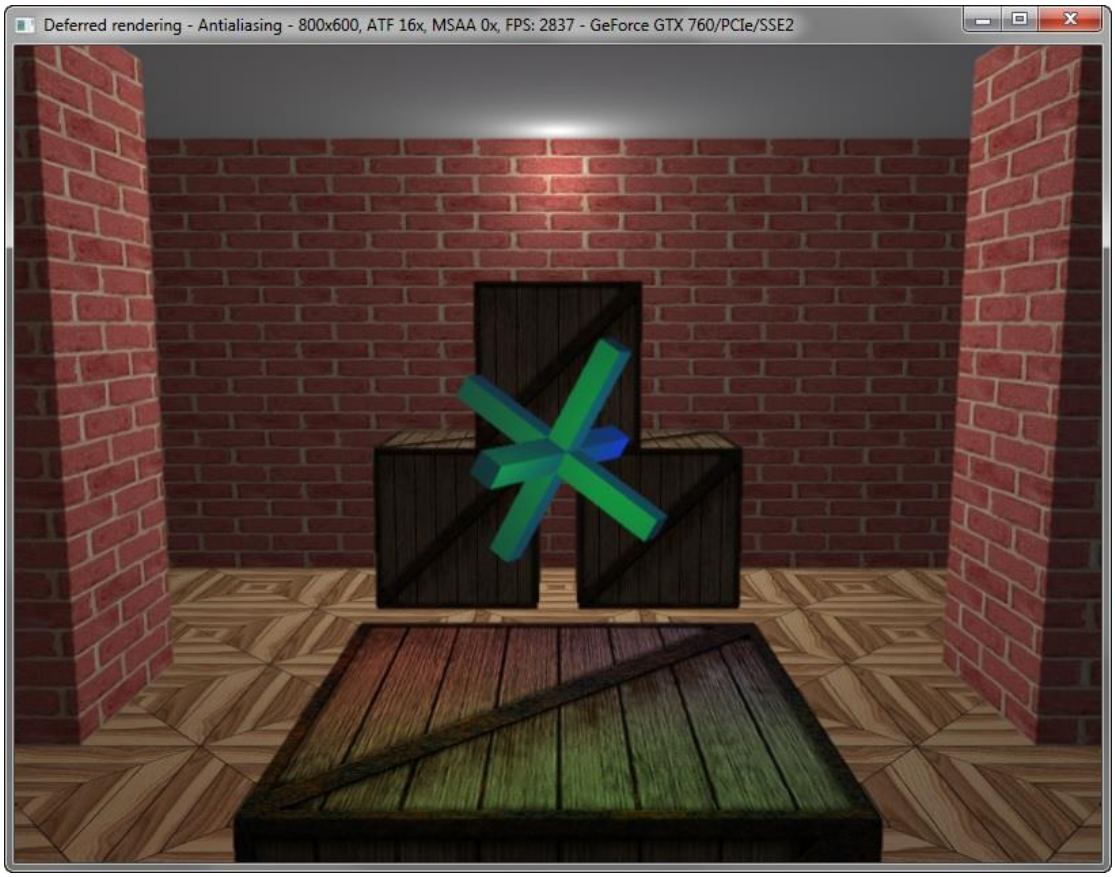
```
m = vec3(0.0f);

buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
```

```
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
```

```
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12); buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12); buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);

buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12); buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);

    glBindBuffer(GL_ARRAY_BUFFER, VBO);
    glBufferData(GL_ARRAY_BUFFER, buffer.GetDataSize(), buffer.GetData(),
GL_STATIC_DRAW);
    glBindBuffer(GL_ARRAY_BUFFER, 0);

    buffer.Empty();
}
```

第三章　延迟着色抗锯齿



第一节 Shader Source

<1>antialiasing.vs

```
#version 120
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

Antialiasing.fs

```
#version 120

uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer;
uniform vec2 PixelSize;

vec2 Offsets[8] = vec2[](
	vec2(-1.0,  1.0),
	vec2( 0.0,  1.0),
	vec2( 1.0,  1.0),
	vec2( 1.0,  0.0),
	vec2( 1.0, -1.0),
	vec2( 0.0, -1.0),
	vec2(-1.0, -1.0),
```

```
        vec2(-1.0,    0.0)
);

void main()
{
        vec2 TexCoords[8];

        for(int i = 0; i < 8; i++)
        {
                TexCoords[i] = gl_TexCoord[0].st + Offsets[i] * PixelSize;
        }

        float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

        vec4 Depths1, Depths2;

        Depths1.x = texture2D(DepthBuffer, TexCoords[0]).r;
        Depths1.y = texture2D(DepthBuffer, TexCoords[1]).r;
        Depths1.z = texture2D(DepthBuffer, TexCoords[2]).r;
        Depths1.w = texture2D(DepthBuffer, TexCoords[3]).r;
        Depths2.x = texture2D(DepthBuffer, TexCoords[4]).r;
        Depths2.y = texture2D(DepthBuffer, TexCoords[5]).r;
        Depths2.z = texture2D(DepthBuffer, TexCoords[6]).r;
        Depths2.w = texture2D(DepthBuffer, TexCoords[7]).r;
        vec4 DepthDeltas1 = abs(Depths1 - Depth);
        vec4 DepthDeltas2 = abs(Depth - Depths2);
        vec4 MinDepthDeltas = max(min(DepthDeltas1, DepthDeltas2), 0.00001);
        vec4 MaxDepthDeltas = max(DepthDeltas1, DepthDeltas2);
        vec4 DepthResults = step(MinDepthDeltas * 25.0, MaxDepthDeltas);
        vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 - 1.0);
        vec4 Dots1, Dots2;
        Dots1.x = dot(normalize(texture2D(NormalBuffer, TexCoords[0]).rgb * 2.0 - 1.0),
Normal);
        Dots1.y = dot(normalize(texture2D(NormalBuffer, TexCoords[1]).rgb * 2.0 - 1.0),
Normal);
        Dots1.z = dot(normalize(texture2D(NormalBuffer, TexCoords[2]).rgb * 2.0 - 1.0),
Normal);
        Dots1.w = dot(normalize(texture2D(NormalBuffer, TexCoords[3]).rgb * 2.0 - 1.0),
Normal);
        Dots2.x = dot(normalize(texture2D(NormalBuffer, TexCoords[4]).rgb * 2.0 - 1.0),
Normal);
        Dots2.y = dot(normalize(texture2D(NormalBuffer, TexCoords[5]).rgb * 2.0 - 1.0),
Normal);
        Dots2.z = dot(normalize(texture2D(NormalBuffer, TexCoords[6]).rgb * 2.0 - 1.0),
```

```
Normal);
    Dots2.w = dot(normalize(texture2D(NormalBuffer, TexCoords[7]).rgb * 2.0 - 1.0),
Normal);

    vec4 DotDeltas = abs(Dots1 - Dots2);
    vec4 NormalResults = step(0.4, DotDeltas);

    vec4 Results = max(NormalResults, DepthResults);
    float EdgeWeight = (Results.x + Results.y + Results.z + Results.w) * 0.25;
    if(EdgeWeight > 0.0)
    {
        vec3 Color = texture2D(ColorBuffer, gl_TexCoord[0].st).rgb;
        vec3 ColorsSum = vec3(0.0);
        for(int i = 0; i < 8; i++)
        {
            ColorsSum += texture2D(ColorBuffer, TexCoords[i]).rgb;
        }
        gl_FragColor = vec4(mix(Color, ColorsSum * 0.125, EdgeWeight), 1.0);
    }
    else
    {
        gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);
    }
}
```

<2>deferredlighting.vs

```
#version 120
void main()
{
    gl_TexCoord[0] = gl_Vertex;
    gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

Deferredlighting.fs

```
#version 120

uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer;
uniform mat4x4 ProjectionBiasMatrixInverse;

void main()
{
    gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);

    float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

    if(Depth < 1.0)
```

```
    {
        vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
1.0);

        vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
        Position /= Position.w;

        vec3 Light = vec3(0.0);

        for(int i = 0; i < 4; i++)
        {
            vec3 LightDirection = gl_LightSource[i].position.xyz - Position.xyz;

            float LightDistance2 = dot(LightDirection, LightDirection);
            float LightDistance = sqrt(LightDistance2);

            LightDirection /= LightDistance;

            float NdotLD = max(dot(Normal, LightDirection), 0.0);

            float Attenuation = gl_LightSource[i].constantAttenuation;

            Attenuation += gl_LightSource[i].linearAttenuation * LightDistance;
            Attenuation += gl_LightSource[i].quadraticAttenuation * LightDistance2;

            Light += (gl_LightSource[i].ambient.rgb + gl_LightSource[i].diffuse.rgb *
NdotLD) / Attenuation;
        }

        gl_FragColor.rgb *= Light;
    }
}
```

<3> preprocess.vs

```
#version 120
varying vec3 Normal;
void main()
{
    gl_FrontColor = gl_Color;
    gl_TexCoord[0] = gl_MultiTexCoord0;
    Normal = gl_NormalMatrix * gl_Normal;
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

Preprocess.fs

```
#version 120
```

```
uniform sampler2D Texture;
uniform bool Texturing;
varying vec3 Normal;
void main()
{
	gl_FragData[0] = gl_Color;
	if(Texturing) gl_FragData[0] *= texture2D(Texture, gl_TexCoord[0].st);
	gl_FragData[1] = vec4(normalize(Normal) * 0.5 + 0.5, 1.0);
}
```

第二节 Source Code Header

```
class COpenGLRenderer
{
protected:
	int Width, Height;
	mat3x3 NormalMatrix;
	mat4x4 ModelMatrix, ViewMatrix, ProjectionMatrix, ProjectionBiasMatrixInverse;

protected:
	CTexture Texture[3];
	CShaderProgram Preprocess, DeferredLighting, Antialiasing;
	GLuint ColorBuffers[2], NormalBuffer, DepthBuffer;
	GLuint VBO, FBO;

public:
	bool CalculateAntialiasing, Pause;
	vec3 LightColors[4], LightPositions[4];

public:
	CString Text;

public:
	COpenGLRenderer();
	~COpenGLRenderer();

	bool Init();
	void Render(float FrameTime);
	void Resize(int Width, int Height);
	void Destroy();

protected:
	void InitArrayBuffers();
};
```

第三节 Source Code Cpp

```
COpenGLRenderer::COpenGLRenderer()
{
	CalculateAntialiasing = true;

	Pause = false;

	Camera.SetViewMatrixPointer(&ViewMatrix);
}

COpenGLRenderer::~COpenGLRenderer()
{
}

bool COpenGLRenderer::Init()
{
	// ------------------------------------------------------------------------
-------------------------------------------

	bool Error = false;

	// ------------------------------------------------------------------------
-------------------------------------------

	if(!GLEW_ARB_texture_non_power_of_two)
	{
		ErrorLog.Append("GL_ARB_texture_non_power_of_two not supported!\r\n");
		Error = true;
	}

	if(!GLEW_ARB_depth_texture)
	{
		ErrorLog.Append("GLEW_ARB_depth_texture not supported!\r\n");
		Error = true;
	}

	if(!GLEW_EXT_framebuffer_object)
	{
		ErrorLog.Append("GL_EXT_framebuffer_object not supported!\r\n");
		Error = true;
	}

	// ------------------------------------------------------------------------
-------------------------------------------
```

```
	char *TextureFileName[] = {"cube.jpg", "floor.jpg", "wall.jpg"};

	for(int i = 0; i < 3; i++)
	{
		Error |= !Texture[i].LoadTexture2D(TextureFileName[i]);
	}

	// ------------------------------------------------------------------------
-------------------------------------------

	Error |= !Preprocess.Load("preprocess.vs", "preprocess.fs");
	Error |= !DeferredLighting.Load("deferredlighting.vs", "deferredlighting.fs");
	Error |= !Antialiasing.Load("antialiasing.vs", "antialiasing.fs");

	// ------------------------------------------------------------------------
-------------------------------------------

	if(Error)
	{
		return false;
	}

	// ------------------------------------------------------------------------
-------------------------------------------

	Preprocess.UniformLocations = new GLuint[1];
	Preprocess.UniformLocations[0] = glGetUniformLocation(Preprocess, "Texturing");

	DeferredLighting.UniformLocations = new GLuint[1];
	DeferredLighting.UniformLocations[0] = glGetUniformLocation(DeferredLighting,
"ProjectionBiasMatrixInverse");

	Antialiasing.UniformLocations = new GLuint[1];
	Antialiasing.UniformLocations[0] = glGetUniformLocation(Antialiasing, "PixelSize");

	// ------------------------------------------------------------------------
-------------------------------------------

	glUseProgram(DeferredLighting);
	glUniform1i(glGetUniformLocation(DeferredLighting, "ColorBuffer"), 0);
	glUniform1i(glGetUniformLocation(DeferredLighting, "NormalBuffer"), 1);
	glUniform1i(glGetUniformLocation(DeferredLighting, "DepthBuffer"), 2);
	glUseProgram(0);
```

```
    glUseProgram(Antialiasing);
    glUniform1i(glGetUniformLocation(Antialiasing, "ColorBuffer"), 0);
    glUniform1i(glGetUniformLocation(Antialiasing, "NormalBuffer"), 1);
    glUniform1i(glGetUniformLocation(Antialiasing, "DepthBuffer"), 2);
    glUseProgram(0);

    // ------------------------------------------------------------------------
-----------------------------------------------

    glGenTextures(2, ColorBuffers);
    glGenTextures(1, &NormalBuffer);
    glGenTextures(1, &DepthBuffer);

