

1. Cel zadania oraz wariant:

Wariant 1. Opracować model kuli z użyciem shadera Water. Opracować shader w języku HLSL pozwalający wykorzystywać kulę na scenie jako billboard

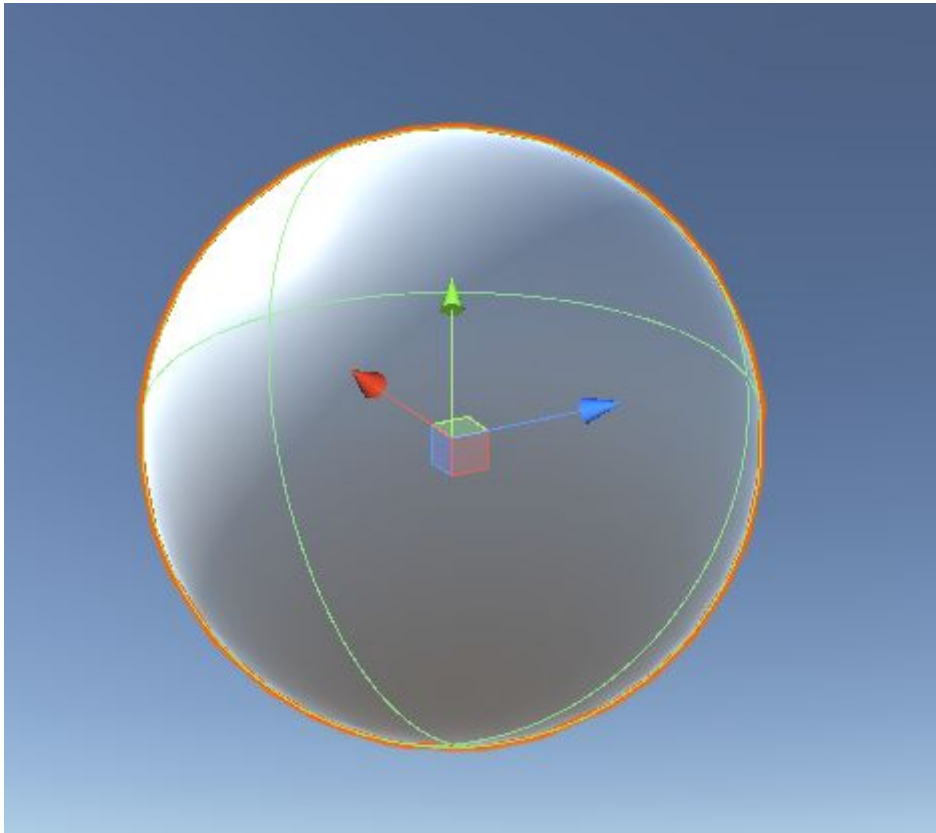
Wariant 2. Opracować model sześcianu z użyciem shadera BumpedSpecular. Opracować shader w języku HLSL pozwalający zmienić kolory RGB zgodnie z $\text{kolor}' = 1 - \text{kolor}$, gdzie kolor' – kolor nowy, kolor – kolor stawy

Wariant 3. Opracować model kapsuli z użyciem shadera BumpedDiffuse. Opracować shader w języku HLSL pozwalający przekształcić kapsulę w kulę

