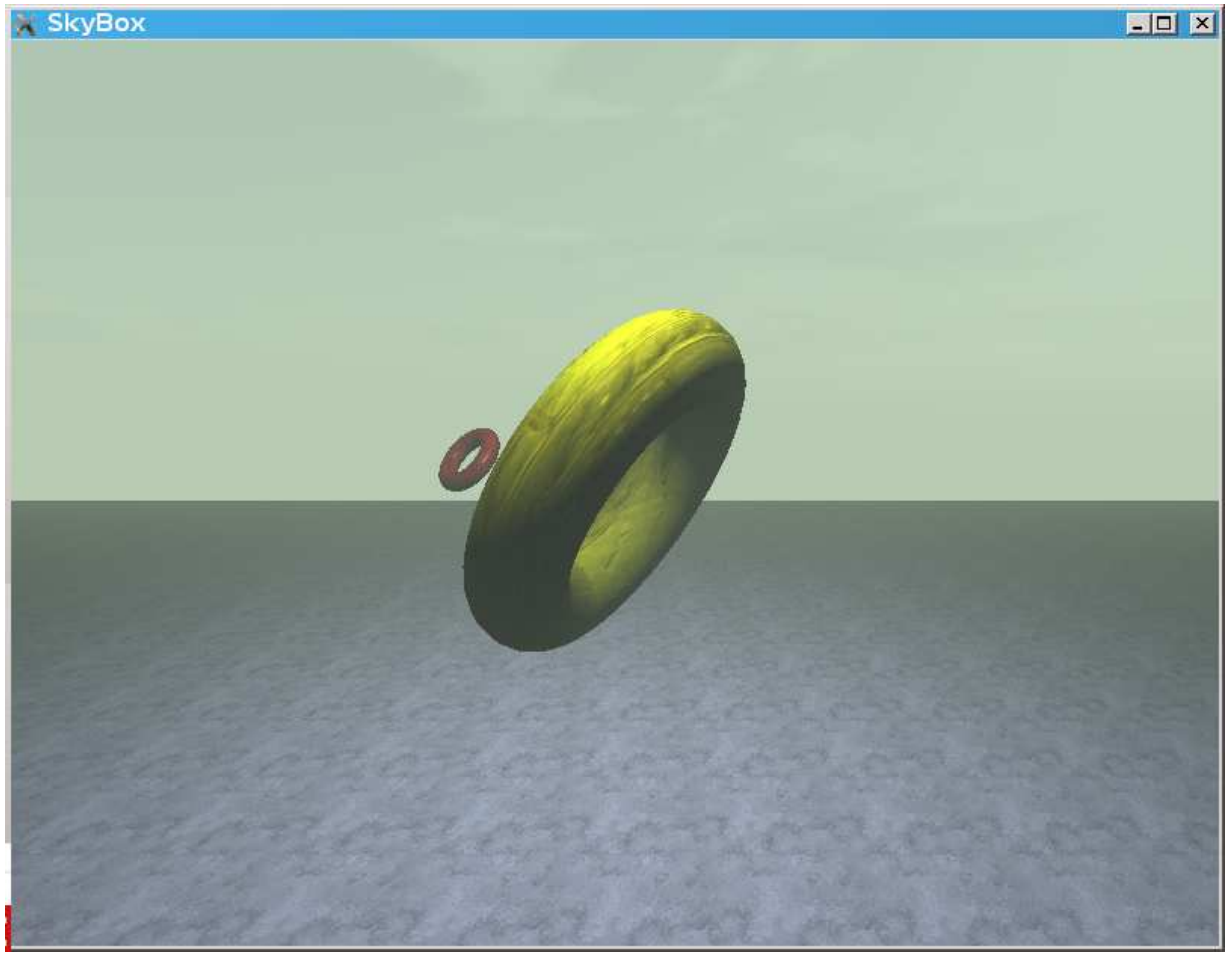
Programowanie 3W grafiki w OpenGL. Modelowanie mgły

Aleksander Denisiuk
Polsko-Japońska Akademia Technik Komputerowych
Wydział Informatyki w Gdańsku
ul. Brzegi 55
80-045 Gdańsk

denisjuk@pja.edu.pl

Najnowsza wersja tego dokumentu dostępna jest pod adresem
http://users.pja.edu.pl/~denisjuk/

Modelowanie



Modelowanie mgły

- Mieszanie koloru piksela z kolorem mgły
- Parametry mgły:
  - kolor
  - gęstość
- Efekt mgły (współczynnik) zależy of odległości fragmetnu od kamery

