

# Programowanie 3W grafiki w OpenGL. Modelowanie przezroczystości. Sprajty punktowe

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

denisjuk@pja.edu.pl

# Modelowanie przezroczystości. Sprajty punktowe

Modelowanie  
przezroczystości

Sprajty  
punktowe

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

Modelowanie  
przezroczystości

Scena

Shadery

Programy

Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe

# Modelowanie przezroczystości

## Modelowanie przezroczystości

Scena

Shadery

Programy

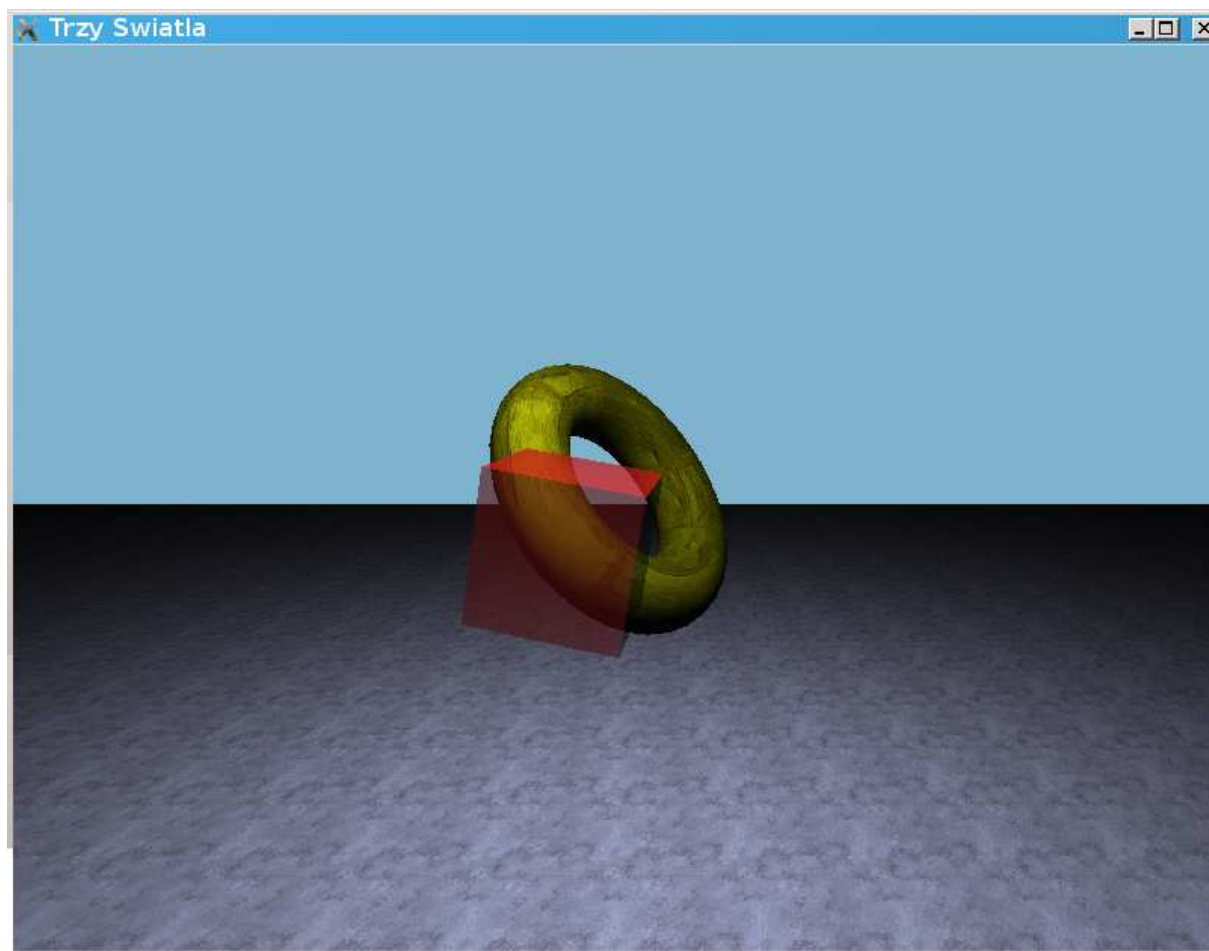
Programy

Sześcian

Funkcje

OpenGL

## Sprajty punktowe



Modelowanie  
przezroczystości

Scena

Shadery

Programy

Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe

- Otekstutowane torus i płaszczyzna
- Przezroczysty cześcian bez tekstury
- Dwa programy cieniujące

