Lab 1 Report:

Visualization using Polygons

CZ2003 – Computer Graphics & Visualization

Wilson Thurman Teng
U1820540H
Lab Group: SSR2

Contents

Lab 1

## 1.1 Different Graphic Modes (Pyramid example)

The Pyramid is made up of 5 points and 5 sides (1 bottom square and 4 triangles).

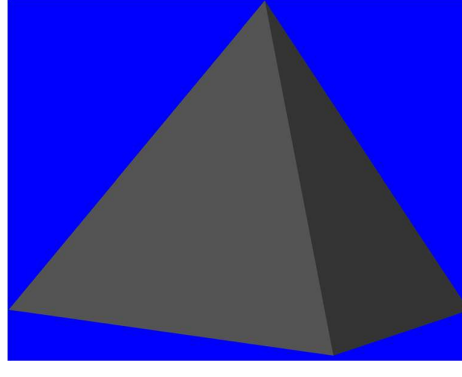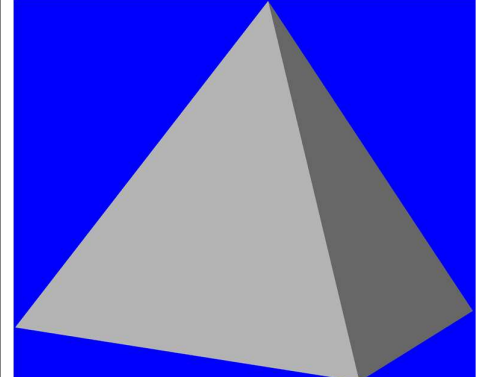| Pyramid.wrl		
Smooth Mode	*Wireframe Mode*	*Hidden Line Mode*
This mode displays the 3D shape and the edges of the shape as they reflect the light clearly.	This mode displays the edges that form the pyramid as white lines.	BSContact uses triangle polygons to make up its shapes. This can be clearly seen by the bottom square made up by 2 triangles. This mode displays the triangles used to make the Pyramid.
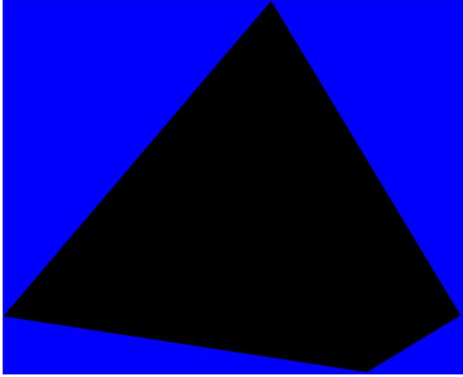
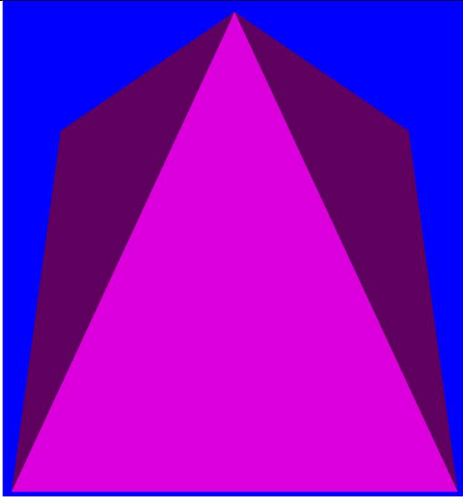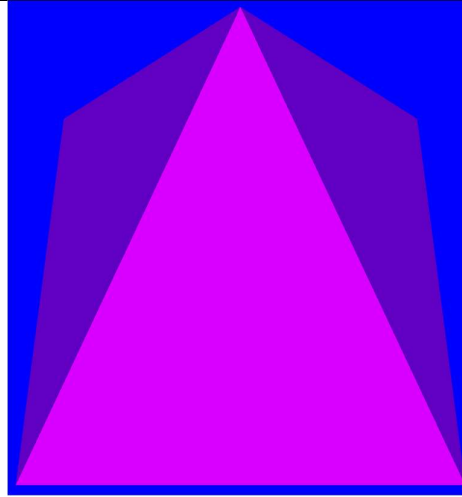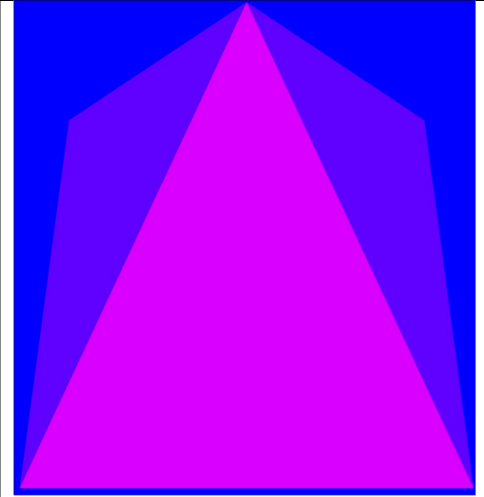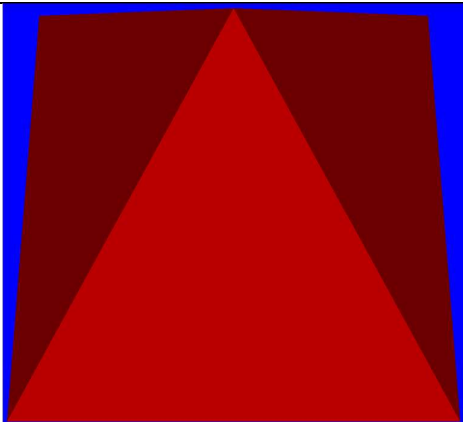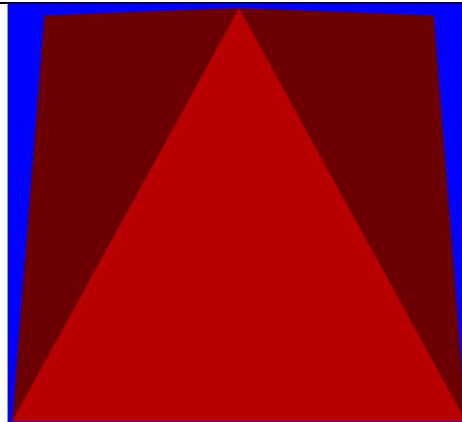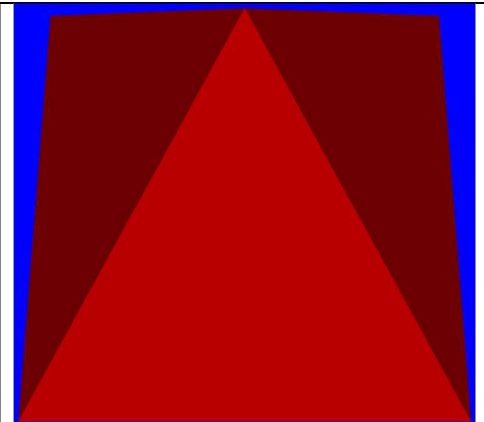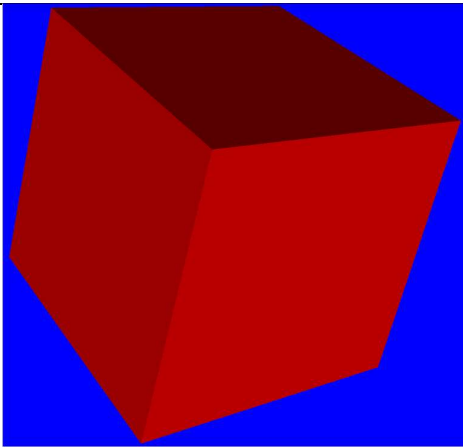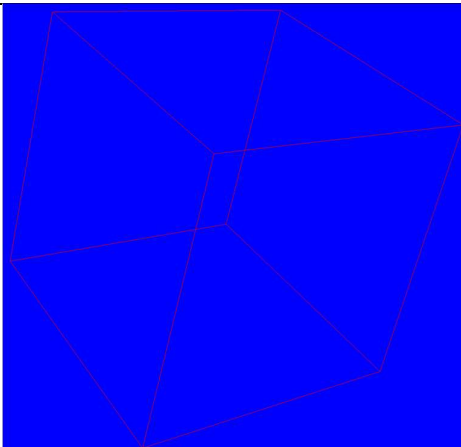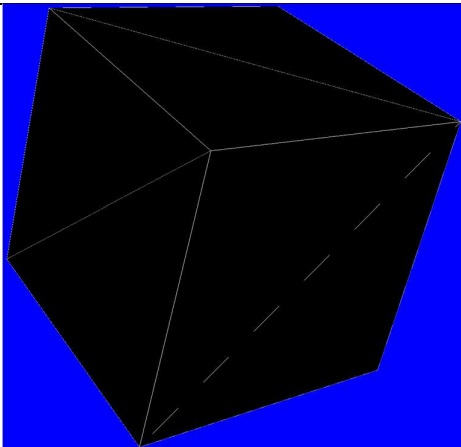| Pyramid.wrl		
diffuseColor(R,G,B) = (0,0,0)	*diffuseColor(R,G,B) = (0.5,0.5,0.5)*	*diffuseColor(R,G,B) = (1,1,1)*
When (R,G,B)=(0,0,0), the shape will be black in colour since there is no colour. When (R,G,B)=(1,1,1), shape will be white in colour as the combination of Red, Green & Blue in maximum equal proportions results in white.		
diffuseColor(R,G,B) = (1,0,1)	*diffuseColor(R,G,B) = (1,0,2)*	*diffuseColor(R,G,B) = (1,0,10)*
As observed, when the value of *diffuseColor* exceeds 1, there is an overexposure in the colours resulting from the reflected light. The shape with a higher value for *diffuseColor* is brighter compared to that of a lower value.		
diffuseColor(R,G,B) = (1,0,0)	*diffuseColor(R,G,B) = (1,0,-1)*	*diffuseColor(R,G,B) = (1,0,-10)*
As observed, regardless of the value of *diffuseColor*, if it is negative, the resulting shape will still have the same colour as a shape with the value of *diffuseColor* set to 0.		