    // ------------------------------------------------------------------------
-----------------------------------------------

    glGenBuffers(1, &VBO);

    InitArrayBuffers();

    // ------------------------------------------------------------------------
-----------------------------------------------

    glGenFramebuffersEXT(1, &FBO);

    // ------------------------------------------------------------------------
-----------------------------------------------

    LightColors[0] = vec3(1.0f, 0.0f, 0.0f);
    LightPositions[0] = vec3(0.0f, 1.5f, 0.33f);
    LightColors[1] = vec3(0.0f, 1.0f, 0.0f);
    LightPositions[1] = rotate(LightPositions[0], 120.0f, vec3(0.0f, 1.0f, 0.0f));
    LightColors[2] = vec3(0.0f, 0.0f, 1.0f);
    LightPositions[2] = rotate(LightPositions[1], 120.0f, vec3(0.0f, 1.0f, 0.0f));
    LightColors[3] = vec3(1.0f, 1.0f, 1.0f);
    LightPositions[3] = vec3(0.0f, 2.75f, -4.75f);

    for(int i = 0; i < 3; i++)
    {
        glLightfv(GL_LIGHT0 + i, GL_AMBIENT, &vec4(LightColors[i] * 0.125f, 1.0f));
        glLightfv(GL_LIGHT0 + i, GL_DIFFUSE, &vec4(LightColors[i] * 0.875f, 1.0f));
        glLightf(GL_LIGHT0 + i, GL_LINEAR_ATTENUATION, 1.0f);
        glLightf(GL_LIGHT0 + i, GL_QUADRATIC_ATTENUATION, 1.0f);
    }
```

```
	glLightfv(GL_LIGHT3, GL_AMBIENT, &vec4(LightColors[3] * 0.25f, 1.0f));
	glLightfv(GL_LIGHT3, GL_DIFFUSE, &vec4(LightColors[3] * 0.75f, 1.0f));
	glLightf(GL_LIGHT3, GL_LINEAR_ATTENUATION, 1.0f / 32.0f);
	glLightf(GL_LIGHT3, GL_QUADRATIC_ATTENUATION, 1.0f / 64.0f);

	// ------------------------------------------------------------------------
-----------------------------------------------

	Camera.Look(vec3(0.0f, 1.75f, 1.875f), vec3(0.0f, 1.5f, 0.0f));

	// ------------------------------------------------------------------------
-----------------------------------------------

	return true;

	// ------------------------------------------------------------------------
-----------------------------------------------
}

void COpenGLRenderer::Render(float FrameTime)
{
	// ------------------------------------------------------------------------
-----------------------------------------------

	GLenum Buffers[] = {GL_COLOR_ATTACHMENT0_EXT,
GL_COLOR_ATTACHMENT1_EXT};

	// render scene to textures ---------------------------------------------
-----------------------------------------

	glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
	glDrawBuffers(2, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT0_EXT,
GL_TEXTURE_2D, ColorBuffers[0], 0);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT1_EXT,
GL_TEXTURE_2D, NormalBuffer, 0);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_DEPTH_ATTACHMENT_EXT,
GL_TEXTURE_2D, DepthBuffer, 0);

	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

	glMatrixMode(GL_MODELVIEW);
	glLoadMatrixf(&ViewMatrix);
```

```
glEnable(GL_DEPTH_TEST);
glEnable(GL_CULL_FACE);

glBindBuffer(GL_ARRAY_BUFFER, VBO);

glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(2, GL_FLOAT, 32, (void*)0);

glEnableClientState(GL_NORMAL_ARRAY);
glNormalPointer(GL_FLOAT, 32, (void*)8);

glEnableClientState(GL_VERTEX_ARRAY);
glVertexPointer(3, GL_FLOAT, 32, (void*)20);

glUseProgram(Preprocess);

glUniform1i(Preprocess.UniformLocations[0], true);

glColor3f(1.0f, 1.0f, 1.0f);

glBindTexture(GL_TEXTURE_2D, Texture[0]);
glDrawArrays(GL_QUADS, 0, 96);

glBindTexture(GL_TEXTURE_2D, Texture[1]);
glDrawArrays(GL_QUADS, 96, 4);

glBindTexture(GL_TEXTURE_2D, Texture[2]);
glDrawArrays(GL_QUADS, 100, 80);

glBindTexture(GL_TEXTURE_2D, 0);

glUniform1i(Preprocess.UniformLocations[0], false);

glDrawArrays(GL_QUADS, 180, 4);

glMultMatrixf(&ModelMatrix);
glColor3f(0.33f, 0.66f, 1.0f);
glDrawArrays(GL_QUADS, 184, 72);

glUseProgram(0);

glDisableClientState(GL_VERTEX_ARRAY);
glDisableClientState(GL_NORMAL_ARRAY);
```

```
        glDisableClientState(GL_TEXTURE_COORD_ARRAY);

        glBindBuffer(GL_ARRAY_BUFFER, 0);

        glDisable(GL_CULL_FACE);
        glDisable(GL_DEPTH_TEST);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

        // set lights positions ------------------------------------------------------
------------------------------------------------------------

        glMatrixMode(GL_MODELVIEW);
        glLoadMatrixf(&ViewMatrix);

        for(int i = 0; i < 4; i++)
        {
            glLightfv(GL_LIGHT0 + i, GL_POSITION, &vec4(LightPositions[i], 1.0f));
        }

        // calculate lighting --------------------------------------------------------
-----------------------------------------

        if(CalculateAntialiasing)
        {
            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
            glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, ColorBuffers[1], 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);
        }

        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffers[0]);
        glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
        glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
        glUseProgram(DeferredLighting);
        glBegin(GL_QUADS);
            glVertex2f(0.0f, 0.0f);
            glVertex2f(1.0f, 0.0f);
            glVertex2f(1.0f, 1.0f);
            glVertex2f(0.0f, 1.0f);
```

```
    glEnd();
    glUseProgram(0);
    glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
    glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
    glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

    if(CalculateAntialiasing)
    {
        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);
    }

    // calculate antialiasing ----------------------------------------------------
----------------------------------------

    if(CalculateAntialiasing)
    {
        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffers[1]);
        glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
        glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
        glUseProgram(Antialiasing);
        glBegin(GL_QUADS);
            glVertex2f(0.0f, 0.0f);
            glVertex2f(1.0f, 0.0f);
            glVertex2f(1.0f, 1.0f);
            glVertex2f(0.0f, 1.0f);
        glEnd();
        glUseProgram(0);
        glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
        glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);
    }

    // rotate object and lights --------------------------------------------------
----------------------------------------

    if(!Pause)
    {
        static float a = 0.0f;

        ModelMatrix = translate(0.0f, 1.5f, 0.0f) * rotate(a, vec3(0.0f, 1.0f, 0.0f)) * rotate(a,
vec3(1.0f, 0.0f, 0.0f));

        a += 22.5f * FrameTime;
```

```
            for(int i = 0; i < 3; i++)
            {
                LightPositions[i] = rotate(LightPositions[i], -180.0f * FrameTime, vec3(0.0f, 1.0f,
0.0f));
            }
        }

        // ------------------------------------------------------------------------
----------------------------------------------
}

void COpenGLRenderer::Resize(int Width, int Height)
{
        this->Width = Width;
        this->Height = Height;

        glViewport(0, 0, Width, Height);

        ProjectionMatrix = perspective(45.0f, (float)Width / (float)Height, 0.125f, 512.0f);

        glMatrixMode(GL_PROJECTION);
        glLoadMatrixf(&ProjectionMatrix);

        ProjectionBiasMatrixInverse = inverse(ProjectionMatrix) * BiasMatrixInverse;

        for(int i = 0; i < 2; i++)
        {
            glBindTexture(GL_TEXTURE_2D, ColorBuffers[i]);
            glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
            glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
            glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
            glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
            glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
            glBindTexture(GL_TEXTURE_2D, 0);
        }

        glBindTexture(GL_TEXTURE_2D, NormalBuffer);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
```

```
	glBindTexture(GL_TEXTURE_2D, 0);

	glBindTexture(GL_TEXTURE_2D, DepthBuffer);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
	glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT24, Width, Height, 0,
GL_DEPTH_COMPONENT, GL_FLOAT, NULL);
	glBindTexture(GL_TEXTURE_2D, 0);

	glUseProgram(DeferredLighting);
	glUniformMatrix4fv(DeferredLighting.UniformLocations[0], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
	glUseProgram(0);

	glUseProgram(Antialiasing);
	glUniform2f(Antialiasing.UniformLocations[0], 1.0f / (float)Width, 1.0f / (float)Height);
	glUseProgram(0);
}

void COpenGLRenderer::Destroy()
{
	for(int i = 0; i < 3; i++)
	{
		Texture[i].Destroy();
	}

	Preprocess.Destroy();
	DeferredLighting.Destroy();
	Antialiasing.Destroy();

	glDeleteBuffers(1, &VBO);

	glDeleteTextures(2, ColorBuffers);
	glDeleteTextures(1, &NormalBuffer);
	glDeleteTextures(1, &DepthBuffer);

	if(GLEW_EXT_framebuffer_object)
	{
		glDeleteFramebuffersEXT(1, &FBO);
	}
}
```

```
void COpenGLRenderer::InitArrayBuffers()
{
	CBuffer buffer;

	vec3 m;

	// cubes

	m = vec3( 0.0f, 0.5f, 0.0f);

	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.0f, 1.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    m = vec3(-0.625f, 0.5f,-4.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.625f, 0.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    // floor

    buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,    5.0f), 12);
    buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,    5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, -5.0f), 12);
```

```
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, -5.0f), 12);

    // walls

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);

    // pillars

    m = vec3(-2.5f, 0.0f,-2.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f,-2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    m = vec3( 2.5f, 0.0f, 2.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3(-2.5f, 0.0f, 2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
```

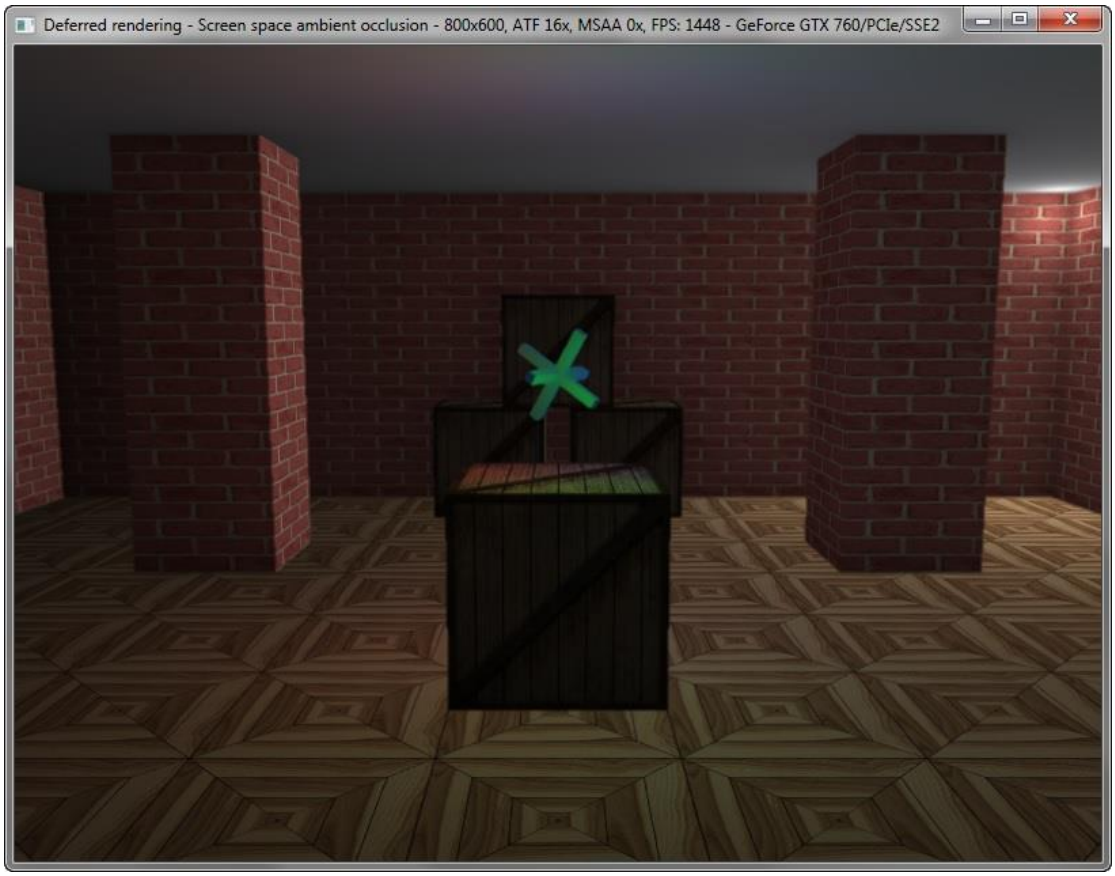
```
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    // ceiling

    buffer.AddData(&vec2( 0.0f,   0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,   0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);