- Jak światło punktowe, tylko bez tekstury

```
struct Vertex {  
    vec3  normal;  
    vec3  light_dir;  
    vec3  view_dir;  
    float dist;  
} frag_vertex;
```

# TransparentMaterialModelProgram

Modelowanie  
przezroczystości

Scena

Shadery

Programy

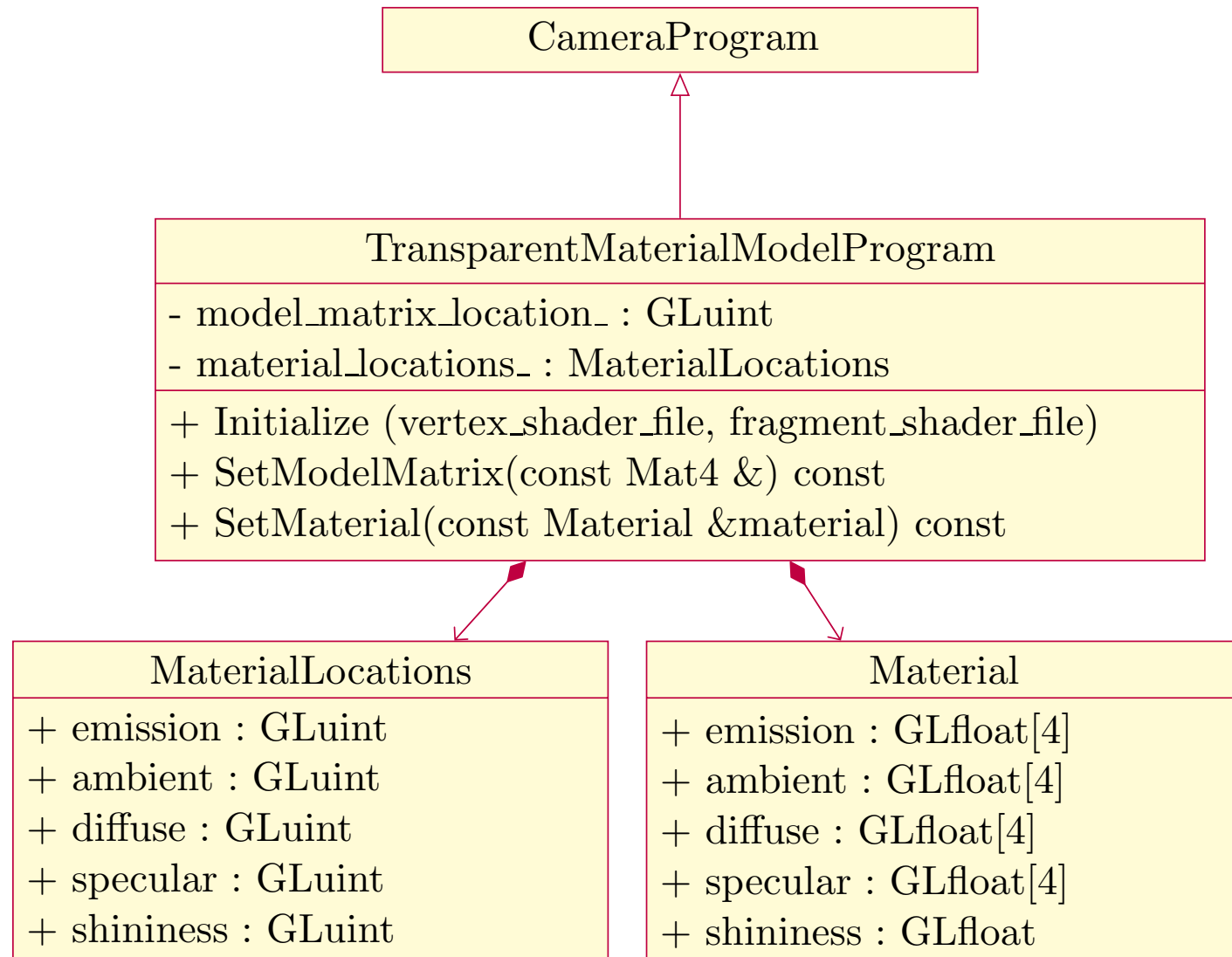
Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe



