

Lab 1 Report

Visualization using Polygons

CZ2003 Computer Graphics and Visualization
Nanyang Technological University

By Trifena Caroline



**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

Graphic Modes

The 3 different graphic modes are experimented on a pyramid (polygons.wrl).

1) Wireframe (Fig 1)

Sets BS Contact into wireframe rendering mode. This show a skeletal grid consisting of lines connecting the vertices that make up the shape.

2) Vertices (Fig 2)

Sets BS Contact into vertices rendering mode. This only shows dots for each corner of the shape so that content authors can get an impression of the complexity of their models. Since this is a simple shape with a few corners, they are thus barely visible.

3) Flat (Fig 3)

Sets BS Contact into flat rendering mode. This disables gouraud shading and faces are thus rendered as flat polygons.

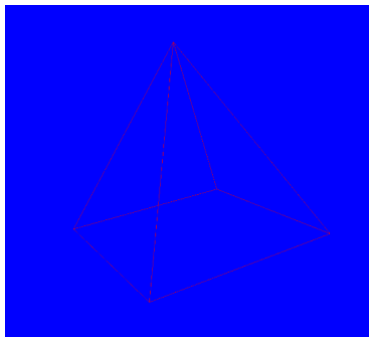


Fig 1. Wireframe



Fig 2. Vertices

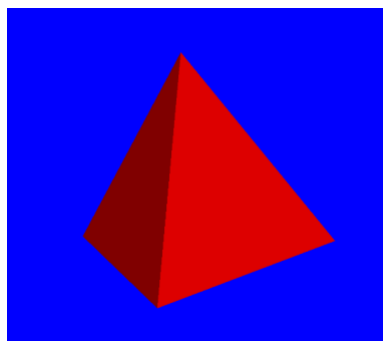


Fig 3. Flat

Colour Changes with Regards to *diffuseColor*

Changes in colour with regards to *diffuseColor* are experimented on a pyramid (polygons.wrl). *diffuseColor* consists of 3 values, ranging between 0 to 1, which relates to the percentage of red, blue and green that the shape will have (Fig 4).

```
material Material {  
  diffuseColor 1 0 0 #red=1, green=0, blue=0  
  specularColor 1 1 1 #red=1, green=1, blue=1  
  transparency 0 # try values between 0 and 1  
  shininess 1 # shiny surface, try values between 0 and 1  
}
```

Fig 4. *diffuseColor* in VRMLPad

The first value corresponds to red (Fig 5), second value to green (Fig 6) and the third value to blue (Fig 7).

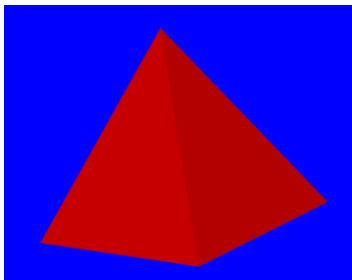


Fig 5. *diffuseColor* 1 0 0

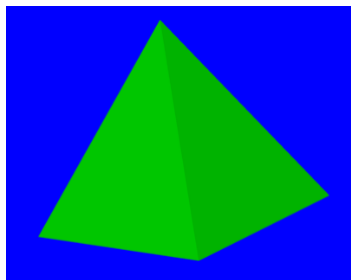


Fig 6. *diffuseColor* 0 1 0

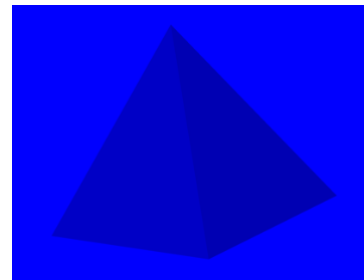


Fig 7. *diffuseColor* 0 0 1

Different combinations of the 3 RGB values produces various colour combinations. For example, *diffuseColour* 1 0 1 forms a purple shape, created by combining red and blue (Fig 9), *diffuseColour* 0 0 0 forms a black shape (Fig 8) and *diffuseColour* 1 1 1 forms a white shape (Fig 10).