Tzry rodzaje mgły

- liniowa:

$$f(d) = \begin{cases} 0 & d < d_{\min}, \\ 1 - \frac{d_{\max} - d}{d_{\max} - d_{\min}} & d_{\min} \leqslant d \leqslant d_{\max}, \\ 1 & d_{\max} < d. \end{cases}$$

- wykładnicza $f(d) = 1 - e^{-\lambda d}$
- kwadratowo wykładnicza $f(d) = 1 - e^{-(\lambda d)^2}$

Implementacja

Mgła w shaderze

```
#version 430 core
#define MIST_LINEAR 0
#define MIST_EXP1 1
#define MIST_EXP2 2
#define MIST_NONE 3

uniform struct Mist
{
    vec4 color; // Mist color
    float start; // This is only for linear fog
    float end; // This is only for linear fog
    float density; // For exp and exp2 equation

    int type; // 0 = linear, 1 = exp,
              // 2 = exp2 3 = no mist at all
} mist;
```

Funkcja mgły


```
float GetMistFactor(Mist params, float coord){
    float res = 0.0;
    switch(params.type){
    case MIST_LINEAR:
        res = (params.end-coord)
              /(params.end-params.start);
        break;
    case MIST_EXP1:
         res = exp(-params.density*coord);
        break;
    case MIST_EXP2:
        res = exp(-pow(params.density*coord, 2.0));
        break;
    }
    return 1.0 - clamp(res, 0.0, 1.0);
}
```

```
.............................
out struct Vertex {
    vec2  texcoord;
    vec3  normal;
    vec3  light_dir;
    vec3  view_dir;
    float light_dist;
    float camera_dist;
} frag_vertex;
.............................

frag_vertex.camera_dist = distance(camera, vertex);
.............................
```

Fragment Shader

```
in struct Vertex {
    vec2  texcoord;
    vec3  normal;
    vec3  light_dir;
    vec3  view_dir;
    float light_dist;
    float camera_dist;
} frag_vertex;

.....................................

if(mist.type!=MIST_NONE) {
   float mist_factor = GetMistFactor(mist,
                                     frag_vertex.camera_dist);
   color = mix(color, mist.color, mist_factor) ;
}

.....................................
```

- $\mathrm{mix}(x, y, \alpha) = (1 - \alpha)x + \alpha y$

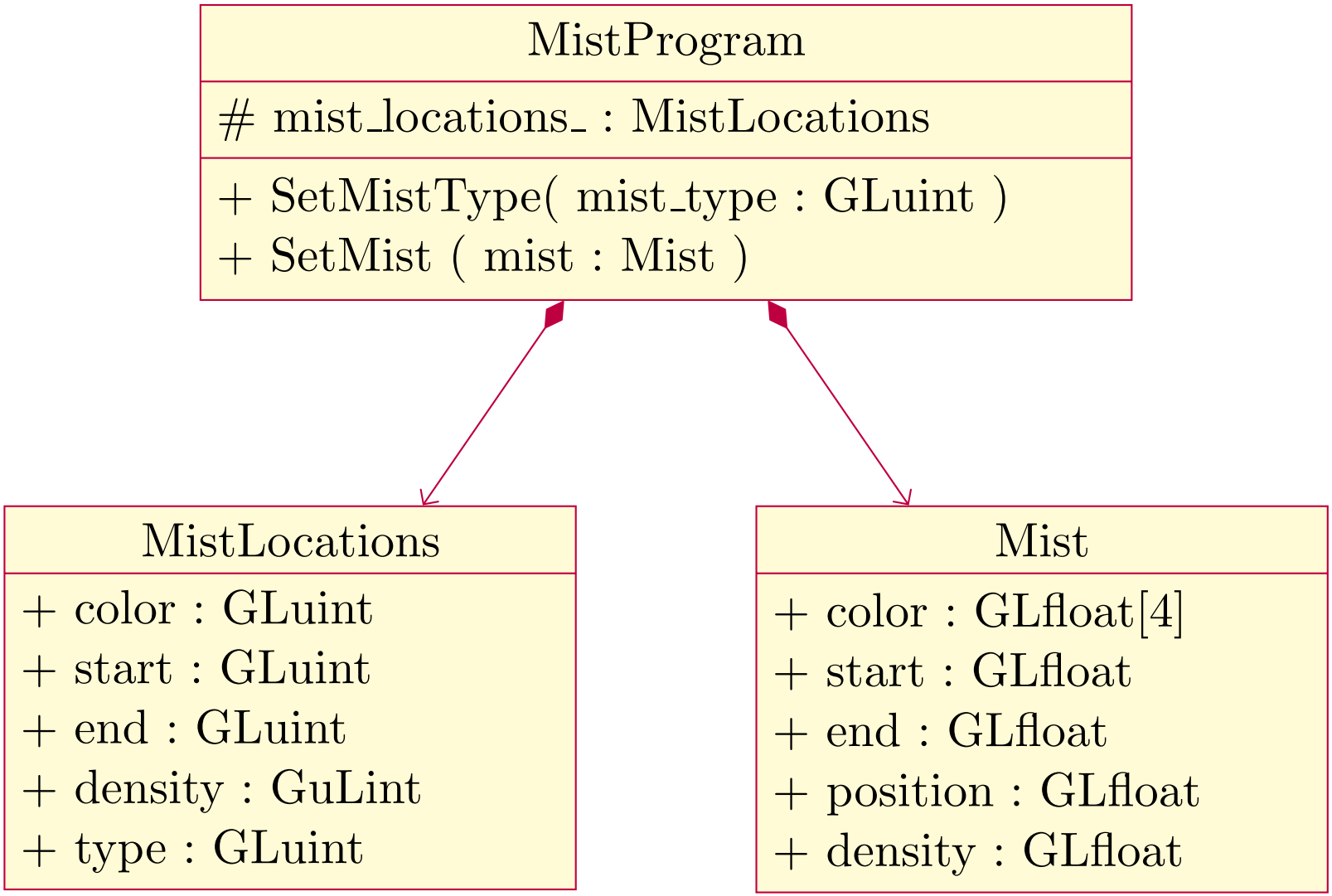
SkyBox Fragment Shader


```
if(mist.type!= MIST_NONE) {
    float mist_coord = 40;
    float mist_factor = GetMistFactor(mist, mist_coord);
    color = mix(color, mist.color, mist_factor) ;
}
```