In conclusion, *diffuseColor* is the main shading colour. It reflects all light sources depending on the angle of the surface with respect to the light source: the more directly the surface faces the light, the more diffuse light reflects.
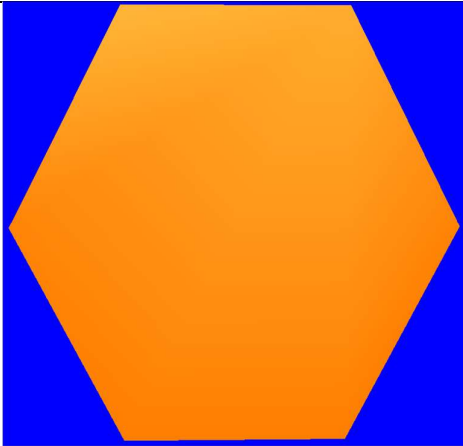
When the value is negative, the value is not considered.

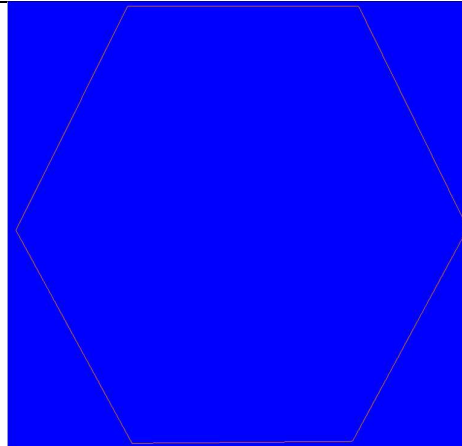When the value is more than 1, shape surfaces are overexposed.

