Informatyka praktyczna. SkyBox

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

SkyBox

Najnowsza wersja tego dokumentu dostępna jest pod adresem

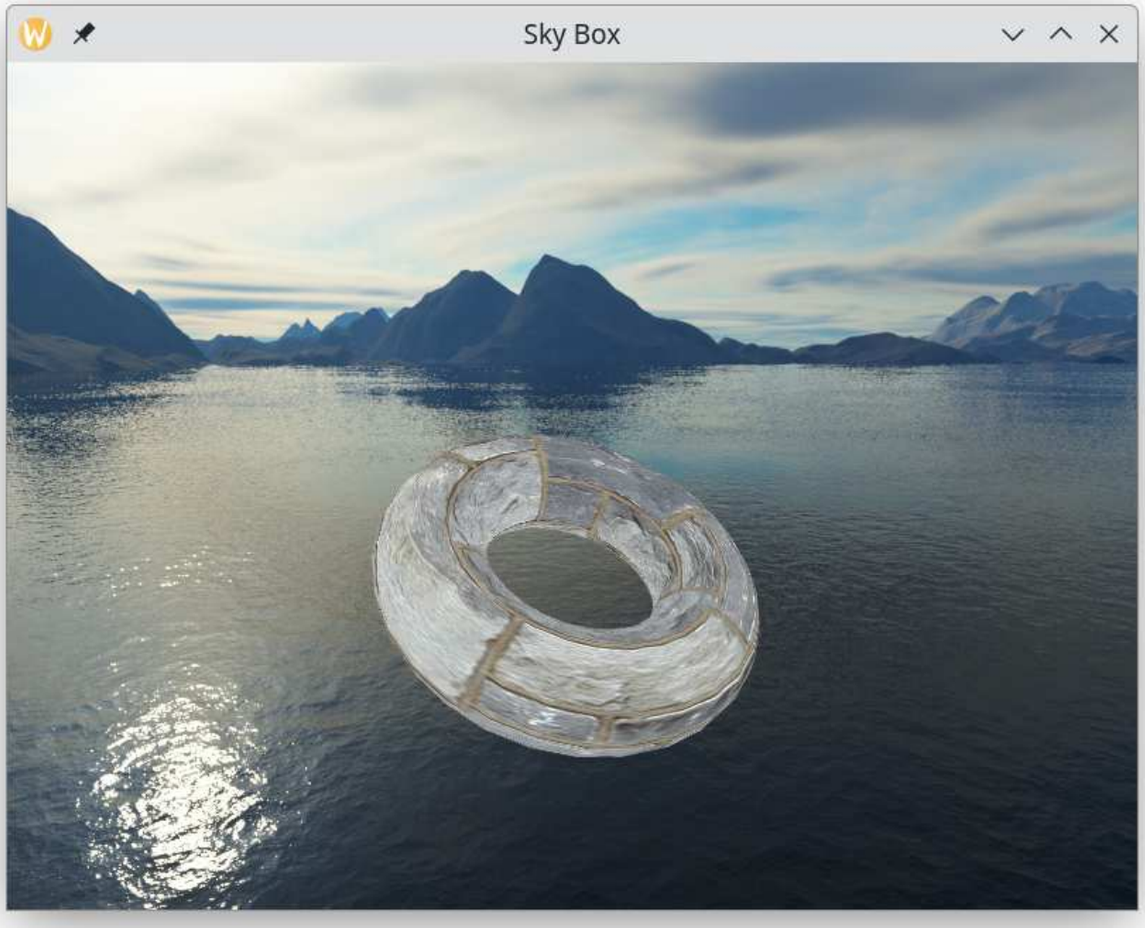
http://wmii.uwm.edu.pl/~denisjuk/uwm

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- `vertices.h`
- `SkyBox`
- `CubeTexture`