    // rotating object

    m = vec3(0.0f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);

	glBindBuffer(GL_ARRAY_BUFFER, VBO);
	glBufferData(GL_ARRAY_BUFFER, buffer.GetDataSize(), buffer.GetData(),
GL_STATIC_DRAW);
	glBindBuffer(GL_ARRAY_BUFFER, 0);

	buffer.Empty();
}
```

第四章延迟着色屏幕空间环境光遮蔽



第一节 Shader Source

<1>antialiasing.vs

```
#version 120
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

Antialiasing.fs

```
#version 120

uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer;
uniform vec2 PixelSize;

vec2 Offsets[8] = vec2[](
	vec2(-1.0,  1.0),
	vec2( 0.0,  1.0),
	vec2( 1.0,  1.0),
	vec2( 1.0,  0.0),
	vec2( 1.0, -1.0),
```

```
        vec2( 0.0, -1.0),
        vec2(-1.0, -1.0),
        vec2(-1.0,    0.0)
);

void main()
{
        vec2 TexCoords[8];

        for(int i = 0; i < 8; i++)
        {
                TexCoords[i] = gl_TexCoord[0].st + Offsets[i] * PixelSize;
        }

        float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

        vec4 Depths1, Depths2;

        Depths1.x = texture2D(DepthBuffer, TexCoords[0]).r;
        Depths1.y = texture2D(DepthBuffer, TexCoords[1]).r;
        Depths1.z = texture2D(DepthBuffer, TexCoords[2]).r;
        Depths1.w = texture2D(DepthBuffer, TexCoords[3]).r;
        Depths2.x = texture2D(DepthBuffer, TexCoords[4]).r;
        Depths2.y = texture2D(DepthBuffer, TexCoords[5]).r;
        Depths2.z = texture2D(DepthBuffer, TexCoords[6]).r;
        Depths2.w = texture2D(DepthBuffer, TexCoords[7]).r;

        vec4 DepthDeltas1 = abs(Depths1 - Depth);
        vec4 DepthDeltas2 = abs(Depth - Depths2);

        vec4 MinDepthDeltas = max(min(DepthDeltas1, DepthDeltas2), 0.00001);
        vec4 MaxDepthDeltas = max(DepthDeltas1, DepthDeltas2);

        vec4 DepthResults = step(MinDepthDeltas * 25.0, MaxDepthDeltas);

        vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 - 1.0);

        vec4 Dots1, Dots2;

        Dots1.x = dot(normalize(texture2D(NormalBuffer, TexCoords[0]).rgb * 2.0 - 1.0),
Normal);
        Dots1.y = dot(normalize(texture2D(NormalBuffer, TexCoords[1]).rgb * 2.0 - 1.0),
Normal);
        Dots1.z = dot(normalize(texture2D(NormalBuffer, TexCoords[2]).rgb * 2.0 - 1.0),
```

```
Normal);
    Dots1.w = dot(normalize(texture2D(NormalBuffer, TexCoords[3]).rgb * 2.0 - 1.0),
Normal);
    Dots2.x = dot(normalize(texture2D(NormalBuffer, TexCoords[4]).rgb * 2.0 - 1.0),
Normal);
    Dots2.y = dot(normalize(texture2D(NormalBuffer, TexCoords[5]).rgb * 2.0 - 1.0),
Normal);
    Dots2.z = dot(normalize(texture2D(NormalBuffer, TexCoords[6]).rgb * 2.0 - 1.0),
Normal);
    Dots2.w = dot(normalize(texture2D(NormalBuffer, TexCoords[7]).rgb * 2.0 - 1.0),
Normal);

    vec4 DotDeltas = abs(Dots1 - Dots2);
    vec4 NormalResults = step(0.4, DotDeltas);
    vec4 Results = max(NormalResults, DepthResults);
    float EdgeWeight = (Results.x + Results.y + Results.z + Results.w) * 0.25;

    if(EdgeWeight > 0.0)
    {
        vec3 Color = texture2D(ColorBuffer, gl_TexCoord[0].st).rgb;
        vec3 ColorsSum = vec3(0.0);
        for(int i = 0; i < 8; i++)
        {
            ColorsSum += texture2D(ColorBuffer, TexCoords[i]).rgb;
        }
        gl_FragColor = vec4(mix(Color, ColorsSum * 0.125, EdgeWeight), 1.0);
    }
    else
    {
        gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);
    }
}
```

<2> deferredlighting.vs

```
#version 120
void main()
{
    gl_TexCoord[0] = gl_Vertex;
    gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

deferredlighting.fs

```
#version 120
uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer, SSAOBuffer;
uniform mat4x4 ProjectionBiasMatrixInverse;
uniform bool CalculateSSAO;
```

```
void main()
{
	gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;
	if(Depth < 1.0)
	{
		vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
1.0);
		vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
		Position /= Position.w;
		float SSAO = CalculateSSAO ? texture2D(SSAOBuffer, gl_TexCoord[0].st).r : 1.0;
		vec3 Light = vec3(0.0);
		for(int i = 0; i < 4; i++)
		{
			vec3 LightDirection = gl_LightSource[i].position.xyz - Position.xyz;

			float LightDistance2 = dot(LightDirection, LightDirection);
			float LightDistance = sqrt(LightDistance2);

			LightDirection /= LightDistance;

			float NdotLD = max(dot(Normal, LightDirection), 0.0);
			float Attenuation = gl_LightSource[i].constantAttenuation;
			Attenuation += gl_LightSource[i].linearAttenuation * LightDistance;
			Attenuation += gl_LightSource[i].quadraticAttenuation * LightDistance2;
			Light += (gl_LightSource[i].ambient.rgb * SSAO + gl_LightSource[i].diffuse.rgb
* NdotLD) / Attenuation;
		}

		gl_FragColor.rgb *= Light;
	}
}
```

<3> preprocess.vs

```
#version 120
varying vec3 Normal;

void main()
{
	gl_FrontColor = gl_Color;
	gl_TexCoord[0] = gl_MultiTexCoord0;
	Normal = gl_NormalMatrix * gl_Normal;
	gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

Preprocess.fs

```
#version 120
uniform sampler2D Texture;
uniform bool Texturing;
varying vec3 Normal;
void main()
{
	gl_FragData[0] = gl_Color;
	if(Texturing) gl_FragData[0] *= texture2D(Texture, gl_TexCoord[0].st);
	gl_FragData[1] = vec4(normalize(Normal) * 0.5 + 0.5, 1.0);
}
```

<4>SSAO.vs

```
#version 120
uniform vec2 Scale;
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_TexCoord[1] = vec4(gl_Vertex.xy * Scale, gl_Vertex.zw);
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

SSAO.fs

```
#version 120
uniform sampler2D NormalBuffer, DepthBuffer, RotationTexture;
uniform mat4x4 ProjectionBiasMatrixInverse;
uniform vec2 Samples[16];
void main()
{
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;
	if(Depth < 1.0)
	{
		vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
1.0);
		vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
		Position.xyz /= Position.w;
		if(dot(Normal, Position.xyz) > 0.0)
		{
			Normal = -Normal;
		}
		vec4 ScaleRotationVector = normalize(texture2D(RotationTexture,
gl_TexCoord[1].st) * 2.0 - 1.0) / length(Position.xyz);

		mat2x2 ScaleRotationMatrix = mat2x2(ScaleRotationVector.xy,
ScaleRotationVector.zw);
		float SSAO = 0.0;
		for(int i = 0; i < 16; i++)
```

```
        {
            vec2 TexCoord = clamp(ScaleRotationMatrix * Samples[i] + gl_TexCoord[0].st,
0.0, 0.999999);
            float depth = texture2D(DepthBuffer, TexCoord).r;
            vec4 position = ProjectionBiasMatrixInverse * vec4(TexCoord, depth, 1.0);
            position.xyz /= position.w;
            vec3 P2P = position.xyz - Position.xyz;
            float Distance2 = dot(P2P, P2P);
            float Weight = 1.0 - Distance2 * 0.25;
            if(Weight > 0.0)
            {
                P2P /= sqrt(Distance2);

                float NdotP2P = dot(Normal, P2P);

                if(NdotP2P > 0.342)
                {
                    SSAO += NdotP2P * Weight;
                }
            }
        }

        gl_FragColor = vec4(vec3(1.0 - SSAO * 0.0625), 1.0);
    }
    else
    {
        gl_FragColor = vec4(vec3(0.0), 1.0);
    }
}
```

<5>SSAOFilter.vs

```
#version 120
void main()
{
    gl_TexCoord[0] = gl_Vertex;
    gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

SSAOFIlterV.fs

```
#version 120

uniform sampler2D SSAOBuffer, DepthBuffer;
uniform float PixelSizeY, fs, fd;

float Offsets[8] = float[](-4.0, -3.0, -2.0, -1.0, 1.0, 2.0, 3.0, 4.0);
float BlurWeights[8] = float[](1.0, 2.0, 3.0, 4.0, 4.0, 3.0, 2.0, 1.0);
```

```
void main()
{
	float BlurWeightsSum = 5.0;

	float SSAO = texture2D(SSAOBuffer, gl_TexCoord[0].st).r * BlurWeightsSum;
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

	float Factor = fs - fd * Depth;

	for(int i = 0; i < 8; i++)
	{
		vec2 TexCoord = vec2(gl_TexCoord[0].s, gl_TexCoord[0].t + Offsets[i] * PixelSizeY);

		float depth = texture2D(DepthBuffer, TexCoord).r;

		if(abs(Depth - depth) < Factor)
		{
			SSAO += texture2D(SSAOBuffer, TexCoord).r * BlurWeights[i];
			BlurWeightsSum += BlurWeights[i];
		}
	}

	gl_FragColor = vec4(vec3(SSAO / BlurWeightsSum), 1.0);
}
```

SSAOFilterH.fs

```
#version 120
uniform sampler2D SSAOBuffer, DepthBuffer;
uniform float PixelSizeX, fs, fd;
float Offsets[8] = float[](-4.0, -3.0, -2.0, -1.0, 1.0, 2.0, 3.0, 4.0);
float BlurWeights[8] = float[](1.0, 2.0, 3.0, 4.0, 4.0, 3.0, 2.0, 1.0);
void main()
{
	float BlurWeightsSum = 5.0;
	float SSAO = texture2D(SSAOBuffer, gl_TexCoord[0].st).r * BlurWeightsSum;
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;
	float Factor = fs - fd * Depth;

	for(int i = 0; i < 8; i++)
	{
		vec2 TexCoord = vec2(gl_TexCoord[0].s + Offsets[i] * PixelSizeX, gl_TexCoord[0].t);
		float depth = texture2D(DepthBuffer, TexCoord).r;
		if(abs(Depth - depth) < Factor)
		{
```

```
                SSAO += texture2D(SSAOBuffer, TexCoord).r * BlurWeights[i];
                BlurWeightsSum += BlurWeights[i];
            }
        }
        gl_FragColor = vec4(vec3(SSAO / BlurWeightsSum), 1.0);
}
```

第二节 Source Code Header

```
class COpenGLRenderer
{
protected:
        int Width, Height;
        mat3x3 NormalMatrix;
        mat4x4 ModelMatrix, ViewMatrix, ProjectionMatrix, ProjectionBiasMatrixInverse;

protected:
        CTexture Texture[3];
        CShaderProgram Preprocess, SSAO, SSAOFilterH, SSAOFilterV, DeferredLighting,
Antialiasing;
        GLuint RotationTexture, ColorBuffers[2], NormalBuffer, DepthBuffer, SSAOBuffers[2];
        GLuint VBO, FBO;

public:
        bool CalculateAntialiasing, CalculateSSAO, ShowSSAO, BlurSSAO, Pause;
        vec3 LightColors[4], LightPositions[4];

public:
        CString Text;

public:
        COpenGLRenderer();
        ~COpenGLRenderer();

        bool Init();
        void Render(float FrameTime);
        void Resize(int Width, int Height);
        void Destroy();

protected:
        void InitArrayBuffers();
};
```

