Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра автоматизированной обработки информации (АОИ)

Работа с туманом и прозрачностью

Отчет о выполнении лабораторной работы
По дисциплине «Компьютерная графика»

Студент гр.__428-2__
___Челпанов Д. А.___
«_____» _____ 20__ г.

Руководитель
канд. техн. наук, доцент каф.АОИ
_____ Т.О. Перемитина
«_____» _____ 20__ г.

Томск 2020

1 Постановка задачи

Цель работы: Сдать долг по лабораторным работам по дисциплине Компьютерная графика.

Выполнение работ: Вариант 20

Лабораторная работа 7

```
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;
using Tao.FreeGlut;
using Tao.OpenGl;
using Tao.Platform.Windows;

namespace lr7
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            holst.InitializeContexts();
            Glut.glutInit();

        }
        double pirX = 0, pirY = 0, angle = 90;
        double conX = 0, conY = 0, alfa =270;

        private void osnova()
        {
            Gl.glViewport(0, 0, holst.Width, holst.Height);
            Gl.glClearColor(0.5f, 0.5f, 0.5f, 1f);

            Gl.glClear(Gl.GL_COLOR_BUFFER_BIT);

            Gl.glEnable(Gl.GL_DEPTH_TEST);
            Gl.glClear(Gl.GL_DEPTH_BUFFER_BIT);

            Gl.glEnable(Gl.GL_LIGHTING);
            Gl.glEnable(Gl.GL_LIGHT0);
            Gl.glEnable(Gl.GL_NORMALIZE);
            Gl.glLightModelf(Gl.GL_LIGHT_MODEL_TWO_SIDE, Gl.GL_TRUE);

            Glut.glutInitDisplayMode(Glut.GLUT_RGB | Glut.GLUT_DOUBLE
| Glut.GLUT_DEPTH | Glut.GLUT_ALPHA);

            Gl.glMatrixMode(Gl.GL_PROJECTION);
            Gl.glLoadIdentity();
            Gl.glFrustum(-0.8, 0.8, -1, 1, 1, 50);
            Gl.glMatrixMode(Gl.GL_MODELVIEW);
```

```
            Gl.glEnable(Gl.GL_FOG);

            Gl.glLoadIdentity();
            Gl.glFogi(Gl.GL_FOG_MODE, Gl.GL_EXP2);
            Gl.glFogf(Gl.GL_FOG_START, 0.0f);
            Gl.glFogf(Gl.GL_FOG_END, 1.0f);
            float[] fogColor = { 0.5f, 0.5f, 0.5f, 1f };
            Gl.glFogfv(Gl.GL_FOG_COLOR, fogColor);
            Gl.glFogf(Gl.GL_FOG_DENSITY, 0.1f);
            Gl.glHint(Gl.GL_FOG_HINT, Gl.GL_DONT_CARE);

        }

        private void holst_Load(object sender, EventArgs e)
        {

        }

        private void dodecahedron()
        {
            Gl.glLoadIdentity();
            float[] color_dod = { 0, 1, 0 };
            float[] light0_dif = { 0.7f, 0.7f, 0.2f };
            float[] light0_pos = { 0.0f, 1.0f, 0.0f, 0.0f };
            Gl.glLightfv(Gl.GL_LIGHT0, Gl.GL_AMBIENT, light0_dif);
            Gl.glMaterialfv(Gl.GL_FRONT_AND_BACK, Gl.GL_AMBIENT,
color_dod);
            Gl.glLightfv(Gl.GL_LIGHT0, Gl.GL_POSITION, light0_pos);

            Gl.glLoadIdentity();
            Gl.glTranslated(0, 0, -15);
            Gl.glRotated(angle, pirX, pirY, 0);
            Gl.glTranslated(0, 0, -8);
            Glut.glutSolidDodecahedron();
            Gl.glLoadIdentity();

            angle += 3;
            pirX += 10;
            pirY -= 10;
            holst.Invalidate();
        }

        private void conus()
        {
            Gl.glLoadIdentity();
            float[] color_con = { 0.7f, 0.5f, 1 };
            float[] light0_dif = { 0.7f, 0.7f, 0.2f };
            float[] light0_pos = { 0.0f, 1.0f, 0.0f, 0.0f };
            Gl.glLightfv(Gl.GL_LIGHT0, Gl.GL_AMBIENT, light0_dif);
```

```
            Gl.glMaterialfv(Gl.GL_FRONT_AND_BACK, Gl.GL_AMBIENT,
color_con);
            Gl.glLightfv(Gl.GL_LIGHT0, Gl.GL_POSITION, light0_pos);

            Gl.glLoadIdentity();
            Gl.glTranslated(0, 0, -15);

            Gl.glRotated(alfa, conX, conY, 0);
            Gl.glTranslated(0, 0, -8);

            Glut.glutSolidCone(1.5, 2, 30, 30);
            Gl.glLoadIdentity();

            alfa += 3;
            conX += 10d;
            conY -= 10d;
            holst.Invalidate();
        }

        private void tor()
        {
            Gl.glEnable(Gl.GL_ALPHA_TEST);
            Gl.glEnable(Gl.GL_BLEND);


            Gl.glBlendFunc(Gl.GL_SRC_COLOR,
Gl.GL_ONE_MINUS_SRC_COLOR);
            Gl.glLoadIdentity();
            float[] color_con = { 0.5f, 0, 1, 0.5f };
            float[] light0_dif = { 0.7f, 0.7f, 0.2f, 0.9f };
            float[] light0_pos = { 0.0f, 1.0f, 0.0f, 0.0f };
            Gl.glLightfv(Gl.GL_LIGHT0, Gl.GL_AMBIENT, light0_dif);
            Gl.glMaterialfv(Gl.GL_FRONT_AND_BACK, Gl.GL_AMBIENT,
color_con);
            Gl.glLightfv(Gl.GL_LIGHT0, Gl.GL_POSITION, light0_pos);

            Gl.glLoadIdentity();
            Gl.glTranslated(0, 0, -10);
            Gl.glRotated(30, 0, 1, 1);

            Glut.glutSolidTorus(1, 2, 20, 20);
            Gl.glLoadIdentity();

            Gl.glDisable(Gl.GL_ALPHA_TEST);
            Gl.glDisable(Gl.GL_BLEND);

            holst.Invalidate();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
```

```
        }

        private void holst_MouseClick_1(object sender, MouseEventArgs
e)
        {
            timer1.Start();
            timer1.Enabled = true;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            osnova();
            dodecahedron();
            conus();
            tor();
            holst.Invalidate();
        }

    }
}
```

