

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/javascript">
var gl;
var shaderProgram;
var uPMatrix;
var vertexPositionBuffer;
var vertexColorBuffer;
function MatrixMul(a,b) //Mnożenie macierzy
{
    c = [
        0,0,0,0,
        0,0,0,0,
        0,0,0,0,
        0,0,0,0
    ]
    for(let i=0;i<4;i++)
    {
        for(let j=0;j<4;j++)
        {
            c[i*4+j] = 0.0;
            for(let k=0;k<4;k++)
            {
                c[i*4+j]+= a[i*4+k] * b[k*4+j];
            }
        }
    }
    return c;
}
function startGL()
{
    alert("StartGL");
    let canvas = document.getElementById("canvas3D"); //wyszukanie obiektu w strukturze
    strony
    gl = canvas.getContext("experimental-webgl"); //pobranie kontekstu OpenGL'u z obiektu
    canvas
    gl.viewportWidth = canvas.width; //przypisanie wybranej przez nas rozdzielczości do
    systemu OpenGL
    gl.viewportHeight = canvas.height;

    //Kod shaderów
    const vertexShaderSource = ` //Znak akcentu z przycisku tyldy - na lewo od przycisku 1 na
    klawiaturze
    precision highp float;
    attribute vec3 aVertexPosition;

```

```

attribute vec3 aVertexColor;
attribute vec2 aVertexCoords;
uniform mat4 uMVMMatrix;
uniform mat4 uPMatrix;
varying vec3 vColor;
varying vec2 vTexUV;
void main(void) {
    gl_Position = uPMatrix * uMVMMatrix * vec4(aVertexPosition, 1.0); //Dokonanie
transformacji położenia punktów z przestrzeni 3D do przestrzeni obrazu (2D)
    vColor = aVertexColor;
    vTexUV = aVertexCoords;
}
`;
const fragmentShaderSource = `
precision highp float;
varying vec3 vColor;
varying vec2 vTexUV;
uniform sampler2D uSampler;
void main(void) {
    // gl_FragColor = vec4(vColor,1.0); //Ustalenie stałego koloru wszystkich punktów sceny
    gl_FragColor = texture2D(uSampler,vTexUV); //Odczytanie punktu tekstury i przypisanie
go jako koloru danego punktu renderowanej figury
}
`;
let fragmentShader = gl.createShader(gl.FRAGMENT_SHADER); //Stworzenie obiektu
shadera
let vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(fragmentShader, fragmentShaderSource); //Podpięcie źródła kodu shader
gl.shaderSource(vertexShader, vertexShaderSource);
gl.compileShader(fragmentShader); //Kompilacja kodu shader
gl.compileShader(vertexShader);
if (!gl.getShaderParameter(fragmentShader, gl.COMPILE_STATUS)) { //Sprawdzenie
ewentualnych błędów kompilacji
    alert(gl.getShaderInfoLog(fragmentShader));
    return null;
}
if (!gl.getShaderParameter(vertexShader, gl.COMPILE_STATUS)) {
    alert(gl.getShaderInfoLog(vertexShader));
    return null;
}

shaderProgram = gl.createProgram(); //Stworzenie obiektu programu
gl.attachShader(shaderProgram, vertexShader); //Podpięcie obu shaderów do naszego
programu wykonywanego na karcie graficznej
gl.attachShader(shaderProgram, fragmentShader);
gl.linkProgram(shaderProgram);

```

```
if (!gl.getProgramParameter(shaderProgram, gl.LINK_STATUS)) alert("Could not initialise  
shaders"); //Sprawdzenie ewentualnych błędów
```

```
//Opis sceny 3D, położenie punktów w przestrzeni 3D w formacie X,Y,Z
```

```
let vertexPosition = [
```

```
/*Słońce          160    odległość 5
```

```
  Merkury          1    odległość 5
```

```
  Wenus-12103 km   2    5
```

```
  Ziemia-12756 km  2    6
```

```
  Mars-6879 km     1    40
```

```
  Jowisz-142984 km 25    50
```

```
  Saturn-120536 km 20    60
```

```
  Uran-51118 km    8    90
```

```
  Neptun-49528 km   8
```

```
  Ariel-1158 km     0.0008          1x1
```

```
  Dione-1118 km     0.0008          1x1
```

```
  Europa-3122 km    0.002           2x2
```

```
  Fobos-24 km       0.0000001       0.5x0.5
```

```
  Io-3643 km        0.002            2x2
```

```
  Tytan-5120 km     0.003            3x3*/
```

```
//merkury - przod, tyl, gora, dol, lewo, prawo
```

```
-0.5, -0.5, +210.0, +0.5, -0.5, +210.0, -0.5, +0.5, +210.0,  
-0.5, +0.5, +210.0, +0.5, +0.5, +210.0, +0.5, -0.5, +210.0,
```