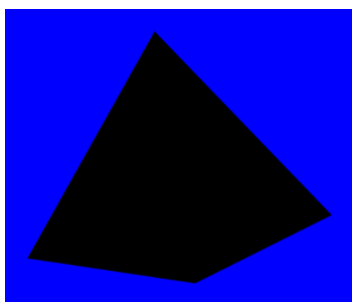


Fig 8. *diffuseColor* 0 0 0

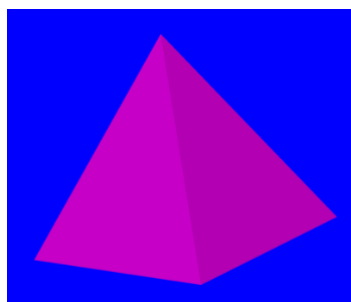


Fig 9. *diffuseColor* 1 0 1

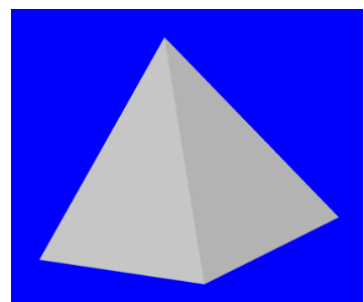


Fig 10. *diffuseColor* 1 1 1

If colour values go beyond the stated range (i.e. less than 0 or more than 1), illumination and realism of the object will be affected, such as the edges of different surfaces becoming less distinct (Fig 11).

The colour of the shape will be the result of normalization of the values (Fig 12).

Furthermore, when all values are set to be less than zero, the object simply becomes dark as there is an absence of reflected light (Fig 13).