第三节 Source Code Cpp

```
COpenGLRenderer::COpenGLRenderer()
```

```
{
	CalculateAntialiasing = true;
	CalculateSSAO = true;

	ShowSSAO = false;
	BlurSSAO = true;

	Pause = false;

	Camera.SetViewMatrixPointer(&ViewMatrix);
}

COpenGLRenderer::~COpenGLRenderer()
{
}

bool COpenGLRenderer::Init()
{
	// ------------------------------------------------------------------------
-----------------------------------------------

	bool Error = false;

	// ------------------------------------------------------------------------
-----------------------------------------------

	if(!GLEW_ARB_texture_non_power_of_two)
	{
		ErrorLog.Append("GL_ARB_texture_non_power_of_two not supported!\r\n");
		Error = true;
	}

	if(!GLEW_ARB_depth_texture)
	{
		ErrorLog.Append("GLEW_ARB_depth_texture not supported!\r\n");
		Error = true;
	}

	if(!GLEW_EXT_framebuffer_object)
	{
		ErrorLog.Append("GL_EXT_framebuffer_object not supported!\r\n");
		Error = true;
	}
```

```
	// ------------------------------------------------------------------------
------------------------------------------------

	char *TextureFileName[] = {"cube.jpg", "floor.jpg", "wall.jpg"};

	for(int i = 0; i < 3; i++)
	{
		Error |= !Texture[i].LoadTexture2D(TextureFileName[i]);
	}

	// ------------------------------------------------------------------------
------------------------------------------------

	Error |= !Preprocess.Load("preprocess.vs", "preprocess.fs");
	Error |= !SSAO.Load("ssao.vs", "ssao.fs");
	Error |= !SSAOFilterH.Load("ssaofilter.vs", "ssaofilterh.fs");
	Error |= !SSAOFilterV.Load("ssaofilter.vs", "ssaofilterv.fs");
	Error |= !DeferredLighting.Load("deferredlighting.vs", "deferredlighting.fs");
	Error |= !Antialiasing.Load("antialiasing.vs", "antialiasing.fs");

	// ------------------------------------------------------------------------
------------------------------------------------

	if(Error)
	{
		return false;
	}

	// ------------------------------------------------------------------------
------------------------------------------------

	Preprocess.UniformLocations = new GLuint[1];
	Preprocess.UniformLocations[0] = glGetUniformLocation(Preprocess, "Texturing");

	SSAO.UniformLocations = new GLuint[2];
	SSAO.UniformLocations[0] = glGetUniformLocation(SSAO, "Scale");
	SSAO.UniformLocations[1] = glGetUniformLocation(SSAO,
"ProjectionBiasMatrixInverse");

	SSAOFilterH.UniformLocations = new GLuint[1];
	SSAOFilterH.UniformLocations[0] = glGetUniformLocation(SSAOFilterH, "PixelSizeX");

	SSAOFilterV.UniformLocations = new GLuint[1];
	SSAOFilterV.UniformLocations[0] = glGetUniformLocation(SSAOFilterV, "PixelSizeY");
```

```
        DeferredLighting.UniformLocations = new GLuint[2];
        DeferredLighting.UniformLocations[0] = glGetUniformLocation(DeferredLighting,
"ProjectionBiasMatrixInverse");
        DeferredLighting.UniformLocations[1] = glGetUniformLocation(DeferredLighting,
"CalculateSSAO");

        Antialiasing.UniformLocations = new GLuint[1];
        Antialiasing.UniformLocations[0] = glGetUniformLocation(Antialiasing, "PixelSize");

        // ------------------------------------------------------------------------
----------------------------------------------

        glUseProgram(SSAO);
        glUniform1i(glGetUniformLocation(SSAO, "NormalBuffer"), 0);
        glUniform1i(glGetUniformLocation(SSAO, "DepthBuffer"), 1);
        glUniform1i(glGetUniformLocation(SSAO, "RotationTexture"), 2);
        glUseProgram(0);

        float s = 128.0f, e = 131070.0f, fs = 1.0f / s, fe = 1.0f / e, fd = fs - fe;

        glUseProgram(SSAOFilterH);
        glUniform1i(glGetUniformLocation(SSAOFilterH, "SSAOBuffer"), 0);
        glUniform1i(glGetUniformLocation(SSAOFilterH, "DepthBuffer"), 1);
        glUniform1f(glGetUniformLocation(SSAOFilterH, "fs"), fs);
        glUniform1f(glGetUniformLocation(SSAOFilterH, "fd"), fd);
        glUseProgram(0);

        glUseProgram(SSAOFilterV);
        glUniform1i(glGetUniformLocation(SSAOFilterV, "SSAOBuffer"), 0);
        glUniform1i(glGetUniformLocation(SSAOFilterV, "DepthBuffer"), 1);
        glUniform1f(glGetUniformLocation(SSAOFilterV, "fs"), fs);
        glUniform1f(glGetUniformLocation(SSAOFilterV, "fd"), fd);
        glUseProgram(0);

        glUseProgram(DeferredLighting);
        glUniform1i(glGetUniformLocation(DeferredLighting, "ColorBuffer"), 0);
        glUniform1i(glGetUniformLocation(DeferredLighting, "NormalBuffer"), 1);
        glUniform1i(glGetUniformLocation(DeferredLighting, "DepthBuffer"), 2);
        glUniform1i(glGetUniformLocation(DeferredLighting, "SSAOBuffer"), 3);
        glUseProgram(0);

        glUseProgram(Antialiasing);
        glUniform1i(glGetUniformLocation(Antialiasing, "ColorBuffer"), 0);
```

```
	glUniform1i(glGetUniformLocation(Antialiasing, "NormalBuffer"), 1);
	glUniform1i(glGetUniformLocation(Antialiasing, "DepthBuffer"), 2);
	glUseProgram(0);

	// ----------------------------------------------------------------------------
-----------------------------------------------

	srand(GetTickCount());

	vec2 *Samples = new vec2[16];
	float RandomAngle = (float)M_PI_4, Radius = 1.0f;

	for(int i = 0; i < 16; i++)
	{
		Samples[i].x = cos(RandomAngle) * (float)(i + 1) / 16.0f * Radius;
		Samples[i].y = sin(RandomAngle) * (float)(i + 1) / 16.0f * Radius;

		RandomAngle += (float)M_PI_2;

		if(((i + 1) % 4) == 0) RandomAngle += (float)M_PI_4;
	}

	glUseProgram(SSAO);
	glUniform2fv(glGetUniformLocation(SSAO, "Samples"), 16, (float*)Samples);
	glUseProgram(0);

	delete [] Samples;

	// ----------------------------------------------------------------------------
-----------------------------------------------

	vec4 *RotationTextureData = new vec4[64 * 64];

	RandomAngle = (float)rand() / (float)RAND_MAX * (float)M_PI * 2.0f;

	for(int i = 0; i < 64 * 64; i++)
	{
		RotationTextureData[i].x = cos(RandomAngle) * 0.5f + 0.5f;
		RotationTextureData[i].y = sin(RandomAngle) * 0.5f + 0.5f;
		RotationTextureData[i].z = -sin(RandomAngle) * 0.5f + 0.5f;
		RotationTextureData[i].w = cos(RandomAngle) * 0.5f + 0.5f;

		RandomAngle += (float)rand() / (float)RAND_MAX * (float)M_PI * 2.0f;
	}
```

```
        glGenTextures(1, &RotationTexture);
        glBindTexture(GL_TEXTURE_2D, RotationTexture);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, 64, 64, 0, GL_RGBA, GL_FLOAT,
(float*)RotationTextureData);

        delete [] RotationTextureData;

        // ------------------------------------------------------------------------
-----------------------------------------------

        glGenTextures(2, ColorBuffers);
        glGenTextures(1, &NormalBuffer);
        glGenTextures(1, &DepthBuffer);
        glGenTextures(2, SSAOBuffers);

        // ------------------------------------------------------------------------
-----------------------------------------------

        glGenBuffers(1, &VBO);

        InitArrayBuffers();

        // ------------------------------------------------------------------------
-----------------------------------------------

        glGenFramebuffersEXT(1, &FBO);

        // ------------------------------------------------------------------------
-----------------------------------------------

        LightColors[0] = vec3(1.0f, 0.0f, 0.0f);
        LightPositions[0] = vec3(0.0f, 1.5f, 0.33f);
        LightColors[1] = vec3(0.0f, 1.0f, 0.0f);
        LightPositions[1] = rotate(LightPositions[0], 120.0f, vec3(0.0f, 1.0f, 0.0f));
        LightColors[2] = vec3(0.0f, 0.0f, 1.0f);
        LightPositions[2] = rotate(LightPositions[1], 120.0f, vec3(0.0f, 1.0f, 0.0f));
        LightColors[3] = vec3(1.0f, 1.0f, 1.0f);
        LightPositions[3] = vec3(0.0f, 2.75f, -4.75f);

        for(int i = 0; i < 3; i++)
        {
```

```
            glLightfv(GL_LIGHT0 + i, GL_AMBIENT, &vec4(LightColors[i] * 0.125f, 1.0f));
            glLightfv(GL_LIGHT0 + i, GL_DIFFUSE, &vec4(LightColors[i] * 0.875f, 1.0f));
            glLightf(GL_LIGHT0 + i, GL_LINEAR_ATTENUATION, 1.0f);
            glLightf(GL_LIGHT0 + i, GL_QUADRATIC_ATTENUATION, 1.0f);
        }

        glLightfv(GL_LIGHT3, GL_AMBIENT, &vec4(LightColors[3] * 0.25f, 1.0f));
        glLightfv(GL_LIGHT3, GL_DIFFUSE, &vec4(LightColors[3] * 0.75f, 1.0f));
        glLightf(GL_LIGHT3, GL_LINEAR_ATTENUATION, 1.0f / 32.0f);
        glLightf(GL_LIGHT3, GL_QUADRATIC_ATTENUATION, 1.0f / 64.0f);

        // ------------------------------------------------------------------------
-------------------------------------------

        Camera.Look(vec3(0.0f, 1.75f, 1.875f), vec3(0.0f, 1.5f, 0.0f));

        // ------------------------------------------------------------------------
-------------------------------------------

        return true;

        // ------------------------------------------------------------------------
-------------------------------------------
}

void COpenGLRenderer::Render(float FrameTime)
{
        // ------------------------------------------------------------------------
-------------------------------------------

        GLenum Buffers[] = {GL_COLOR_ATTACHMENT0_EXT,
GL_COLOR_ATTACHMENT1_EXT};

        // render scene to textures ---------------------------------------------
-----------------------------------------

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
        glDrawBuffers(2, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT0_EXT,
GL_TEXTURE_2D, ColorBuffers[0], 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT1_EXT,
GL_TEXTURE_2D, NormalBuffer, 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_DEPTH_ATTACHMENT_EXT,
GL_TEXTURE_2D, DepthBuffer, 0);
```

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

glMatrixMode(GL_PROJECTION);
glLoadMatrixf(&ProjectionMatrix);

glMatrixMode(GL_MODELVIEW);
glLoadMatrixf(&ViewMatrix);

glEnable(GL_DEPTH_TEST);
glEnable(GL_CULL_FACE);

glBindBuffer(GL_ARRAY_BUFFER, VBO);

glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(2, GL_FLOAT, 32, (void*)0);

glEnableClientState(GL_NORMAL_ARRAY);
glNormalPointer(GL_FLOAT, 32, (void*)8);

glEnableClientState(GL_VERTEX_ARRAY);
glVertexPointer(3, GL_FLOAT, 32, (void*)20);

glUseProgram(Preprocess);

glUniform1i(Preprocess.UniformLocations[0], true);

glColor3f(1.0f, 1.0f, 1.0f);

glBindTexture(GL_TEXTURE_2D, Texture[0]);
glDrawArrays(GL_QUADS, 0, 96);

glBindTexture(GL_TEXTURE_2D, Texture[1]);
glDrawArrays(GL_QUADS, 96, 4);

glBindTexture(GL_TEXTURE_2D, Texture[2]);
glDrawArrays(GL_QUADS, 100, 80);

glBindTexture(GL_TEXTURE_2D, 0);

glUniform1i(Preprocess.UniformLocations[0], false);

glDrawArrays(GL_QUADS, 180, 4);
```

```
	glMultMatrixf(&ModelMatrix);
	glColor3f(0.33f, 0.66f, 1.0f);
	glDrawArrays(GL_QUADS, 184, 72);

	glUseProgram(0);

	glDisableClientState(GL_VERTEX_ARRAY);
	glDisableClientState(GL_NORMAL_ARRAY);
	glDisableClientState(GL_TEXTURE_COORD_ARRAY);

	glBindBuffer(GL_ARRAY_BUFFER, 0);

	glDisable(GL_CULL_FACE);
	glDisable(GL_DEPTH_TEST);

	glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

	// calculate screen space ambient occlusion ----------------------------------------
----------------------------------------

	if(CalculateSSAO || ShowSSAO)
	{
		glViewport(0, 0, Width / 2, Height / 2);

		glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
		glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
		glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, SSAOBuffers[0], 0);
		glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
		glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);

		glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
		glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
		glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, RotationTexture);
		glUseProgram(SSAO);
		glBegin(GL_QUADS);
			glVertex2f(0.0f, 0.0f);
			glVertex2f(1.0f, 0.0f);
			glVertex2f(1.0f, 1.0f);
			glVertex2f(0.0f, 1.0f);
		glEnd();
		glUseProgram(0);
```

```
        glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
        glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

        // blur filter with edge detection ----------------------------------------
----------------------------------------

        if(BlurSSAO)
        {
            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
            glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, SSAOBuffers[1], 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);

            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D,
SSAOBuffers[0]);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
            glUseProgram(SSAOFilterH);
            glBegin(GL_QUADS);
                glVertex2f(0.0f, 0.0f);
                glVertex2f(1.0f, 0.0f);
                glVertex2f(1.0f, 1.0f);
                glVertex2f(0.0f, 1.0f);
            glEnd();
            glUseProgram(0);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, SSAOBuffers[0], 0);

            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D,
SSAOBuffers[1]);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
            glUseProgram(SSAOFilterV);
            glBegin(GL_QUADS);
                glVertex2f(0.0f, 0.0f);
                glVertex2f(1.0f, 0.0f);
```

```
                glVertex2f(1.0f, 1.0f);
                glVertex2f(0.0f, 1.0f);
            glEnd();
            glUseProgram(0);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);
        }

        glViewport(0, 0, Width, Height);
    }

    // set lights positions ------------------------------------------------------------
----------------------------------------

    glMatrixMode(GL_MODELVIEW);
    glLoadMatrixf(&ViewMatrix);

    for(int i = 0; i < 4; i++)
    {
        glLightfv(GL_LIGHT0 + i, GL_POSITION, &vec4(LightPositions[i], 1.0f));
    }

    // ---------------------------------------------------------------------------------
----------------------------------------------

    if(ShowSSAO)
    {
        // display SSAO --------------------------------------------------------------
----------------------------------------------

        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

        glColor3f(1.0f, 1.0f, 1.0f);

        glEnable(GL_TEXTURE_2D);
        glBindTexture(GL_TEXTURE_2D, SSAOBuffers[0]);
        glBegin(GL_QUADS);
            glTexCoord2f(0.0f, 0.0f); glVertex2f(-1.0f, -1.0f);
```

```
                glTexCoord2f(1.0f, 0.0f); glVertex2f( 1.0f, -1.0f);
                glTexCoord2f(1.0f, 1.0f); glVertex2f( 1.0f,  1.0f);
                glTexCoord2f(0.0f, 1.0f); glVertex2f(-1.0f,  1.0f);
            glEnd();
            glBindTexture(GL_TEXTURE_2D, 0);
            glDisable(GL_TEXTURE_2D);
        }
        else
        {
            // calculate lighting ---------------------------------------------------
-----------------------------------------

            if(CalculateAntialiasing)
            {
                glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
                glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
                glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, ColorBuffers[1], 0);
                glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
                glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);
            }