```
-0.5, -0.5, +209.0, +0.5, -0.5, +209.0, -0.5, +0.5, +209.0,  
-0.5, +0.5, +209.0, +0.5, +0.5, +209.0, +0.5, -0.5, +209.0,
```

```
-0.5, +0.5, +209.0, +0.5, +0.5, +210.0, +0.5, +0.5, +209.0,  
-0.5, +0.5, +209.0, -0.5, +0.5, +210.0, +0.5, +0.5, +210.0,
```

```
-0.5, -0.5, +210.0, +0.5, -0.5, +210.0, +0.5, -0.5, +209.0,  
+0.5, -0.5, +209.0, -0.5, -0.5, +210.0, -0.5, -0.5, +209.0,
```

```
-0.5, -0.5, +210.0, -0.5, -0.5, +209.0, -0.5, +0.5, +209.0,  
-0.5, +0.5, +209.0, -0.5, -0.5, +210.0, -0.5, +0.5, +210.0,
```

```
+0.5, -0.5, +210.0, +0.5, -0.5, +209.0, +0.5, +0.5, +209.0,  
+0.5, +0.5, +209.0, +0.5, -0.5, +210.0, +0.5, +0.5, +210.0,
```

```
//wenus - przod, tyl, gora, dol, lewo, prawo
```

```
-1.0, -1.0, +204.0, +1.0, -1.0, +204.0, -1.0, +1.0, +204.0,  
-1.0, +1.0, +204.0, +1.0, +1.0, +204.0, +1.0, -1.0, +204.0,
```

