

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

УТВЕРЖДАЮ

Доцент департамента программной
инженерии, канд. техн. наук

Академический руководитель
образовательной программы
«Программная инженерия» профессор
департамента программной инженерии,
канд. техн. наук

_____ Р. З. Ахметсафина
«_____» _____ 2020 г.

_____ В. В. Шилов
«_____» _____ 2020 г.

ВИЗУАЛИЗАТОР ТРЕХМЕРНОЙ ПРОЕКЦИИ
ЧЕТЫРЕХМЕРНОГО ТЕЛА

Текст программы

Лист УТВЕРЖДЕНИЯ

RU.17701729.04.13-01 12 01-1-ЛУ

Исполнитель: Студент группы БПИ194
_____ М. Г. Савинов
«_____» _____ 2020 г.

УТВЕРЖДЁН
RU.17701729.04.13-01 12 01-1-ЛУ

ВИЗУАЛИЗАТОР ТРЕХМЕРНОЙ ПРОЕКЦИИ
ЧЕТЫРЕХМЕРНОГО ТЕЛА

Текст программы

RU.17701729.04.13-01 12 01-1

Листов 9

Инов. № подл	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

Содержание

1	Текст программы	3
1.1	Main.cs	3
1.2	AbstractMesh.cs	4
1.3	TesseractMesh.cs	6

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. Текст программы

Программа состоит из 3х классов.

1.1. Main.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Main : MonoBehaviour
{
    public Material edgeColor;
    public Material innerColor;
    private GameObject projection;
    private static AbstractMesh meshObject;
    /// <summary>
    /// Создаем тессеракт
    /// </summary>
    void Start()
    {
        meshObject = TesseractMesh.GenerateHypercube();
        projection = new GameObject(meshObject.Name);
        projection.AddComponent<MeshFilter>();
        MeshRenderer meshrenderer = projection.AddComponent<MeshRenderer>();
        meshrenderer.materials = new Material[] { edgeColor, innerColor };
    }
    /// <summary>
    /// вызываем вращение тессеракта каждый кадр
    /// </summary>
    void Update()
    {
        Mesh mesh = projection.GetComponent<MeshFilter>().mesh;
        meshObject.UpdateMesh(mesh);
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1.2. AbstractMesh.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public abstract class AbstractMesh
{
    protected string _name;
    public string Name
    {
        get
        {
            return _name;
        }
    }
    public AbstractMesh(string name)
    {
        _name = name;
    }

    public abstract void UpdateMesh(Mesh mesh);

    /// <summary>
    /// считаем нормали для каждой грани,
    /// записываем их как сумму для каждой вершины
    /// </summary>
    /// <param name="vertices">список вершин-векторов</param>
    /// <param name="quads">грани</param>
    /// <returns></returns>
    protected static Vector3[] CalculateNormals(Vector3[] vertices,
        int[] quads)
    {
        Vector3[] normals = new Vector3[vertices.Length];
        for (int i = 0; i < quads.Length; i += 4)
        {
            int v1 = quads[i];
            int v2 = quads[i + 1];
            int v3 = quads[i + 2];
            int v4 = quads[i + 3];
            Vector3 normal = CalculateNormal(vertices[v1],
                vertices[v2], vertices[v3]);