1.3 New Shapes

Cube.wrl		
Smooth Mode	*Wireframe Mode*	*Hidden Lines Mode*

The Cube is made up of 8 points and 6 sides (Squares of equal sizes). The 8 points are:

```
# bottom vertices
1    -1    -1    #vertex 0
-1   -1    -1    #vertex 1
-1    1    -1    #vertex 2
1     1    -1    #vertex 3
# upper vertices
1    -1     1    #vertex 4
-1   -1     1    #vertex 5
-1    1     1    #vertex 6
1     1     1    #vertex 7
```
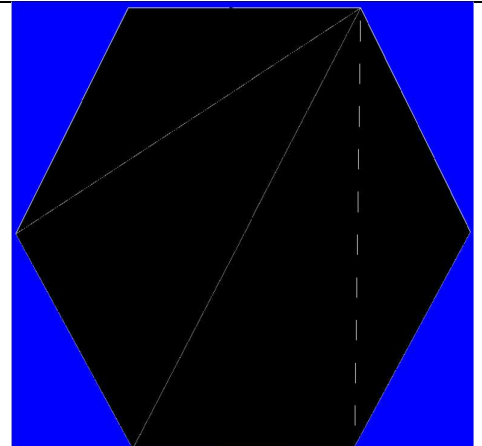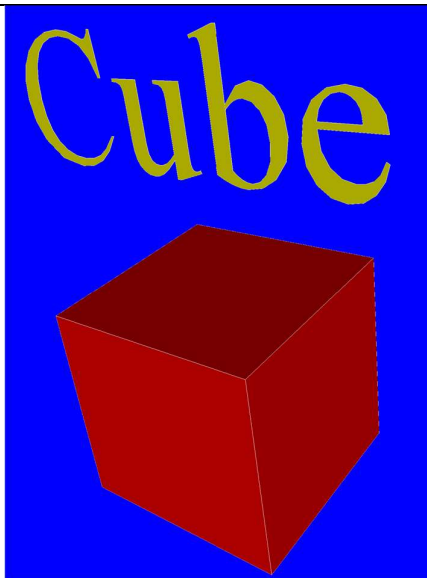
Hexagon_2D.wrl		
Smooth Mode	*Wireframe Mode*	*Hidden Lines Mode*

The Cube is made up of 6 points and 2 sides (Squares of equal sizes). The 6 points are:

0.5	1	0	#vertex 0
-0.5	1	0	#vertex 1
-1	0	0	#vertex 2
-0.5	-1	0	#vertex 3
0.5	-1	0	#vertex 4
1	0	0	#vertex 5

1.4 Order of Vertices

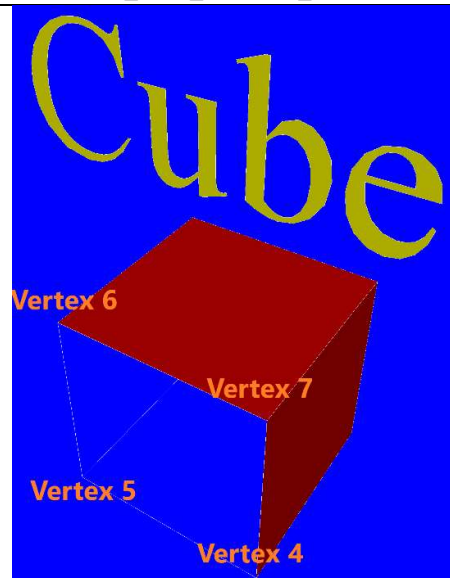| Cube.wrl	Cube_with_invisible_side.wrl
This 3D cube is defined based on 8 vertices. Each visible side of the cube is defined such that the normal vector is facing outwards so that their visible side are displayed to the viewer.	This 3D cube defines the same 8 vertices in **Cube.wrl**. However, the order of bottom square is now "4, 5, 6, 7" instead of "7, 6, 5, 4", causing the normal vector to face inwards and thus be invisible. Also, the silhouette edge mode is used to show how the cube would have looked like if the side was visible.