Assignment 11: Simple Vertex and Fragment Shader

In this assignment you have to write a Vertex and a Fragment Shader for showing a starship with a rotating palette of colors.

The *Vertex Shader* receives the cartesian local coordinates of each vertex in variable

```
vec3 inPosition;
```

and respectively the world, view an projection matrices in components

```
mat4 worldMat;
mat4 viewMat;
mat4 prjMat;
```

of the black called `ubo` . The clipping coordinates of the vertex should be returned in variable

```
vec4 gl_Position;
```

The *Fragment Shader* must compute a time dependent color and return it in the four component variable

```
vec4 outColor;
```

The first three of the four elements are respectively the red, green and blue components of the color, all in the 0-1 range. The last element of the vector is the transparency, and should be always set to 1 to avoid graphic card dependent showing problems. The fragment shader can use the

```
float time;
```

component of the black called `gubo` to have a timer count. In particular, it is a number in the range 0-1, representing the current fraction of second. The way in which the color alternates is left to the student, and can be anything. An interesting reference can be taken here:

https://iquilezles.org/articles/palettes/
(since the URL has become no-longer trustable, I had printed the page and included in the ZIP file of the assignment).

Shaders are found in files `Shader.vert` and `Shader.frag` of the `shaders` folder. Please remember to compile them (as done in Assignment 10), to allow the main program loading them. In particular, the following naming conventions should be used:

Source ->	**Binary**
`Shader.frag`	`Frag.spv`
`Shader.vert`	`Vert.spv`