SkyBox

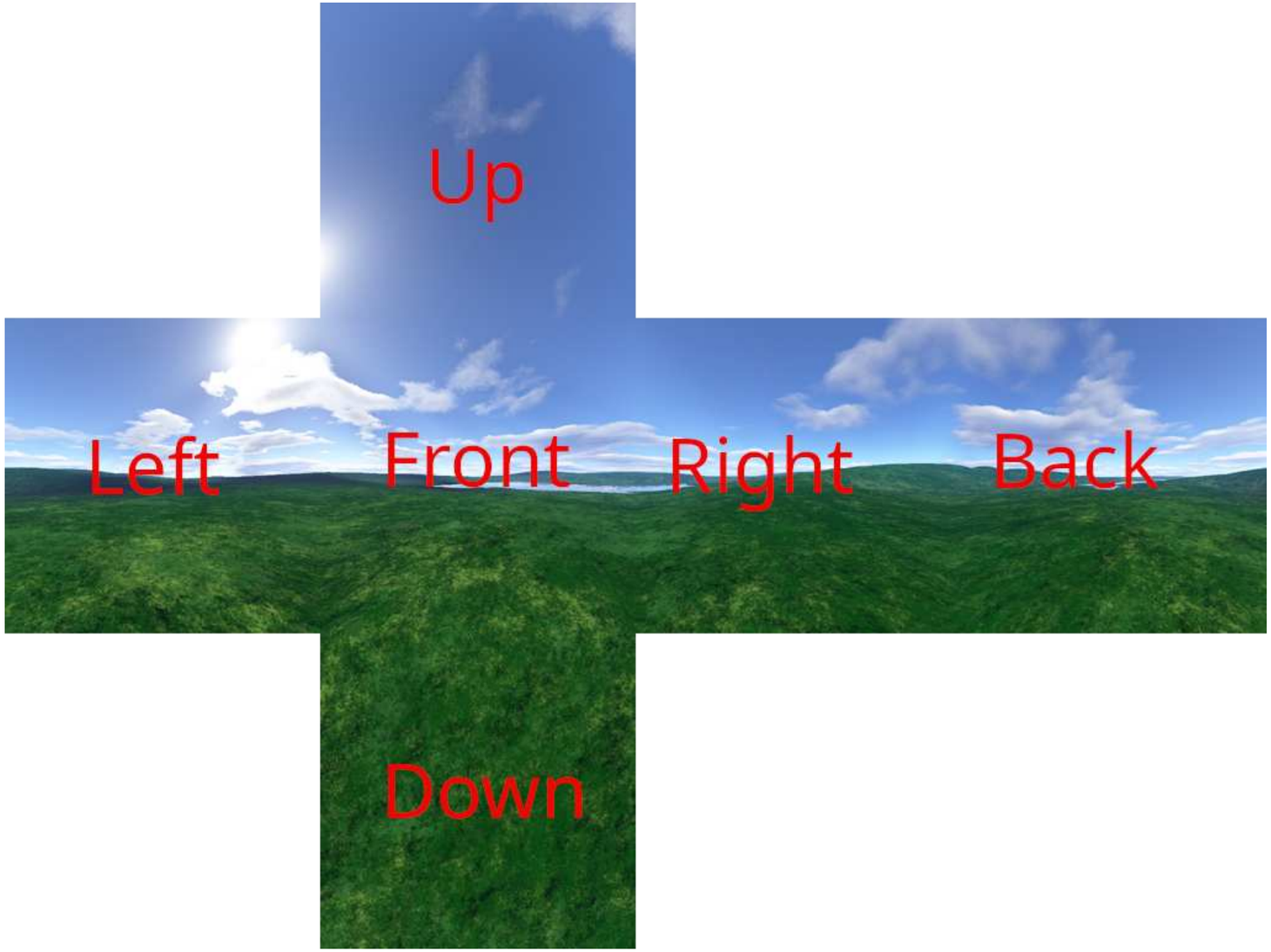
Modelowanie środowiska

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- `vertices.h`
- `SkyBox`
- `CubeTexture`