```
-1.0, -1.0, +202.0, +1.0, -1.0, +202.0, -1.0, +1.0, +202.0,
```

-1.0, +1.0, +202.0, +1.0, +1.0, +202.0, +1.0, -1.0, +202.0,

-1.0, +1.0, +204.0, +1.0, +1.0, +204.0, +1.0, +1.0, +202.0,
+1.0, +1.0, +202.0, -1.0, +1.0, +204.0, -1.0, +1.0, +202.0,

-1.0, -1.0, +204.0, +1.0, -1.0, +204.0, +1.0, -1.0, +202.0,
+1.0, -1.0, +202.0, -1.0, -1.0, +204.0, -1.0, -1.0, +202.0,

-1.0, -1.0, +204.0, -1.0, -1.0, +202.0, -1.0, +1.0, +202.0,
-1.0, +1.0, +202.0, -1.0, -1.0, +204.0, -1.0, +1.0, +204.0,

+1.0, -1.0, +204.0, +1.0, -1.0, +202.0, +1.0, +1.0, +202.0,
+1.0, +1.0, +202.0, +1.0, -1.0, +204.0, +1.0, +1.0, +204.0,

//ziemia - przod, tyl, gora, dol, lewo, prawo

-1.0, -1.0, +197.0, +1.0, -1.0, +197.0, -1.0, +1.0, +197.0,
-1.0, +1.0, +197.0, +1.0, +1.0, +197.0, +1.0, -1.0, +197.0,

-1.0, -1.0, +195.0, +1.0, -1.0, +195.0, -1.0, +1.0, +195.0,
-1.0, +1.0, +195.0, +1.0, +1.0, +195.0, +1.0, -1.0, +195.0,

-1.0, +1.0, +197.0, +1.0, +1.0, +197.0, +1.0, +1.0, +195.0,
+1.0, +1.0, +195.0, -1.0, +1.0, +197.0, -1.0, +1.0, +195.0,

-1.0, -1.0, +197.0, +1.0, -1.0, +197.0, +1.0, -1.0, +195.0,
+1.0, -1.0, +195.0, -1.0, -1.0, +197.0, -1.0, -1.0, +195.0,

-1.0, -1.0, +197.0, -1.0, -1.0, +195.0, -1.0, +1.0, +195.0,
-1.0, +1.0, +195.0, -1.0, -1.0, +197.0, -1.0, +1.0, +197.0,

+1.0, -1.0, +197.0, +1.0, -1.0, +195.0, +1.0, +1.0, +195.0,
+1.0, +1.0, +195.0, +1.0, -1.0, +197.0, +1.0, +1.0, +197.0,

//księżyc - przod, tyl, gora, dol, lewo, prawo

+2.0, +2.0, +197.0, +2.6, +2.0, +197.0, +2.0, +2.6, +197.0,
+2.0, +2.6, +197.0, +2.6, +2.6, +197.0, +2.6, +2.6, +197.0,

+2.0, +2.0, +196.4, +2.6, +2.0, +196.4, +2.0, +2.6, +196.4,
+2.0, +2.6, +196.4, +2.6, +2.6, +196.4, +2.6, +2.0, +196.4,

+2.0, +2.6, +197.0, +2.6, +2.6, +197.0, +2.6, +2.6, +196.4,
+2.6, +2.6, +196.4, +2.0, +2.6, +197.0, +2.0, +2.6, +196.4,

+2.0, +2.0, +197.0, +2.6, +2.0, +197.0, +2.6, +2.0, +196.4,
+2.6, +2.0, +196.4, +2.0, +2.0, +197.0, +2.0, +2.0, +196.4,

+2.0, +2.0, +197.0, +2.0, +2.0, +196.4, +2.0, +2.6, +196.4,
+2.0, +2.6, +196.4, +2.0, +2.0, +197.0, +2.0, +2.6, +197.0,

+2.6, +2.0, +197.0, +2.6, +2.0, +196.4, +2.6, +2.6, +196.4,
+2.6, +2.6, +196.4, +2.6, +2.0, +197.0, +2.6, +2.6, +197.0,

//mars - przod, tyl, gora, dol, lewo, prawo

-0.5, -0.5, +189.0, +0.5, -0.5, +189.0, -0.5, +0.5, +189.0,
-0.5, +0.5, +189.0, +0.5, +0.5, +189.0, +0.5, -0.5, +189.0,

-0.5, -0.5, +188.0, +0.5, -0.5, +188.0, -0.5, +0.5, +188.0,
-0.5, +0.5, +188.0, +0.5, +0.5, +188.0, +0.5, -0.5, +188.0,

-0.5, +0.5, +188.0, +0.5, +0.5, +189.0, +0.5, +0.5, +188.0,
-0.5, +0.5, +188.0, -0.5, +0.5, +189.0, +0.5, +0.5, +189.0,

-0.5, -0.5, +189.0, +0.5, -0.5, +189.0, +0.5, -0.5, +188.0,
+0.5, -0.5, +188.0, -0.5, -0.5, +189.0, -0.5, -0.5, +188.0,

-0.5, -0.5, +189.0, -0.5, -0.5, +188.0, -0.5, +0.5, +188.0,
-0.5, +0.5, +188.0, -0.5, -0.5, +189.0, -0.5, +0.5, +189.0,

+0.5, -0.5, +189.0, +0.5, -0.5, +188.0, +0.5, +0.5, +188.0,
+0.5, +0.5, +188.0, +0.5, -0.5, +189.0, +0.5, +0.5, +189.0,

//jowisz - przod, tyl, gora, dol, lewo, prawo

-12.0, -12.0, +148.0, +12.0, -12.0, +148.0, -12.0, +12.0, +148.0,
-12.0, +12.0, +148.0, +12.0, +12.0, +148.0, +12.0, -12.0, +148.0,

-12.0, -12.0, +122.0, +12.0, -12.0, +122.0, -12.0, +12.0, +122.0,
-12.0, +12.0, +122.0, +12.0, +12.0, +122.0, +12.0, -12.0, +122.0,

-12.0, +12.0, +122.0, +12.0, +12.0, +148.0, +12.0, +12.0, +122.0,
-12.0, +12.0, +122.0, -12.0, +12.0, +148.0, +12.0, +12.0, +148.0,

-12.0, -12.0, +148.0, +12.0, -12.0, +148.0, +12.0, -12.0, +122.0,