Рис 1

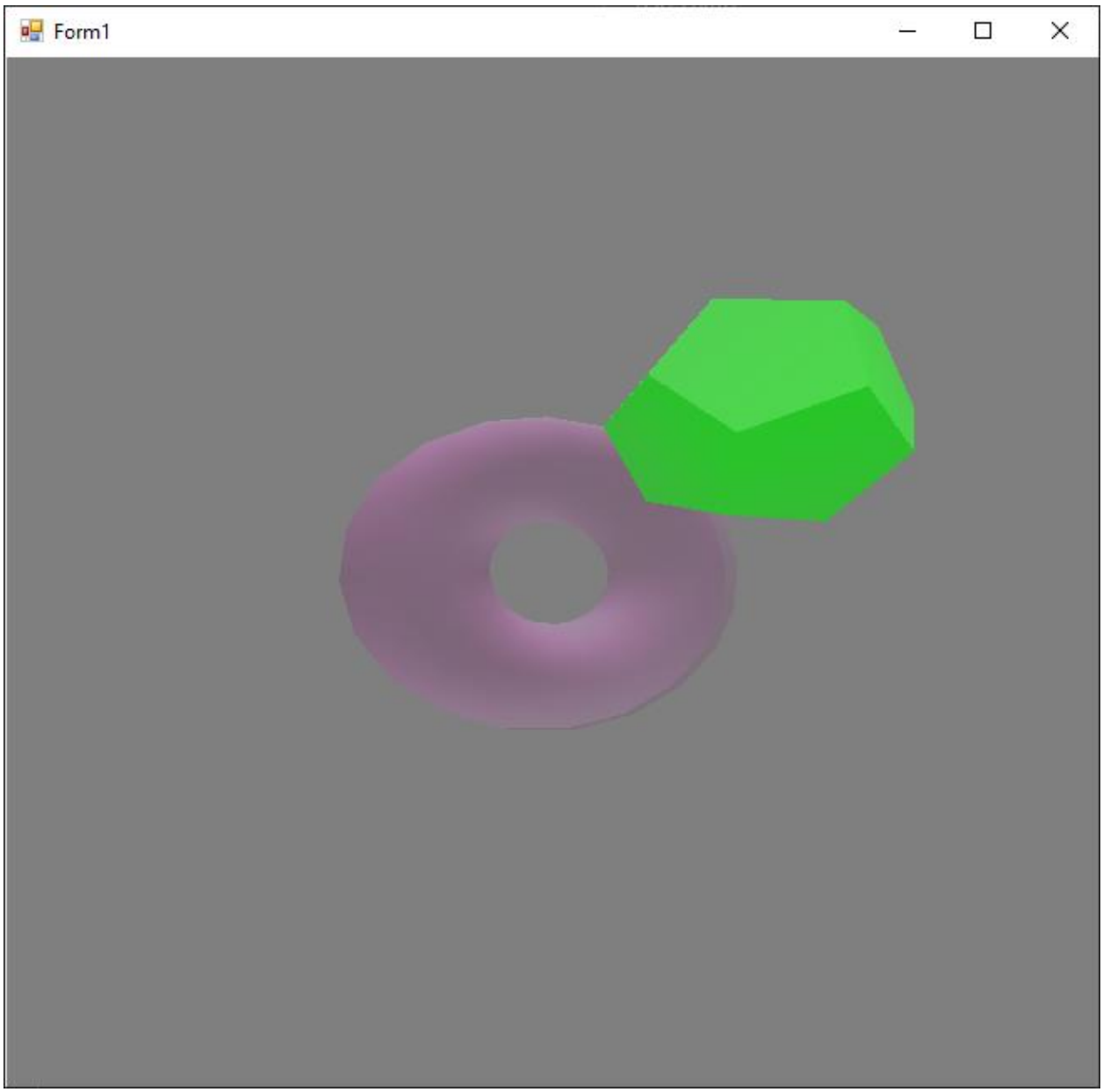


Рис 2

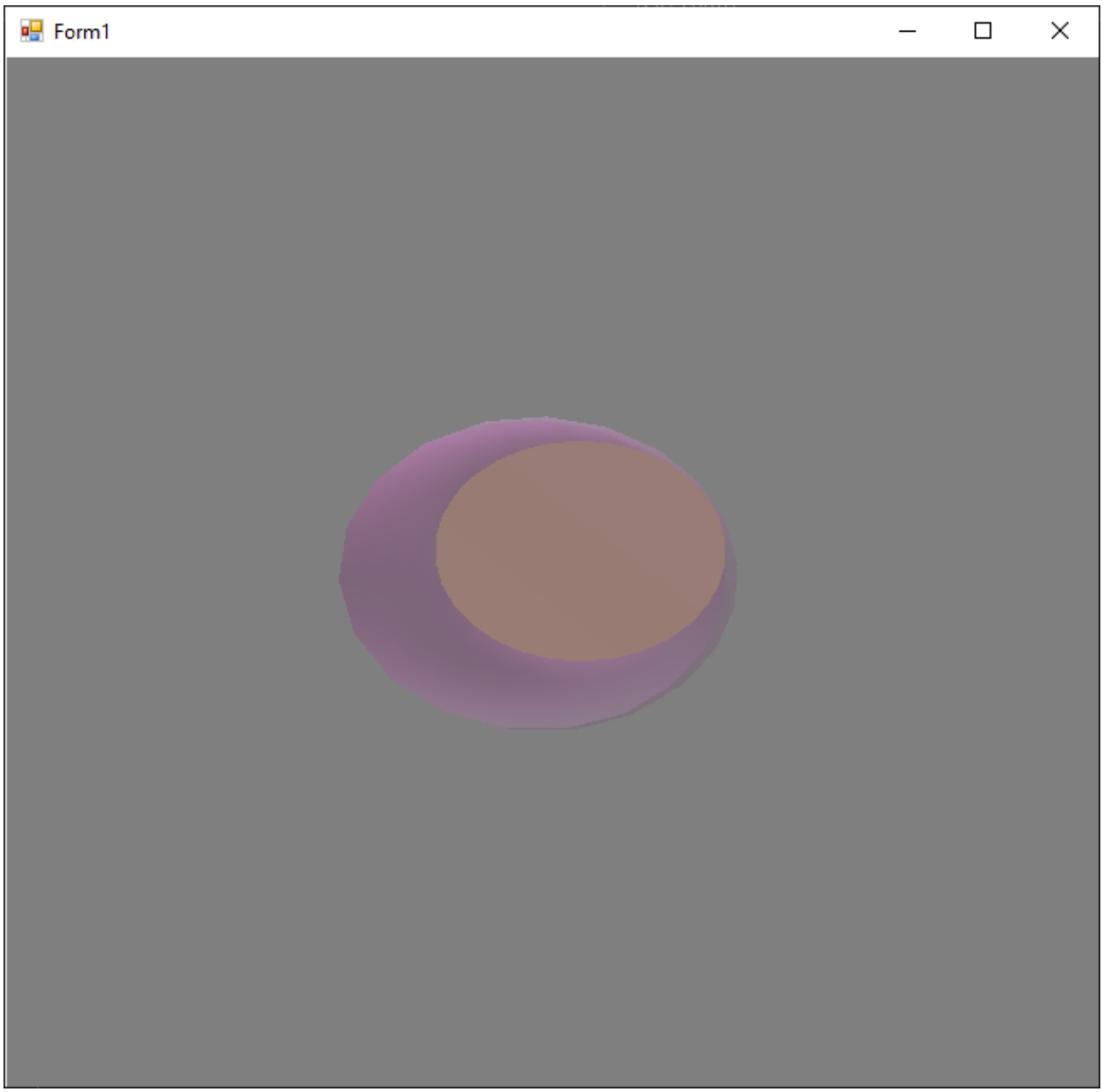


Рис 3

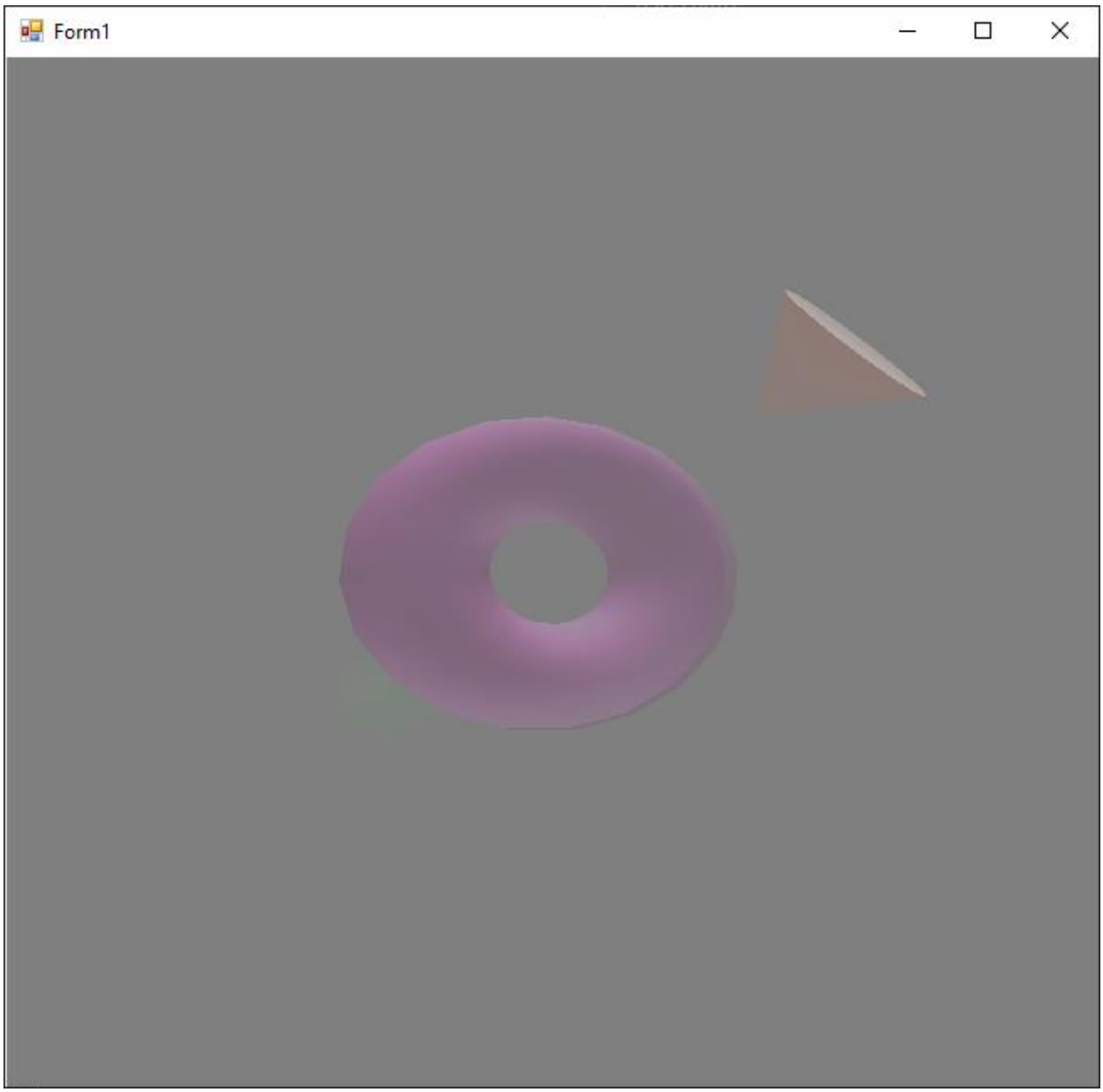


Рис 4

Лабораторная работа 8

Скрипт для движения игрока

```
using UnityEngine;
using System.Collections;
public class MouseLook : MonoBehaviour
{
    public enum RotationAxes
    {
        MouseXAndY = 0,
        MouseX = 1,
        MouseY = 2
    }
    public RotationAxes axes = RotationAxes.MouseXAndY;
    public float sensitivityHor = 9.0f;
    public float sensitivityVert = 9.0f;
    public float minimumVert = -45.0f;
    public float maximumVert = 45.0f;
    private float _rotationX = 0;
    void Start()
    {
        Rigidbody body = GetComponent<Rigidbody>();
        if (body != null)
            body.freezeRotation = true;
    }
    void Update()
    {
        if (axes == RotationAxes.MouseX)