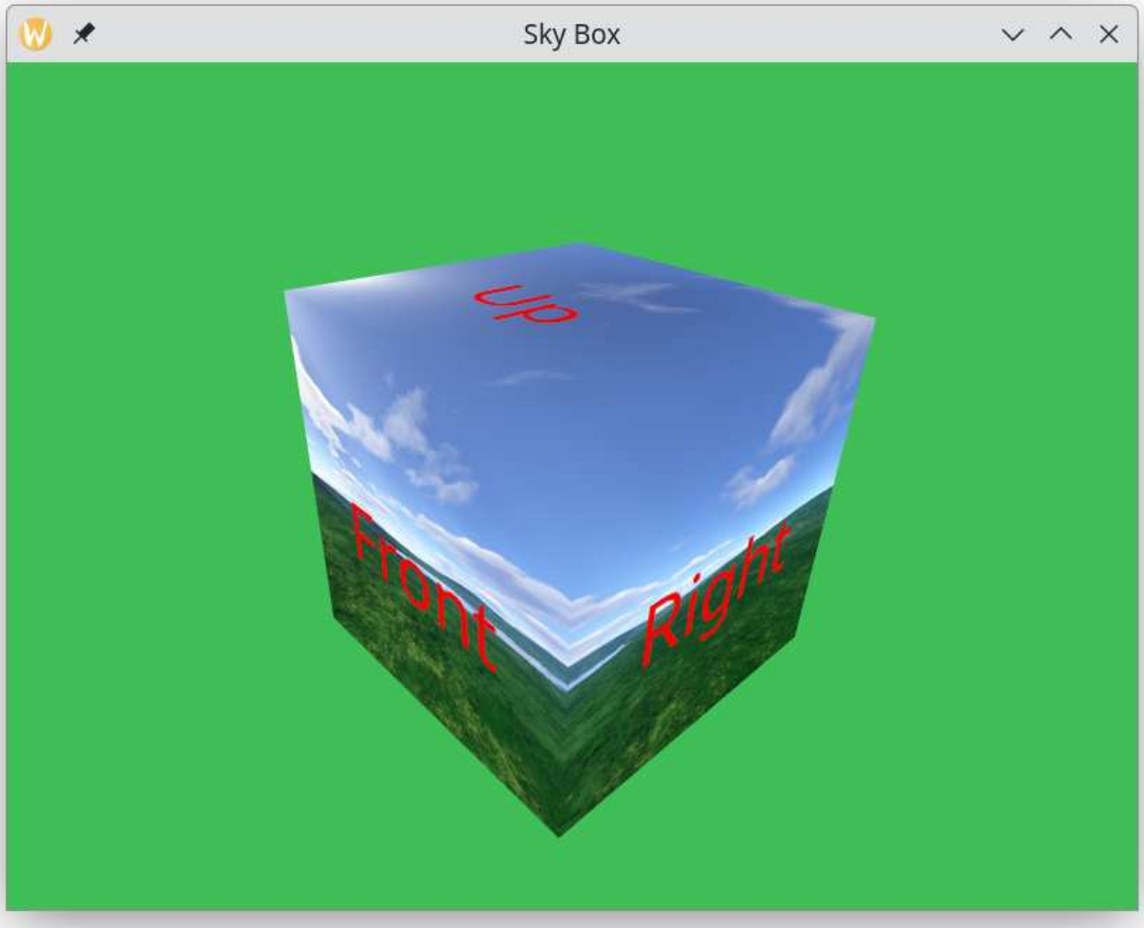
Tekstura Sześcienna

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- `vertices.h`
- `SkyBox`
- `CubeTexture`