+12.0, -12.0, +122.0, -12.0, -12.0, +148.0, -12.0, -12.0, +122.0,

-12.0, -12.0, +148.0, -12.0, -12.0, +122.0, -12.0, +12.0, +122.0,
-12.0, +12.0, +122.0, -12.0, -12.0, +148.0, -12.0, +12.0, +148.0,

+12.0, -12.0, +148.0, +12.0, -12.0, +122.0, +12.0, +12.0, +122.0,

+12.0, +12.0, +122.0, +12.0, -12.0, +148.0, +12.0, +12.0, +148.0,

//saturn - przod, tyl, gora, dol, lewo, prawo

-10.0, -10.0, +72.0, +10.0, -10.0, +72.0, -10.0, +10.0, +72.0,
-10.0, +10.0, +72.0, +10.0, +10.0, +72.0, +10.0, -10.0, +72.0,

-10.0, -10.0, +52.0, +10.0, -10.0, +52.0, -10.0, +10.0, +52.0,
-10.0, +10.0, +52.0, +10.0, +10.0, +52.0, +10.0, -10.0, +52.0,

-10.0, +10.0, +52.0, +10.0, +10.0, +72.0, +10.0, +10.0, +52.0,
-10.0, +10.0, +52.0, -10.0, +10.0, +72.0, +10.0, +10.0, +72.0,

-10.0, -10.0, +72.0, +10.0, -10.0, +72.0, +10.0, -10.0, +52.0,
+10.0, -10.0, +52.0, -10.0, -10.0, +72.0, -10.0, -10.0, +52.0,

-10.0, -10.0, +72.0, -10.0, -10.0, +52.0, -10.0, +10.0, +52.0,
-10.0, +10.0, +52.0, -10.0, -10.0, +72.0, -10.0, +10.0, +72.0,

+10.0, -10.0, +72.0, +10.0, -10.0, +52.0, +10.0, +10.0, +52.0,
+10.0, +10.0, +52.0, +10.0, -10.0, +72.0, +10.0, +10.0, +72.0,

//uran - przod, tyl, gora, dol, lewo, prawo

-4.0, -4.0, -8.0, +4.0, -4.0, -8.0, -4.0, +4.0, -8.0,
-4.0, +4.0, -8.0, +4.0, +4.0, -8.0, +4.0, -4.0, -8.0,

-4.0, -4.0, -16.0, +4.0, -4.0, -16.0, -4.0, +4.0, -16.0,
-4.0, +4.0, -16.0, +4.0, +4.0, -16.0, +4.0, -4.0, -16.0,

-4.0, +4.0, -16.0, +4.0, +4.0, -8.0, +4.0, +4.0, -16.0,
-4.0, +4.0, -16.0, -4.0, +4.0, -8.0, +4.0, +4.0, -8.0,

-4.0, -4.0, -8.0, +4.0, -4.0, -8.0, +4.0, -4.0, -16.0,
+4.0, -4.0, -16.0, -4.0, -4.0, -8.0, -4.0, -4.0, -16.0,

-4.0, -4.0, -8.0, -4.0, -4.0, -16.0, -4.0, +4.0, -16.0,
-4.0, +4.0, -16.0, -4.0, -4.0, -8.0, -4.0, +4.0, -8.0,

+4.0, -4.0, -8.0, +4.0, -4.0, -16.0, +4.0, +4.0, -16.0,
+4.0, +4.0, -16.0, +4.0, -4.0, -8.0, +4.0, +4.0, -8.0,

//neptun - przod, tyl, gora, dol, lewo, prawo

-4.0, -4.0, -106.0, +4.0, -4.0, -106.0, -4.0, +4.0, -106.0,

-4.0, +4.0, -106.0, +4.0, +4.0, -106.0, +4.0, -4.0, -106.0,

-4.0, -4.0, -114.0, +4.0, -4.0, -114.0, -4.0, +4.0, -114.0,
-4.0, +4.0, -114.0, +4.0, +4.0, -114.0, +4.0, -4.0, -114.0,

-4.0, +4.0, -114.0, +4.0, +4.0, -106.0, +4.0, +4.0, -114.0,
-4.0, +4.0, -114.0, -4.0, +4.0, -106.0, +4.0, +4.0, -106.0,

-4.0, -4.0, -106.0, +4.0, -4.0, -106.0, +4.0, -4.0, -114.0,
+4.0, -4.0, -114.0, -4.0, -4.0, -106.0, -4.0, -4.0, -114.0,

-4.0, -4.0, -106.0, -4.0, -4.0, -114.0, -4.0, +4.0, -114.0,
-4.0, +4.0, -114.0, -4.0, -4.0, -106.0, -4.0, +4.0, -106.0,

+4.0, -4.0, -106.0, +4.0, -4.0, -114.0, +4.0, +4.0, -114.0,
+4.0, +4.0, -114.0, +4.0, -4.0, -106.0, +4.0, +4.0, -106.0,

//słonce - przod, tyl, gora, dol, lewo, prawo

-40.0, -40.0, +320.0, +40.0, -40.0, +320.0, -40.0, +40.0, +320.0,
-40.0, +40.0, +320.0, +40.0, +40.0, +320.0, +40.0, -40.0, +320.0,

-40.0, -40.0, +240.0, +40.0, -40.0, +240.0, -40.0, +40.0, +240.0,
-40.0, +40.0, +240.0, +40.0, +40.0, +240.0, +40.0, -40.0, +240.0,

-40.0, +40.0, +240.0, +40.0, +40.0, +320.0, +40.0, +40.0, +240.0,
-40.0, +40.0, +240.0, -40.0, +40.0, +320.0, +40.0, +40.0, +320.0,

-40.0, -40.0, +320.0, +40.0, -40.0, +320.0, +40.0, -40.0, +240.0,
+40.0, -40.0, +240.0, -40.0, -40.0, +310.0, -40.0, -40.0, +240.0,

-40.0, -40.0, +320.0, -40.0, -40.0, +240.0, -40.0, +40.0, +240.0,
-40.0, +40.0, +240.0, -40.0, -40.0, +320.0, -40.0, +40.0, +320.0,

+40.0, -40.0, +320.0, +40.0, -40.0, +240.0, +40.0, +40.0, +240.0,
+40.0, +40.0, +240.0, +40.0, -40.0, +320.0, +40.0, +40.0, +320.0,

]

