Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 4**

**по дисциплине**

**«Программирование компьютерной графики»**

**Тема: «Введение в OpenGL»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Жигалов И.Е.

Владимир, 2020

ЦЕЛЬ РАБОТЫ

Изучение принципов применения библиотеки OpenGL при разработке приложений в C#.

ЗАДАНИЕ

ХОД РАБОТЫ

TAO framework установить не удалось поэтому я скачала zip архив, добавила в переменную среды PATH каталог C:\Program Files\TaoFramework\lib. Далее я создала новый проект и добавила в него ссылки на dll файлы (Рисунок 1). После добавила в панель элементов новый элементSimpleOpenGLControl (Рисунок 2).

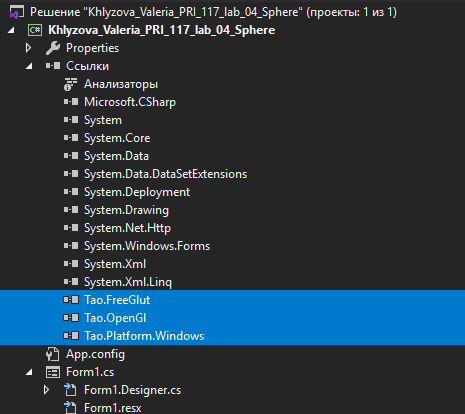


Рисунок 1. Добавленные библиотеки.

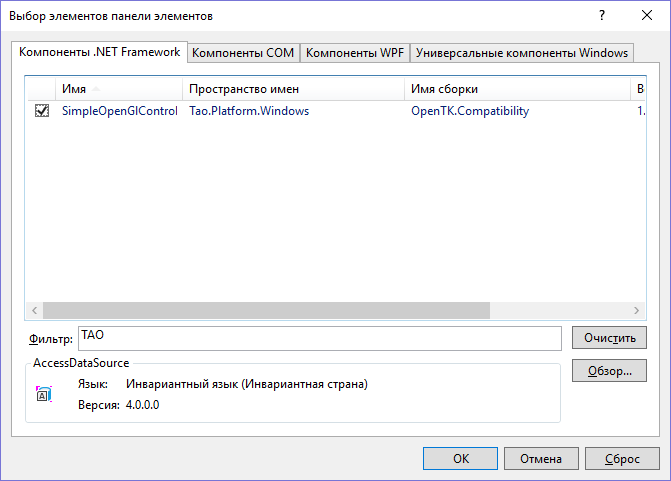


Рисунок 2. Добавление нового элемента.

Код программы (Вариант 15):

using System;

using System.Windows.Forms;

using Tao.OpenGl;

using Tao.FreeGlut;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_04\_Sphere

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void Form1\_Load(object sender, EventArgs e)

{

Glut.glutInit();

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

Glu.gluPerspective(45, (float)AnT.Width / (float)AnT.Height, 0.1, 200);

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

Gl.glLoadIdentity();

Gl.glEnable(Gl.GL\_COLOR\_MATERIAL);

Gl.glEnable(Gl.GL\_DEPTH\_TEST);

Gl.glEnable(Gl.GL\_LIGHTING);

Gl.glEnable(Gl.GL\_LIGHT0);

}

private void simpleOpenGlControl1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glColor3ub(255, 0, 0);

Gl.glPushMatrix();

Gl.glTranslated(0, 0, -6);

Gl.glRotated(45, 1, 1, 0);

Glut.glutWireSphere(2, 32, 32);

Gl.glPopMatrix();

Gl.glFlush();

AnT.Invalidate();

}

private void Khlyzova\_Valeria\_PRI\_117\_lab\_04(object sender, EventArgs e)

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glColor3ub(0, 0, 255);

Gl.glPushMatrix();

Gl.glTranslated(0, 0, -6);

Gl.glRotated(45, 1, 1, 0);

Glut.glutSolidSphere(2, 16, 18);

Gl.glPopMatrix();

Gl.glFlush();

AnT.Invalidate();

}

}

}

Выполнение программы:

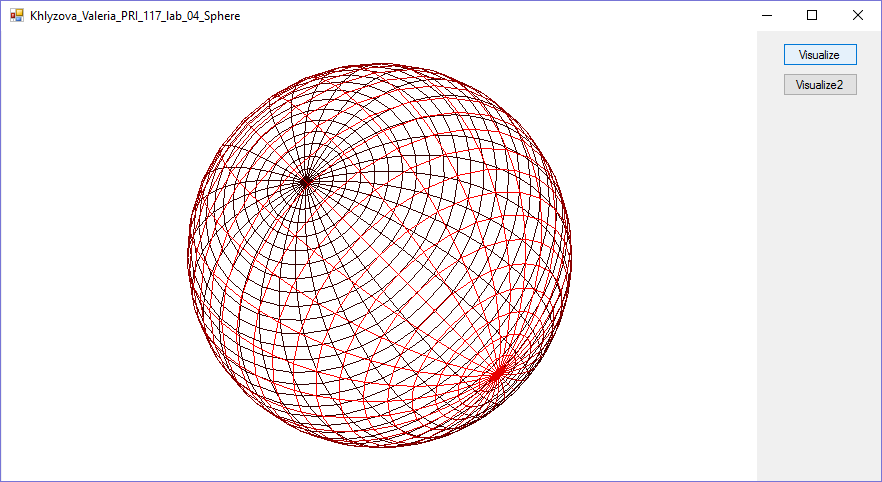


Рисунок 3. Разобранный пример.

