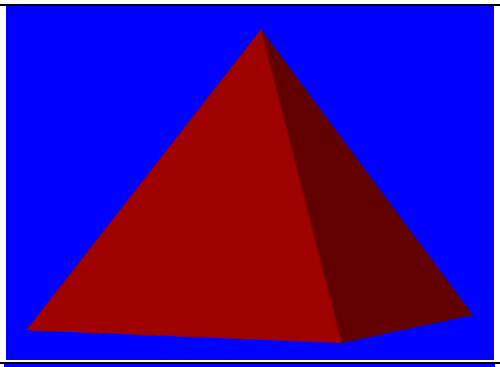
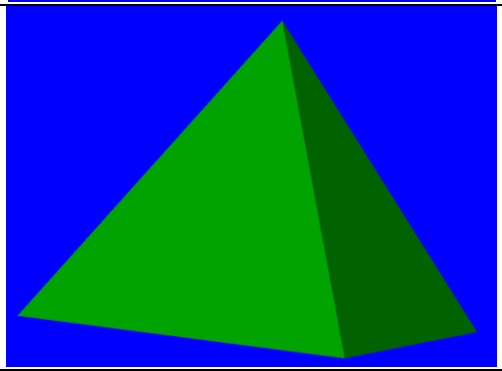
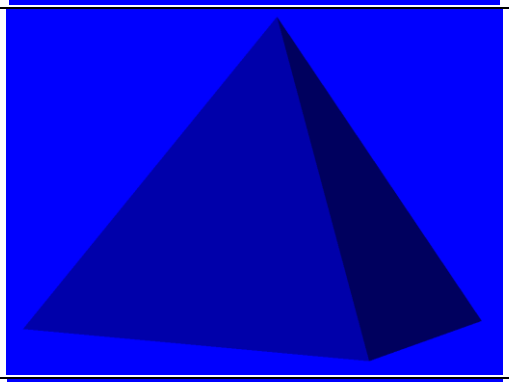
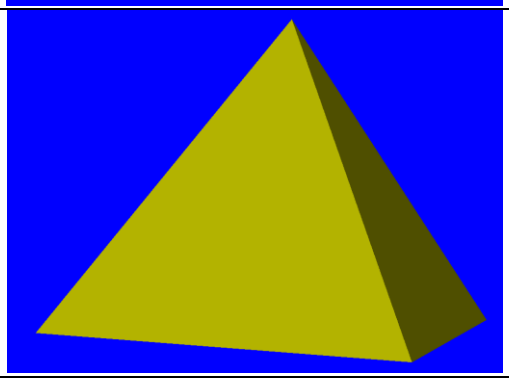
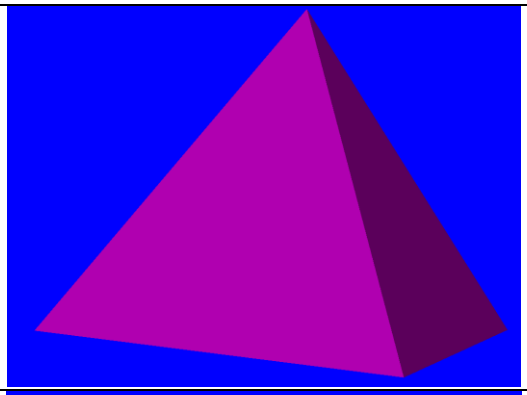
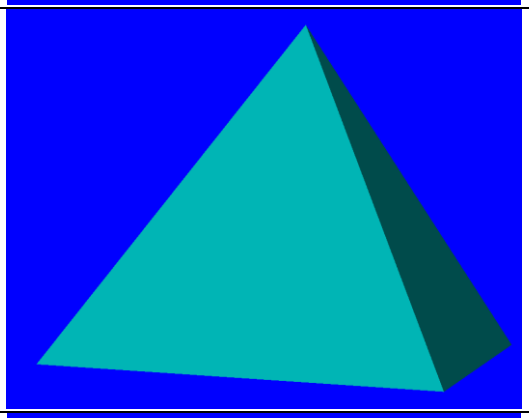
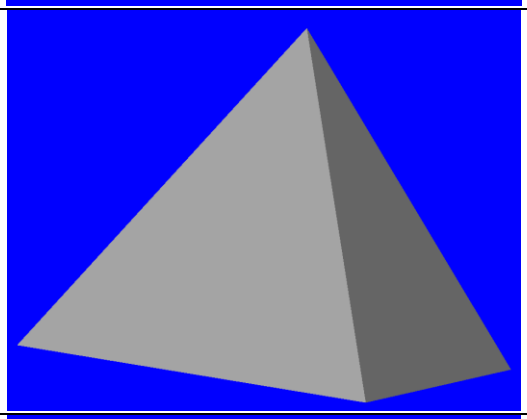
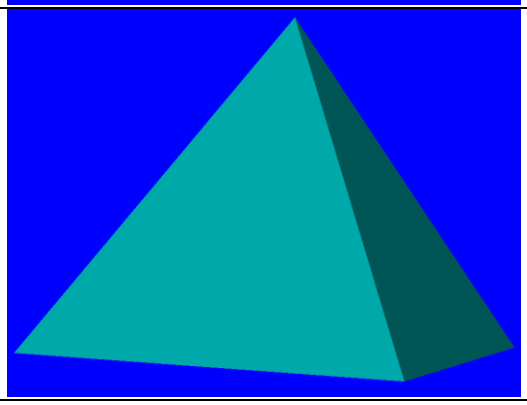
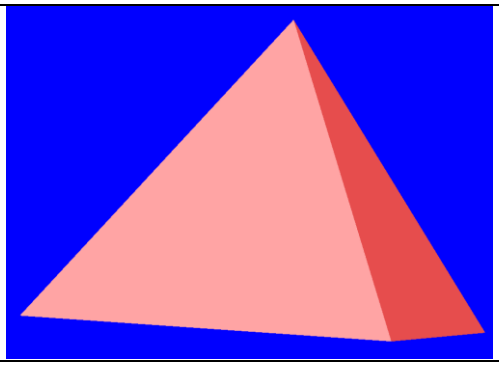
Effects of diffuseColor on color of shapes