# PointLightTransparentProgram

Modelowanie  
przezroczystości

Scena

Shadery

Programy

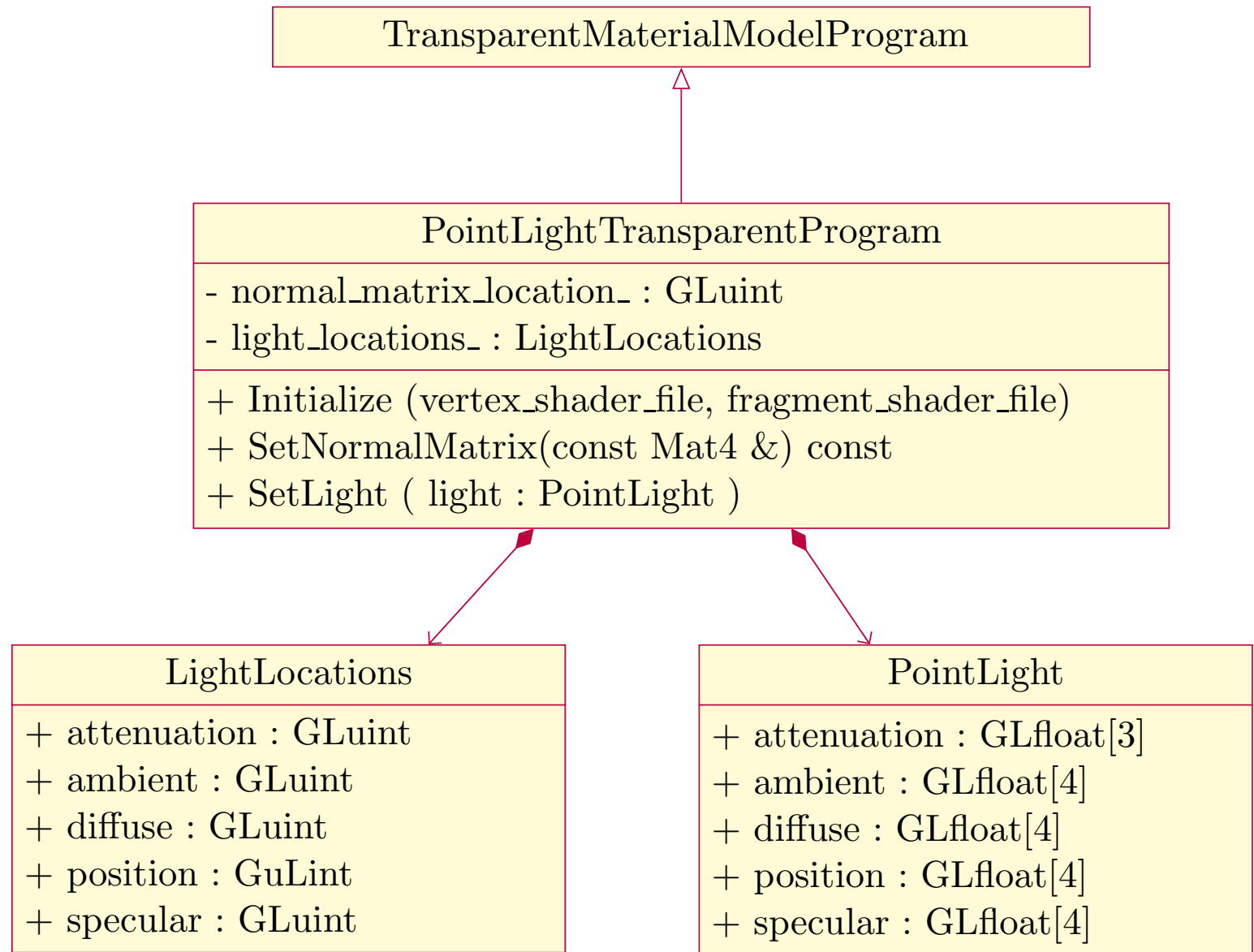
Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe





Modelowanie  
przezroczystości

Scena

Shadery

Programy

Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe

```
typedef struct MaterialVertex{  
    float position[4];  
    float normal[3];  
} MaterialVertex;
```