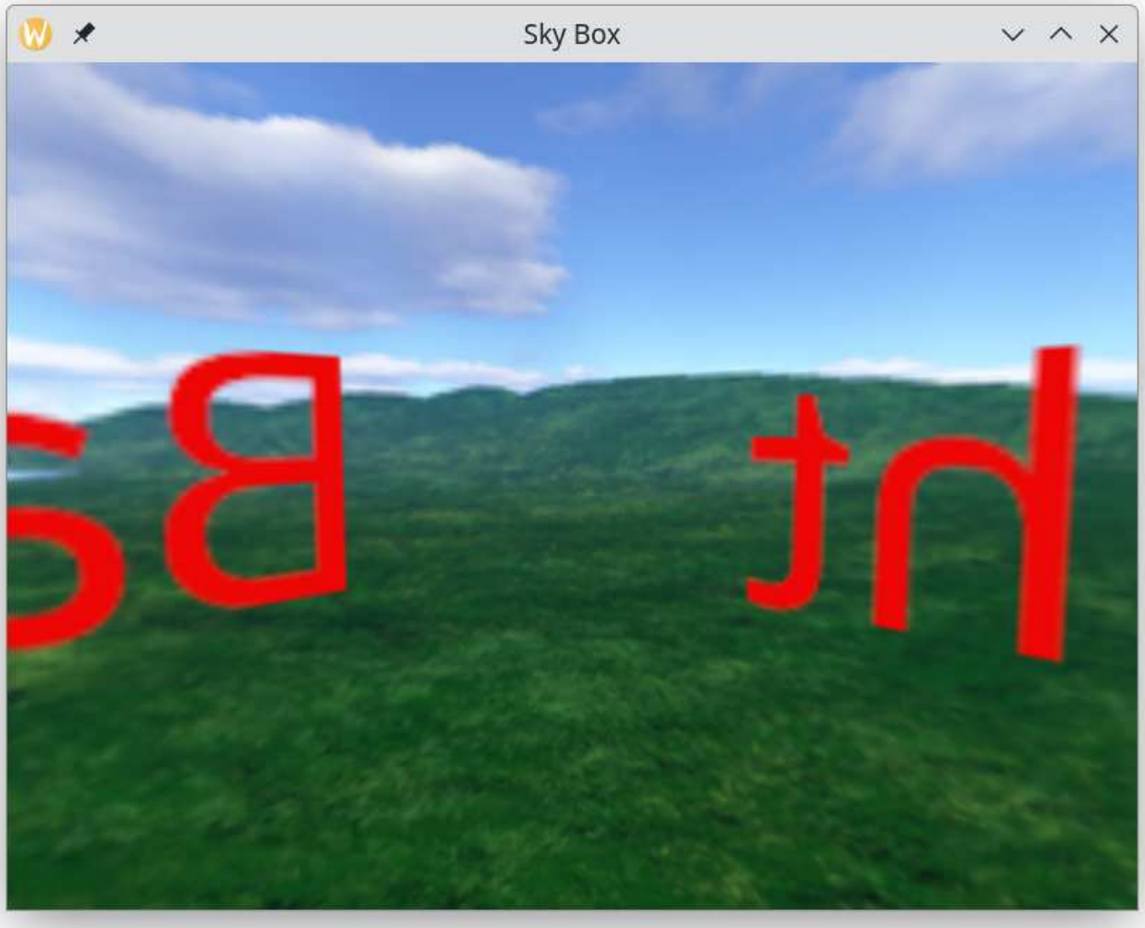
SkyBox od zewnątrz

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- `vertices.h`
- `SkyBox`
- `CubeTexture`



SkyBox od wewnątrz

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- `vertices.h`
- `SkyBox`
- `CubeTexture`



Shader Wierzchołków

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

```
#version 430 core
layout(location=0) in vec4 in_position;
layout(location=2) in vec3 in_texture;

uniform mat4 model_matrix;
uniform mat4 view_matrix;
uniform mat4 projection_matrix;

out vec3 tex_coord;

void main(void){
  gl_Position = (projection_matrix * view_matrix
     * model_matrix) * in_position;
  tex_coord = in_texture;
}
```

Shader Fragmentów

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

```
#version 430 core

layout (location = 0) out vec4 color;

in vec3 tex_coord;

uniform samplerCube texture_unit;

void main(void){
    color = texture(texture_unit, tex_coord);
}
```

Nowe klasy C++

SkyBox
❖ Modelowanie środowiska
❖ Shadery
❖ Nowe klasy C++
❖ vertices.h
❖ SkyBox
❖ CubeTexture

- `SkyBox` — model nieba (sześcian) z teksturą
- `CubeTexture` — tekstura sześcienna (sześć obrazków)