```
vertexPositionBuffer = gl.createBuffer(); //Stworzenie tablicy w pamieci karty graficznej
gl.bindBuffer(gl.ARRAY_BUFFER, vertexPositionBuffer);
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertexPosition), gl.STATIC_DRAW);
vertexPositionBuffer.itemSize = 3; //zdefiniowanie liczby współrzędnych per wierzchołek
```

```
vertexPositionBuffer.numItems = 12*10; //Zdefiniowanie liczby punktów w naszym buforze
```

```
//Opis sceny 3D, kolor każdego z wierzchołków
```

```
let vertexColor = [
```

```
]
```

```
vertexColorBuffer = gl.createBuffer();
```

```
gl.bindBuffer(gl.ARRAY_BUFFER, vertexColorBuffer);
```

```
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertexColor), gl.STATIC_DRAW);
```

```
vertexColorBuffer.itemSize = 3;
```

```
vertexColorBuffer.numItems = 12*10;
```

```
let vertexCoords = [
```

```
    //merkury
```

```
    0.0, 0.0,  0.1, 0.0,  0.0, 1.0,
```

```
    0.0, 1.0,  0.1, 1.0,  0.1, 0.0,
```

```
    0.0, 0.0,  0.1, 0.0,  0.0, 1.0,
```

```
    0.0, 1.0,  0.1, 1.0,  0.1, 0.0,
```

```
    0.0, 0.0,  0.1, 0.0,  0.1, 1.0,
```

```
    0.1, 1.0,  0.0, 0.0,  0.0, 1.0,
```

```
    0.0, 0.0,  0.1, 0.0,  0.1, 1.0,
```

```
    0.1, 1.0,  0.0, 0.0,  0.0, 1.0,
```

```
    0.0, 0.0,  0.1, 0.0,  0.1, 1.0,
```

```
    0.1, 1.0,  0.0, 0.0,  0.0, 1.0,
```

```
    0.0, 0.0,  0.1, 0.0,  0.1, 1.0,
```

```
    0.1, 1.0,  0.0, 0.0,  0.0, 1.0,
```

```
// wenus
```


0.1, 0.0, 0.2, 0.0, 0.1, 1.0,
0.1, 1.0, 0.2, 1.0, 0.2, 0.0,

0.1, 0.0, 0.2, 0.0, 0.1, 1.0,
0.1, 1.0, 0.2, 1.0, 0.2, 0.0,

0.1, 0.0, 0.2, 0.0, 0.2, 1.0,
0.2, 1.0, 0.1, 0.0, 0.1, 1.0,

0.1, 0.0, 0.2, 0.0, 0.2, 1.0,
0.2, 1.0, 0.1, 0.0, 0.1, 1.0,

0.1, 0.0, 0.2, 0.0, 0.2, 1.0,
0.2, 1.0, 0.1, 0.0, 0.1, 1.0,

0.1, 0.0, 0.2, 0.0, 0.2, 1.0,
0.2, 1.0, 0.1, 0.0, 0.1, 1.0,

// ziemia

0.2, 0.0, 0.3, 0.0, 0.2, 1.0,
0.2, 1.0, 0.3, 1.0, 0.3, 0.0,

0.2, 0.0, 0.3, 0.0, 0.2, 1.0,
0.2, 1.0, 0.3, 1.0, 0.3, 0.0,

0.2, 0.0, 0.3, 0.0, 0.3, 1.0,
0.3, 1.0, 0.2, 0.0, 0.2, 1.0,

0.2, 0.0, 0.3, 0.0, 0.3, 1.0,
0.3, 1.0, 0.2, 0.0, 0.2, 1.0,