Modelowanie  
przezroczystości

Scena

Shadery

Programy

Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe

```
Material RedMaterial={  
    {0.2f, 0.2f, 0.2f, 0.25f}, //Ambient  
    {0.9f, 0.1f, 0.1f, 0.25f}, //Diffuse  
    {0.5f, 0.5f, 0.5f, 0.25f}, //Specular  
    {0.0f, 0.0f, 0.0f, 0.25f}, //Emission  
    20.0f //Shininess  
};
```

```
.....  
glEnable(GL_BLEND);  
glBlendFunc(GL_SRC_ALPHA,  
             GL_ONE_MINUS_SRC_ALPHA);  
glDepthMask(0);  
  
glDrawElements(GL_TRIANGLES, 36,  
               GL_UNSIGNED_INT, (GLvoid*)0);  
  
glDisable(GL_BLEND);  
glDepthMask(1);  
.....
```

■ Przezroczyste — jako ostatnie

```
void glBlendFunc(GLenum sfactor,  
                 GLenum dfactor);
```

- **sfactor** — współczynnik przy źródłowym kolorze
- **dfactor** — współczynnik przy kolorze docelowym
  - $\text{Kolor} = C_s \cdot \text{sfactor} + C_d \cdot \text{dfactor}$
- przykładowe współczynniki:
  - GL\_ZERO, GL\_ONE
  - GL\_SRC\_COLOR, GL\_ONE\_MINUS\_SRC\_COLOR
  - GL\_DST\_COLOR, GL\_ONE\_MINUS\_DST\_COLOR
  - GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA
  - GL\_DST\_ALPHA, GL\_ONE\_MINUS\_DST\_ALPHA
  - GL\_CONSTANT\_COLOR, GL\_ONE\_MINUS\_CONSTANT\_COLOR
  - GL\_CONSTANT\_ALPHA, GL\_ONE\_MINUS\_CONSTANT\_ALPHA