            normals[v1] = normals[v1] + normal;
            normals[v2] = normals[v2] + normal;
            normals[v3] = normals[v3] + normal;
            normals[v4] = normals[v4] + normal;
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
    //считаем нормаль к каждой вершине
    for (int i = 0; i < normals.Length; i++)
    {
        normals[i] = (normals[i]).normalized;
    }
    return normals;
}
/// <summary>
/// находим нормаль к заданной через 3 точки плоскости
///по формуле через векторное произведение
/// </summary>
/// <param name="v0">первая точка</param>
/// <param name="v1">вторая точка</param>
/// <param name="v2">третья точка</param>
/// <returns></returns>
private static Vector3 CalculateNormal(Vector3 v0, Vector3 v1, Vector3 v2)
{
    Vector3 d0 = v1 - v0;
    Vector3 d1 = v2 - v0;
    return Vector3.Cross(d0, d1);
}
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1.3. TesseractMesh.cs

```

using System;
using System.Collections.Generic;
using UnityEngine;

public class TesseractMesh : AbstractMesh
{
    private static int[] lines;
    private static int[] quads;
    private static Vector4[] vertices;

    public TesseractMesh(string name) : base(name)
    {
    }

    public override void UpdateMesh(Mesh mesh)
    {
        //вычисляем угол поворота с помощью времени работы программы
        float angle = (Time.time * 0.5f) % 360.0f;
        float cos = Mathf.Cos(angle);
        float sin = Mathf.Sin(angle);

        Vector3[] transformedVertices = new Vector3[vertices.Length];
        for (int i = 0; i < transformedVertices.Length; i++)
        {
            Vector4 oldVert = vertices[i];
            //для каждой вершины считаем ее поворот
            float newX = (oldVert.x * cos) + (oldVert.y * sin);
            float newY = (oldVert.y * cos) - (oldVert.x * sin);
            float newz = (oldVert.z * cos) + (oldVert.w * sin);
            float newW = (oldVert.w * cos) - (oldVert.z * sin);
            //считаем коэффициент матрицы проекции
            float wProj = 5.0f / (4.0f - newW);
            //проектируем на трехмерное пространство
            transformedVertices[i] = wProj * (new Vector3(newX, newY, newz));
        }
        //очищаем старую сетку
        mesh.Clear();
        //задаем вершины проекции
        mesh.vertices = transformedVertices;
        //задаем нормали проекции
        mesh.normals = CalculateNormals(transformedVertices, quads);
        mesh.subMeshCount = 2;
        //задаем ребра проекции
        mesh.SetIndices(lines, MeshTopology.Lines, 0);
        //задаем грани проекции
        mesh.SetIndices(quads, MeshTopology.Quads, 1);
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

/// <summary>
/// создание тессеракта
/// </summary>
/// <returns></returns>
public static TesseractMesh GenerateHypercube()
{
    //задаем вершины
    vertices = new Vector4[] {
        new Vector4(-1.0f, -1.0f, -1.0f, 1.0f),
        new Vector4( 1.0f, -1.0f, -1.0f, 1.0f),
        new Vector4( 1.0f, -1.0f, 1.0f, 1.0f),
        new Vector4(-1.0f, -1.0f, 1.0f, 1.0f),

        new Vector4(-1.0f, 1.0f, 1.0f, 1.0f),
        new Vector4( 1.0f, 1.0f, 1.0f, 1.0f),
        new Vector4( 1.0f, 1.0f, -1.0f, 1.0f),
        new Vector4(-1.0f, 1.0f, -1.0f, 1.0f),

        new Vector4(-1.0f, -1.0f, -1.0f, -1.0f),
        new Vector4( 1.0f, -1.0f, -1.0f, -1.0f),
        new Vector4( 1.0f, -1.0f, 1.0f, -1.0f),
        new Vector4(-1.0f, -1.0f, 1.0f, -1.0f),

        new Vector4(-1.0f, 1.0f, 1.0f, -1.0f),
        new Vector4( 1.0f, 1.0f, 1.0f, -1.0f),
        new Vector4( 1.0f, 1.0f, -1.0f, -1.0f),
        new Vector4(-1.0f, 1.0f, -1.0f, -1.0f),
    };
    //задаем грани проекции
    quads = new int[] {
        0, 1, 2, 3,
        4, 5, 6, 7,
        7, 6, 1, 0,
        6, 5, 2, 1,
        5, 4, 3, 2,
        4, 7, 0, 3,

        8, 9, 10, 11,
        12, 13, 14, 15,
        15, 14, 9, 8,
        14, 13, 10, 9,
        13, 12, 11, 10,
        12, 15, 8, 11,
    };
    //задаем ребра проекции
    lines = new int[] {
        0, 1, 1, 2, 2, 3, 3, 0,
        4, 5, 5, 6, 6, 7, 7, 4,
    };
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

0, 7, 1, 6, 2, 5, 3, 4,

8, 9, 9, 10, 10, 11, 11, 8,
12, 13, 13, 14, 14, 15, 15, 12,
8, 15, 9, 14, 10, 13, 11, 12,

0, 8, 1, 9, 2, 10, 3, 11,
4, 12, 5, 13, 6, 14, 7, 15,

};

//создаем новый объект класса TesseractMesh

TesseractMesh tesseract = new TesseractMesh("Tesseract");

return tesseract;

}

}

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Лист регистрации изменений

[illegible]