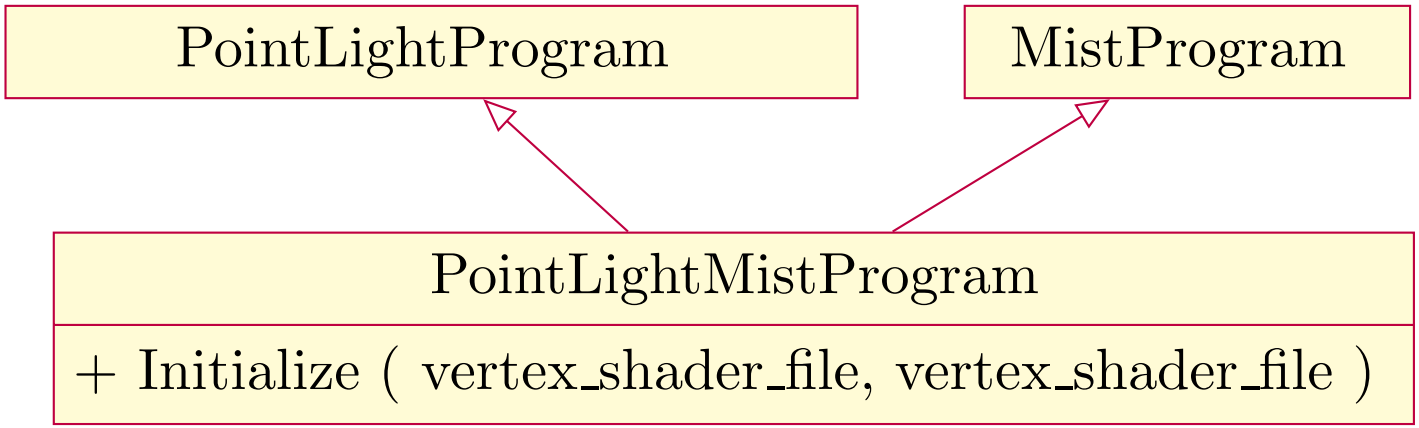
Typ i struktura dla mgły (`mist.h`)


```
#define MIST_LINEAR 0
#define MIST_EXP1 1
#define MIST_EXP2 2
#define MIST_NONE 3

typedef struct Mist{
  GLfloat color[4];
  GLfloat start;
  GLfloat end;
  GLfloat density;
} Mist;
```