        {
            transform.Rotate(0, Input.GetAxis("Mouse X") *
sensitivityHor, 0);
        }
        else if (axes == RotationAxes.MouseY)
        {
            _rotationX -= Input.GetAxis("Mouse Y") * sensitivityVert;
            _rotationX = Mathf.Clamp(_rotationX, minimumVert,
maximumVert);
            float rotationY = transform.localEulerAngles.y;
            transform.localEulerAngles = new Vector3(_rotationX,
rotationY, 0);
        }
        else
        {
            _rotationX -= Input.GetAxis("Mouse Y") * sensitivityVert;
            _rotationX = Mathf.Clamp(_rotationX, minimumVert,
maximumVert);
            float delta = Input.GetAxis("Mouse X") * sensitivityHor;
            float rotationY = transform.localEulerAngles.y + delta;
```

```
            transform.localEulerAngles = new Vector3(_rotationX,
rotationY, 0);
        }
    }
                                    }


using UnityEngine;
using System.Collections;
[RequireComponent(typeof(CharacterController))]
[AddComponentMenu("Control Script/FPS Input")]
public class FPSInput : MonoBehaviour
{
    public float speed = 6.0f;
    public float gravity = -9.8f;
    private CharacterController _charController;
    void Start()
    {
        _charController = GetComponent<CharacterController>();
    }
    void Update()
    {
        float deltaX = Input.GetAxis("Horizontal") * speed;
        float deltaZ = Input.GetAxis("Vertical") * speed;
        Vector3 movement = new Vector3(deltaX, 0, deltaZ);
        movement = Vector3.ClampMagnitude(movement, speed);
        movement.y = gravity;
        movement *= Time.deltaTime;
        movement = transform.TransformDirection(movement);
        _charController.Move(movement);
    }
}
```