Polygon No.	Results	Notes
Polygon 1		In polygons.wrl, diffuseColor = 1 0 0.
Polygon 2		In polygons1.wrl, diffuseColor = 0 1 0.
Polygon 3		In polygons2.wrl, diffuseColor = 0 0 1.
Polygon 4		In polygons3.wrl, diffuseColor = 1 1 0.

Polygon 5		In polygons4.wrl, diffuseColor = 1 0 1.
Polygon 6		In polygons5.wrl, diffuseColor = 0 1 1.
Polygon 7		In polygons6.wrl, diffuseColor = 1 1 1.
Polygon 8		In polygons7.wrl, diffuseColor = -1 1 1.

Polygon 9		In polygons8.wrl, diffuseColor = 3 1 1.

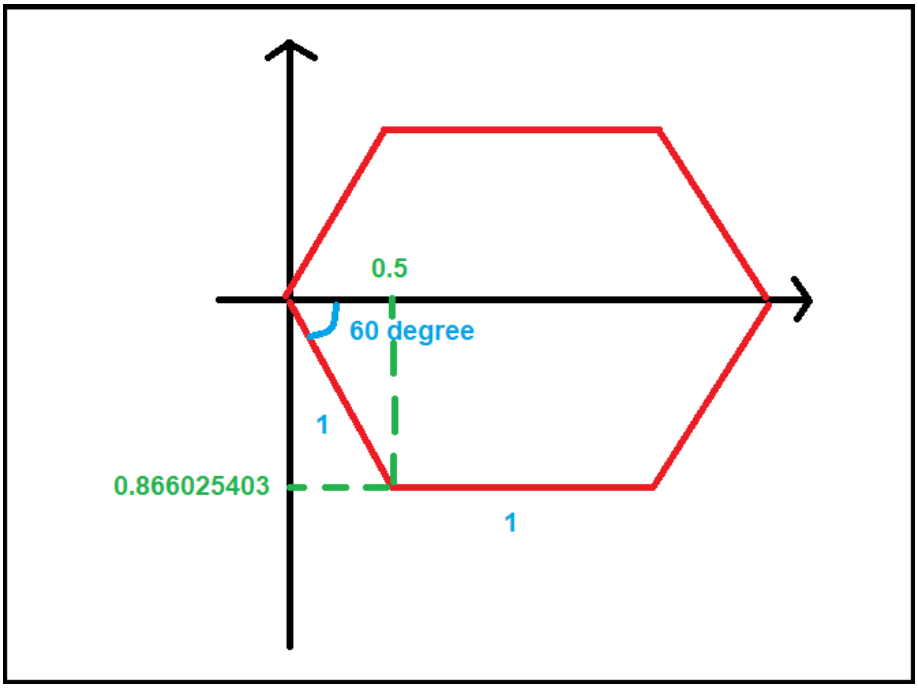
The **diffuseColor** contains three parameters which are Red Green Blue. In polygon 1, when red has a value of 1, while the rest of the colors parameters have a value of 0, only red color will be applied on the polygon.

In polygon 4, two colors, red and green, have a value of 1 while blue parameter have a value of 0. This causes red and green color to be applied to the polygon. A combination of red and green color produces yellow color as since in the result of polygon 4.

When one of the color parameter have a value of -1, it will be treated as value of 0. This can be seen between polygon 6 and polygon 8.