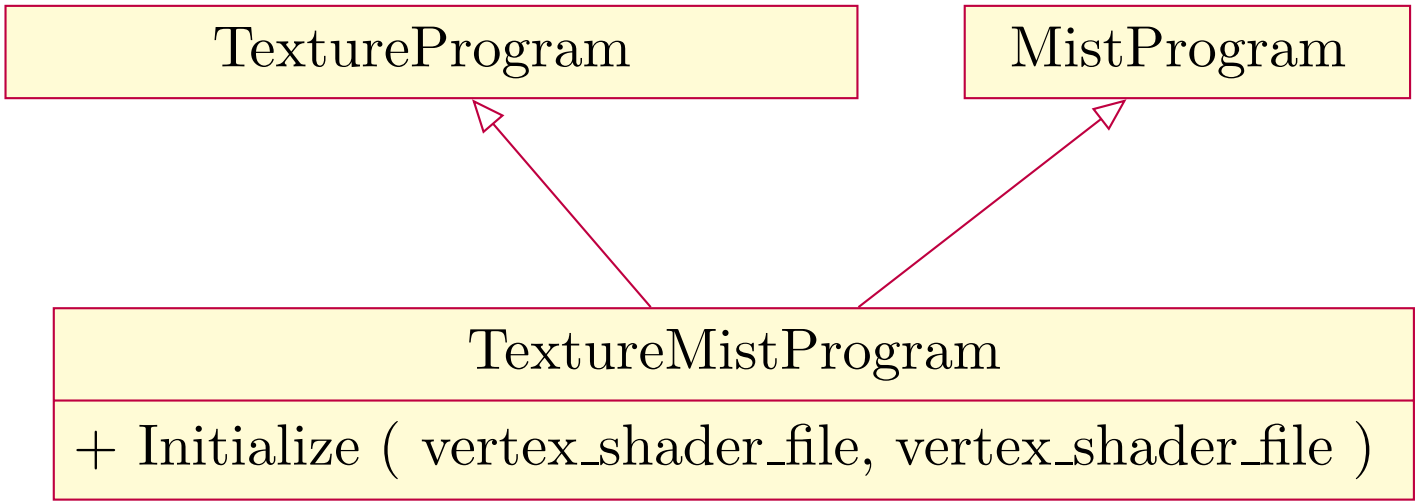
PointLightMistProgram



PointLightMistProgram: inicjalizacja


```
void PointLightMistProgram::Initialize(
        const char *vertex_shader_file,
        const char *fragment_shader_file){
    PointLightProgram::Initialize(
       vertex_shader_file, fragment_shader_file);
    mist_locations_.color
      = GetUniformLocationOrDie("mist.color");
    mist_locations_.start
      = GetUniformLocationOrDie("mist.start");
    mist_locations_.end
      = GetUniformLocationOrDie("mist.end");
    mist_locations_.density
      = GetUniformLocationOrDie("mist.density");
    mist_locations_.type
      = GetUniformLocationOrDie("mist.type");
}
```

Skybox (TextureMistProgram)



- Inicjalizacja: analogicznie

Mgła i nowe shadery


```
const Mist kMist = {
    0.8f, 0.8f, 0.7f, 1.0f, //color
    3.0f, //start
    60.0f, //end
    0.05f //density
};

const char* kPointLightVertexShader
        = "PointLightMist.vertex.glsl";
const char* kPointLightFragmentShader
        = "PointLightMist.fragment.glsl";

const char* kSkyBoxVertexShader="SkyBox.vertex.glsl";
const char* kSkyBoxFragmentShader
        = "SkyBoxMist.fragment.glsl";
```

Inicjalizacja programów (InitPrograms)


```
 ..............................
point_program_.Initialize(
    kPointLightVertexShader,
    kPointLightFragmentShader);
glUseProgram(point_program_);
point_program_.SetMist(kMist);
point_program_.SetMistType(MIST_LINEAR);
point_program_.SetLight(kPointLight);
point_program_.SetTextureUnit(0);
point_program_.SetProjectionMatrix(projection_matrix_);
point_program_.SetViewMatrix(view_matrix_);
 ..............................
```

Zmiana typu mgły (SpecialKeyPressed)


```
switch (key){
    case GLUT_KEY_F1:
    SetMistType(MIST_NONE);
break;
    case GLUT_KEY_F2:
    SetMistType(MIST_LINEAR);
break;
 ..............................
```

```
void Window::SetMistType(GLuint mist_type){
    glUseProgram(point_program_);
    point_program_.SetMistType(mist_type);
    glUseProgram(sky_program_);
    sky_program_.SetMistType(mist_type);
    glUseProgram(0);
}

 ..............................
```