# Pisanie do buforu głębokości

Modelowanie  
przezroczystości

Scena

Shadery

Programy

Programy

Sześcian

Funkcje

OpenGL

Sprajty  
punktowe

```
void glDepthMask(GLboolean flag);
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

# Sprajty punktowe

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

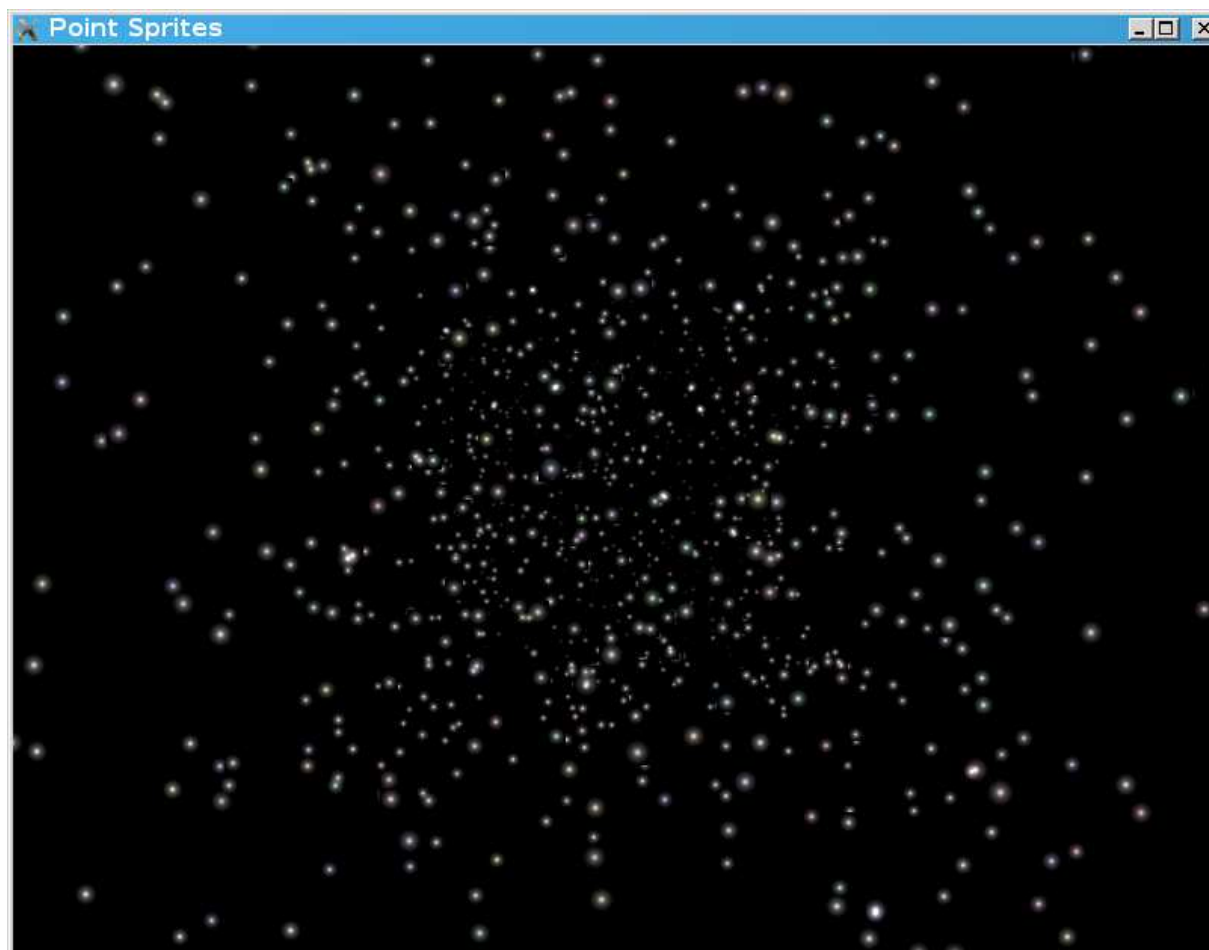
Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy



Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

- Punkty jako piksele  $n \times n$
- Tekstura na punktach
- Program cieniujący
- Animacja



# Shader wierzchołków. Parametry i zmienne

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
#version 430 core
```

```
layout (location = 0) in vec4 in_position;  
layout (location = 1) in vec4 in_color;
```

```
uniform float time;  
uniform mat4 projection_matrix;
```

```
flat out vec4 star_color;
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
void main(void){  
    vec4 vertex = in_position;  
  
    vertex.z += time;  
    vertex.z = fract(vertex.z);  
  
    float size = (20.0 * vertex.z * vertex.z);  
  
    star_color = smoothstep(1.0, 7.0, size) * in_color;  
  
    vertex.z = (999.9 * vertex.z) - 1000.0;  
    gl_Position = projection_matrix * vertex;  
    glPointSize = size;  
}
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
#version 430 core
```

```
layout (location = 0) out vec4 color;
```

```
uniform sampler2D texture_unit;
```

```
flat in vec4 star_color;
```

```
void main(void){
```

```
    color = star_color
```

```
        * texture(texture_unit, gl_PointCoord);
```

```
}
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
typedef struct ColorVertex{  
    float Position[4];  
    float Color[4];  
} StarVertex;
```

■ Już było

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
class ArrayModel{
public:
    ~ArrayModel();
protected:
    GLuint vao_;
    GLuint vertex_buffer_;
};
ArrayModel::~~ArrayModel(){
    glDisableVertexAttribArray(1);
    glDisableVertexAttribArray(0);
    glBindBuffer(GL_ARRAY_BUFFER, 0);
    glDeleteBuffers(1, &vertex_buffer_);
    glBindVertexArray(0);
    glDeleteVertexArrays(1, &vao_);
}
```

- Może dodać `glDisableVertexAttribArray(2), 3, 4?`

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

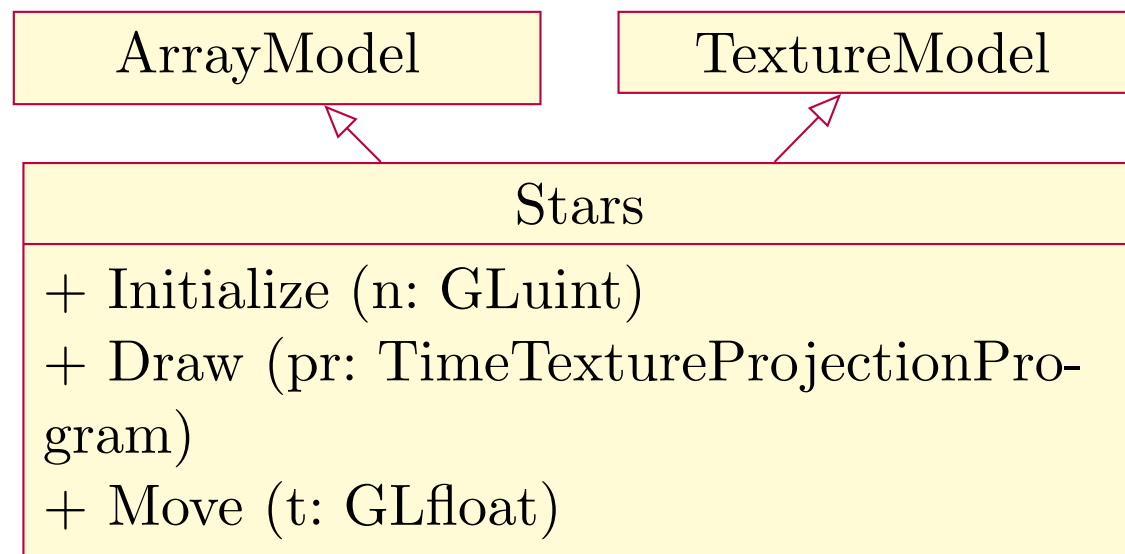
Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy



