Министерство образования и науки Российской

**Федерации Федеральное государственное бюджетное образовательное учреждение 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»**

**По теме: «Введение в OpenGL**

Выполнила:

ст. гр. ПРИ-118

Левченко В.П

Принял:

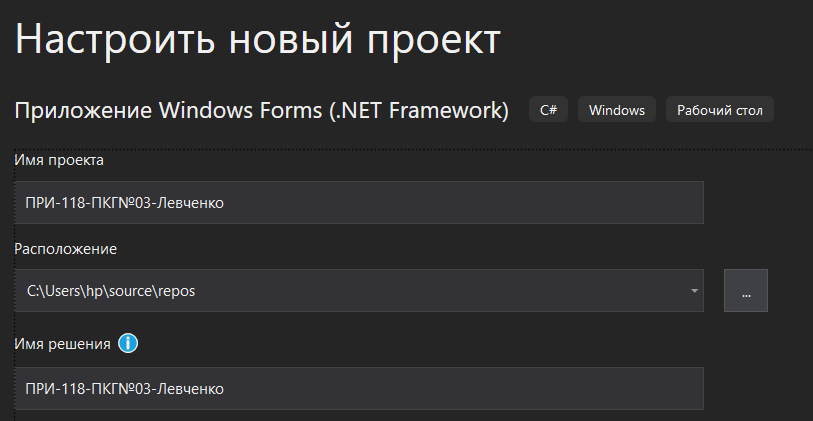
Жигалов И.Е

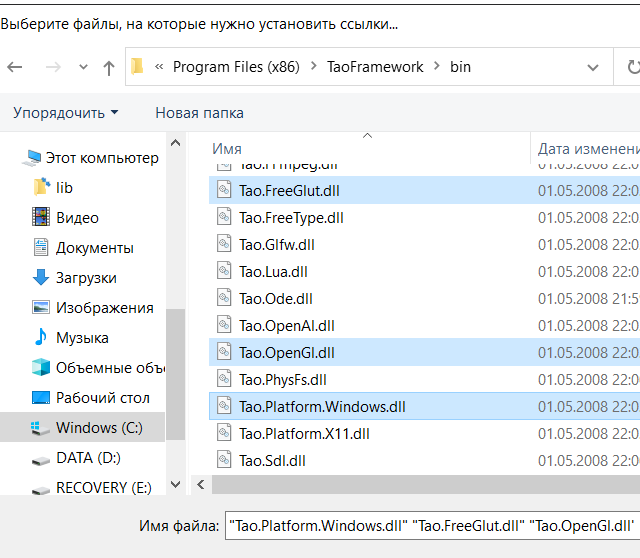
Владимир, 2021

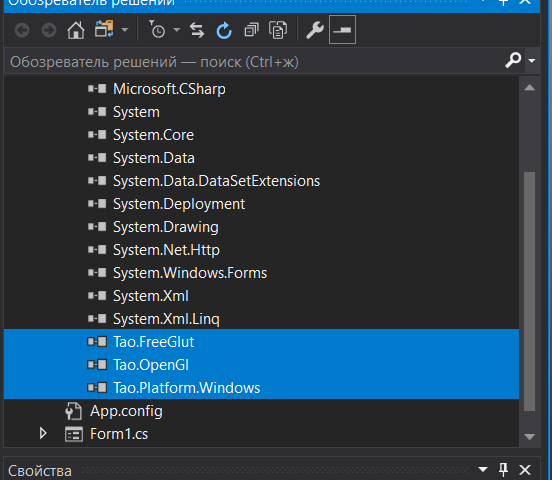
**Цель работы**

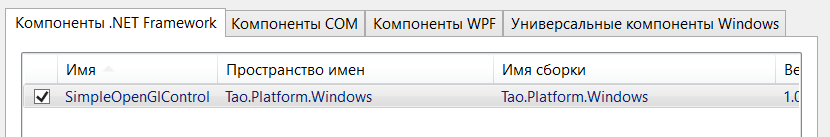
Изучение принципов применения библиотеки OpenGL при разработке приложений в C#.

