РИСОВАНИЕ В КОМПОНЕНТЕ GLCONTROL
В ПРОЕКТЕ ТИПА WINFORMS

1. Создать проект типа WinForms
2. Подключить к проекту библиотеки OpenTK
   и System.Drawing 4.0 (в пункте Сборки)
3. Отправить на форму groupBox'ы для управляющих кнопок
   и для области рисования
4. В Панели элементов нажать правую кнопку мыши
   и выбрать "Добавить вкладку". Назвать ее, например OpenGL.
5. На вкладке OpenGL по нажатию правой кнопки
   "Выбрать элементы" -> Обзор -> OpenTK.GLControl -> OK
   Во вкладке OpenGL появится компонент GLControl
6. Перенести компонент GLControl в groupBox для рисования.
7. В свойствах GLControl можно определить цвет фона и рамку.
8. Поместим на поле кнопок comboBox.
   В свойстве Items укажем для него варианты X, Y, Z
   (это названия осей, вокруг которых будем в дальнейшем вращать)
9. Выберем в свойствах формы функцию load (нажмем на нее в свойствах формы, закладка для функций, она появится автоматически), выполняемую при загрузке формы.
   Заполним ее следующим образом:

```
bool loaded = false;
private void Form1_Load(object sender, EventArgs e)
{
   loaded = true;
   SetupViewport(glControl1);    //установка вида
   comboBox1.SelectedIndex = 0;  //выбор оси для вращения (X)
}
```

   Функция SetupViewport определяет свойства вида на форме.
   Определим ее выше функции load, например, так:

```
private void SetupViewport(GLControl glControl)
{
   float aspectRatio = (float)glControl.Width / (float)glControl.Height;
   GL.Viewport(0, 0, glControl.Width, glControl.Height);
   GL.MatrixMode(MatrixMode.Projection);
   GL.LoadIdentity();

   Matrix4 perspective =
Matrix4.CreatePerspectiveFieldOfView(MathHelper.PiOver4,
                                     aspectRatio,
                                     1f,
```

```
                                                5000000000);
            GL.MultMatrix(ref perspective);
            GL.MatrixMode(MatrixMode.Modelview);
            GL.LoadIdentity();
        }
```

Добавим также в описание библиотек:

```
using OpenTK;
using OpenTK.Graphics;
    using OpenTK.Input;
```

10. В свойствах компонента GLControl выберем функцию Paint (нажмем на нее в свойствах GLControl, она появится автоматически) и заполним ее, например, так:

```
        double angle = 0;
        double dangle = 0;

        private void glControl1_Paint(object sender, PaintEventArgs e)
        {
            if (!loaded)
                return;

            //onload
            GL.ClearColor(0.1f, 0.2f, 0.5f, 0.0f);//цвет фона
            GL.Enable(EnableCap.DepthTest);

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

            //выбор вида
            Matrix4 modelview = Matrix4.LookAt
              (new Vector3(-300, 300, 200), new Vector3(0, 0, 0), new Vector3(0, 0, 1));

            //Vid(comboBox1.SelectedIndex.ToString());

            GL.MatrixMode(MatrixMode.Modelview);
            GL.LoadMatrix(ref modelview);

            //выбор оси вращения
            switch (comboBox1.Text)
            {
                case "X": GL.Rotate(angle, 1, 0, 0); break;
                case "Y": GL.Rotate(angle, 0, 1, 0); break;
                case "Z": GL.Rotate(angle, 0, 0, 1); break;
```

```
        }

        Vector3 clr1 = new Vector3(1.0f, 1.0f, 0.0f);
        Vector3 clr2 = new Vector3(1.0f, 0.0f, 0.0f);
        Vector3 clr3 = new Vector3(0.2f, 0.9f, 1.0f);

        //Рисуем оси
        GL.Begin(BeginMode.Lines);
        GL.Color3(clr1); //X-желтая
        GL.Vertex3(-300.0f, 0.0f, 0.0f);
        GL.Vertex3(300.0f, 0.0f, 0.0f);
        GL.End();

        GL.Begin(BeginMode.Lines);
        GL.Color3(clr2); //Y-красная
        GL.Vertex3(0.0f, -300.0f, 0.0f);
        GL.Vertex3(0.0f, 300.0f, 0.0f);
        GL.End();

        GL.Begin(BeginMode.Lines);
        GL.Color3(clr3); //Z-голубая
        GL.Vertex3(0.0f, 0.0f, -300f);
        GL.Vertex3(0.0f, 0.0f, 300.0f);
        GL.End();

        //Рисуем треугольник
        float z = 40f;
        //  рисуем разноцветный треугольник
        GL.Begin(BeginMode.Triangles);
        GL.Color3(clr1);
        GL.Vertex3(-100.0f, -100.0f, z);
        GL.Color3(clr2);
        GL.Vertex3(100.0f, -100.0f, z);
        GL.Color3(clr3);
        GL.Vertex3(0.0f, 100.0f, z);
        GL.End();

        glControl1.SwapBuffers();

    }//paint
```

Здесь мы определяем начальные условия для рисования, после чего рисуем три оси (X, Y, Z).

Далее рисуем треугольник.

Выше функции заданы начальные значения углов вращения.

