

Zaawansowane systemy programowania grafiki. Modelowanie mgły

Aleksander Denisiuk
Uniwersytet Warmińsko-Mazurski
Olsztyn, ul. Słoneczna 54
denisjuk@matman.uwm.edu.pl

19 kwietnia 2021

Modelowanie mgły

Modelowanie mgły

Implementacja mgły

Najnowsza wersja tego dokumentu dostępna jest pod adresem

<http://wmii.uwm.edu.pl/~denisjuk/uwm>

Modelowanie mgły

❖ Scena

❖ Równania mgły

Implementacja mgły

Modelowanie mgły

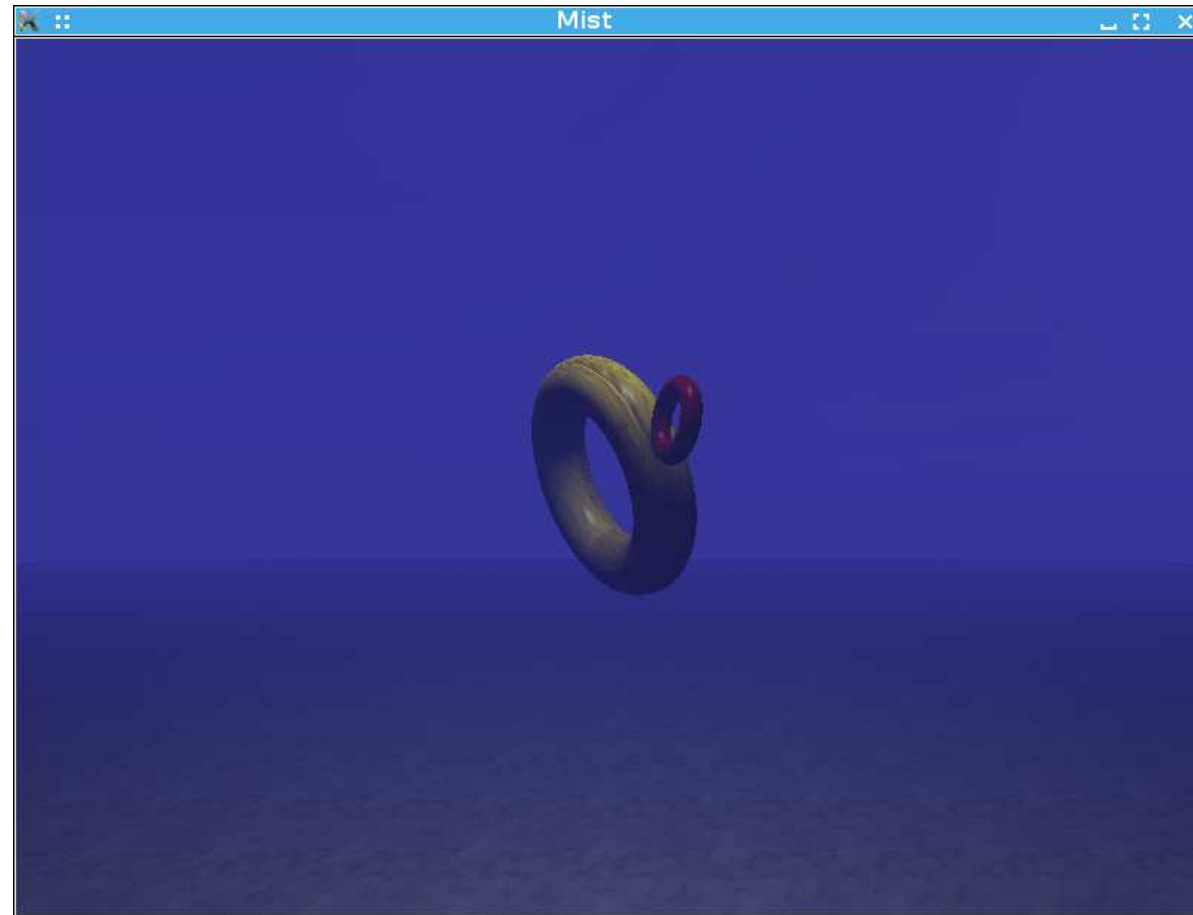
Scena

Modelowanie mgły

❖ Scena

❖ Równania mgły

Implementacja mgły



Modelowanie mgły

Modelowanie mgły

❖ Scena

❖ Równania mgły

Implementacja mgły

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

Tzry rodzaje mgły

Modelowanie mgły

❖ Scena

❖ Rownania mgły

Implementacja mgły

- liniowa:

