Real Time Graphics Lab C.

# Week 3 – Lab C

## Exercise 1.

Transform the cube into following shape using scaling transformation.

### Solution:

SimpleVertex vertices[] =

{

{ XMFLOAT3( -1.0f, 3.0f, -1.0f ), XMFLOAT4( 0.0f, 0.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, 3.0f, -1.0f ), XMFLOAT4( 0.0f, 1.0f, 0.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, 3.0f, 1.0f ), XMFLOAT4( 0.0f, 1.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( -1.0f, 3.0f, 1.0f ), XMFLOAT4( 1.0f, 0.0f, 0.0f, 1.0f ) },

{ XMFLOAT3( -1.0f, -3.0f, -1.0f ), XMFLOAT4( 1.0f, 0.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, -3.0f, -1.0f ), XMFLOAT4( 1.0f, 1.0f, 0.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, -3.0f, 1.0f ), XMFLOAT4( 1.0f, 1.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( -1.0f, -3.0f, 1.0f ), XMFLOAT4( 0.0f, 0.0f, 0.0f, 1.0f ) },

};

### Sample Output:

### Solution:

SimpleVertex vertices[] =

{

{ XMFLOAT3( -1.0f, -3.0f, -1.0f ), XMFLOAT4( 0.0f, 0.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, -3.0f, -1.0f ), XMFLOAT4( 0.0f, 1.0f, 0.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, -3.0f, 1.0f ), XMFLOAT4( 0.0f, 1.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( -1.0f, -3.0f, 1.0f ), XMFLOAT4( 1.0f, 0.0f, 0.0f, 1.0f ) },

{ XMFLOAT3( -1.0f, -3.0f, -1.0f ), XMFLOAT4( 1.0f, 0.0f, 1.0f, 1.0f ) },

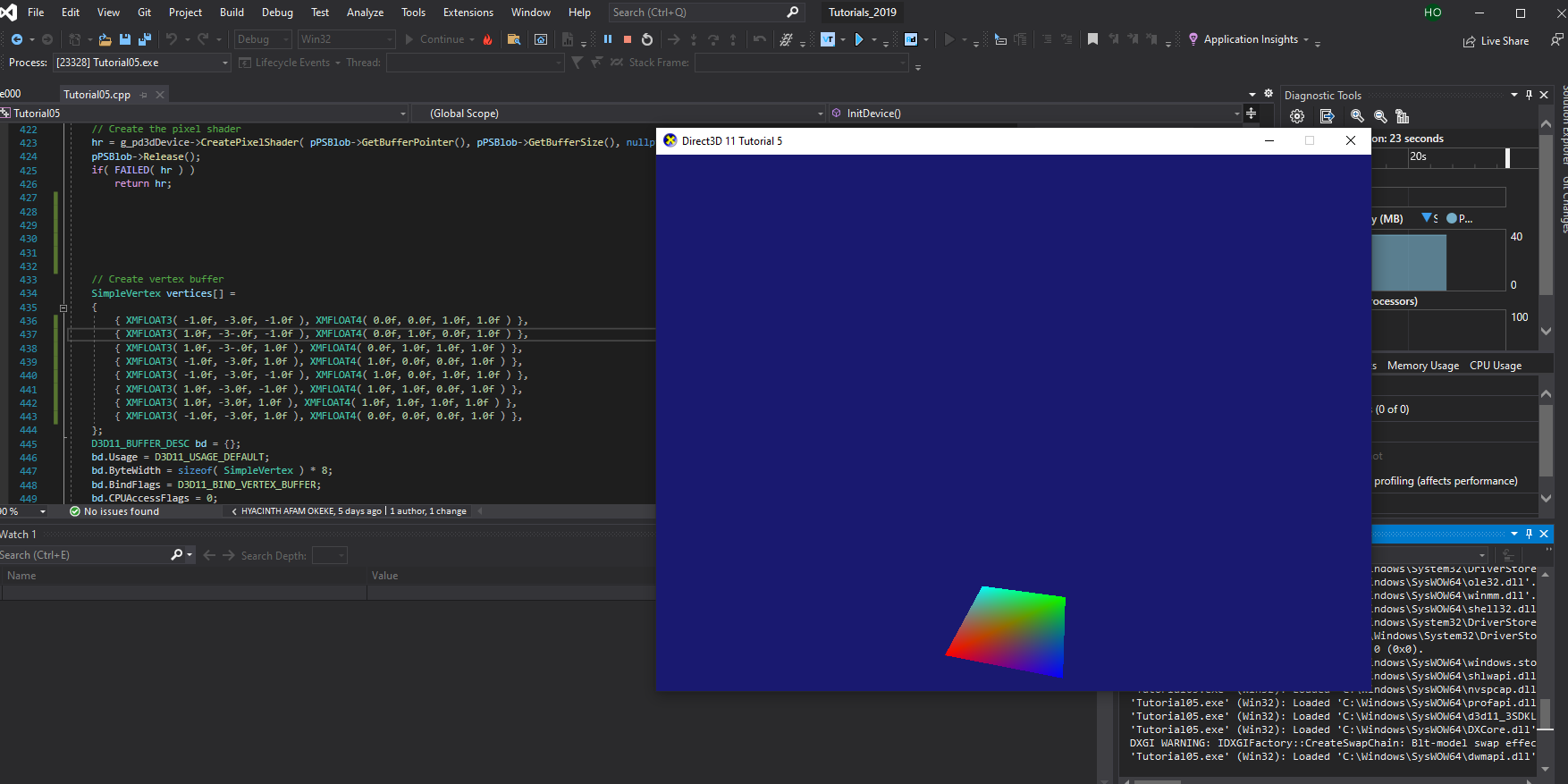
{ XMFLOAT3( 1.0f, -3.0f, -1.0f ), XMFLOAT4( 1.0f, 1.0f, 0.0f, 1.0f ) },

{ XMFLOAT3( 1.0f, -3.0f, 1.0f ), XMFLOAT4( 1.0f, 1.0f, 1.0f, 1.0f ) },

{ XMFLOAT3( -1.0f, -3.0f, 1.0f ), XMFLOAT4( 0.0f, 0.0f, 0.0f, 1.0f ) },

};

Sample Output:



### Test data:

N/A

### Reflection:

### Metadata:

N/A

### Further information:

N/A

**Exercise 2:**

Perform scaling, translation and rotation transformation to achieve the following effects: (1) a cube   
rotates by a vertical rotation axis (as shown in figure (a)); (2). Two cubes rotate by two different   
rotational axes with different rotational speeds respectively.

Solution:

Sample Output:

**Exercise 3:**

In Exercise 2(a), scale the small rotating cube into a long-thin stick and rote the stick by a rotation

axis such that the stick is always tangent to the rotation path, as shown below. If you are able to do

this, also consider how to get the stick flying along a general curve.

**Exercise 4:**

Scale the cube into different sizes corresponding to the Sun, the Earth and the Moon respectively   
and then combine a set of rotation and translation transformations to animate a simple solar   
system.

**Exercise 5:**

In the Tutorial04, the view transformation and projection transformation are created from the   
XMMatrixLookAtLH( ) method and the XMMatrixPerspectiveFovLH() method. To develop a better   
understanding of the two transformations, you can directly define the matrices in your c++ program   
and observe if you can get exactly the same effect. You can define the two matrices directly using   
XMMatrixSet( ) method.