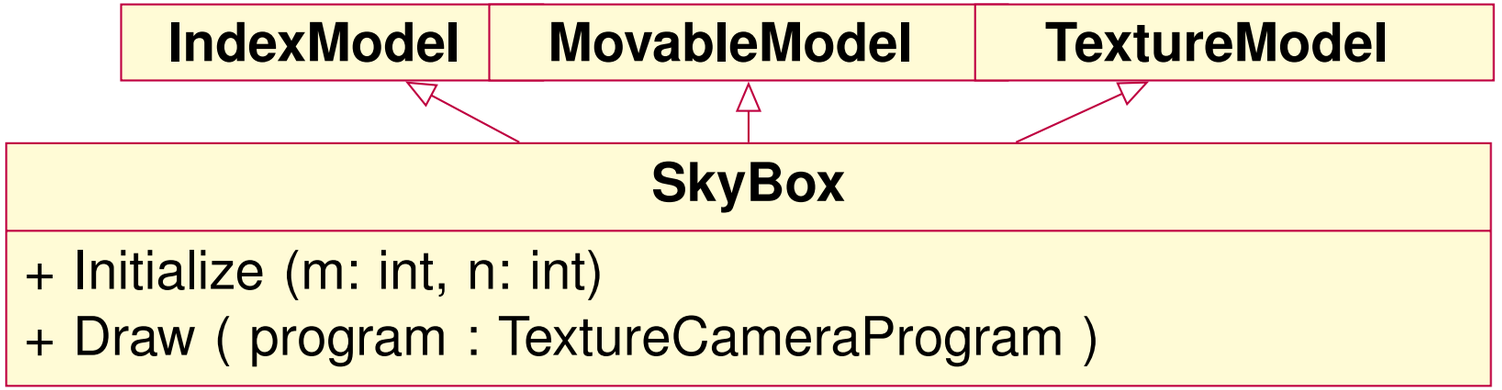
Nowa struktura dla wierzchołków

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

```
typedef struct SkyTextureVertex{
    float position[4];
    float texture[3];
} SkyTextureVertex;
```

Klasa SkyBox

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture




```
class SkyBox: public IndexModel,
    public MovableModel, public TextureModel{
public:
     void Initialize();
     void Draw(const TextureCameraProgram & program) const;
};
```

Wierzchołki

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

```
const SkyTextureVertex kVertices[8] =
{
  //front
  {{-1.0f, 1.0f, 1.0f, 1.0f}, {-1.0f, 1.0f, 1.0f}}, //0
  {{ 1.0f, 1.0f, 1.0f, 1.0f}, {1.0f, 1.0f, 1.0f}},  //1
  {{ 1.0f,-1.0f, 1.0f, 1.0f}, {1.0f, -1.0f, 1.0f}}, //2
  {{-1.0f,-1.0f, 1.0f, 1.0f}, {-1.0f, -1.0f, 1.0f}},//3
  // back
  {{ 1.0f, 1.0f,-1.0f, 1.0f}, {1.0f, 1.0f, -1.0f}}, //4
  {{-1.0f, 1.0f,-1.0f, 1.0f}, {-1.0f, 1.0f, -1.0f}}, //5
  {{-1.0f,-1.0f,-1.0f, 1.0f}, {-1.0f, -1.0f, -1.0f}}, //6
  {{ 1.0f,-1.0f,-1.0f, 1.0f}, {1.0f, -1.0f, -1.0f}} //7
};
```

- Możliwa optymalizacja: wykorzystać współrzędne (x, y, z) jako współrzędne teksturowe