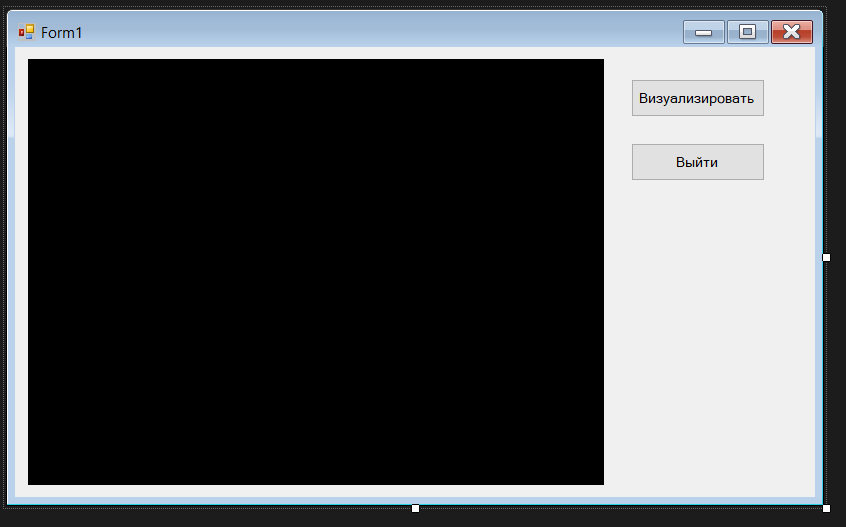
**Выполнение работы**

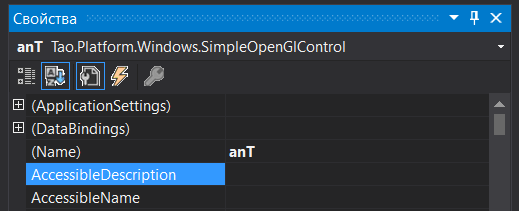
  
Рисунок 1. Создание проекта

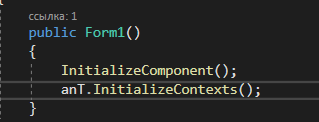
  
Рисунок 2. Процесс добавления библиотек

  
Рисунок 3. Результат добавления

  
Рисунок 4. Добавление элемента SimpleOpenGLControl

  
Рисунок 5. Результат размещения элементов

  
Рисунок 6. Изменение свойства

  
Рисунок 7. Инициализация работы элемента

Полный код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

// для работы с библиотекой OpenGL

using Tao.OpenGl;

// для работы с библиотекой FreeGLUT

using Tao.FreeGlut;

// для работы с элементом управления SimpleOpenGLControl

using Tao.Platform.Windows;

namespace ПРИ\_118\_ПКГ\_03\_Левченко

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

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

{

// инициализация Glut

Glut.glutInit();

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

// очитка окна

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

// установка порта вывода в соотвествии с размерами элемента anT

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

// настройка проекции

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();

// настройка параметров OpenGL для визуализации

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

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

Gl.glEnable(Gl.GL\_LIGHTING);

Gl.glEnable(Gl.GL\_LIGHT0);

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

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);

// рисуем сферу с помощью библиотеки FreeGLUT

Glut.glutWireSphere(2, 32, 32);

Gl.glPopMatrix();

Gl.glFlush();

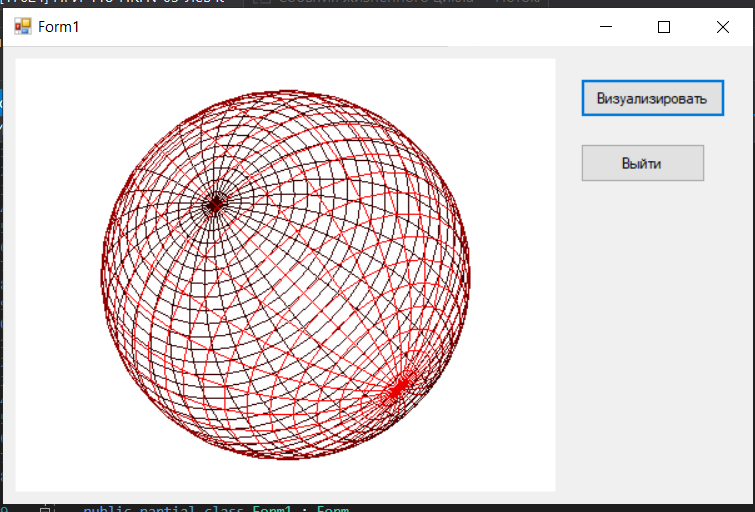
AnT.Invalidate();