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

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

Modelowanie mgły

Implementacja mgły

- ❖ Czynności
- ❖ Shadery
- ❖ Mist
- ❖ Programy
- ❖ Window

Implementacja mgły

Czynności

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

- Oddzielny shader z funkcją mgły
- Dołączamy do programu *point light*
- Dołączamy do programu *skybox light*
 - ◆ Jeżeli skybox jest ukryty mgłą, można go wyłączyć

Mgła w shaderze

(...Mist.fragment.glsl)

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
#version 430 core
#define LINEAR 0
#define EXP1 1
#define EXP2 2
#define 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; // exp and exp2 equation

    int type; // 0 = linear, 1 = exp, 2 = exp2,
              // 3 = nothing at all
} mist;
```

Funkcja mgły

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
float GetMistFactor(Mist params, float coord) {  
    float res = 0.0;  
    switch(params.type) {  
        case LINEAR:  
            res = (params.end-coord) /  
                (params.end-params.start);  
            break;  
        case EXP1:  
            res = exp(-params.density*coord);  
            break;  
        case EXP2:  
            res = exp(-pow(params.density*coord, 2.0));  
            break;  
    }  
    return 1.0 - clamp(res, 0.0, 1.0);  
}
```

Vertex Shader

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
.....  
out struct Vertex {  
    vec2    texcoord;  
    vec3    normal;  
    vec3    light_dir;  
    vec3    view_dir;  
    vec4    view_coord;  
    float   dist;  
} frag_vertex;
```

```
.....  
  
vec4 vertex = model_matrix * in_position;  
frag_vertex.view_coord = view_matrix * vertex;  
gl_Position = projection_matrix  
                * frag_vertex.view_coord;
```

Fragment Shader

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
in struct Vertex {  
    vec2 texcoord;  
    vec3 normal;  
    vec3 light_dir;  
    vec3 view_dir;  
    vec4 view_coord;  
    float dist;  
} frag_vertex;  
  
.....
```

Fragment Shader

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
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;  
} mist;  
  
float GetMistFactor(Mist params, float coord);  
  
.....
```

Fragment Shader (koniec main)

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
if (mist.type != NONE) {  
    float mist_coord  
        = abs(frag_vertex.view_coord.z  
              / frag_vertex.view_coord.w);  
    float mist_factor  
        = GetMistFactor(mist, mist_coord);  
    color = mix(color, mist.color, mist_factor);  
}
```

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

SkyBox Fragment Shader

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
#define 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; // exp and exp2 equation

    int type;
} mist;

float GetMistFactor(Mist params, float coord);

.....
```

SkyBox Fragment Shader (w main)

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
if (mist.type!=NONE) {  
    float mist_coord = 40;  
    float mist_factor  
        =GetMistFactor(mist, mist_coord);  
    color = mix(color, mist.color, mist_factor) ;  
}
```