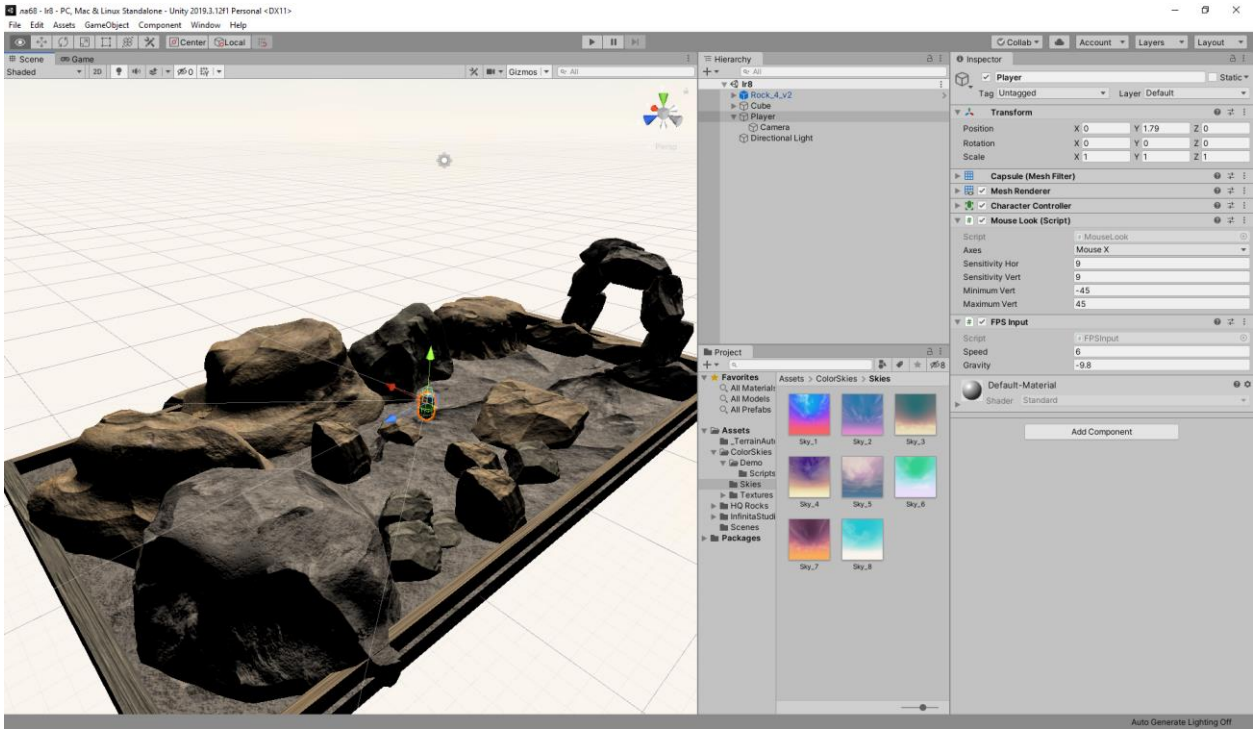


Рис 5

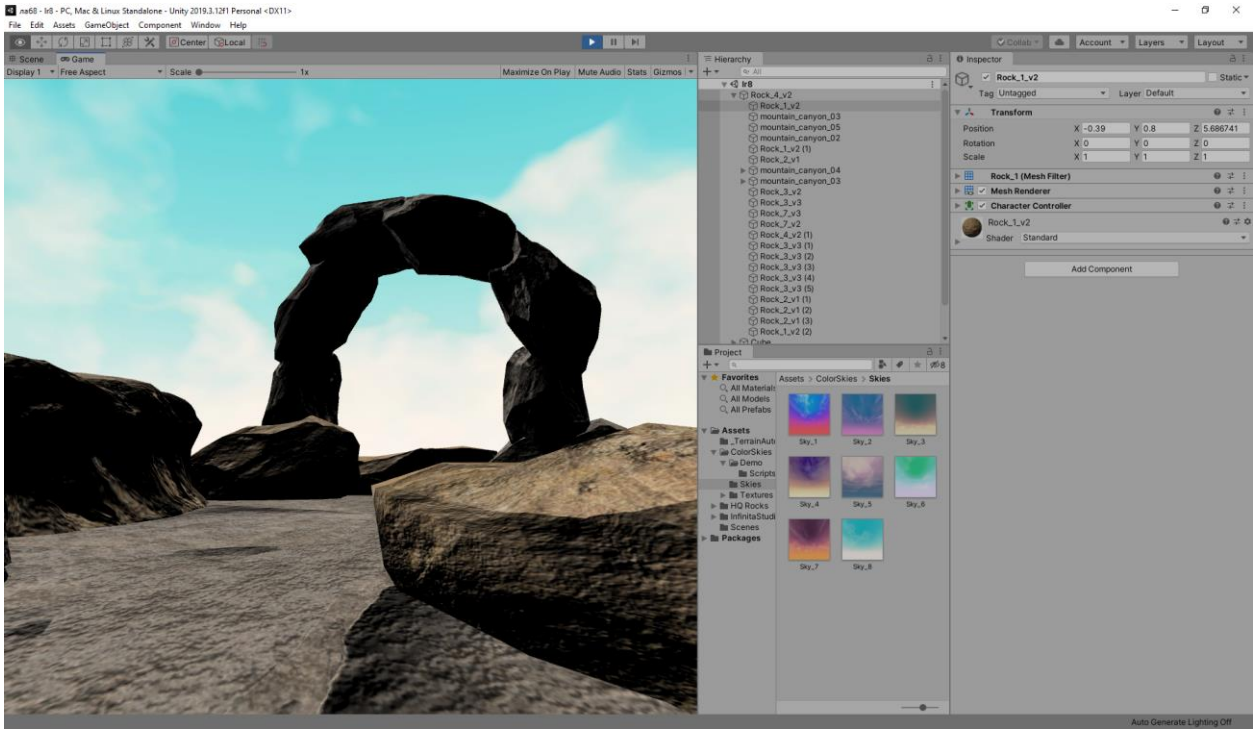