}

}

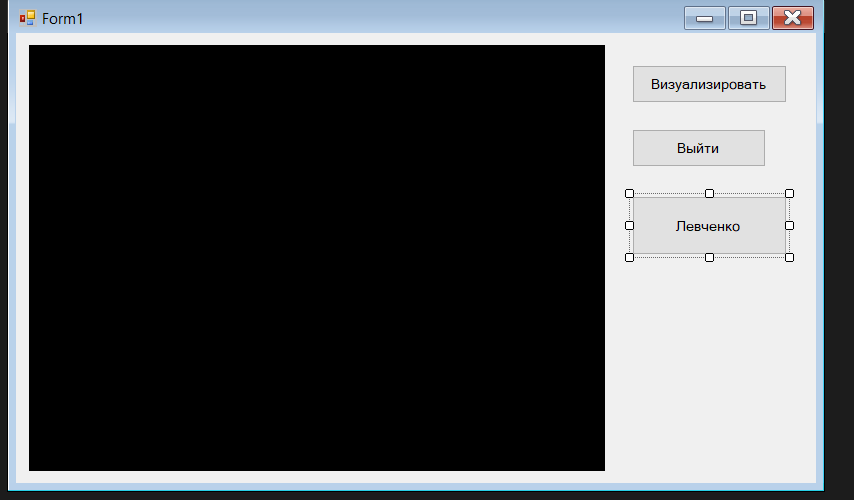
}

Результат запуска и нажатия кнопки «Визуализировать»:

  
Рисунок 8. Результат запуска

***Индивидуальное задание***

  
Рисунок 9. Вариант 5

  
Рисунок 10. Добавление кнопки

private void button3\_Click(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);

// рисуем сферу с помощью библиотеки FreeGLUT

Glut.glutSolidSphere(1, 8, 32);

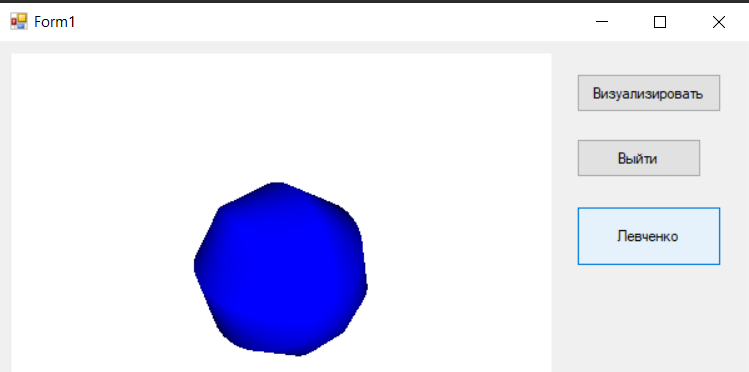
Gl.glPopMatrix();

Gl.glFlush();

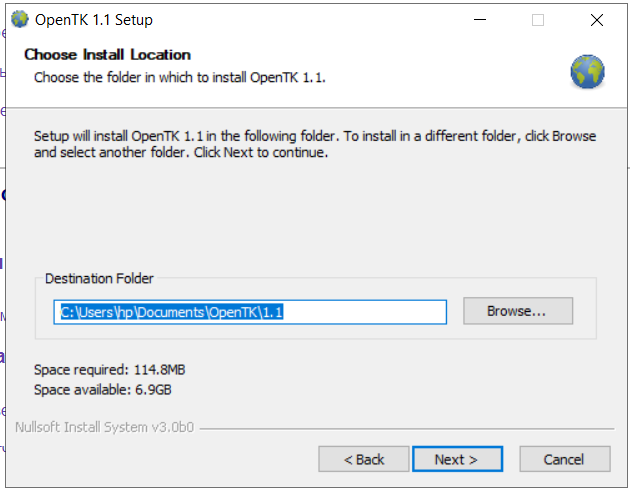
AnT.Invalidate();