            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffers[0]);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
            glActiveTexture(GL_TEXTURE3); glBindTexture(GL_TEXTURE_2D, SSAOBuffers[0]);
            glUseProgram(DeferredLighting);
            glUniform1i(DeferredLighting.UniformLocations[1], CalculateSSAO);
            glBegin(GL_QUADS);
                glVertex2f(0.0f, 0.0f);
                glVertex2f(1.0f, 0.0f);
                glVertex2f(1.0f, 1.0f);
                glVertex2f(0.0f, 1.0f);
            glEnd();
            glUseProgram(0);
            glActiveTexture(GL_TEXTURE3); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

            if(CalculateAntialiasing)
            {
```

```
            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);
        }

        // calculate antialiasing ----------------------------------------------------
-----------------------------------------

        if(CalculateAntialiasing)
        {
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D,
ColorBuffers[1]);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D,
NormalBuffer);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
            glUseProgram(Antialiasing);
            glBegin(GL_QUADS);
                glVertex2f(0.0f, 0.0f);
                glVertex2f(1.0f, 0.0f);
                glVertex2f(1.0f, 1.0f);
                glVertex2f(0.0f, 1.0f);
            glEnd();
            glUseProgram(0);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);
        }
    }

    // rotate object and lights --------------------------------------------------
-----------------------------------------

    if(!Pause)
    {
        static float a = 0.0f;

        ModelMatrix = translate(0.0f, 1.5f, 0.0f) * rotate(a, vec3(0.0f, 1.0f, 0.0f)) * rotate(a,
vec3(1.0f, 0.0f, 0.0f));

        a += 22.5f * FrameTime;

        for(int i = 0; i < 3; i++)
        {
            LightPositions[i] = rotate(LightPositions[i], -180.0f * FrameTime, vec3(0.0f, 1.0f,
0.0f));
        }
```

```
	}

	// ------------------------------------------------------------------------
----------------------------------------------
}

void COpenGLRenderer::Resize(int Width, int Height)
{
	this->Width = Width;
	this->Height = Height;

	glViewport(0, 0, Width, Height);

	ProjectionMatrix = perspective(45.0f, (float)Width / (float)Height, 0.125f, 512.0f);

	ProjectionBiasMatrixInverse = inverse(ProjectionMatrix) * BiasMatrixInverse;

	for(int i = 0; i < 2; i++)
	{
		glBindTexture(GL_TEXTURE_2D, ColorBuffers[i]);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
		glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
		glBindTexture(GL_TEXTURE_2D, 0);
	}

	glBindTexture(GL_TEXTURE_2D, NormalBuffer);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
	glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
	glBindTexture(GL_TEXTURE_2D, 0);

	glBindTexture(GL_TEXTURE_2D, DepthBuffer);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
	glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT24, Width, Height, 0,
```

```
GL_DEPTH_COMPONENT, GL_FLOAT, NULL);
	glBindTexture(GL_TEXTURE_2D, 0);

	for(int i = 0; i < 2; i++)
	{
		glBindTexture(GL_TEXTURE_2D, SSAOBuffers[i]);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
		glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
		glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width / 2, Height / 2, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
		glBindTexture(GL_TEXTURE_2D, 0);
	}

	glUseProgram(SSAO);
	glUniform2f(SSAO.UniformLocations[0], (float)Width / 2.0f / 64.0f, (float)Height / 2.0f /
64.0f);
	glUniformMatrix4fv(SSAO.UniformLocations[1], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
	glUseProgram(0);

	glUseProgram(SSAOFilterH);
	glUniform1f(SSAOFilterH.UniformLocations[0], 2.0f / (float)Width);
	glUseProgram(SSAOFilterV);
	glUniform1f(SSAOFilterV.UniformLocations[0], 2.0f / (float)Height);
	glUseProgram(0);

	glUseProgram(DeferredLighting);
	glUniformMatrix4fv(DeferredLighting.UniformLocations[0], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
	glUseProgram(0);

	glUseProgram(Antialiasing);
	glUniform2f(Antialiasing.UniformLocations[0], 1.0f / (float)Width, 1.0f / (float)Height);
	glUseProgram(0);
}

void COpenGLRenderer::Destroy()
{
	for(int i = 0; i < 3; i++)
	{
		Texture[i].Destroy();
	}
```

```
	Preprocess.Destroy();
	SSAO.Destroy();
	SSAOFilterH.Destroy();
	SSAOFilterV.Destroy();
	DeferredLighting.Destroy();
	Antialiasing.Destroy();

	glDeleteBuffers(1, &VBO);

	glDeleteTextures(1, &RotationTexture);
	glDeleteTextures(2, ColorBuffers);
	glDeleteTextures(1, &NormalBuffer);
	glDeleteTextures(1, &DepthBuffer);
	glDeleteTextures(2, SSAOBuffers);

	if(GLEW_EXT_framebuffer_object)
	{
		glDeleteFramebuffersEXT(1, &FBO);
	}
}

void COpenGLRenderer::InitArrayBuffers()
{
	CBuffer buffer;

	vec3 m;

	// cubes

	m = vec3( 0.0f, 0.5f, 0.0f);

	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.0f, 1.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
```

```
m = vec3(-0.625f, 0.5f,-4.5f);

buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3( 0.625f, 0.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    // floor

    buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,    5.0f), 12);
    buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,    5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, -5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, -5.0f), 12);

    // walls

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
    buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
```

```
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
	buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
	buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
	buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
	buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
	buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
	buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);

	// pillars

	m = vec3(-2.5f, 0.0f,-2.5f);

	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f,-2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
```

```
m = vec3( 2.5f, 0.0f, 2.5f);

buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

m = vec3(-2.5f, 0.0f, 2.5f);

buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    // ceiling

    buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
    buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
    buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);

    // rotating object

    m = vec3(0.0f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
```

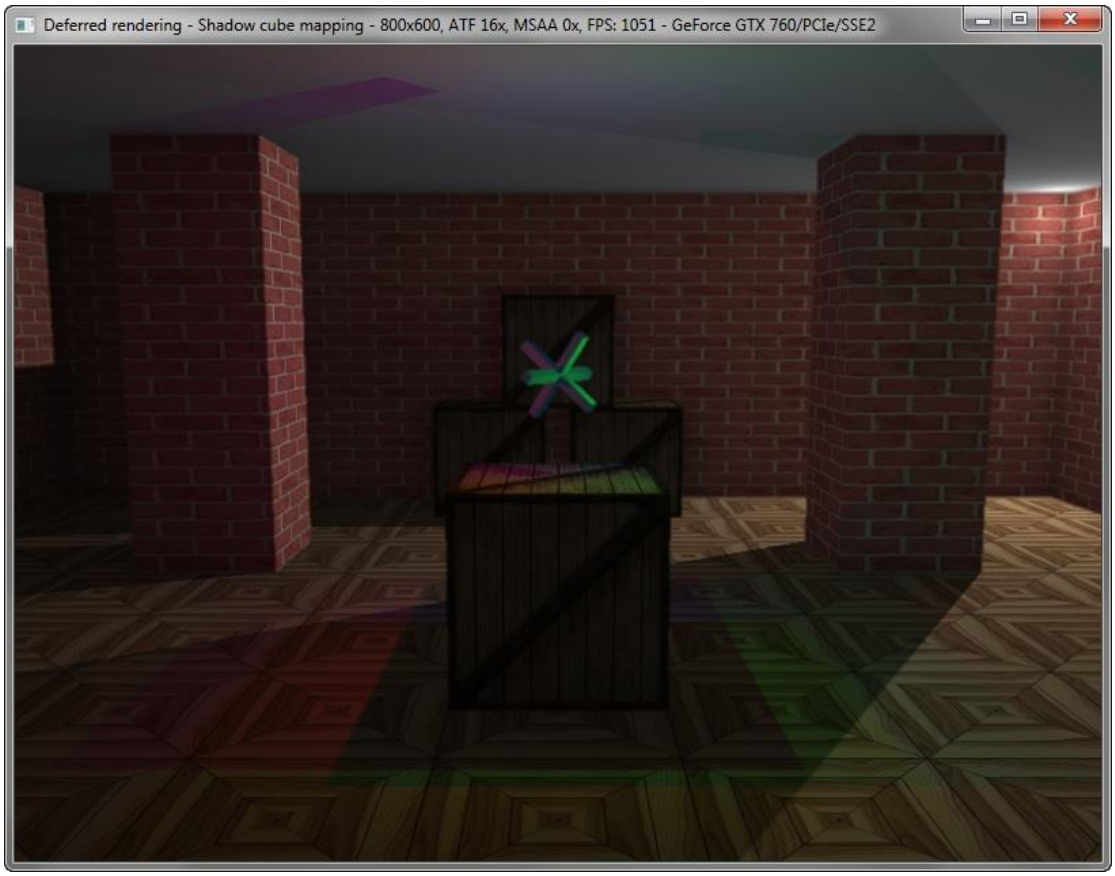
```
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
```

```
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
```

```
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);

        glBindBuffer(GL_ARRAY_BUFFER, VBO);
        glBufferData(GL_ARRAY_BUFFER, buffer.GetDataSize(), buffer.GetData(),
GL_STATIC_DRAW);
        glBindBuffer(GL_ARRAY_BUFFER, 0);

        buffer.Empty();
}
```

第六章　延迟着色万向阴影



第一节 Shader Source

<1>Antialiasing.vs

```
#version 120
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

Antialiasing.fs

```
#version 120

uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer;
uniform vec2 PixelSize;

vec2 Offsets[8] = vec2[](
	vec2(-1.0,  1.0),
	vec2( 0.0,  1.0),
	vec2( 1.0,  1.0),
	vec2( 1.0,  0.0),
	vec2( 1.0, -1.0),
```

```
        vec2( 0.0, -1.0),
        vec2(-1.0, -1.0),
        vec2(-1.0,    0.0)
);

void main()
{
        vec2 TexCoords[8];

        for(int i = 0; i < 8; i++)
        {
                TexCoords[i] = gl_TexCoord[0].st + Offsets[i] * PixelSize;
        }
        float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

        vec4 Depths1, Depths2;
        Depths1.x = texture2D(DepthBuffer, TexCoords[0]).r;
        Depths1.y = texture2D(DepthBuffer, TexCoords[1]).r;
        Depths1.z = texture2D(DepthBuffer, TexCoords[2]).r;
        Depths1.w = texture2D(DepthBuffer, TexCoords[3]).r;
        Depths2.x = texture2D(DepthBuffer, TexCoords[4]).r;
        Depths2.y = texture2D(DepthBuffer, TexCoords[5]).r;
        Depths2.z = texture2D(DepthBuffer, TexCoords[6]).r;
        Depths2.w = texture2D(DepthBuffer, TexCoords[7]).r;

        vec4 DepthDeltas1 = abs(Depths1 - Depth);
        vec4 DepthDeltas2 = abs(Depth - Depths2);
        vec4 MinDepthDeltas = max(min(DepthDeltas1, DepthDeltas2), 0.00001);
        vec4 MaxDepthDeltas = max(DepthDeltas1, DepthDeltas2);
        vec4 DepthResults = step(MinDepthDeltas * 25.0, MaxDepthDeltas);
        vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 - 1.0);
        vec4 Dots1, Dots2;

        Dots1.x = dot(normalize(texture2D(NormalBuffer, TexCoords[0]).rgb * 2.0 - 1.0),
Normal);
        Dots1.y = dot(normalize(texture2D(NormalBuffer, TexCoords[1]).rgb * 2.0 - 1.0),
Normal);
        Dots1.z = dot(normalize(texture2D(NormalBuffer, TexCoords[2]).rgb * 2.0 - 1.0),
Normal);
        Dots1.w = dot(normalize(texture2D(NormalBuffer, TexCoords[3]).rgb * 2.0 - 1.0),
Normal);
        Dots2.x = dot(normalize(texture2D(NormalBuffer, TexCoords[4]).rgb * 2.0 - 1.0),
Normal);
        Dots2.y = dot(normalize(texture2D(NormalBuffer, TexCoords[5]).rgb * 2.0 - 1.0),
```

```
Normal);
	Dots2.z = dot(normalize(texture2D(NormalBuffer, TexCoords[6]).rgb * 2.0 - 1.0),
Normal);
	Dots2.w = dot(normalize(texture2D(NormalBuffer, TexCoords[7]).rgb * 2.0 - 1.0),
Normal);

	vec4 DotDeltas = abs(Dots1 - Dots2);
	vec4 NormalResults = step(0.4, DotDeltas);
	vec4 Results = max(NormalResults, DepthResults);
	float EdgeWeight = (Results.x + Results.y + Results.z + Results.w) * 0.25;

	if(EdgeWeight > 0.0)
	{
		vec3 Color = texture2D(ColorBuffer, gl_TexCoord[0].st).rgb;
		vec3 ColorsSum = vec3(0.0);
		for(int i = 0; i < 8; i++)
		{
			ColorsSum += texture2D(ColorBuffer, TexCoords[i]).rgb;
		}
		gl_FragColor = vec4(mix(Color, ColorsSum * 0.125, EdgeWeight), 1.0);
	}
	else
	{
		gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);
	}
}
```

<2> deferredlighting.vs

```
#version 120
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

deferredlighting.fs