Рис 6

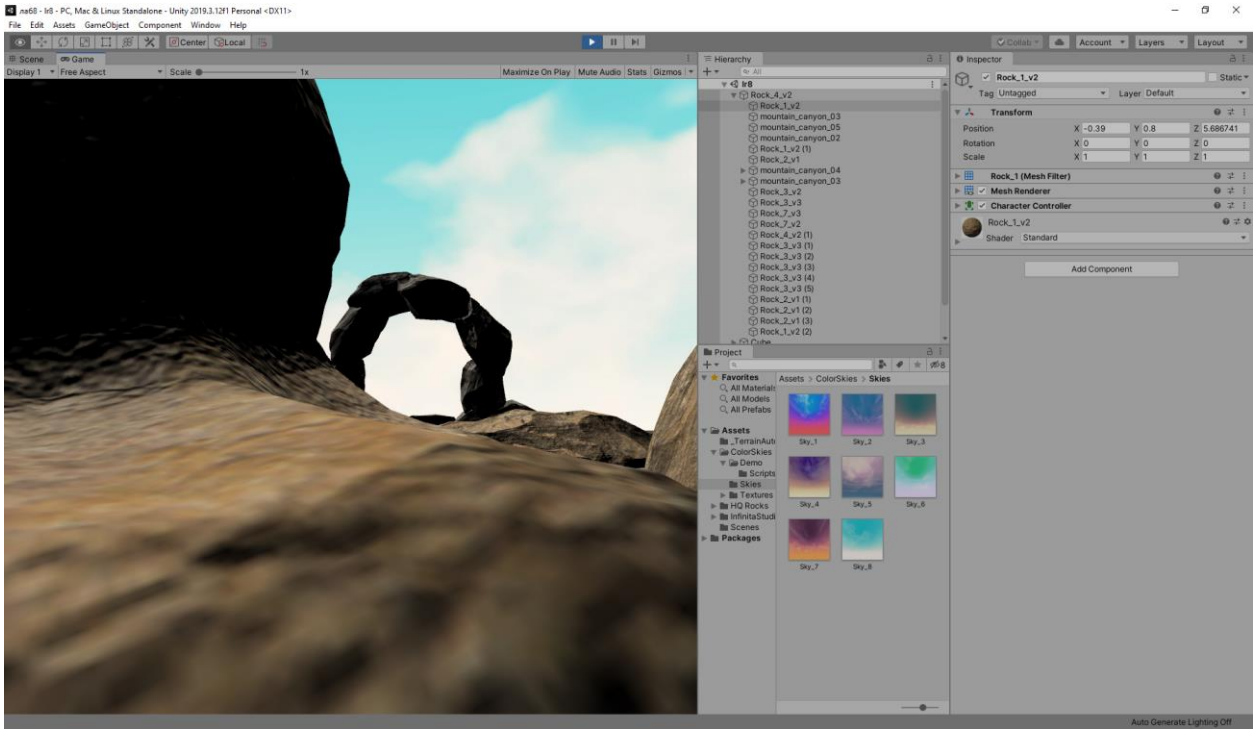
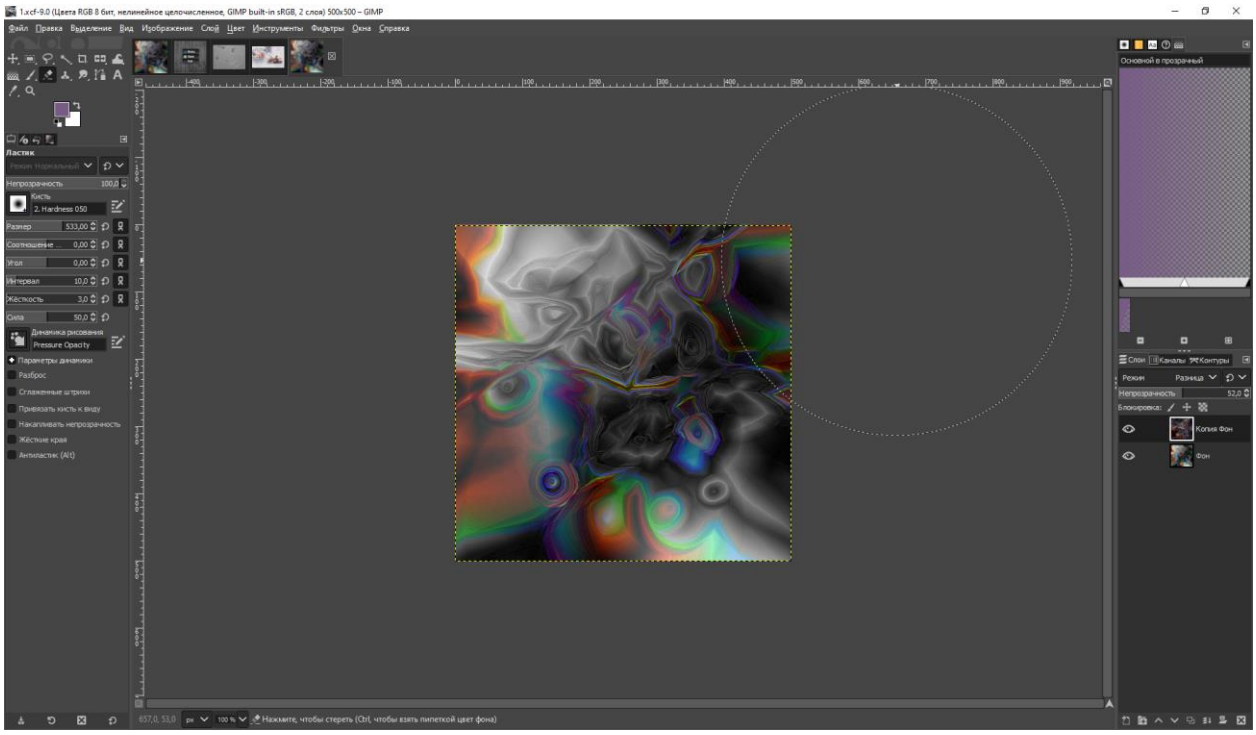