```
void Stars::Move(GLfloat t){
    time_ = t;
}
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
void Stars::Initialize(GLuint n){  
    n_ = n;  
    time_ = 0.0f;  
    srand(time(NULL));  
    glGenVertexArrays(1, &vao_);  
    glBindVertexArray(vao_);  
  
    glGenBuffers(1, &vertex_buffer_);  
    glBindBuffer(GL_ARRAY_BUFFER, vertex_buffer_);
```

# Mapowanie pamięci GPU na CPU

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
glBufferData(GL_ARRAY_BUFFER,
             n_ * sizeof(ColorVertex),
             NULL,
             GL_STATIC_DRAW);
ColorVertex * star
    = (ColorVertex*) glMapBufferRange(
        GL_ARRAY_BUFFER, 0,
        GL_MAP_WRITE_BIT | GL_MAP_INVALIDATE_BUFFER_BIT);
```



Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
for (unsigned int i = 0; i < n_; i++){
    star[i].position[0]
        = (random_float() * 2.0f - 1.0f) * 100.0f;
    star[i].position[1]
        = (random_float() * 2.0f - 1.0f) * 100.0f;
    star[i].position[2] = random_float();
    star[i].position[3] = 1.0f;
    star[i].color[0] = 0.8f + random_float() * 0.2f;
    star[i].color[1] = 0.8f + random_float() * 0.2f;
    star[i].color[2] = 0.8f + random_float() * 0.2f;
    star[i].color[3] = 1.0f;
}
glUnmapBuffer(GL_ARRAY_BUFFER);
```

# Argumenty shadera wierzchołków

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
glVertexAttribPointer(0, 4, GL_FLOAT,  
    GL_FALSE, sizeof(ColorVertex), NULL);  
glVertexAttribPointer(1, 4, GL_FLOAT,  
    GL_FALSE, sizeof(ColorVertex),  
    (void *) sizeof(star[0].position));  
glEnableVertexAttribArray(0);  
glEnableVertexAttribArray(1);
```

■ gdzie

```
float random_float(){  
    return ((float)rand() / RAND_MAX);  
}
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
void Stars::Draw(  
    const TimeTextureProjectionProgram &pr){  
    glUseProgram(pr);  
    pr.SetTime(time_);  
    glBindVertexArray(vao_);  
    glActiveTexture(texture_unit_);  
    glBindTexture(GL_TEXTURE_2D, texture_);  
    glEnable(GL_BLEND);  
    glBlendFunc(GL_ONE, GL_ONE);  
    glEnable(GL_PROGRAM_POINT_SIZE);  
    glDrawArrays(GL_POINTS, 0, n_);  
    glDisable(GL_BLEND);  
    glDisable(GL_PROGRAM_POINT_SIZE);  
    glBindVertexArray(0);  
    glUseProgram(0);  
}
```