0.2, 0.0, 0.3, 0.0, 0.3, 1.0,
0.3, 1.0, 0.2, 0.0, 0.2, 1.0,

0.2, 0.0, 0.3, 0.0, 0.3, 1.0,
0.3, 1.0, 0.2, 0.0, 0.2, 1.0,

//ksiezyc

0.9, 0.0, 1.0, 0.0, 0.9, 1.0,
0.9, 1.0, 1.0, 1.0, 1.0, 0.0,

0.9, 0.0, 1.0, 0.0, 0.9, 1.0,
0.9, 1.0, 1.0, 1.0, 1.0, 0.0,

0.9, 0.0, 1.0, 0.0, 1.0, 1.0,
1.0, 1.0, 0.9, 0.0, 0.9, 1.0,

0.9, 0.0, 1.0, 0.0, 1.0, 1.0,
1.0, 1.0, 0.9, 0.0, 0.9, 1.0,

0.9, 0.0, 1.0, 0.0, 1.0, 1.0,
1.0, 1.0, 0.9, 0.0, 0.9, 1.0,

0.9, 0.0, 1.0, 0.0, 1.0, 1.0,
1.0, 1.0, 0.9, 0.0, 0.9, 1.0,

// mars

0.3, 0.0, 0.4, 0.0, 0.3, 1.0,
0.3, 1.0, 0.4, 1.0, 0.4, 0.0,

0.3, 0.0, 0.4, 0.0, 0.3, 1.0,
0.3, 1.0, 0.4, 1.0, 0.4, 0.0,

0.3, 0.0, 0.4, 0.0, 0.4, 1.0,
0.4, 1.0, 0.3, 0.0, 0.3, 1.0,

0.3, 0.0, 0.4, 0.0, 0.4, 1.0,
0.4, 1.0, 0.3, 0.0, 0.3, 1.0,

0.3, 0.0, 0.4, 0.0, 0.4, 1.0,
0.4, 1.0, 0.3, 0.0, 0.3, 1.0,

0.3, 0.0, 0.4, 0.0, 0.4, 1.0,
0.4, 1.0, 0.3, 0.0, 0.3, 1.0,

//jowisz

0.4, 0.0, 0.5, 0.0, 0.4, 1.0,
0.4, 1.0, 0.5, 1.0, 0.5, 0.0,

0.4, 0.0, 0.5, 0.0, 0.4, 1.0,
0.4, 1.0, 0.5, 1.0, 0.5, 0.0,

0.4, 0.0, 0.5, 0.0, 0.5, 1.0,
0.5, 1.0, 0.4, 0.0, 0.4, 1.0,

0.4, 0.0, 0.5, 0.0, 0.5, 1.0,
0.5, 1.0, 0.4, 0.0, 0.4, 1.0,

0.4, 0.0, 0.5, 0.0, 0.5, 1.0,

0.5, 1.0, 0.4, 0.0, 0.4, 1.0,

0.4, 0.0, 0.5, 0.0, 0.5, 1.0,
0.5, 1.0, 0.4, 0.0, 0.4, 1.0,

//saturn

0.5, 0.0, 0.6, 0.0, 0.5, 1.0,
0.5, 1.0, 0.6, 1.0, 0.6, 0.0,

0.5, 0.0, 0.6, 0.0, 0.5, 1.0,
0.5, 1.0, 0.6, 1.0, 0.6, 0.0,

0.5, 0.0, 0.6, 0.0, 0.6, 1.0,
0.6, 1.0, 0.5, 0.0, 0.5, 1.0,

0.5, 0.0, 0.6, 0.0, 0.6, 1.0,
0.6, 1.0, 0.5, 0.0, 0.5, 1.0,

0.5, 0.0, 0.6, 0.0, 0.6, 1.0,
0.6, 1.0, 0.5, 0.0, 0.5, 1.0,

0.5, 0.0, 0.6, 0.0, 0.6, 1.0,
0.6, 1.0, 0.5, 0.0, 0.5, 1.0,

//uran

0.6, 0.0, 0.7, 0.0, 0.6, 1.0,
0.6, 1.0, 0.7, 1.0, 0.7, 0.0,

0.6, 0.0, 0.7, 0.0, 0.6, 1.0,
0.6, 1.0, 0.7, 1.0, 0.7, 0.0,

0.6, 0.0, 0.7, 0.0, 0.7, 1.0,
0.7, 1.0, 0.6, 0.0, 0.6, 1.0,

0.6, 0.0, 0.7, 0.0, 0.7, 1.0,
0.7, 1.0, 0.6, 0.0, 0.6, 1.0,

0.6, 0.0, 0.7, 0.0, 0.7, 1.0,
0.7, 1.0, 0.6, 0.0, 0.6, 1.0,

0.6, 0.0, 0.7, 0.0, 0.7, 1.0,
0.7, 1.0, 0.6, 0.0, 0.6, 1.0,

//neptun