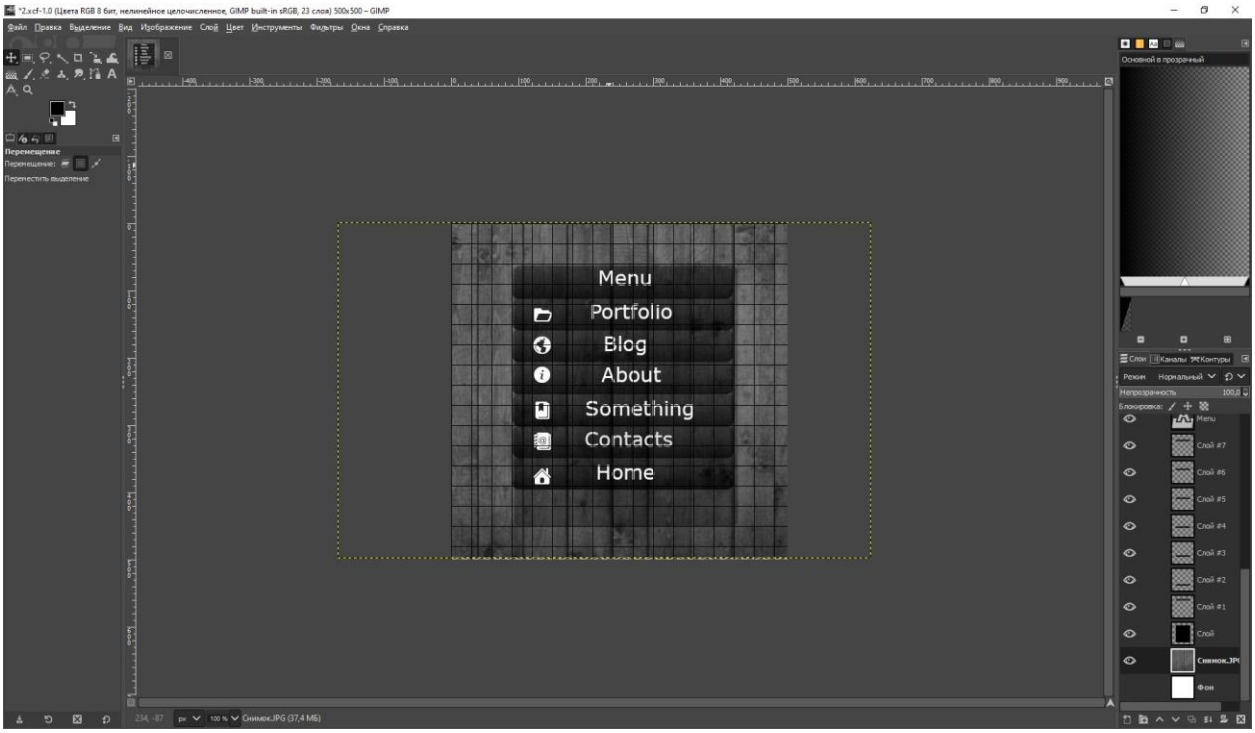
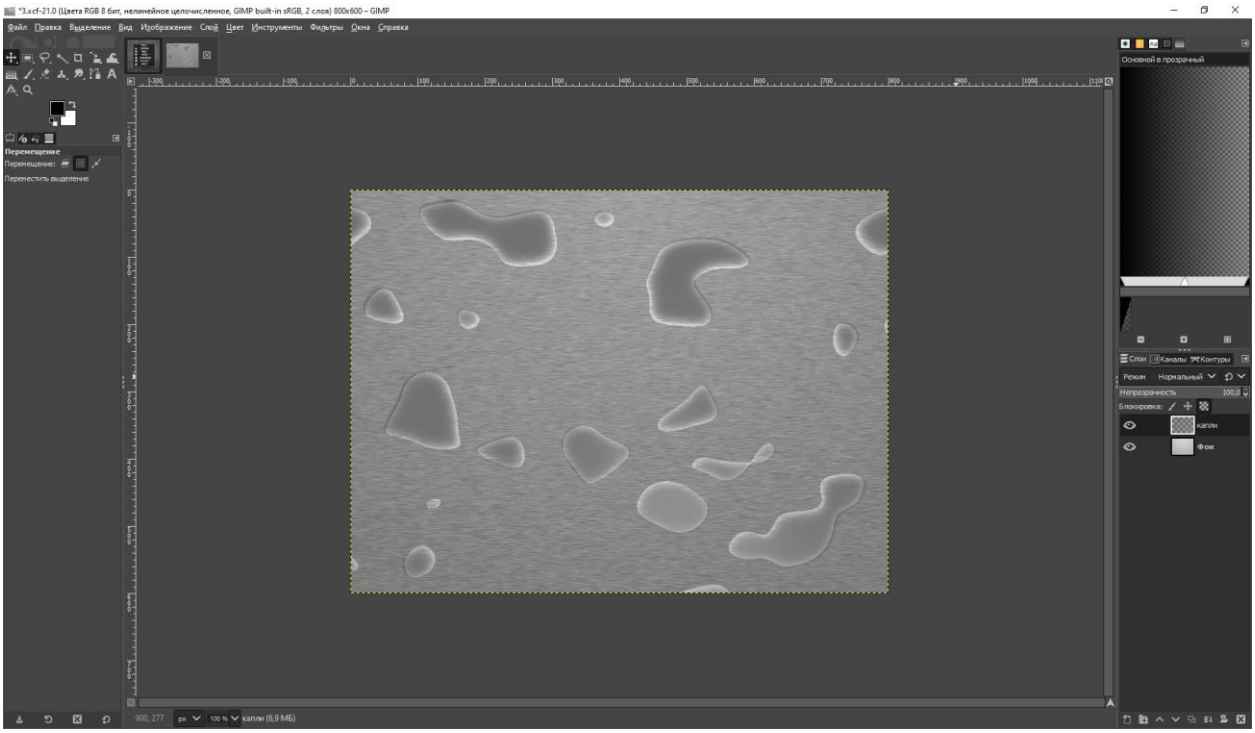