Struktura dla mgły

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

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

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

Klasa pomocniczna *MistProgram*

Modelowanie mgły

Implementacja mgły

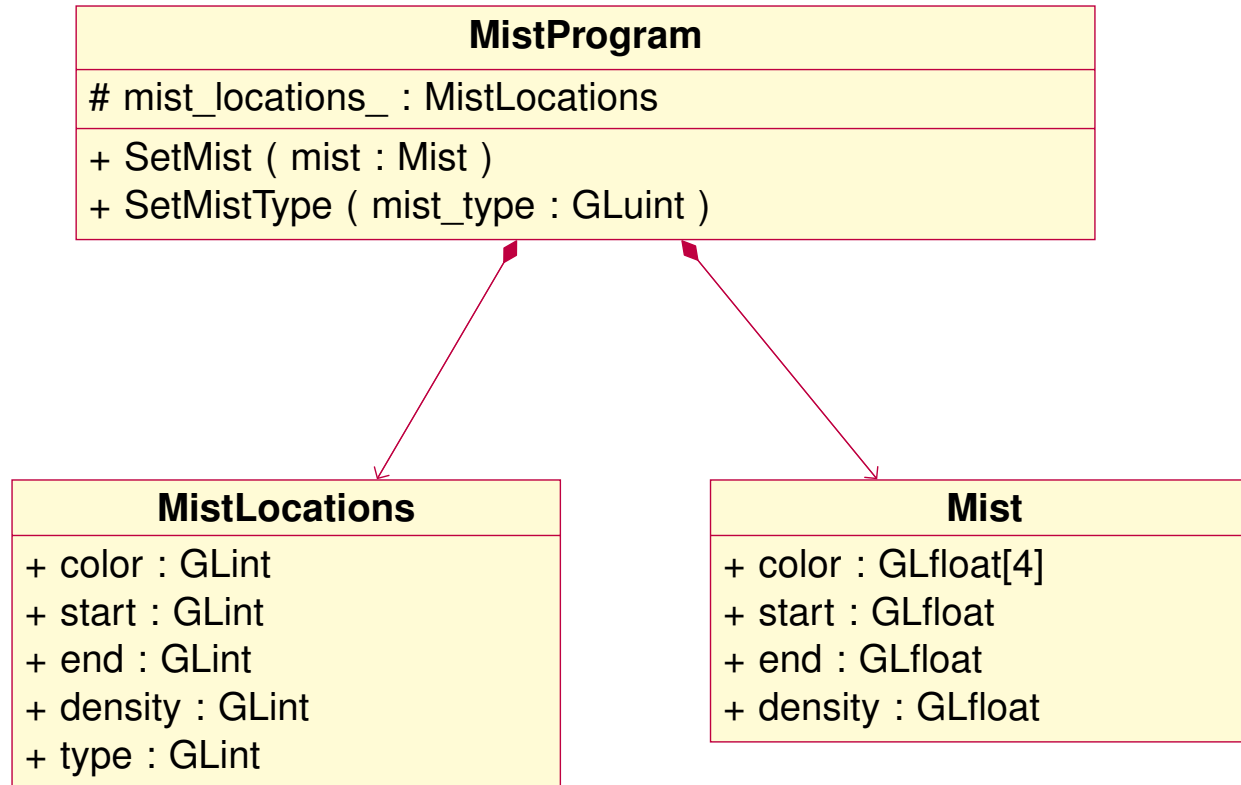
❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window