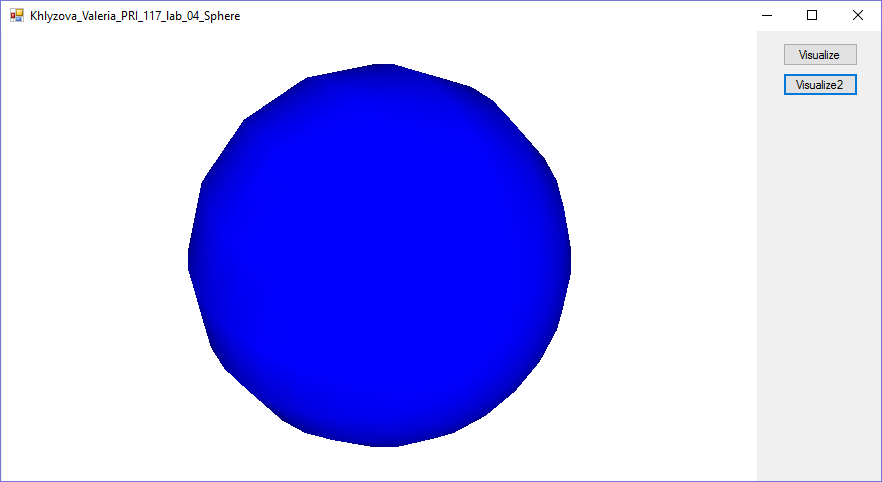


Рисунок 4. Выполненное задание.

Процесс установки OpenTK:

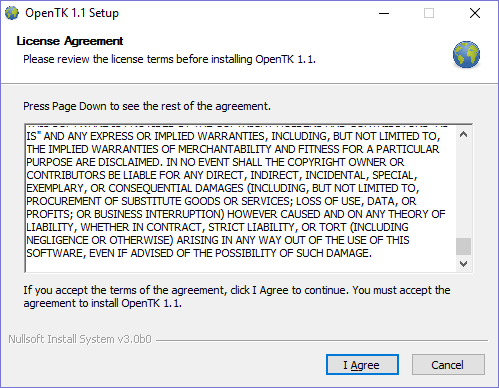


Рисунок 5. Диалог с лицензионным соглашением для библиотеки OpenTK.

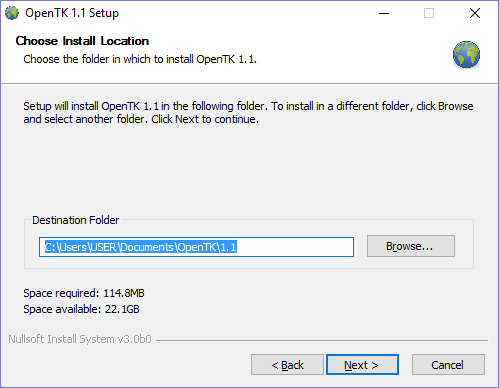


Рисунок 6. Диалог выбора каталога для установки библиотеки OpenTK.

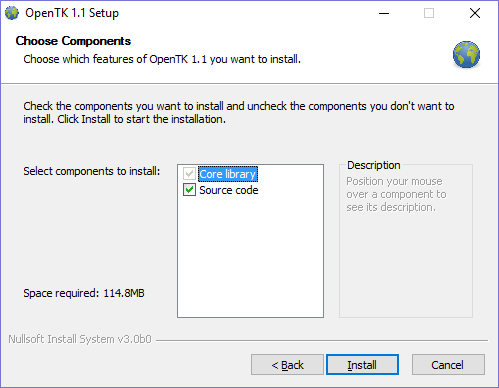


Рисунок 7. Выбор устанавливаемых компонентов библиотеки OpenTK.

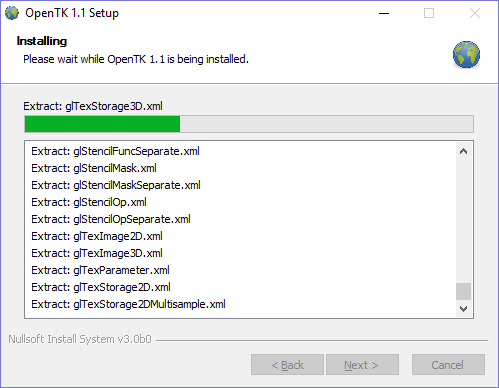


Рисунок 8. Диалог с информацией о процессе установки выбранных компонентов библиотеки OpenTK.