Для компонента GLControl создадимм также функцию Resize (нажмем на нее в свойствах) и заполним ее:

```
private void glControl1_Resize(object sender, EventArgs e)
{
    if (!loaded) return;
    SetupViewport(glControl1);
}//resize
```

11. Можно собрать проект и посмотреть, что рисуется три оси (X, Y, Z) и треугольник.

12. Для вращения добавим в область кнопок две кнопки. Назовем их "Rotate" (вращать)
и "Angle=0" (возврат угла вращения в ноль).

```
//вращать
private void button1_Click(object sender, EventArgs e)
{
    dangle = (double)numericUpDown1.Value;
    angle += dangle;
    glControl1.Invalidate();
}

//обнулить угол вращения
private void button2_Click(object sender, EventArgs e)
{
    angle = 0;
    glControl1.Invalidate();
}
```

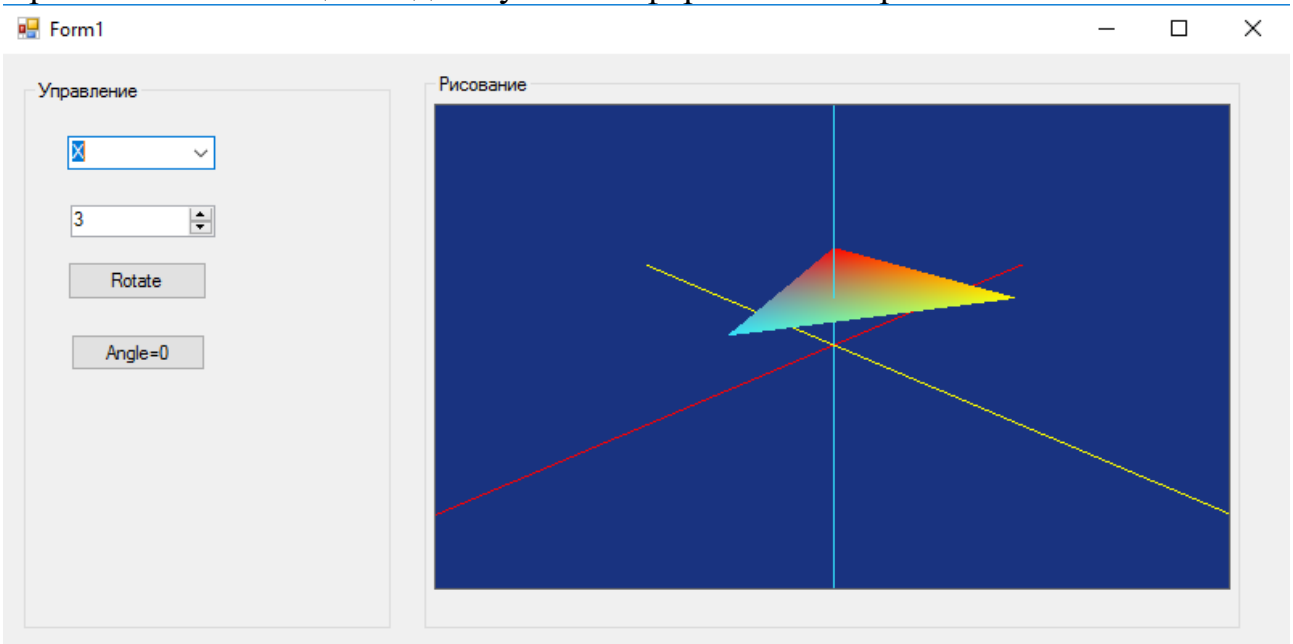
Для задания угла вращения поместим на форму компонент numericUpDown1,
в свойство Value которого запишем например значение 3 (угол поворота).

В функции Matrix4.LookAt определяется направление вида и посредством Vector3(-300, 300, 200) границы области рисования в пикселах.

Вид полученной формы в Приложении 1.

Другие способы представления направления вида в приложении 2.

Приложение 1. Общий вид полученной формы При



Приложение 2. Примеры задания прочих направлений видов

```
//примеры прочих видов
 Matrix4 Vid(String pos)
  {
  switch (pos)
   {
   case "1": //X (вид сбоку сверху)":
    return Matrix4.LookAt(new Vector3(300,230,0), new Vector3(0,50,0), new Vector3(0,1,0));
   case "2": //Y (вид сверху)":
    return Matrix4.LookAt(new Vector3(0,300,0), new Vector3(1,0,0), new Vector3(0,1,0));
   case "3": //Z (вид сбоку прямо)":
    return Matrix4.LookAt(new Vector3(0,150,400), new Vector3(0,0,0), new Vector3(0,1,0));
   case "4": //-X (вид сбоку прямо)":
    return Matrix4.LookAt(new Vector3(-300,300,200),new Vector3(0,0,0), new Vector3(0,1,0));
   case "5": //-Y (вид снизу)":
    return Matrix4.LookAt(new Vector3(0,-100,0), new Vector3(1,0,0), new Vector3(0,1,0));
   case "6": //-Z (вид снизу сбоку)":
    return Matrix4.LookAt(new Vector3(0,100,-400),new Vector3(0,0,0), new Vector3(0,1,0));
   default:
    return Matrix4.LookAt(new Vector3(0,0,0), new Vector3(0,0,0), new Vector3(0,1,0));
    }
   }
```