

Grafika komputerowa

Lista zadań nr 3

Napisz aplikację implementującą obrót obiektu trójwymiarowego. Zalecana technologia implementacji: DirectX.

Kryteria oceny:

- ocena dostateczna: obrót czworościanu i sześciianu,
- ocena dobra: obrót walca,
- ocena bardzo dobra: obrót kuli utworzonej z n trójkątów,
- ocena celująca: dodanie efektu świetlnych.

Marcin Masalski

Kod pomocniczy:

```
array<CustomVertex::PositionColored>^ czworoscian;  
array<CustomVertex::PositionColored>^ szescian;  
float obrot;  
Device^ device;  
  
void InicjalizujUrzadzenie() {  
    SetStyle(ControlStyles::DoubleBuffer, true);  
    SetStyle(ControlStyles::AllPaintingInWmPaint, true);  
    SetStyle(ControlStyles::UserPaint, true);  
    SetStyle(ControlStyles::SupportsTransparentBackColor, false);  
    SetStyle(ControlStyles::Opaque, false);  
    SetStyle(ControlStyles::OptimizedDoubleBuffer, true);  
    SetStyle(ControlStyles::ResizeRedraw, true);  
  
    FormBorderStyle = Windows::Forms::FormBorderStyle::None;  
    Top = 0; Left = 0;  
    Width = Screen::PrimaryScreen::get()->Bounds.Width;  
    Height = Screen::PrimaryScreen::get()->Bounds.Height;  
  
    PresentParameters^ pp = gcnew PresentParameters();  
    pp->SwapEffect = SwapEffect::Discard;  
    pp->Windowed = true;
```

```

pp->EnableAutoDepthStencil = true;
pp->AutoDepthStencilFormat = DepthFormat::D16;
device = gcnew Device(0, DeviceType::Hardware, this, CreateFlags::SoftwareVertexProcessing, pp);
device->Transform->View = Matrix::LookAtLH(Vector3(0, 0, -8), Vector3(0, 0, 0), Vector3(0, 1, 0));
device->Transform->Projection = Matrix::PerspectiveFovLH((float)Math::PI / 4.0f, (float)ClientSize.Width /
ClientSize.Height, 1.0f, 100.0f);
device->Transform->World = Matrix::Identity;
device->RenderState->CullMode = Cull::CounterClockwise;
device->RenderState->Lighting = false;
obrot = 0.0f;
}

```

```

void UtworzObjekt() {
czworoscian = gcnew array<CustomVertex::PositionColored>(12);
czworoscian[0] = CustomVertex::PositionColored(-1.0f, -1.0f, -1.0f, Color::Green.ToArgb());
czworoscian[1] = CustomVertex::PositionColored(0.0f, 1.0f, 0.0f, Color::Red.ToArgb());
czworoscian[2] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Blue.ToArgb());

czworoscian[3] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Blue.ToArgb());
czworoscian[4] = CustomVertex::PositionColored(0.0f, 1.0f, 0.0f, Color::Red.ToArgb());
czworoscian[5] = CustomVertex::PositionColored(1.0f, -1.0f, 1.0f, Color::Green.ToArgb());

czworoscian[6] = CustomVertex::PositionColored(1.0f, -1.0f, 1.0f, Color::Green.ToArgb());
czworoscian[7] = CustomVertex::PositionColored(0.0f, 1.0f, 0.0f, Color::Red.ToArgb());
czworoscian[8] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Blue.ToArgb());

czworoscian[9] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Blue.ToArgb());
czworoscian[10] = CustomVertex::PositionColored(0.0f, 1.0f, 0.0f, Color::Red.ToArgb());
czworoscian[11] = CustomVertex::PositionColored(-1.0f, -1.0f, -1.0f, Color::Green.ToArgb());

szescian = gcnew array<CustomVertex::PositionColored>(48);
szescian[0] = CustomVertex::PositionColored(-1.0f, -1.0f, -1.0f, Color::Blue.ToArgb());
szescian[1] = CustomVertex::PositionColored(-1.0f, 1.0f, -1.0f, Color::Blue.ToArgb());
szescian[2] = CustomVertex::PositionColored(1.0f, 1.0f, -1.0f, Color::Blue.ToArgb());
szescian[3] = CustomVertex::PositionColored(1.0f, 1.0f, -1.0f, Color::Blue.ToArgb());
szescian[4] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Blue.ToArgb());
szescian[5] = CustomVertex::PositionColored(-1.0f, -1.0f, -1.0f, Color::Blue.ToArgb());

szescian[6] = CustomVertex::PositionColored(1.0f, -1.0f, 1.0f, Color::Blue.ToArgb());
szescian[7] = CustomVertex::PositionColored(1.0f, 1.0f, 1.0f, Color::Blue.ToArgb());