Przenoszenie środka tam gdzie jest kamera

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

- Kamera:

```
view_matrix_.Translate(0, -3, -10);
```

- SkyBox:

```
model_matrix_.Translate(0, 3, 10);
```

Wyświetlenie

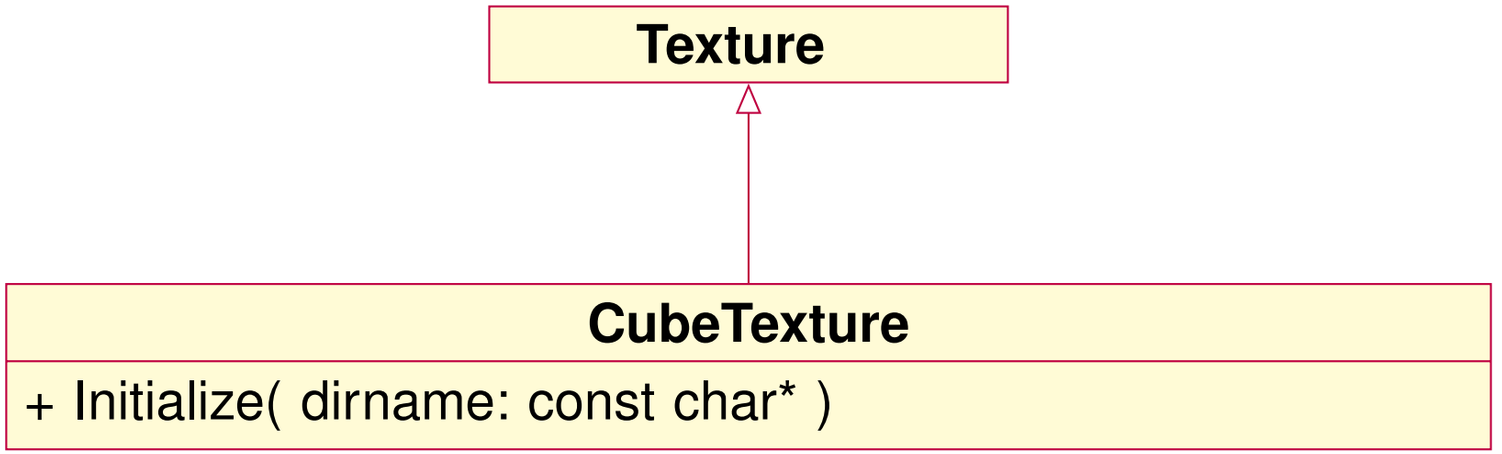
SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

- Jako pierwszy, nie piszemy do buforu głębokości:

```
glDepthMask(GL_FALSE);
glDrawElements(GL_TRIANGLES, 36, GL_UNSIGNED_INT, nullptr);
glDepthMask(GL_TRUE);
```

Klasa `CubeTexture`

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- `vertices.h`
- `SkyBox`
- `CubeTexture`




```
class CubeTexture: public Texture{
public:
    void Initialize(const char *dirname);
};
```

Inicjalizacja

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

```
glGenTextures(1, &texture_);

glBindTexture(GL_TEXTURE_CUBE_MAP, texture_);

glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_MIN_FILTER, GL_LI
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_MAG_FILTER, GL_LI

glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_S, GL_CLAMP_
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_T, GL_CLAMP_
glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_R, GL_CLAMP_
```

6 obrazków

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

Bok	W nazwie pliku	OpenGL target
Left	x−	GL_TEXTURE_CUBE_MAP_NEGATIVE_X
Right	x+	GL_TEXTURE_CUBE_MAP_POSITIVE_X
Down	y−	GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
Up	y+	GL_TEXTURE_CUBE_MAP_POSITIVE_Y
Back	z−	GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
Front	z+	GL_TEXTURE_CUBE_MAP_POSITIVE_Z

6 obrazków. Kod

SkyBox
- Modelowanie środowiska
- Shadery
- Nowe klasy C++
- vertices.h
- SkyBox
- CubeTexture

```
for (const auto & entry : fs::directory_iterator(dirname)){
 std::string file_name = entry.path().string();
 if( file_name.find(".tga") !=std::string::npos){
  if( file_name.find("x+") !=std::string::npos){
    LoadTGAFileOrDie(GL_TEXTURE_CUBE_MAP_POSITIVE_X,
          file_name.c_str());
  }
  else if( file_name.find("x-") !=std::string::npos){
    LoadTGAFileOrDie(GL_TEXTURE_CUBE_MAP_NEGATIVE_X,
          file_name.c_str());
  }
  else if( file_name.find("y+") !=std::string::npos){
    LoadTGAFileOrDie(GL_TEXTURE_CUBE_MAP_POSITIVE_Y,
          file_name.c_str());
  }

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