```
0.7, 0.0, 0.8, 0.0, 0.7, 1.0,  
0.7, 1.0, 0.8, 1.0, 0.8, 0.0,
```

```
0.7, 0.0, 0.8, 0.0, 0.7, 1.0,  
0.7, 1.0, 0.8, 1.0, 0.8, 0.0,
```

```
0.7, 0.0, 0.8, 0.0, 0.8, 1.0,  
0.8, 1.0, 0.7, 0.0, 0.7, 1.0,
```

```
0.7, 0.0, 0.8, 0.0, 0.8, 1.0,  
0.8, 1.0, 0.7, 0.0, 0.7, 1.0,
```

```
0.7, 0.0, 0.8, 0.0, 0.8, 1.0,  
0.8, 1.0, 0.7, 0.0, 0.7, 1.0,
```

```
0.7, 0.0, 0.8, 0.0, 0.8, 1.0,  
0.8, 1.0, 0.7, 0.0, 0.7, 1.0,
```

```
//slonce
```

```
0.8, 0.0, 0.9, 0.0, 0.8, 1.0,  
0.8, 1.0, 0.9, 1.0, 0.9, 0.0,
```

```
0.8, 0.0, 0.9, 0.0, 0.8, 1.0,  
0.8, 1.0, 0.9, 1.0, 0.9, 0.0,
```

```
0.8, 0.0, 0.9, 0.0, 0.9, 1.0,  
0.9, 1.0, 0.8, 0.0, 0.8, 1.0,
```

```
0.8, 0.0, 0.9, 0.0, 0.9, 1.0,  
0.9, 1.0, 0.8, 0.0, 0.8, 1.0,
```

```
0.8, 0.0, 0.9, 0.0, 0.9, 1.0,  
0.9, 1.0, 0.8, 0.0, 0.8, 1.0,
```

```
0.8, 0.0, 0.9, 0.0, 0.9, 1.0,  
0.9, 1.0, 0.8, 0.0, 0.8, 1.0,
```

```
//
```

```
];
```

```
vertexCoordsBuffer = gl.createBuffer();  
gl.bindBuffer(gl.ARRAY_BUFFER, vertexCoordsBuffer);
```

```

gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertexCoords), gl.STATIC_DRAW);
vertexCoordsBuffer.itemSize = 2;
vertexCoordsBuffer.numItems = 12*10;

textureBuffer = gl.createTexture();
var textureImg = new Image();
textureImg.onload = function() { //Wykonanie kodu automatycznie po załadowaniu obrazka
    gl.bindTexture(gl.TEXTURE_2D, textureBuffer);
    gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE,
textureImg); //Faktyczne załadowanie danych obrazu do pamięci karty graficznej
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_S, gl.CLAMP_TO_EDGE);
//Ustawienie parametrów próbkowania tekstury
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_T, gl.CLAMP_TO_EDGE);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR);
}
textureImg.src="planety.png"; //Nazwa obrazka // jest na github //

//Macierze opisujące położenie wirtualnej kamery w przestrzeni 3D
let aspect = gl.viewportWidth/gl.viewportHeight;
let fov = 45.0 * Math.PI / 180.0; //Określenie pola widzenia kamery
let zFar = 600.0; //Ustalenie zakresów renderowania sceny 3D (od obiektu najbliższego
zNear do najdalszego zFar)
let zNear = 0.1;
uPMatrix = [
    1.0/(aspect*Math.tan(fov/2)),0,0,0,
    0,1.0/(Math.tan(fov/2)),0,0,
    0,0,-(zFar+zNear)/(zFar-zNear),-1,
    0,0,-(2*zFar*zNear)/(zFar-zNear),0.0,
];
Tick();
}

//let angle = 45.0; //Macierz transformacji świata - określenie położenia kamery
var angleZ = 0.0;
var angleY = -45.0;
var angleX = 0.0;
var tz = -240.0;
var ty = 0.0;
var tx = -70.0;
function Tick()
{
    let uMVMatrix = [
    1,0,0,0, //Macierz jednostkowa
    0,1,0,0,
    0,0,1,0,
    0,0,0,1

```

```
];
```

```
let uMVRotZ = [  
+Math.cos(angleZ*Math.PI/180.0),+Math.sin(angleZ*Math.PI/180.0),0,0,  
-Math.sin(angleZ*Math.PI/180.0),+Math.cos(angleZ*Math.PI/180.0),0,0,  
0,0,1,0,  
0,0,0,1  
];
```

```
let uMVRotY = [  
+Math.cos(angleY*Math.PI/180.0),0,-Math.sin(angleY*Math.PI/180.0),0,  
0,1,0,0,  
+Math.sin(angleY*Math.PI/180.0),0,+Math.cos(angleY*Math.PI/180.0),0,  
0,0,0,1  
];
```