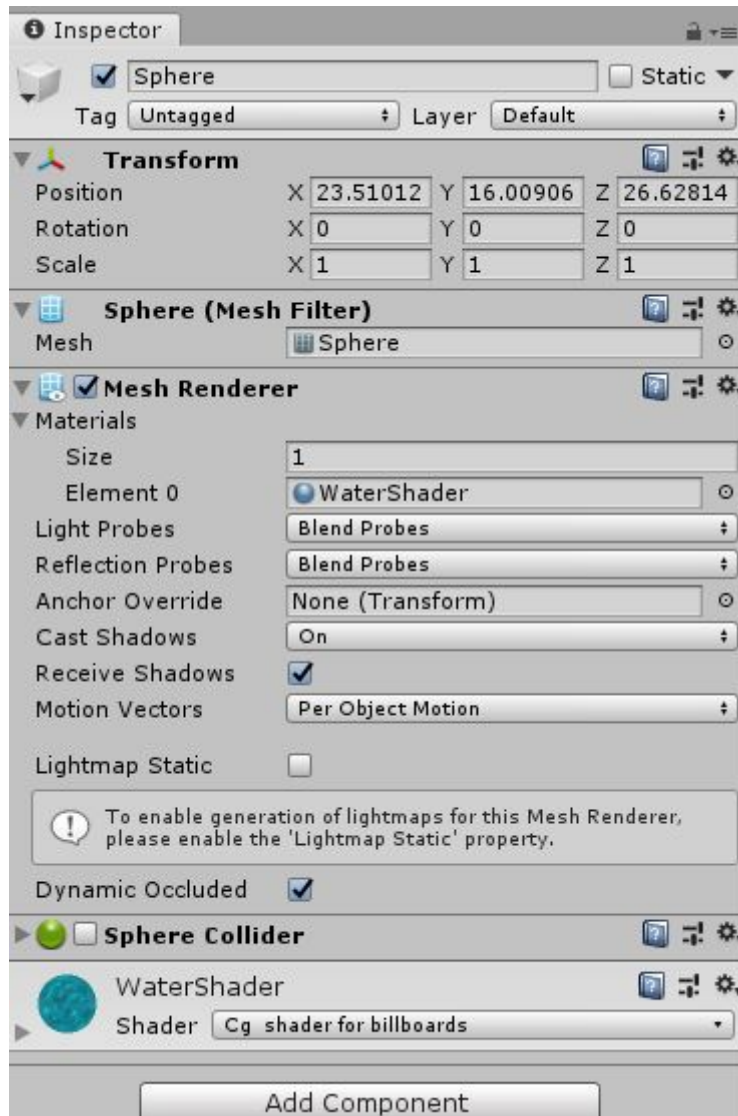
2. Przebieg ćwiczenia:

Język HLSL jest językiem opracowanym w celu programowania shaderów. W niniejszym zadaniu, celem ćwiczenia będzie utworzenie kuli z teksturą wody oraz zaimplementowanie odpowiedniego shadera.

Na początek dodajemy Kulę (3D GameObject):



Następnie dodajemy odpowiednie elementy:



Najważniejszym elementem całego obiektu będzie skrypt C# (Cg shader for billboards).

Zaimplementowany skrypt jest podstawowym skryptem dla shaderów.

Budowa skryptu:

Nazwa shadera:

```
Shader "Cg shader for billboards" {
```

Właściwości shadera:

```
Properties{
    _MainTex("Texture Image", 2D) = "white" {}
    _ScaleX("Scale X", Float) = 1.0
    _ScaleY("Scale Y", Float) = 1.0
}
```