```
#version 120

#extension GL_EXT_texture_array : enable

uniform sampler2D ColorBuffer, NormalBuffer, DepthBuffer, SSAOBuffer;
uniform sampler2DArrayShadow ShadowCubeMaps;
uniform mat4x4 ProjectionBiasMatrixInverse, ViewMatrixInverse, ShadowMatrices[24];
uniform bool CalculateSSAO, CalculateShadows;

void main()
```

```
{
    gl_FragColor = texture2D(ColorBuffer, gl_TexCoord[0].st);

    float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

    if(Depth < 1.0)
    {
        vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
1.0);

        vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
        Position /= Position.w;

        float SSAO = CalculateSSAO ? texture2D(SSAOBuffer, gl_TexCoord[0].st).r : 1.0;

        vec3 Light = vec3(0.0);

        for(int i = 0; i < 4; i++)
        {
            vec3 LightDirection = gl_LightSource[i].position.xyz - Position.xyz;

            float LightDistance2 = dot(LightDirection, LightDirection);
            float LightDistance = sqrt(LightDistance2);

            LightDirection /= LightDistance;

            float NdotLD = max(dot(Normal, LightDirection), 0.0);

            float Attenuation = gl_LightSource[i].constantAttenuation;

            Attenuation += gl_LightSource[i].linearAttenuation * LightDistance;
            Attenuation += gl_LightSource[i].quadraticAttenuation * LightDistance2;

            float Shadow = 1.0;

            if(CalculateShadows)
            {
                LightDirection = (ViewMatrixInverse * vec4(LightDirection, 0.0)).xyz;

                float Axis[6];

                Axis[0] = -LightDirection.x;
                Axis[1] = LightDirection.x;
                Axis[2] = -LightDirection.y;
```

```
                    Axis[3] = LightDirection.y;
                    Axis[4] = -LightDirection.z;
                    Axis[5] = LightDirection.z;

                    int MaxAxisID = 0;

                    for(int ii = 1; ii < 6; ii++)
                    {
                        if(Axis[ii] > Axis[MaxAxisID])
                        {
                            MaxAxisID = ii;
                        }
                    }

                    int Index = i * 6 + MaxAxisID;

                    vec4 ShadowTexCoord = ShadowMatrices[Index] * vec4(Position.xyz, 1.0);
                    ShadowTexCoord.xyz /= ShadowTexCoord.w;
                    ShadowTexCoord.w = ShadowTexCoord.z;
                    ShadowTexCoord.z = float(Index);

                    Shadow = shadow2DArray(ShadowCubeMaps, ShadowTexCoord).r;
                }

                Light += (gl_LightSource[i].ambient.rgb * SSAO + gl_LightSource[i].diffuse.rgb
* NdotLD * Shadow) / Attenuation;
            }

            gl_FragColor.rgb *= Light;
        }
}
```

<3>preprocess.vs

```
#version 120
varying vec3 Normal;
void main()
{
    gl_FrontColor = gl_Color;
    gl_TexCoord[0] = gl_MultiTexCoord0;
    Normal = gl_NormalMatrix * gl_Normal;
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

Preprocess.fs

```
#version 120
uniform sampler2D Texture;
```

```
uniform bool Texturing;
varying vec3 Normal;
void main()
{
	gl_FragData[0] = gl_Color;
	if(Texturing) gl_FragData[0] *= texture2D(Texture, gl_TexCoord[0].st);
	gl_FragData[1] = vec4(normalize(Normal) * 0.5 + 0.5, 1.0);
}
```

<4>SSAO.vs

```
#version 120
uniform vec2 Scale;
void main()
{
	gl_TexCoord[0] = gl_Vertex;
	gl_TexCoord[1] = vec4(gl_Vertex.xy * Scale, gl_Vertex.zw);
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

SSAO.fs

```
#version 120

uniform sampler2D NormalBuffer, DepthBuffer, RotationTexture;
uniform mat4x4 ProjectionBiasMatrixInverse;
uniform vec2 Samples[16];

void main()
{
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

	if(Depth < 1.0)
	{
		vec3 Normal = normalize(texture2D(NormalBuffer, gl_TexCoord[0].st).rgb * 2.0 -
1.0);

		vec4 Position = ProjectionBiasMatrixInverse * vec4(gl_TexCoord[0].st, Depth, 1.0);
		Position.xyz /= Position.w;

		if(dot(Normal, Position.xyz) > 0.0)
		{
			Normal = -Normal;
		}

		vec4 ScaleRotationVector = normalize(texture2D(RotationTexture,
gl_TexCoord[1].st) * 2.0 - 1.0) / length(Position.xyz);
```

```
        mat2x2 ScaleRotationMatrix = mat2x2(ScaleRotationVector.xy,
ScaleRotationVector.zw);

        float SSAO = 0.0;

        for(int i = 0; i < 16; i++)
        {
            vec2 TexCoord = clamp(ScaleRotationMatrix * Samples[i] + gl_TexCoord[0].st,
0.0, 0.999999);

            float depth = texture2D(DepthBuffer, TexCoord).r;

            vec4 position = ProjectionBiasMatrixInverse * vec4(TexCoord, depth, 1.0);
            position.xyz /= position.w;

            vec3 P2P = position.xyz - Position.xyz;

            float Distance2 = dot(P2P, P2P);

            float Weight = 1.0 - Distance2 * 0.25;

            if(Weight > 0.0)
            {
                P2P /= sqrt(Distance2);

                float NdotP2P = dot(Normal, P2P);

                if(NdotP2P > 0.342)
                {
                    SSAO += NdotP2P * Weight;
                }
            }
        }

        gl_FragColor = vec4(vec3(1.0 - SSAO * 0.0625), 1.0);
    }
    else
    {
        gl_FragColor = vec4(vec3(0.0), 1.0);
    }
}
```

<5>SSAOFilter.vs

```
#version 120
void main()
```

```
{
	gl_TexCoord[0] = gl_Vertex;
	gl_Position = gl_Vertex * 2.0 - 1.0;
}
```

SSAOFIlterV.fs

```
#version 120

uniform sampler2D SSAOBuffer, DepthBuffer;
uniform float PixelSizeY, fs, fd;

float Offsets[8] = float[](-4.0, -3.0, -2.0, -1.0, 1.0, 2.0, 3.0, 4.0);
float BlurWeights[8] = float[](1.0, 2.0, 3.0, 4.0, 4.0, 3.0, 2.0, 1.0);

void main()
{
	float BlurWeightsSum = 5.0;

	float SSAO = texture2D(SSAOBuffer, gl_TexCoord[0].st).r * BlurWeightsSum;
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

	float Factor = fs - fd * Depth;

	for(int i = 0; i < 8; i++)
	{
		vec2 TexCoord = vec2(gl_TexCoord[0].s, gl_TexCoord[0].t + Offsets[i] * PixelSizeY);

		float depth = texture2D(DepthBuffer, TexCoord).r;

		if(abs(Depth - depth) < Factor)
		{
			SSAO += texture2D(SSAOBuffer, TexCoord).r * BlurWeights[i];
			BlurWeightsSum += BlurWeights[i];
		}
	}

	gl_FragColor = vec4(vec3(SSAO / BlurWeightsSum), 1.0);
}
```

SSAOFIlterH.fs

```
#version 120

uniform sampler2D SSAOBuffer, DepthBuffer;
uniform float PixelSizeX, fs, fd;

float Offsets[8] = float[](-4.0, -3.0, -2.0, -1.0, 1.0, 2.0, 3.0, 4.0);
```

```
float BlurWeights[8] = float[](1.0, 2.0, 3.0, 4.0, 4.0, 3.0, 2.0, 1.0);

void main()
{
	float BlurWeightsSum = 5.0;

	float SSAO = texture2D(SSAOBuffer, gl_TexCoord[0].st).r * BlurWeightsSum;
	float Depth = texture2D(DepthBuffer, gl_TexCoord[0].st).r;

	float Factor = fs - fd * Depth;

	for(int i = 0; i < 8; i++)
	{
		vec2 TexCoord = vec2(gl_TexCoord[0].s + Offsets[i] * PixelSizeX, gl_TexCoord[0].t);

		float depth = texture2D(DepthBuffer, TexCoord).r;

		if(abs(Depth - depth) < Factor)
		{
			SSAO += texture2D(SSAOBuffer, TexCoord).r * BlurWeights[i];
			BlurWeightsSum += BlurWeights[i];
		}
	}

	gl_FragColor = vec4(vec3(SSAO / BlurWeightsSum), 1.0);
                                                }
```

第二节 Source Code Header

```
#define SHADOW_CUBE_MAP_SIZE 512

class COpenGLRenderer
{
protected:
	int Width, Height;
	mat3x3 NormalMatrix;
	mat4x4 ModelMatrix, ViewMatrix, ViewMatrixInverse, ProjectionMatrix,
ProjectionBiasMatrixInverse, LightViewMatrices[24], LightProjectionMatrix,
ShadowMatrices[24];

protected:
	CTexture Texture[3];
	CShaderProgram Preprocess, SSAO, SSAOFilterH, SSAOFilterV, DeferredLighting,
Antialiasing;
	GLuint RotationTexture, ShadowCubeMaps, ColorBuffers[2], NormalBuffer, DepthBuffer,
```

```
SSAOBuffers[2];
	GLuint VBO, FBO;

public:
	bool CalculateAntialiasing, CalculateSSAO, CalculateShadows, ShowSSAO, BlurSSAO,
Pause;
	vec3 LightColors[4], LightPositions[4];

public:
	CString Text;

public:
	COpenGLRenderer();
	~COpenGLRenderer();

	bool Init();
	void Render(float FrameTime);
	void Resize(int Width, int Height);
	void Destroy();

protected:
	void InitArrayBuffers();
                                                        };
```

第三节 Source Code Cpp

```
COpenGLRenderer::COpenGLRenderer()
{
	CalculateAntialiasing = true;
	CalculateSSAO = true;
	CalculateShadows = true;

	ShowSSAO = false;
	BlurSSAO = true;

	Pause = false;

	Camera.SetViewMatrixPointer(&ViewMatrix, &ViewMatrixInverse);
}

COpenGLRenderer::~COpenGLRenderer()
{
}

bool COpenGLRenderer::Init()
```

```
{
    // ------------------------------------------------------------------------
------------------------------------------

    bool Error = false;

    // ------------------------------------------------------------------------
------------------------------------------

    if(!GLEW_ARB_texture_non_power_of_two)
    {
        ErrorLog.Append("GL_ARB_texture_non_power_of_two not supported!\r\n");
        Error = true;
    }

    if(!GLEW_ARB_depth_texture)
    {
        ErrorLog.Append("GLEW_ARB_depth_texture not supported!\r\n");
        Error = true;
    }

    if(!GLEW_EXT_framebuffer_object)
    {
        ErrorLog.Append("GL_EXT_framebuffer_object not supported!\r\n");
        Error = true;
    }

    if(!GLEW_EXT_texture_array)
    {
        ErrorLog.Append("GL_EXT_texture_array not supported!\r\n");
        Error = true;
    }

    // ------------------------------------------------------------------------
------------------------------------------

    char *TextureFileName[] = {"cube.jpg", "floor.jpg", "wall.jpg"};

    for(int i = 0; i < 3; i++)
    {
        Error |= !Texture[i].LoadTexture2D(TextureFileName[i]);
    }

    // ------------------------------------------------------------------------
```

```
-------------------------------------------

	Error |= !Preprocess.Load("preprocess.vs", "preprocess.fs");
	Error |= !SSAO.Load("ssao.vs", "ssao.fs");
	Error |= !SSAOFilterH.Load("ssaofilter.vs", "ssaofilterh.fs");
	Error |= !SSAOFilterV.Load("ssaofilter.vs", "ssaofilterv.fs");
	Error |= !DeferredLighting.Load("deferredlighting.vs", "deferredlighting.fs");
	Error |= !Antialiasing.Load("antialiasing.vs", "antialiasing.fs");

	// ------------------------------------------------------------------------
-------------------------------------------

	if(Error)
	{
		return false;
	}

	// ------------------------------------------------------------------------
-------------------------------------------

	Preprocess.UniformLocations = new GLuint[1];
	Preprocess.UniformLocations[0] = glGetUniformLocation(Preprocess, "Texturing");

	SSAO.UniformLocations = new GLuint[2];
	SSAO.UniformLocations[0] = glGetUniformLocation(SSAO, "Scale");
	SSAO.UniformLocations[1] = glGetUniformLocation(SSAO,
"ProjectionBiasMatrixInverse");

	SSAOFilterH.UniformLocations = new GLuint[1];
	SSAOFilterH.UniformLocations[0] = glGetUniformLocation(SSAOFilterH, "PixelSizeX");

	SSAOFilterV.UniformLocations = new GLuint[1];
	SSAOFilterV.UniformLocations[0] = glGetUniformLocation(SSAOFilterV, "PixelSizeY");

	DeferredLighting.UniformLocations = new GLuint[5];
	DeferredLighting.UniformLocations[0] = glGetUniformLocation(DeferredLighting,
"ProjectionBiasMatrixInverse");
	DeferredLighting.UniformLocations[1] = glGetUniformLocation(DeferredLighting,
"ViewMatrixInverse");
	DeferredLighting.UniformLocations[2] = glGetUniformLocation(DeferredLighting,
"ShadowMatrices");
	DeferredLighting.UniformLocations[3] = glGetUniformLocation(DeferredLighting,
"CalculateSSAO");
	DeferredLighting.UniformLocations[4] = glGetUniformLocation(DeferredLighting,
```

```
"CalculateShadows");