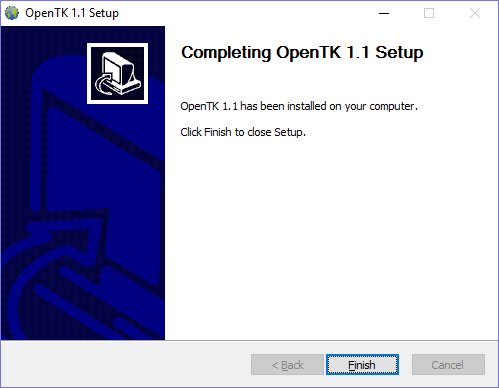


Рисунок 9. Диалог с информацией об успешной установке библиотеки OpenTK.

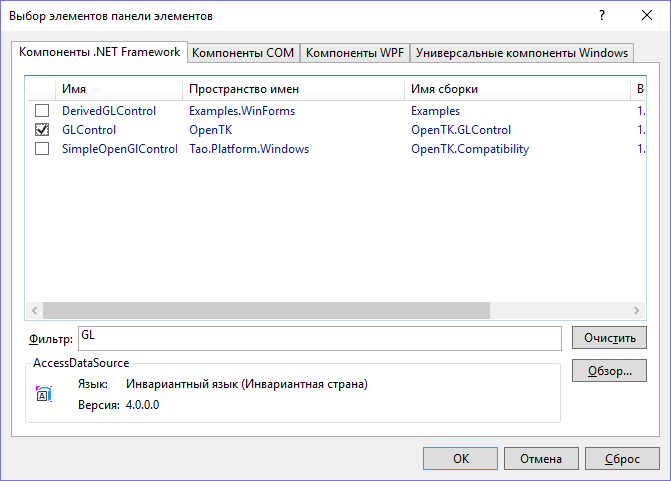


Рисунок 10. Выбираем элемент ля дальнейшей работы.

Код программы:

using System;

using System.Windows.Forms;

using OpenTK.Graphics.OpenGL;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_04

{

public partial class Form1 : Form

{

float AngleX = 30;

float AngleY = 30;

float AngleZ = 30;

const float AngleDl = 5;

public Form1()

{

InitializeComponent();

}

private void glControl1\_Load(object sender, EventArgs e)

{

GL.Viewport(0, 0, glControl1.Width, glControl1.Height);

GL.MatrixMode(MatrixMode.Projection);

GL.LoadIdentity();

GL.Ortho(-1, 1, -1, 1, -1, 1);

GL.MatrixMode(MatrixMode.Modelview);

glControl1.Invalidate();

}

private void glControl1\_Paint(object sender, PaintEventArgs e)

{

GL.ClearColor(0.5f, 0.5f, 0.75f, 1.0f);

GL.Clear(ClearBufferMask.ColorBufferBit | ClearBufferMask.DepthBufferBit);

GL.LoadIdentity();

GL.Rotate(AngleX, 1.0, 0.0, 0.0);

GL.Rotate(AngleY, 0.0, 1.0, 0.0);

GL.Rotate(AngleZ, 0.0, 0.0, 1.0);

GL.Begin(BeginMode.Lines);

GL.Color3(1f, 0f, 0f); GL.Vertex3(-1f, 0f, 0f); GL.Vertex3(1f, 0f, 0f);

GL.Color3(0f, 1f, 0f); GL.Vertex3(0f, -1f, 0f); GL.Vertex3(0f, 1f, 0f);

GL.Color3(0f, 0f, 1f); GL.Vertex3(0f, 0f, -1f); GL.Vertex3(0f, 0f, 1f);

GL.End();

GL.Flush();

GL.Finish();

glControl1.SwapBuffers();

}

private void glControl1\_KeyDown(object sender, KeyEventArgs e)

{

switch (e.KeyCode)

{

case Keys.Insert:

AngleX += AngleDl;

break;

case Keys.Delete:

AngleX -= AngleDl;

break;

case Keys.Home:

AngleY += AngleDl;

break;

case Keys.End:

AngleY -= AngleDl;

break;

case Keys.Prior:

AngleZ += AngleDl;

break;

case Keys.Next:

AngleZ -= AngleDl;

break;

}

glControl1.Invalidate();

}

}

}

Выполнение программы:

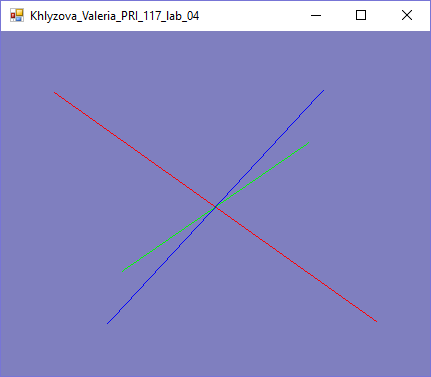


Рисунок 11. Результат выполнения программы.

ВЫВОД

В ходе выполнения лабораторной работы были изучены принципы применения библиотеки OpenGL при разработке приложений в C#.