Subshader dla warstwy sprzętowej:

```
SubShader{
    Pass {
        CGPROGRAM

        #pragma vertex vert
        #pragma fragment frag

uniform sampler2D _MainTex;
uniform float _ScaleX;
uniform float _ScaleY;

struct vertexInput {
    float4 vertex : POSITION;
    float4 tex : TEXCOORD0;
};
struct vertexOutput {
    float4 pos : SV_POSITION;
    float4 tex : TEXCOORD0;
};

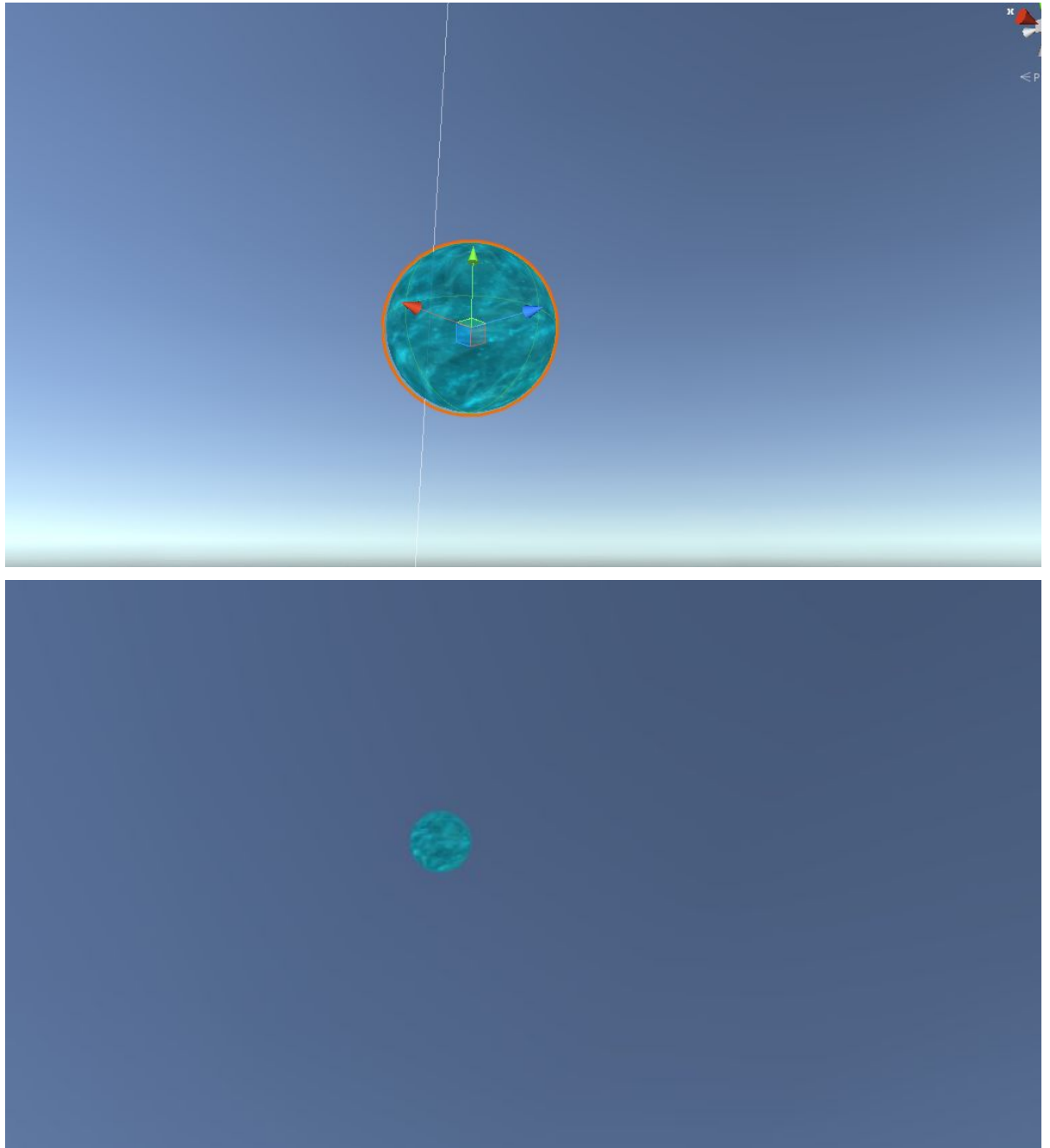
vertexOutput vert(vertexInput input)
{
    vertexOutput output;

    output.pos = mul(UNITY_MATRIX_P,
        mul(UNITY_MATRIX_MV, float4(0.0, 0.0, 0.0, 1.0))
        + float4(input.vertex.x, input.vertex.y, 0.0, 0.0)
        * float4(_ScaleX, _ScaleY, 1.0, 1.0));

    output.tex = input.tex;

    return output;
}
```

Zrzut z działania programu:



3. Wnioski:

Stworzenie prostego Shadera w Unity jest stosunkowo proste. Dzięki tutorialom oraz pomocy od strony społeczności tego środowiska. Pisanie tego typu skryptów staje się dużo prostsze.