        Antialiasing.UniformLocations = new GLuint[1];
        Antialiasing.UniformLocations[0] = glGetUniformLocation(Antialiasing, "PixelSize");

        // ------------------------------------------------------------------------
------------------------------------------

        glUseProgram(SSAO);
        glUniform1i(glGetUniformLocation(SSAO, "NormalBuffer"), 0);
        glUniform1i(glGetUniformLocation(SSAO, "DepthBuffer"), 1);
        glUniform1i(glGetUniformLocation(SSAO, "RotationTexture"), 2);
        glUseProgram(0);

        float s = 128.0f, e = 131070.0f, fs = 1.0f / s, fe = 1.0f / e, fd = fs - fe;

        glUseProgram(SSAOFilterH);
        glUniform1i(glGetUniformLocation(SSAOFilterH, "SSAOBuffer"), 0);
        glUniform1i(glGetUniformLocation(SSAOFilterH, "DepthBuffer"), 1);
        glUniform1f(glGetUniformLocation(SSAOFilterH, "fs"), fs);
        glUniform1f(glGetUniformLocation(SSAOFilterH, "fd"), fd);
        glUseProgram(0);

        glUseProgram(SSAOFilterV);
        glUniform1i(glGetUniformLocation(SSAOFilterV, "SSAOBuffer"), 0);
        glUniform1i(glGetUniformLocation(SSAOFilterV, "DepthBuffer"), 1);
        glUniform1f(glGetUniformLocation(SSAOFilterV, "fs"), fs);
        glUniform1f(glGetUniformLocation(SSAOFilterV, "fd"), fd);
        glUseProgram(0);

        glUseProgram(DeferredLighting);
        glUniform1i(glGetUniformLocation(DeferredLighting, "ColorBuffer"), 0);
        glUniform1i(glGetUniformLocation(DeferredLighting, "NormalBuffer"), 1);
        glUniform1i(glGetUniformLocation(DeferredLighting, "DepthBuffer"), 2);
        glUniform1i(glGetUniformLocation(DeferredLighting, "SSAOBuffer"), 3);
        glUniform1i(glGetUniformLocation(DeferredLighting, "ShadowCubeMaps"), 4);
        glUseProgram(0);

        glUseProgram(Antialiasing);
        glUniform1i(glGetUniformLocation(Antialiasing, "ColorBuffer"), 0);
        glUniform1i(glGetUniformLocation(Antialiasing, "NormalBuffer"), 1);
        glUniform1i(glGetUniformLocation(Antialiasing, "DepthBuffer"), 2);
        glUseProgram(0);
```

```
    // ------------------------------------------------------------------------
------------------------------------------------

    srand(GetTickCount());

    vec2 *Samples = new vec2[16];
    float RandomAngle = (float)M_PI_4, Radius = 1.0f;

    for(int i = 0; i < 16; i++)
    {
        Samples[i].x = cos(RandomAngle) * (float)(i + 1) / 16.0f * Radius;
        Samples[i].y = sin(RandomAngle) * (float)(i + 1) / 16.0f * Radius;

        RandomAngle += (float)M_PI_2;

        if(((i + 1) % 4) == 0) RandomAngle += (float)M_PI_4;
    }

    glUseProgram(SSAO);
    glUniform2fv(glGetUniformLocation(SSAO, "Samples"), 16, (float*)Samples);
    glUseProgram(0);

    delete [] Samples;

    // ------------------------------------------------------------------------
------------------------------------------------

    vec4 *RotationTextureData = new vec4[64 * 64];

    RandomAngle = (float)rand() / (float)RAND_MAX * (float)M_PI * 2.0f;

    for(int i = 0; i < 64 * 64; i++)
    {
        RotationTextureData[i].x = cos(RandomAngle) * 0.5f + 0.5f;
        RotationTextureData[i].y = sin(RandomAngle) * 0.5f + 0.5f;
        RotationTextureData[i].z = -sin(RandomAngle) * 0.5f + 0.5f;
        RotationTextureData[i].w = cos(RandomAngle) * 0.5f + 0.5f;

        RandomAngle += (float)rand() / (float)RAND_MAX * (float)M_PI * 2.0f;
    }

    glGenTextures(1, &RotationTexture);
    glBindTexture(GL_TEXTURE_2D, RotationTexture);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
```

```
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, 64, 64, 0, GL_RGBA, GL_FLOAT,
(float*)RotationTextureData);

        delete [] RotationTextureData;

        // ------------------------------------------------------------------------
-----------------------------------------------

        LightProjectionMatrix = perspective(90.0f, 1.0f, 0.03125f, 32.0f);

        // ------------------------------------------------------------------------
-----------------------------------------------

        glGenTextures(1, &ShadowCubeMaps);
        glBindTexture(GL_TEXTURE_2D_ARRAY_EXT, ShadowCubeMaps);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_TEXTURE_WRAP_S,
GL_CLAMP_TO_EDGE);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_TEXTURE_WRAP_T,
GL_CLAMP_TO_EDGE);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_TEXTURE_COMPARE_MODE,
GL_COMPARE_R_TO_TEXTURE);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_TEXTURE_COMPARE_FUNC,
GL_LEQUAL);
        glTexParameteri(GL_TEXTURE_2D_ARRAY_EXT, GL_DEPTH_TEXTURE_MODE,
GL_INTENSITY);
        glTexImage3D(GL_TEXTURE_2D_ARRAY_EXT, 0, GL_DEPTH_COMPONENT24,
SHADOW_CUBE_MAP_SIZE, SHADOW_CUBE_MAP_SIZE, 24, 0, GL_DEPTH_COMPONENT,
GL_FLOAT, NULL);
        glBindTexture(GL_TEXTURE_2D_ARRAY_EXT, 0);

        // ------------------------------------------------------------------------
-----------------------------------------------

        glGenTextures(2, ColorBuffers);
        glGenTextures(1, &NormalBuffer);
        glGenTextures(1, &DepthBuffer);
        glGenTextures(2, SSAOBuffers);

        // ------------------------------------------------------------------------
-----------------------------------------------
```

```
    glGenBuffers(1, &VBO);

    InitArrayBuffers();

    // ------------------------------------------------------------------------
-----------------------------------------------

    glGenFramebuffersEXT(1, &FBO);

    // ------------------------------------------------------------------------
-------------------------------------------

    LightColors[0] = vec3(1.0f, 0.0f, 0.0f);
    LightPositions[0] = vec3(0.0f, 1.5f, 0.33f);
    LightColors[1] = vec3(0.0f, 1.0f, 0.0f);
    LightPositions[1] = rotate(LightPositions[0], 120.0f, vec3(0.0f, 1.0f, 0.0f));
    LightColors[2] = vec3(0.0f, 0.0f, 1.0f);
    LightPositions[2] = rotate(LightPositions[1], 120.0f, vec3(0.0f, 1.0f, 0.0f));
    LightColors[3] = vec3(1.0f, 1.0f, 1.0f);
    LightPositions[3] = vec3(0.0f, 2.75f, -4.75f);

    for(int i = 0; i < 3; i++)
    {
        glLightfv(GL_LIGHT0 + i, GL_AMBIENT, &vec4(LightColors[i] * 0.125f, 1.0f));
        glLightfv(GL_LIGHT0 + i, GL_DIFFUSE, &vec4(LightColors[i] * 0.875f, 1.0f));
        glLightf(GL_LIGHT0 + i, GL_LINEAR_ATTENUATION, 1.0f);
        glLightf(GL_LIGHT0 + i, GL_QUADRATIC_ATTENUATION, 1.0f);
    }

    glLightfv(GL_LIGHT3, GL_AMBIENT, &vec4(LightColors[3] * 0.25f, 1.0f));
    glLightfv(GL_LIGHT3, GL_DIFFUSE, &vec4(LightColors[3] * 0.75f, 1.0f));
    glLightf(GL_LIGHT3, GL_LINEAR_ATTENUATION, 1.0f / 32.0f);
    glLightf(GL_LIGHT3, GL_QUADRATIC_ATTENUATION, 1.0f / 64.0f);

    // ------------------------------------------------------------------------
-----------------------------------------------

    Camera.Look(vec3(0.0f, 1.75f, 1.875f), vec3(0.0f, 1.5f, 0.0f));

    // ------------------------------------------------------------------------
-----------------------------------------------

    return true;
```

```
	// ------------------------------------------------------------------------
-----------------------------------------------
}

void COpenGLRenderer::Render(float FrameTime)
{
	// ------------------------------------------------------------------------
-----------------------------------------------

	GLenum Buffers[] = {GL_COLOR_ATTACHMENT0_EXT,
GL_COLOR_ATTACHMENT1_EXT};

	// render scene to textures ---------------------------------------------
----------------------------------------

	glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
	glDrawBuffers(2, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT0_EXT,
GL_TEXTURE_2D, ColorBuffers[0], 0);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_COLOR_ATTACHMENT1_EXT,
GL_TEXTURE_2D, NormalBuffer, 0);
	glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT, GL_DEPTH_ATTACHMENT_EXT,
GL_TEXTURE_2D, DepthBuffer, 0);

	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

	glMatrixMode(GL_PROJECTION);
	glLoadMatrixf(&ProjectionMatrix);

	glMatrixMode(GL_MODELVIEW);
	glLoadMatrixf(&ViewMatrix);

	glEnable(GL_DEPTH_TEST);
	glEnable(GL_CULL_FACE);

	glBindBuffer(GL_ARRAY_BUFFER, VBO);

	glEnableClientState(GL_TEXTURE_COORD_ARRAY);
	glTexCoordPointer(2, GL_FLOAT, 32, (void*)0);

	glEnableClientState(GL_NORMAL_ARRAY);
	glNormalPointer(GL_FLOAT, 32, (void*)8);

	glEnableClientState(GL_VERTEX_ARRAY);
```

```
        glVertexPointer(3, GL_FLOAT, 32, (void*)20);

        glUseProgram(Preprocess);

        glUniform1i(Preprocess.UniformLocations[0], true);

        glColor3f(1.0f, 1.0f, 1.0f);

        glBindTexture(GL_TEXTURE_2D, Texture[0]);
        glDrawArrays(GL_QUADS, 0, 96);

        glBindTexture(GL_TEXTURE_2D, Texture[1]);
        glDrawArrays(GL_QUADS, 96, 4);

        glBindTexture(GL_TEXTURE_2D, Texture[2]);
        glDrawArrays(GL_QUADS, 100, 80);

        glBindTexture(GL_TEXTURE_2D, 0);

        glUniform1i(Preprocess.UniformLocations[0], false);

        glDrawArrays(GL_QUADS, 180, 4);

        glMultMatrixf(&ModelMatrix);
        glColor3f(0.33f, 0.66f, 1.0f);
        glDrawArrays(GL_QUADS, 184, 72);

        glUseProgram(0);

        glDisableClientState(GL_VERTEX_ARRAY);
        glDisableClientState(GL_NORMAL_ARRAY);
        glDisableClientState(GL_TEXTURE_COORD_ARRAY);

        glBindBuffer(GL_ARRAY_BUFFER, 0);

        glDisable(GL_CULL_FACE);
        glDisable(GL_DEPTH_TEST);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

        // calculate shadow cube maps matrices ----------------------------------------
-----------------------------------------

        if(CalculateShadows && !ShowSSAO)
```

```
    {
        for(int i = 0; i < 4; i++)
        {
            LightViewMatrices[i * 6 + 0] = look(LightPositions[i], LightPositions[i] +
vec3( 1.0f, 0.0f, 0.0f), vec3(0.0f, 1.0f, 0.0f));
            LightViewMatrices[i * 6 + 1] = look(LightPositions[i], LightPositions[i] + vec3(-
1.0f, 0.0f, 0.0f), vec3(0.0f, 1.0f, 0.0f));
            LightViewMatrices[i * 6 + 2] = look(LightPositions[i], LightPositions[i] +
vec3( 0.0f, 1.0f, 0.0f), vec3(0.0f, 0.0f, 1.0f));
            LightViewMatrices[i * 6 + 3] = look(LightPositions[i], LightPositions[i] +
vec3( 0.0f,-1.0f, 0.0f), vec3(0.0f, 0.0f,-1.0f));
            LightViewMatrices[i * 6 + 4] = look(LightPositions[i], LightPositions[i] +
vec3( 0.0f, 0.0f, 1.0f), vec3(0.0f, 1.0f, 0.0f));
            LightViewMatrices[i * 6 + 5] = look(LightPositions[i], LightPositions[i] +
vec3( 0.0f, 0.0f,-1.0f), vec3(0.0f, 1.0f, 0.0f));

            for(int ii = 0; ii < 6; ii++)
            {
                ShadowMatrices[i * 6 + ii] = BiasMatrix * LightProjectionMatrix *
LightViewMatrices[i * 6 + ii] * ViewMatrixInverse;
            }
        }
    }