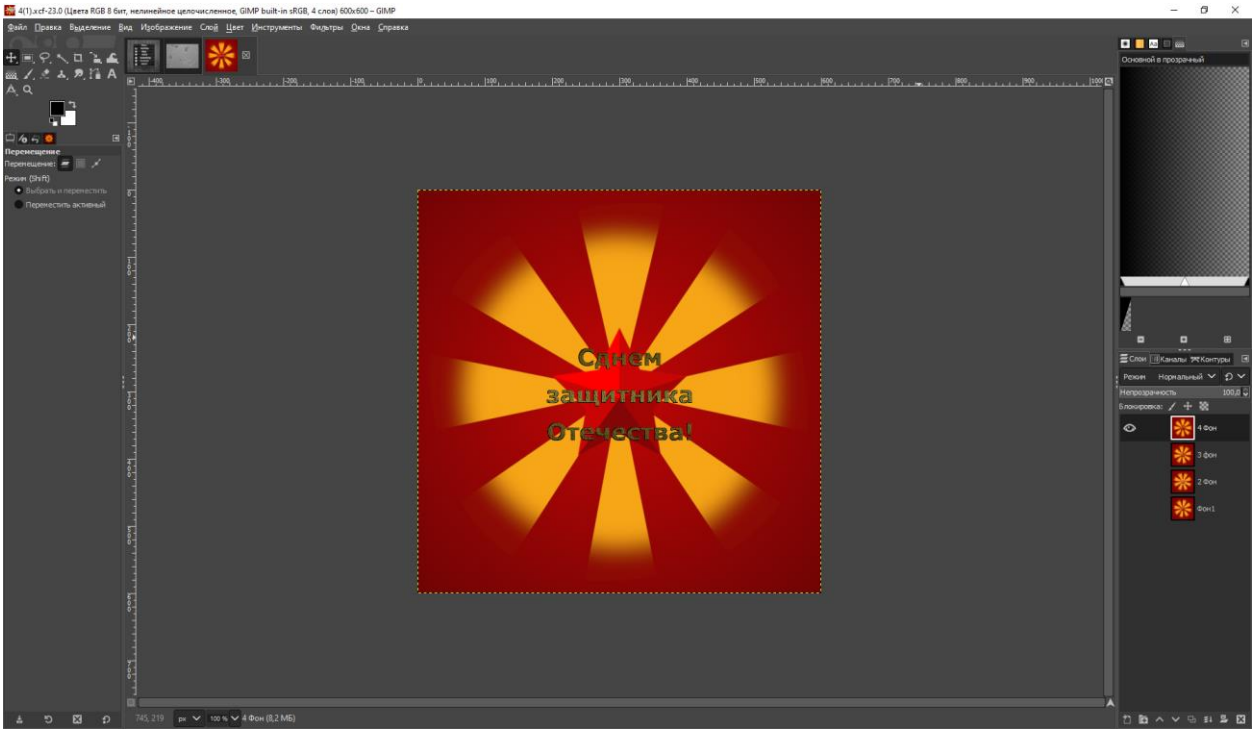
Рис 7

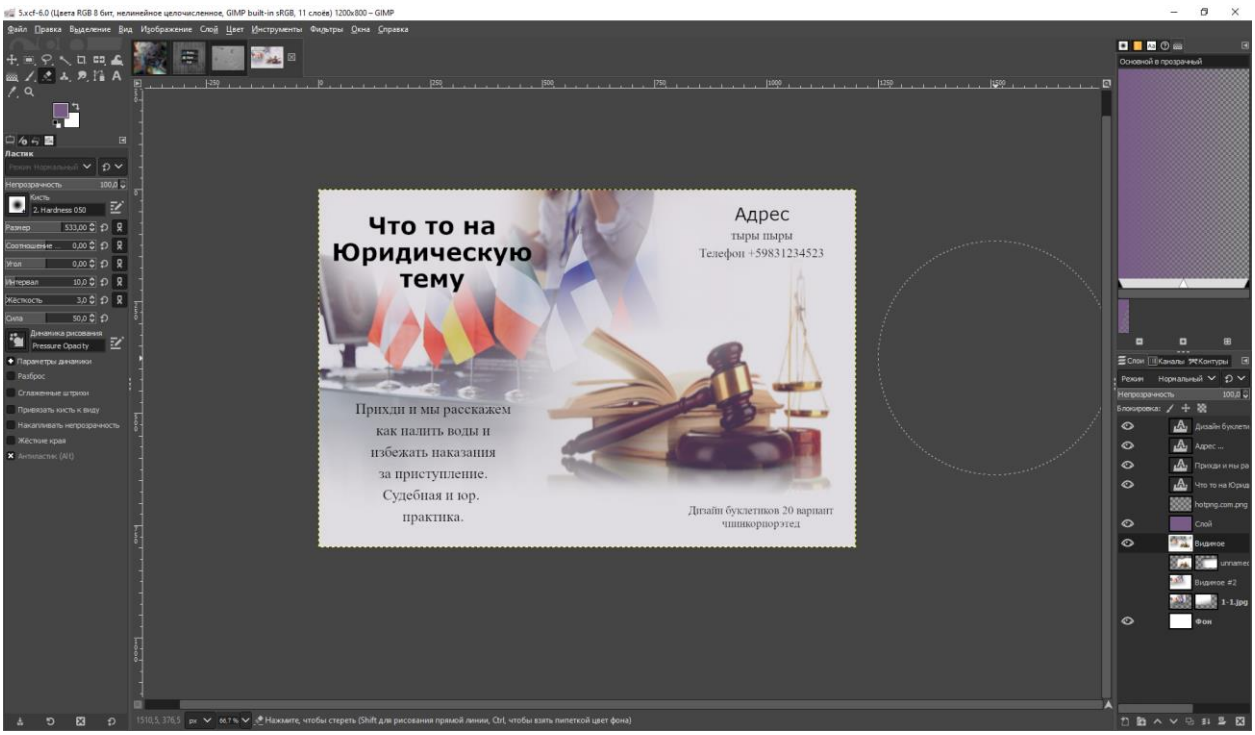
Лабораторная работа 1




Menu
Portfolio
Blog
About
Something
Contacts
Home




Сднем
защитника
Отечества!


Что то на Юридическую тему
Адрес
тыры пыры
Телефон +59831234523
Прихди и мы рассекажем как палить воды и избежать наказания за приступление. Судебная и юр. практика.
Дизайн буклетиков 20 вариант чшикорпорэтед