**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 6**

**Дисциплина:** Компьютерная графика

**Тема:** Трипланарное наложение

**Выполнил:** Вышегородских Виктор Егорович, **студент группы:** 211-728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2023**

На основе теоретического материала написали шейдер.

На основе шейдера был создан и настроен материал.

В Unity3D была создана и настроена сцена. Добавили на сцену сферу, применили к ней созданный ранее материал на основе доработанного шейдера.

# Исходный код шейдера:

Shader "CG Labs/Lab 6"

{

    Properties

    {

        \_Color("Tint", Color) = (0, 0, 0, 1)

        \_MainTex("Texture", 2D) = "white" {}

        \_Sharpness("Blend sharpness", Range(1,64)) = 1

    }

    SubShader

    {

        Tags

        {

            "RenderType" = "Opaque"

            "Queue" = "Geometry"

        }

        Pass

        {

            CGPROGRAM

            #pragma vertex vert

            #pragma fragment frag

            #include "UnityCG.cginc"

            sampler2D \_MainTex;

            float4 \_MainTex\_ST;

            fixed4 \_Color;

            float \_Sharpness;

            struct appdata

            {

                float4 vertex : POSITION;

                float3 normal: NORMAL;

            };

            struct v2f

            {

                float4 position : SV\_POSITION;

                float3 worldPos : TEXCOORD0;

                float3 normal: NORMAL;

            };

            v2f vert(appdata v)

            {

                v2f o;

                o.position = UnityObjectToClipPos(v.vertex);

                float4 worldPos = mul(unity\_ObjectToWorld, v.vertex);

                o.worldPos = worldPos.xyz;

                float3 worldNormal = mul(v.normal, (float3x3)unity\_WorldToObject);

                o.normal = normalize(worldNormal);

                return o;

            }

            fixed4 frag(v2f i) : SV\_Target

            {

                float2 uv\_front = TRANSFORM\_TEX(i.worldPos.xy, \_MainTex);

                float2 uv\_side = TRANSFORM\_TEX(i.worldPos.zy, \_MainTex);

                float2 uv\_top = TRANSFORM\_TEX(i.worldPos.xz, \_MainTex);

                fixed4 col\_front = tex2D(\_MainTex, uv\_front);

                fixed4 col\_side = tex2D(\_MainTex, uv\_side);

                fixed4 col\_top = tex2D(\_MainTex, uv\_top);

                float3 weights = i.normal;

                weights = abs(weights);

                weights = pow(weights, \_Sharpness);

                weights = weights / (weights.x + weights.y + weights.z);

                col\_front \*= weights.z;

                col\_side \*= weights.x;

                col\_top \*= weights.y;

                fixed4 col = col\_front + col\_side + col\_top;

                col \*= \_Color;

                return col;

            }

            ENDCG

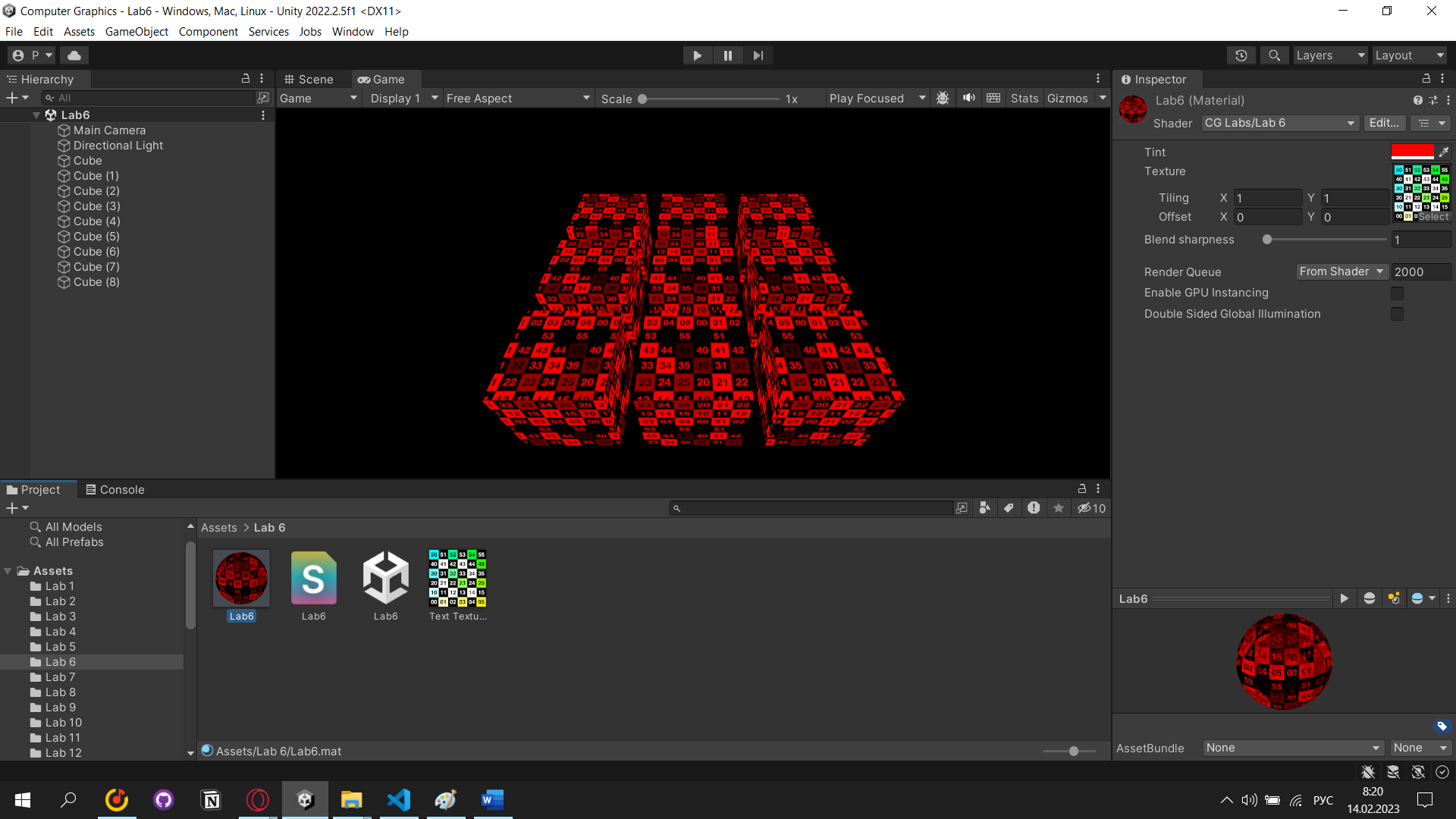
        }

    }

    FallBack "Standard"

}

# Скриншоты итоговой сцены в игровом движке Unity3D:



Изображение выглядит как текст, монитор, экран, снимок экрана

Автоматически созданное описание