**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 № 7**

**Дисциплина:** Компьютерная графика

**Тема:** Градиент

**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 7"

{

    Properties

    {

        \_ColorA("Color A", Color) = (0, 0, 0, 1)

        \_ColorB("Color B", Color) = (1, 1, 1, 1)

        \_ColorStart("Color Start", Range(0, 1)) = 0

        \_ColorEnd("Color End", Range(0, 1)) = 1

    }

    SubShader

    {

        Tags

        {

            "RenderType" = "Opaque"

        }

        Pass

        {

            CGPROGRAM

            #pragma vertex vert

            #pragma fragment frag

            #include "UnityCG.cginc"

            #define INTERPOLATION

            float4 \_ColorA;

            float4 \_ColorB;

            float \_ColorStart;

            float \_ColorEnd;

            struct MeshData

            {

                float4 vertex : POSITION;

                float3 normal : NORMAL;

                float2 uv0 : TEXCOORD0;

            };

            struct Interpolator

            {

                float4 vertex : SV\_POSITION;

                float3 normal : TEXCOORD0;

                float2 uv : TEXCOORD1;

            };

            Interpolator vert(MeshData v)

            {

                Interpolator o;

                o.vertex = UnityObjectToClipPos(v.vertex);

                o.normal = UnityObjectToWorldNormal(v.normal);

                o.uv = v.uv0;

                return o;

            }

            float InverseLerp(

                float4 a,

                float4 b,

                float v)

            {

                return(v - a) / (b - a);

            }

            fixed4 frag(Interpolator i) : SV\_Target

            {

                float t = InverseLerp(\_ColorStart, \_ColorEnd, i.uv.x);

                float4 col = lerp(\_ColorA, \_ColorB, t);

                return col;

            }

            ENDCG

        }

    }

}

# Скриншоты итоговой сцены в игровом движке Unity3D:

Изображение выглядит как текст, снимок экрана, монитор

Автоматически созданное описание

Изображение выглядит как текст, монитор, экран, снимок экрана

Автоматически созданное описание