PointLightMistProgram

Modelowanie mgły

Implementacja mgły

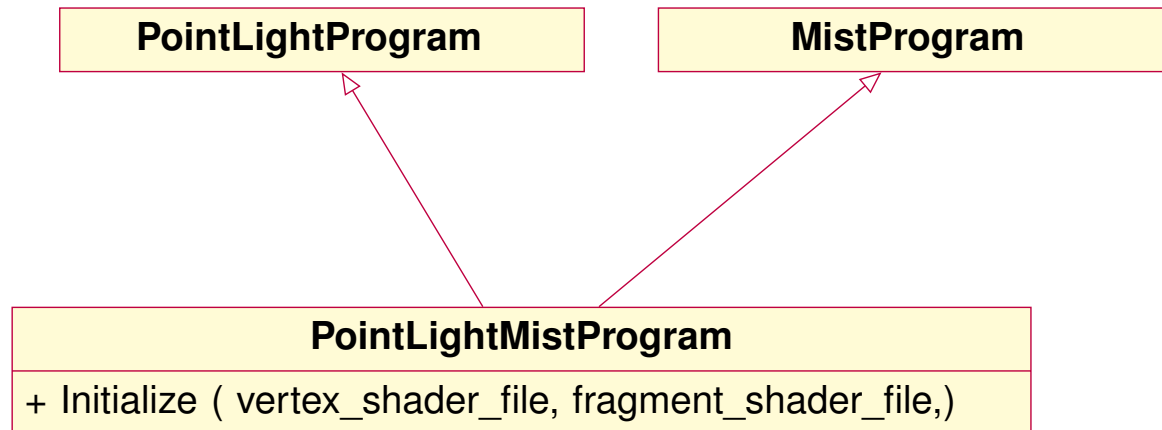
❖ Czynności

❖ Shadery

❖ Mist

❖ **Programy**

❖ Window



TextureMistCameraProgram

Modelowanie mgły

Implementacja mgły

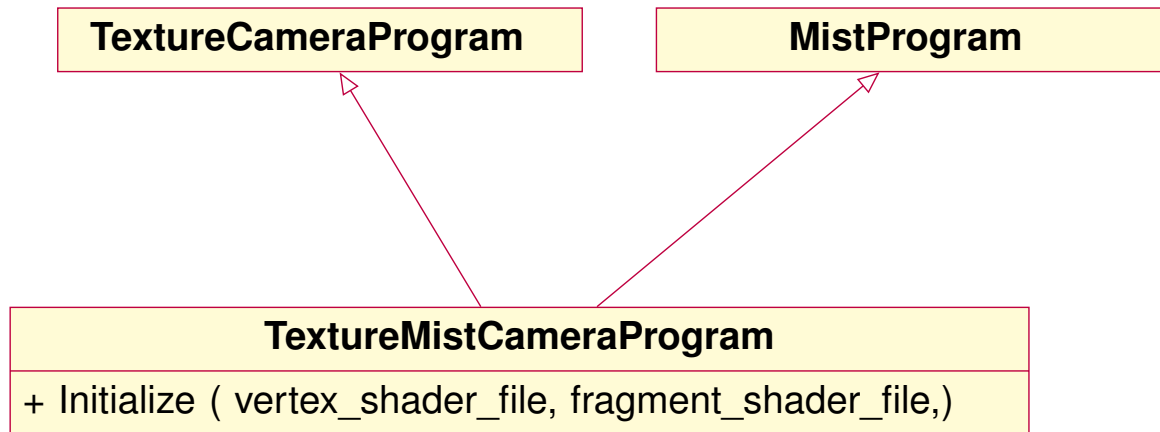
❖ Czynności

❖ Shadery

❖ Mist

❖ **Programy**

❖ Window



Klasa Window

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

- window.h

```
GLuint mist_type_;  
void SetMistType();
```

- window.cpp

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

Inicjalizacja

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
mist_type_ = MIST_LINEAR;
```

- 0 — liniowa
- 1 — wykładnicza
- 2 — kwadratowo-wykładnicza
- 3 — brak

Inicjalizacja, funkcje

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
point_program_.Initialize(  
    kPointLightVertexShader,  
    kPointLightFragmentShader);  
  
. . . . .  
point_program_.SetMist(kMist);  
point_program_.SetMistType(MIST_LINEAR);  
  
. . . . .  
  
sky_program_.Initialize(kSkyVertexShader,  
    kSkyFragmentShader, kMistShader);  
  
. . . . .  
sky_program_.SetMist(kMist);  
sky_program_.SetMistType(MIST_LINEAR);
```

Zdarzenia klawiatury

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

```
case GLFW_KEY_F1 :  
    SetMistType (MIST_NONE) ;  
break ;  
case GLFW_KEY_F2 :  
    SetMistType (MIST_LINEAR) ;  
break ;  
case GLFW_KEY_F3 :  
    SetMistType (MIST_EXP1) ;  
break ;  
case GLFW_KEY_F4 :  
    SetMistType (MIST_EXP2) ;  
break ;
```

.....

Funkcja SetMistType

Modelowanie mgły

Implementacja mgły

❖ Czynności

❖ Shadery

❖ Mist

❖ Programy

❖ Window

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