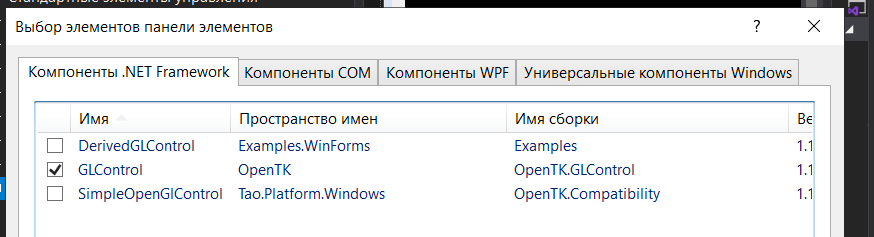
}

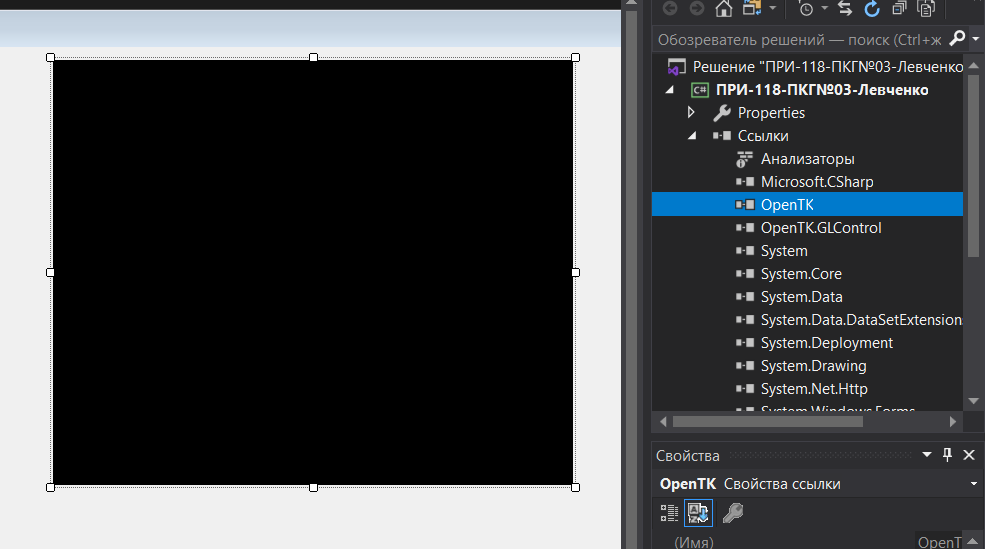
Результат:

  
Рисунок 11. Результат

***Использование библиотеки OpenTK d проектах Microsoft Visual Studio***

  
Рисунок 12. Каталог для OpenTK

  
Рисунок 13. Добавление графического элемента управления GLControl

  
Рисунок 14. Результат добавления на рабочую область

Код:

namespace ПРИ\_118\_ПКГ\_03\_Левченко

{

public partial class Form1 : Form

{

float AngleX = 30;

float AngleY = 30;

float AngleZ = 0;

const float AngleDl = 5;

public Form1()

{

InitializeComponent();

}

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

{

this.Controls.Add(glControl1);

}

private void glControl1\_Resize(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(PrimitiveType.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();

}

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

{

glControl1.Paint += new System.Windows.Forms.PaintEventHandler(this.glControl1\_Paint);

glControl1.KeyDown += new System.Windows.Forms.KeyEventHandler(this.glControl1\_KeyDown);

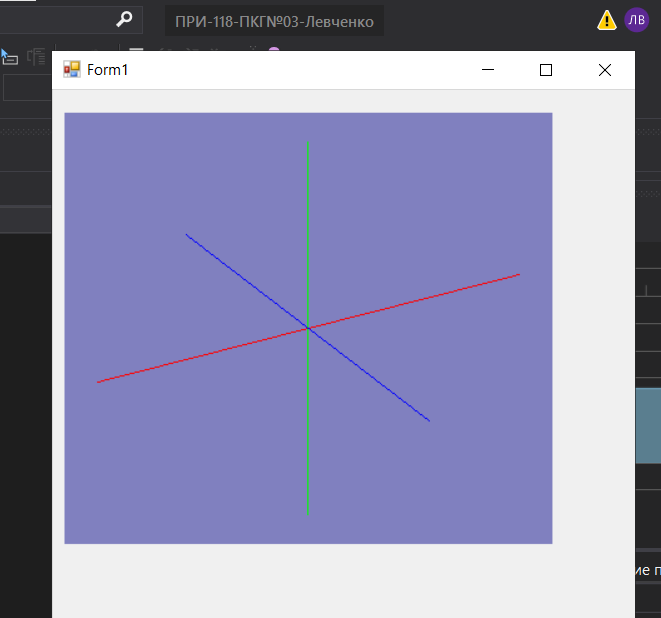
glControl1.Resize += new System.EventHandler(this.glControl1\_Resize);

this.Controls.Add(glControl1);

}

}

}

  
Рисунок 15. Результат

**Вывод**

Изучены принципы применения библиотеки OpenGL при разработке приложений в C#.