    // render scene to shadow cube maps ----------------------------------------
----------------------------------------

    if(CalculateShadows && !ShowSSAO)
    {
        glViewport(0, 0, SHADOW_CUBE_MAP_SIZE, SHADOW_CUBE_MAP_SIZE);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
        glDrawBuffers(0, NULL); glReadBuffer(GL_NONE);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, 0, 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);

        glMatrixMode(GL_PROJECTION);
        glLoadMatrixf(&LightProjectionMatrix);

        glEnable(GL_DEPTH_TEST);
```

```
            glEnable(GL_CULL_FACE);

            glCullFace(GL_FRONT);

            glBindBuffer(GL_ARRAY_BUFFER, VBO);

            glEnableClientState(GL_VERTEX_ARRAY);
            glVertexPointer(3, GL_FLOAT, 32, (void*)20);

            for(int i = 0; i < 24; i++)
            {
                glFramebufferTextureLayerEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, ShadowCubeMaps, 0, i);

                glClear(GL_DEPTH_BUFFER_BIT);

                glMatrixMode(GL_MODELVIEW);
                glLoadMatrixf(&LightViewMatrices[i]);

                glDrawArrays(GL_QUADS, 0, 96);
                glDrawArrays(GL_QUADS, 96, 4);
                glDrawArrays(GL_QUADS, 100, 80);
                glDrawArrays(GL_QUADS, 180, 4);
                glMultMatrixf(&ModelMatrix);
                glDrawArrays(GL_QUADS, 184, 72);
            }

            glDisableClientState(GL_VERTEX_ARRAY);

            glBindBuffer(GL_ARRAY_BUFFER, 0);

            glCullFace(GL_BACK);

            glDisable(GL_CULL_FACE);
            glDisable(GL_DEPTH_TEST);

            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

            glViewport(0, 0, Width, Height);
        }

        // calculate screen space ambient occlusion ----------------------------------
----------------------------------------
```

```
    if(CalculateSSAO || ShowSSAO)
    {
        glViewport(0, 0, Width / 2, Height / 2);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
        glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, SSAOBuffers[0], 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
        glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);

        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
        glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
        glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, RotationTexture);
        glUseProgram(SSAO);
        glBegin(GL_QUADS);
            glVertex2f(0.0f, 0.0f);
            glVertex2f(1.0f, 0.0f);
            glVertex2f(1.0f, 1.0f);
            glVertex2f(0.0f, 1.0f);
        glEnd();
        glUseProgram(0);
        glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
        glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
        glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

        glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);

        // blur filter with edge detection ----------------------------------------
----------------------------------------

        if(BlurSSAO)
        {
            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
            glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, SSAOBuffers[1], 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);
```

```
                glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D,
SSAOBuffers[0]);
                glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
                glUseProgram(SSAOFilterH);
                glBegin(GL_QUADS);
                    glVertex2f(0.0f, 0.0f);
                    glVertex2f(1.0f, 0.0f);
                    glVertex2f(1.0f, 1.0f);
                    glVertex2f(0.0f, 1.0f);
                glEnd();
                glUseProgram(0);
                glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
                glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

                glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, SSAOBuffers[0], 0);

                glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D,
SSAOBuffers[1]);
                glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
                glUseProgram(SSAOFilterV);
                glBegin(GL_QUADS);
                    glVertex2f(0.0f, 0.0f);
                    glVertex2f(1.0f, 0.0f);
                    glVertex2f(1.0f, 1.0f);
                    glVertex2f(0.0f, 1.0f);
                glEnd();
                glUseProgram(0);
                glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
                glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

                glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);
            }

            glViewport(0, 0, Width, Height);
        }

        // set lights positions ----------------------------------------------------
----------------------------------------

        glMatrixMode(GL_MODELVIEW);
        glLoadMatrixf(&ViewMatrix);

        for(int i = 0; i < 4; i++)
```

```
    {
        glLightfv(GL_LIGHT0 + i, GL_POSITION, &vec4(LightPositions[i], 1.0f));
    }

    // ------------------------------------------------------------------------
-------------------------------------------

    if(ShowSSAO)
    {
        // display SSAO ---------------------------------------------------------
-----------------------------------------

        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

        glColor3f(1.0f, 1.0f, 1.0f);

        glEnable(GL_TEXTURE_2D);
        glBindTexture(GL_TEXTURE_2D, SSAOBuffers[0]);
        glBegin(GL_QUADS);
            glTexCoord2f(0.0f, 0.0f); glVertex2f(-1.0f, -1.0f);
            glTexCoord2f(1.0f, 0.0f); glVertex2f( 1.0f, -1.0f);
            glTexCoord2f(1.0f, 1.0f); glVertex2f( 1.0f,  1.0f);
            glTexCoord2f(0.0f, 1.0f); glVertex2f(-1.0f,  1.0f);
        glEnd();
        glBindTexture(GL_TEXTURE_2D, 0);
        glDisable(GL_TEXTURE_2D);
    }
    else
    {
        // calculate lighting ---------------------------------------------------
-----------------------------------------

        if(CalculateAntialiasing)
        {
            glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, FBO);
            glDrawBuffers(1, Buffers); glReadBuffer(GL_COLOR_ATTACHMENT0_EXT);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT0_EXT, GL_TEXTURE_2D, ColorBuffers[1], 0);
            glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_COLOR_ATTACHMENT1_EXT, GL_TEXTURE_2D, 0, 0);
```

```
                glFramebufferTexture2DEXT(GL_FRAMEBUFFER_EXT,
GL_DEPTH_ATTACHMENT_EXT, GL_TEXTURE_2D, 0, 0);
            }

            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, ColorBuffers[0]);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, NormalBuffer);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
            glActiveTexture(GL_TEXTURE3); glBindTexture(GL_TEXTURE_2D, SSAOBuffers[0]);
            glActiveTexture(GL_TEXTURE4); glBindTexture(GL_TEXTURE_2D_ARRAY_EXT,
ShadowCubeMaps);
            glUseProgram(DeferredLighting);
            if(CalculateShadows) glUniformMatrix4fv(DeferredLighting.UniformLocations[1], 1,
GL_FALSE, &ViewMatrixInverse);
            if(CalculateShadows) glUniformMatrix4fv(DeferredLighting.UniformLocations[2],
24, GL_FALSE, (float*)ShadowMatrices);
            glUniform1i(DeferredLighting.UniformLocations[3], CalculateSSAO);
            glUniform1i(DeferredLighting.UniformLocations[4], CalculateShadows);
            glBegin(GL_QUADS);
                glVertex2f(0.0f, 0.0f);
                glVertex2f(1.0f, 0.0f);
                glVertex2f(1.0f, 1.0f);
                glVertex2f(0.0f, 1.0f);
            glEnd();
            glUseProgram(0);
            glActiveTexture(GL_TEXTURE4); glBindTexture(GL_TEXTURE_2D_ARRAY_EXT, 0);
            glActiveTexture(GL_TEXTURE3); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);

            if(CalculateAntialiasing)
            {
                glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, 0);
            }

            // calculate antialiasing ------------------------------------------------
----------------------------------------

            if(CalculateAntialiasing)
            {
                glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D,
ColorBuffers[1]);
                glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D,
NormalBuffer);
```

```
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, DepthBuffer);
            glUseProgram(Antialiasing);
            glBegin(GL_QUADS);
                glVertex2f(0.0f, 0.0f);
                glVertex2f(1.0f, 0.0f);
                glVertex2f(1.0f, 1.0f);
                glVertex2f(0.0f, 1.0f);
            glEnd();
            glUseProgram(0);
            glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, 0);
            glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, 0);
        }
    }

    // rotate object and lights ----------------------------------------------------
-----------------------------------------

    if(!Pause)
    {
        static float a = 0.0f;

        ModelMatrix = translate(0.0f, 1.5f, 0.0f) * rotate(a, vec3(0.0f, 1.0f, 0.0f)) * rotate(a,
vec3(1.0f, 0.0f, 0.0f));

        a += 22.5f * FrameTime;

        for(int i = 0; i < 3; i++)
        {
            LightPositions[i] = rotate(LightPositions[i], -180.0f * FrameTime, vec3(0.0f, 1.0f,
0.0f));
        }
    }

    // ----------------------------------------------------------------------------
-----------------------------------------------
}

void COpenGLRenderer::Resize(int Width, int Height)
{
    this->Width = Width;
    this->Height = Height;

    glViewport(0, 0, Width, Height);
```

```
    ProjectionMatrix = perspective(45.0f, (float)Width / (float)Height, 0.125f, 512.0f);

    ProjectionBiasMatrixInverse = inverse(ProjectionMatrix) * BiasMatrixInverse;

    for(int i = 0; i < 2; i++)
    {
        glBindTexture(GL_TEXTURE_2D, ColorBuffers[i]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
        glBindTexture(GL_TEXTURE_2D, 0);
    }

    glBindTexture(GL_TEXTURE_2D, NormalBuffer);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width, Height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
    glBindTexture(GL_TEXTURE_2D, 0);

    glBindTexture(GL_TEXTURE_2D, DepthBuffer);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT24, Width, Height, 0,
GL_DEPTH_COMPONENT, GL_FLOAT, NULL);
    glBindTexture(GL_TEXTURE_2D, 0);

    for(int i = 0; i < 2; i++)
    {
        glBindTexture(GL_TEXTURE_2D, SSAOBuffers[i]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA8, Width / 2, Height / 2, 0, GL_RGBA,
GL_UNSIGNED_BYTE, NULL);
```

```
            glBindTexture(GL_TEXTURE_2D, 0);
        }

        glUseProgram(SSAO);
        glUniform2f(SSAO.UniformLocations[0], (float)Width / 2.0f / 64.0f, (float)Height / 2.0f /
64.0f);
        glUniformMatrix4fv(SSAO.UniformLocations[1], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
        glUseProgram(0);

        glUseProgram(SSAOFilterH);
        glUniform1f(SSAOFilterH.UniformLocations[0], 2.0f / (float)Width);
        glUseProgram(SSAOFilterV);
        glUniform1f(SSAOFilterV.UniformLocations[0], 2.0f / (float)Height);
        glUseProgram(0);

        glUseProgram(DeferredLighting);
        glUniformMatrix4fv(DeferredLighting.UniformLocations[0], 1, GL_FALSE,
&ProjectionBiasMatrixInverse);
        glUseProgram(0);

        glUseProgram(Antialiasing);
        glUniform2f(Antialiasing.UniformLocations[0], 1.0f / (float)Width, 1.0f / (float)Height);
        glUseProgram(0);
}

void COpenGLRenderer::Destroy()
{
        for(int i = 0; i < 3; i++)
        {
            Texture[i].Destroy();
        }

        Preprocess.Destroy();
        SSAO.Destroy();
        SSAOFilterH.Destroy();
        SSAOFilterV.Destroy();
        DeferredLighting.Destroy();
        Antialiasing.Destroy();

        glDeleteBuffers(1, &VBO);

        glDeleteTextures(1, &RotationTexture);
        glDeleteTextures(1, &ShadowCubeMaps);
```

```
	glDeleteTextures(2, ColorBuffers);
	glDeleteTextures(1, &NormalBuffer);
	glDeleteTextures(1, &DepthBuffer);
	glDeleteTextures(2, SSAOBuffers);

	if(GLEW_EXT_framebuffer_object)
	{
		glDeleteFramebuffersEXT(1, &FBO);
	}
}

void COpenGLRenderer::InitArrayBuffers()
{
	CBuffer buffer;

	vec3 m;

	// cubes

	m = vec3( 0.0f, 0.5f, 0.0f);

	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
	buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
	buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    m = vec3( 0.0f, 1.5f,-4.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        m = vec3(-0.625f, 0.5f,-4.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

    m = vec3( 0.625f, 0.5f,-4.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x,-0.5f + m.y, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.5f + m.y, 0.5f + m.z), 12);

        // floor

        buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,    5.0f), 12);
        buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,    5.0f), 12);
        buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, -5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, -5.0f), 12);

        // walls

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
        buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
        buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
        buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
        buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f,    0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f, 5.0f), 12);
        buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f, 5.0f), 12);
        buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, -1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 0.0f,-5.0f), 12);
        buffer.AddData(&vec2(10.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 0.0f,-5.0f), 12);
```

```
        buffer.AddData(&vec2(10.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,    1.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);

        // pillars

        m = vec3(-2.5f, 0.0f,-2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f,-2.5f);
```

```
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        m = vec3( 2.5f, 0.0f, 2.5f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

    m = vec3(-2.5f, 0.0f, 2.5f);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f,-0.5f + m.z), 12);
```

```
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f,-0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 0.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.5f + m.x, 3.0f, 0.5f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 3.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.5f + m.x, 3.0f, 0.5f + m.z), 12);

        // ceiling

        buffer.AddData(&vec2( 0.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f,-5.0f), 12);
        buffer.AddData(&vec2(10.0f,    0.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f,-5.0f), 12);
        buffer.AddData(&vec2(10.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3( 5.0f, 3.0f, 5.0f), 12);
        buffer.AddData(&vec2( 0.0f, 10.0f), 8); buffer.AddData(&vec3(0.0f, -1.0f, 0.0f), 12);
buffer.AddData(&vec3(-5.0f, 3.0f, 5.0f), 12);

        // rotating object

        m = vec3(0.0f);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
```

```
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x,-0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.25f + m.x, 0.025f + m.y, 0.025f + m.z), 12);

        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
        buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
```

```
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y,-0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);
    buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.25f + m.y, 0.025f + m.z), 12);

    buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
    buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
```

```
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3(-1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 1.0f, 0.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f,-1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 1.0f, 0.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f,-1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y,-0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 0.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x,-0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 1.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3( 0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
buffer.AddData(&vec2( 0.0f, 1.0f), 8); buffer.AddData(&vec3( 0.0f, 0.0f, 1.0f), 12);
buffer.AddData(&vec3(-0.025f + m.x, 0.025f + m.y, 0.25f + m.z), 12);
```

```
        glBindBuffer(GL_ARRAY_BUFFER, VBO);
        glBufferData(GL_ARRAY_BUFFER, buffer.GetDataSize(), buffer.GetData(),
GL_STATIC_DRAW);
        glBindBuffer(GL_ARRAY_BUFFER, 0);

        buffer.Empty();
}
```