If a color parameter has a value more than 1, more layers of that color will be applied to the polygon. In polygon 9, red has a value of 3 while green and blue have values of 1 respectively. As a result, light red color is seen on the polygon as compared to all colors having values of 1s in polygon 7 which result in grey color.

Hexagon



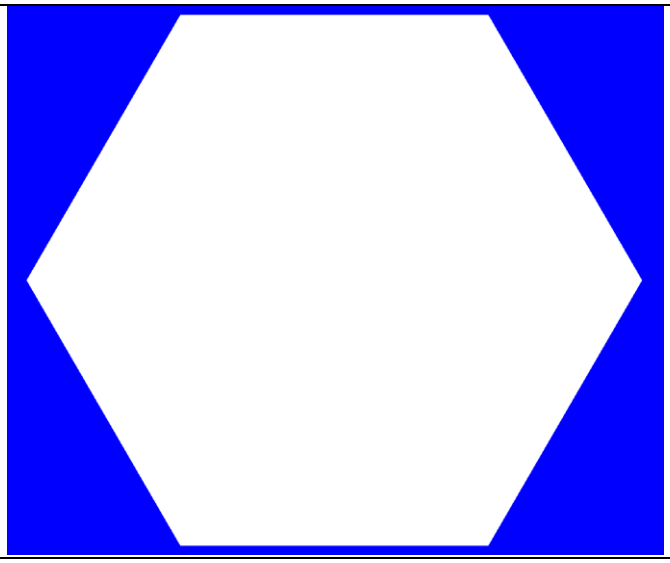
The coordinates of the hexagon was obtains with the use of SOH CAH TOA where sin θ = Opposite/Hypotenuse, cos θ = Adjacent/Hypotenuse, and tan θ = Opposite/ Adjacent respectively.

In my solution, I assigned each line drawn to have a length of 1. This value of 1 will be used as the Hypotenuse for the figure above. As hexagon have a total angle of 720degree, each corner of the hexagon will have an angle of 120degree. Next, 120degree was divided by 2 to obtain 60degree as the x-axis divided the corner of the hexagon into half. Through the use of SOH CAH, the coordinates were obtained as x coordinate is Adjacent in cos θ = Adjacent/Hypotenuse, while y coordinate is the Opposite in sin θ = Opposite/Hypotenuse.

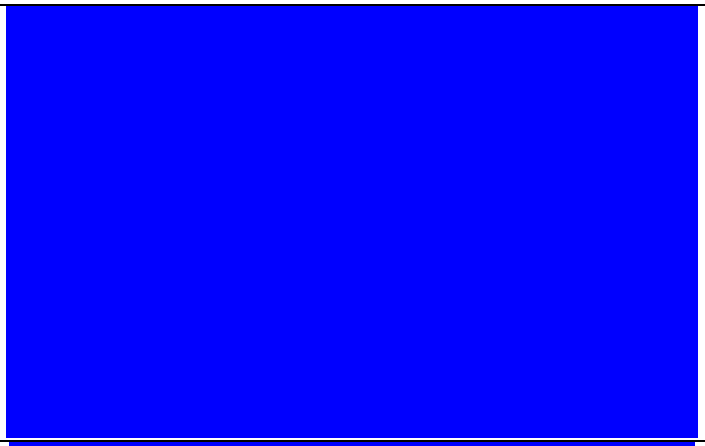
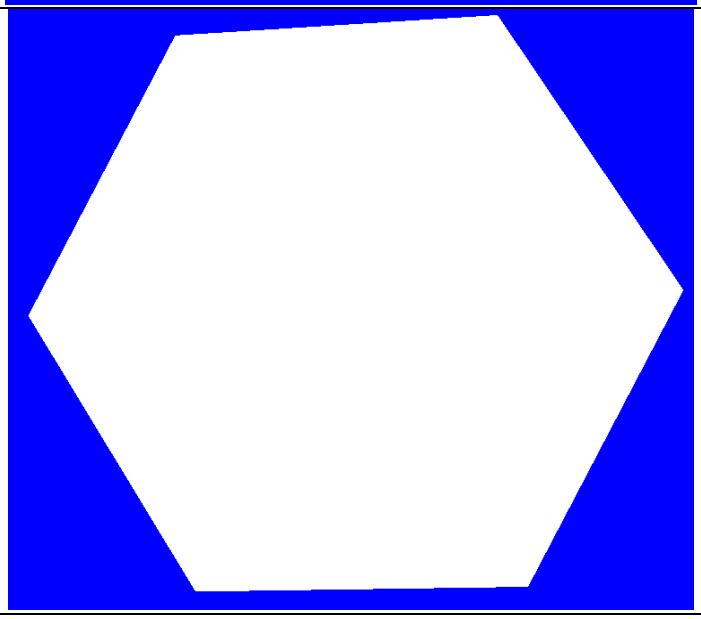
Vertices values:

- 0 = 0.0 0.0 0.0
- 1 = 0.5 -0.86602540378 0.0
- 2 = 1.5 -0.86602540378 0.0
- 3 = 2.0 0.0 0.0
- 4 = 1.5 0.86602540378 0.0
- 5 = 0.5 0.86602540378 0.0

In hexagon1.wrl, order of vertices: 0, 1, 2, 3, 4, 5, -1.