```
let uMVRotX = [  
1,0,0,0,  
0,+Math.cos(angleX*Math.PI/180.0),+Math.sin(angleX*Math.PI/180.0),0,  
0,-Math.sin(angleX*Math.PI/180.0),+Math.cos(angleX*Math.PI/180.0),0,  
0,0,0,1  
];
```

```
let uMVTranslateZ = [  
1,0,0,0,  
0,1,0,0,  
0,0,1,0,  
0,0,tz,1  
];
```

```
let uMVTranslateY = [  
1,0,0,0,  
0,1,0,0,  
0,0,1,0,  
0,ty,0,1  
];
```

```
let uMVTranslateX = [  
1,0,0,0,  
0,1,0,0,  
0,0,1,0,  
tx,0,0,1  
];
```

```
uMVMatrix = MatrixMul(uMVMatrix,uMVTranslateZ);
```

```
uMVMatrix = MatrixMul(uMVMatrix,uMVTranslateY);
uMVMatrix = MatrixMul(uMVMatrix,uMVTranslateX);
uMVMatrix = MatrixMul(uMVMatrix,uMVRotX);
uMVMatrix = MatrixMul(uMVMatrix,uMVRotY);
uMVMatrix = MatrixMul(uMVMatrix,uMVRotZ);
```

```
//alert(uPMatrix);
```

```
//Render Scene
```

```
gl.viewport(0, 0, gl.viewportWidth, gl.viewportHeight);
gl.clearColor(0.0,0.0,0.0,1.0); //Wyczyszczenie obrazu kolorem czerwonym
gl.clearDepth(1.0);           //Wyczyścienie bufora głębi najdalszym planem
gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
gl.useProgram(shaderProgram) //Użycie przygotowanego programu shaderowego
```

```
gl.enable(gl.DEPTH_TEST);      // Włączenie testu głębi - obiekty bliższe mają
przykrywać obiekty dalsze
gl.depthFunc(gl.LEQUAL);      //
```

```
gl.uniformMatrix4fv(gl.getUniformLocation(shaderProgram, "uPMatrix"), false, new
Float32Array(uPMatrix)); //Wgranie macierzy kamery do pamięci karty graficznej
gl.uniformMatrix4fv(gl.getUniformLocation(shaderProgram, "uMVMatrix"), false, new
Float32Array(uMVMatrix));
```

```
gl.enableVertexAttribArray(gl.getAttribLocation(shaderProgram, "aVertexPosition"));
//Przekazanie położenia
gl.bindBuffer(gl.ARRAY_BUFFER, vertexPositionBuffer);
gl.vertexAttribPointer(gl.getAttribLocation(shaderProgram, "aVertexPosition"),
vertexPositionBuffer.itemSize, gl.FLOAT, false, 0, 0);
```

```
gl.enableVertexAttribArray(gl.getAttribLocation(shaderProgram, "aVertexColor"));
//Przekazanie kolorów
gl.bindBuffer(gl.ARRAY_BUFFER, vertexColorBuffer);
gl.vertexAttribPointer(gl.getAttribLocation(shaderProgram, "aVertexColor"),
vertexColorBuffer.itemSize, gl.FLOAT, false, 0, 0);
```

```
gl.enableVertexAttribArray(gl.getAttribLocation(shaderProgram, "aVertexCoords")); //Pass
the geometry
gl.bindBuffer(gl.ARRAY_BUFFER, vertexCoordsBuffer);
gl.vertexAttribPointer(gl.getAttribLocation(shaderProgram, "aVertexCoords"),
vertexCoordsBuffer.itemSize, gl.FLOAT, false, 0, 0);
```

```

gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, textureBuffer);
gl.uniform1i(gl.getUniformLocation(shaderProgram, "uSampler"), 0);

gl.drawArrays(gl.TRIANGLES, 0,
vertexPositionBuffer.numItems*vertexPositionBuffer.itemSize); //Faktyczne wywołanie
rendrowania

    setTimeout(Tick, 100);
}
function handlekeydown(e)
{
    if(e.keyCode==73) tz=tz+5.0; //i - przod
    if(e.keyCode==75) tz=tz-5.0; //k - tyl
    if(e.keyCode==74) tx=tx+2.0; //j - lewo
    if(e.keyCode==76) tx=tx-2.0; //l - prawo
    if(e.keyCode==68) angleY=angleY+3.0; //a
    if(e.keyCode==65) angleY=angleY-3.0; //d

    //alert(e.keyCode);
    //alert(angleX);
}
</script>
</head>
<body onload="startGL()" onkeydown="handlekeydown(event)">
<canvas id="canvas3D" width="640" height="480" style="border: solid black 1px"></canvas>

</body>
</html>

```