```

```

szescian[8] = CustomVertex::PositionColored(-1.0f, 1.0f, 1.0f, Color::Blue.ToArgb());
szescian[9] = CustomVertex::PositionColored(-1.0f, 1.0f, 1.0f, Color::Blue.ToArgb());
szescian[10] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Blue.ToArgb());
szescian[11] = CustomVertex::PositionColored(1.0f, -1.0f, 1.0f, Color::Blue.ToArgb());

szescian[12] = CustomVertex::PositionColored(-1.0f, 1.0f, -1.0f, Color::Red.ToArgb());
szescian[13] = CustomVertex::PositionColored(-1.0f, 1.0f, 1.0f, Color::Red.ToArgb());
szescian[14] = CustomVertex::PositionColored(1.0f, 1.0f, 1.0f, Color::Red.ToArgb());
szescian[15] = CustomVertex::PositionColored(1.0f, 1.0f, 1.0f, Color::Red.ToArgb());
szescian[16] = CustomVertex::PositionColored(1.0f, 1.0f, -1.0f, Color::Red.ToArgb());
szescian[17] = CustomVertex::PositionColored(-1.0f, 1.0f, -1.0f, Color::Red.ToArgb());

szescian[18] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Red.ToArgb());
szescian[19] = CustomVertex::PositionColored(1.0f, -1.0f, 1.0f, Color::Red.ToArgb());
szescian[20] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Red.ToArgb());
szescian[21] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Red.ToArgb());
szescian[22] = CustomVertex::PositionColored(-1.0f, -1.0f, -1.0f, Color::Red.ToArgb());
szescian[23] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Red.ToArgb());

szescian[24] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Green.ToArgb());
szescian[25] = CustomVertex::PositionColored(-1.0f, 1.0f, 1.0f, Color::Green.ToArgb());
szescian[26] = CustomVertex::PositionColored(-1.0f, 1.0f, -1.0f, Color::Green.ToArgb());
szescian[27] = CustomVertex::PositionColored(-1.0f, 1.0f, -1.0f, Color::Green.ToArgb());
szescian[28] = CustomVertex::PositionColored(-1.0f, -1.0f, -1.0f, Color::Green.ToArgb());
szescian[29] = CustomVertex::PositionColored(-1.0f, -1.0f, 1.0f, Color::Green.ToArgb());

szescian[30] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Green.ToArgb());
szescian[31] = CustomVertex::PositionColored(1.0f, 1.0f, -1.0f, Color::Green.ToArgb());
szescian[32] = CustomVertex::PositionColored(1.0f, 1.0f, 1.0f, Color::Green.ToArgb());
szescian[33] = CustomVertex::PositionColored(1.0f, 1.0f, 1.0f, Color::Green.ToArgb());
szescian[34] = CustomVertex::PositionColored(1.0f, -1.0f, 1.0f, Color::Green.ToArgb());
szescian[35] = CustomVertex::PositionColored(1.0f, -1.0f, -1.0f, Color::Green.ToArgb());
}
void Rysuj() {
    device->Clear(ClearFlags::Target | ClearFlags::ZBuffer, Color::White, 1.0f, 0);
    device->BeginScene();
    device->VertexFormat = CustomVertex::PositionColored::Format;
    device->Transform->World = Matrix::RotationY(obrot) * Matrix::Translation(-2.0f, 0.0f, 0.0f);
    device->DrawUserPrimitives(PrimitiveType::TriangleList, czworoscian->Length/3, czworoscian);
}

```

```
    device->Transform->World = Matrix::RotationYawPitchRoll(0.0f, obrot, obrot) * Matrix::Translation(2.0f, 0.0f,
0.0f);
    device->DrawUserPrimitives(PrimitiveType::TriangleList, szescian->Length/3, szescian);
    device->EndScene();
    device->Present();
}
```