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

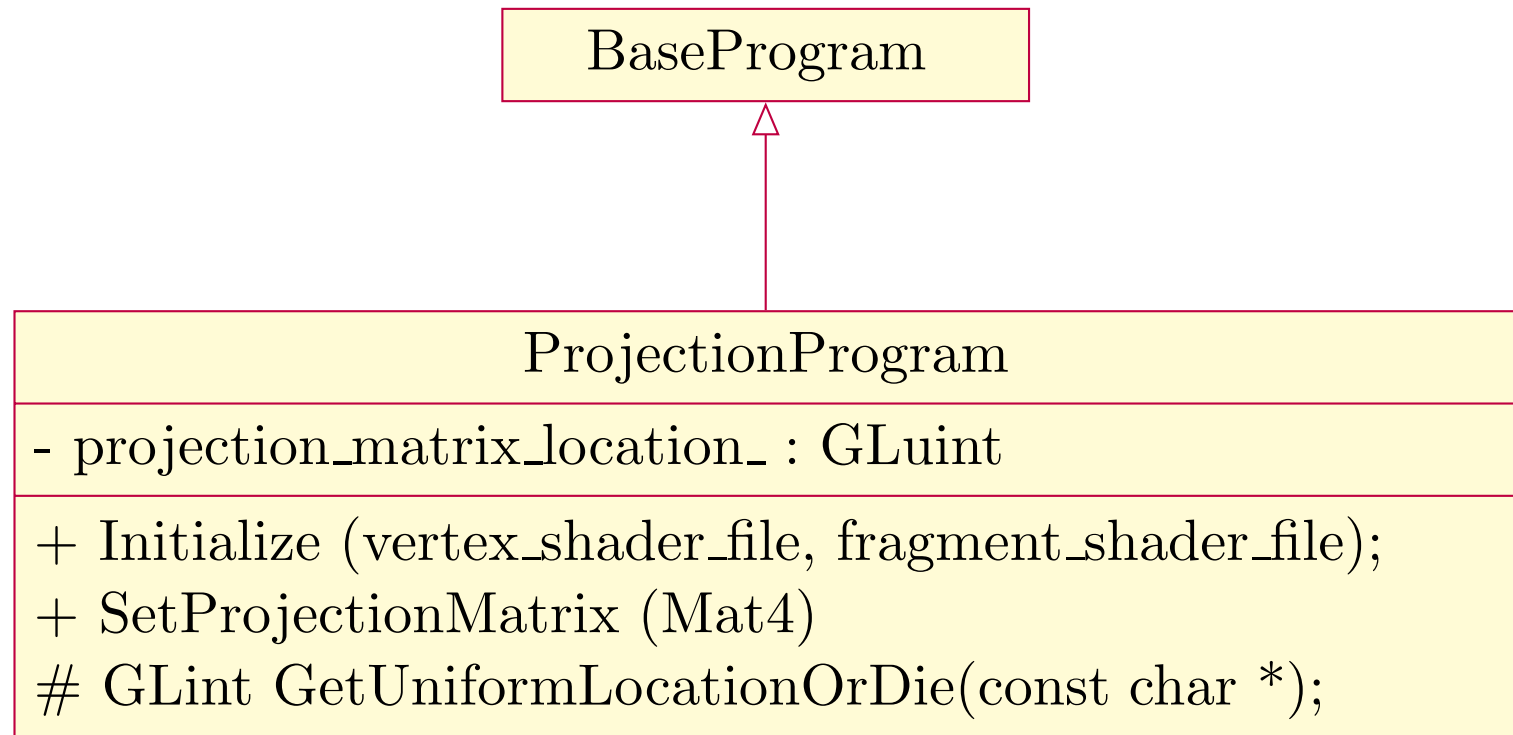
**Tekstura**

Program

Window

Tryb  
pełnoekranowy





# TextureProjectionProgram

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

ProjectionProgram



TextureProjectionProgram

- texture\_unit\_location\_ : GLuint

+ Initialize (vertex\_shader\_file, fragment\_shader\_file);

+ SetTextureUnit (GLfloat)

# TimeTextureProjectionProgram

TextureProjectionProgram



TimeTextureProjectionProgram

- time\_location\_ : GLuint

+ Initialize (vertex\_shader\_file, fragment\_shader\_file);

+ SetTime (GLfloat)

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
void Window::Render(){
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    stars_.Move(0.1*(float)clock()/CLOCKS_PER_SEC);
    stars_.Draw(program_);

    glutSwapBuffers();
    glutPostRedisplay();
}
```



Modelowanie  
przezroczystości

Sprajty  
punktowe

Scena

Shadery

Punkty

Tekstura

Program

Window

Tryb  
pełnoekranowy

```
void glutFullScreen();
```