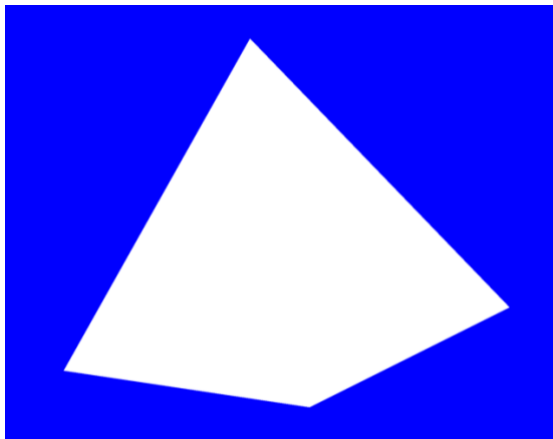


Fig 11. *diffuseColor 2 2 2*

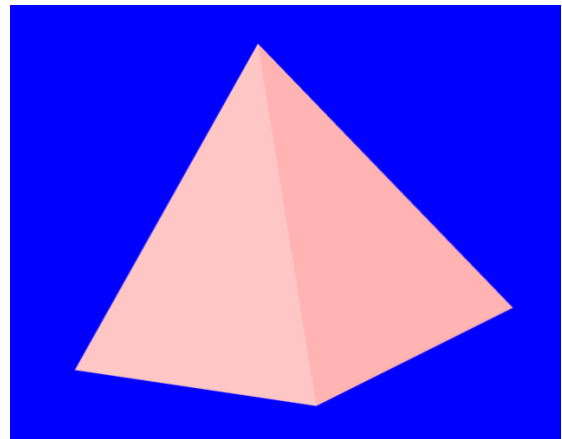


Fig 12. *diffuseColor 2 1 1*

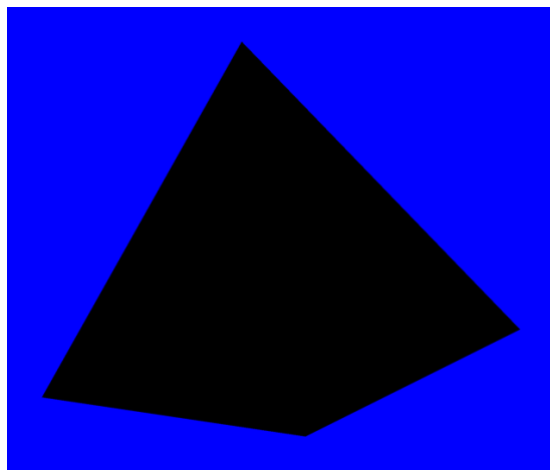


Fig 12. *diffuseColor -1 -1 -1*

2D Hexagon

Hexagon (Fig 13) is a six-sided equilateral and equiangular polygon. It is formed by first defining 6 vertices and then creating the shape by joining the vertices in the correct order.

As it is a 2D shape, the hexagon may not be visible from certain viewpoints as polygons may be hidden when viewed from the wrong angles.

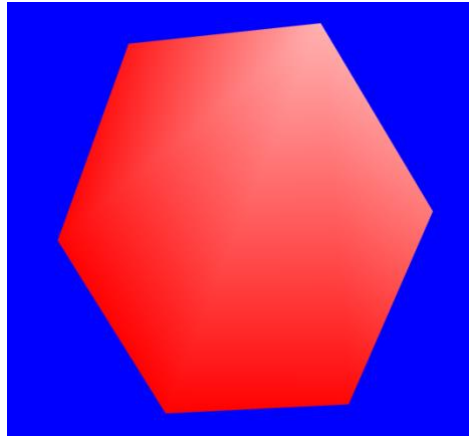


Fig 13. 2D Hexagon viewed from a correct angle

3D Cube

The cube (Fig 14) is formed by first defining 8 vertices, followed by 6 squares using the previously defined vertices. All squares must be defined such that it faces outwards in order for the viewer to be able to view it as a cube.

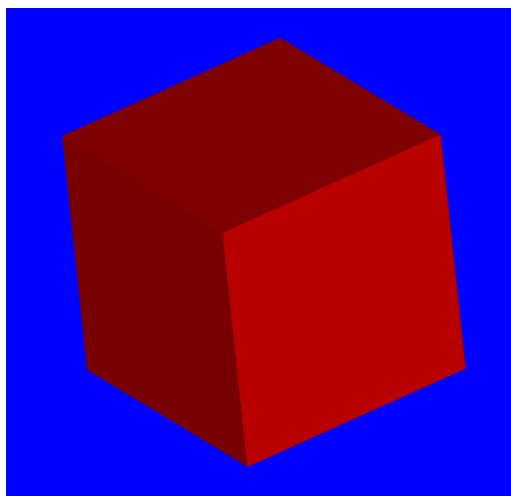


Fig 14. 3D Cube

Order of Vertices

The order of vertices plays a significant role in determining how the shape will appear to the viewer. The Right-Hand System (RHS) (Fig 15) is adopted in ensuring that all shapes face outwards for the viewer to view it as a closed shape.

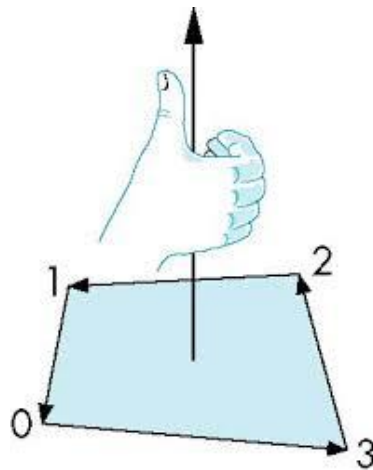


Fig 15. The arrows direct the order in which the vertices should be defined for the surface to point upwards (where the thumb points to)

Description of Files

1. polygons.wrl - Given template. Shows a pyramid shape as per Fig 5's specifications.
2. hexagon.wrl - Shows a 2D hexagon as per Fig 13.
3. cube.wrl - Shows a 3D cube as per Fig 14.