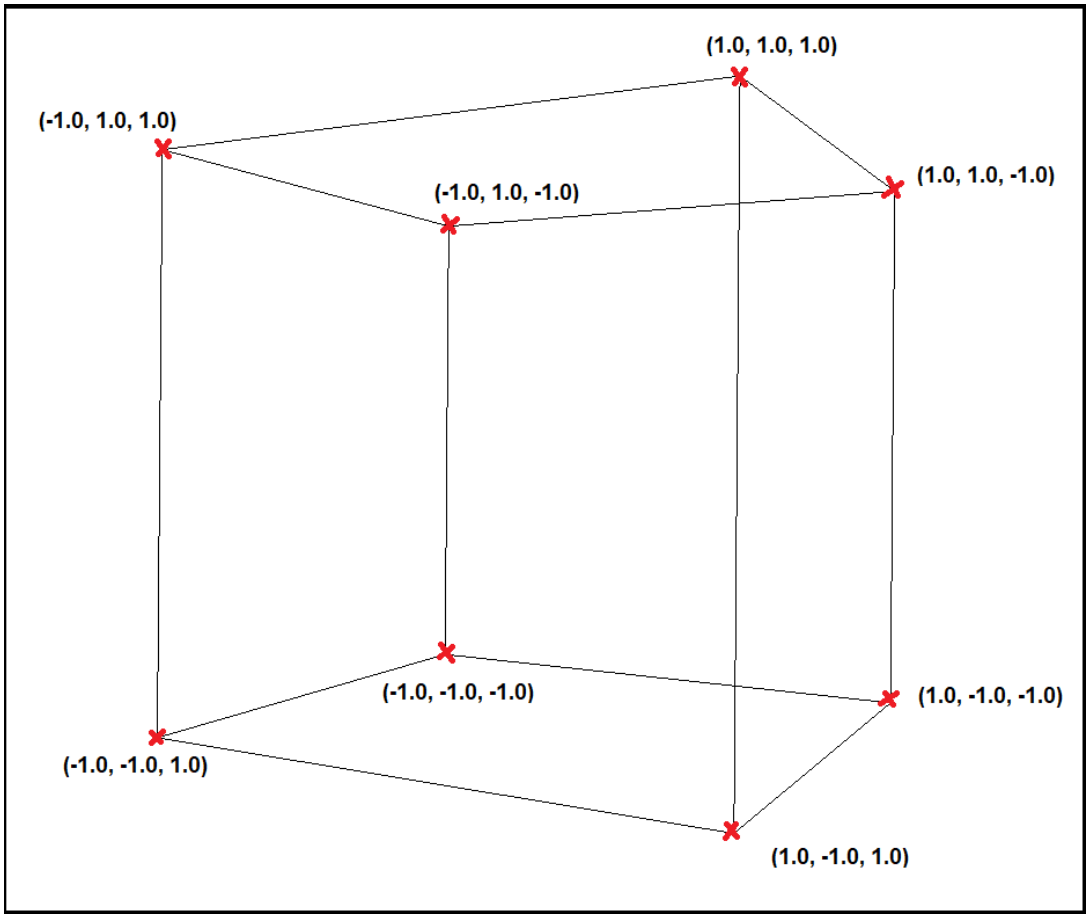
At original view position	

In hexagon2.wrl, order of vertices: 5, 4, 3, 2, 1, 0, -1.

At original view position	
After rotation of view position	

When assigning the vertices, the order of the vertices must be in counter clockwise. When the order of the vertices is reversed as shown in hexagon2.wrl, we were not able to see the hexagon until we rotated it.

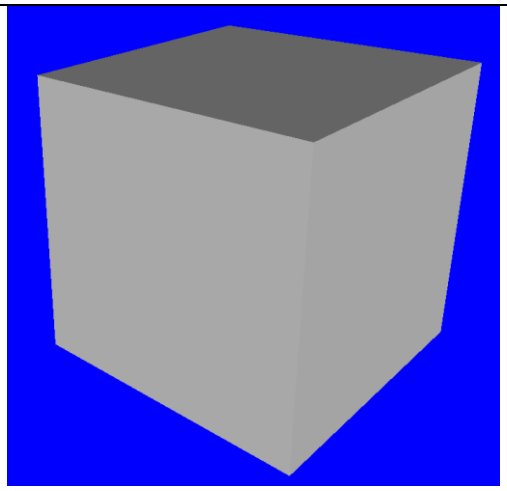
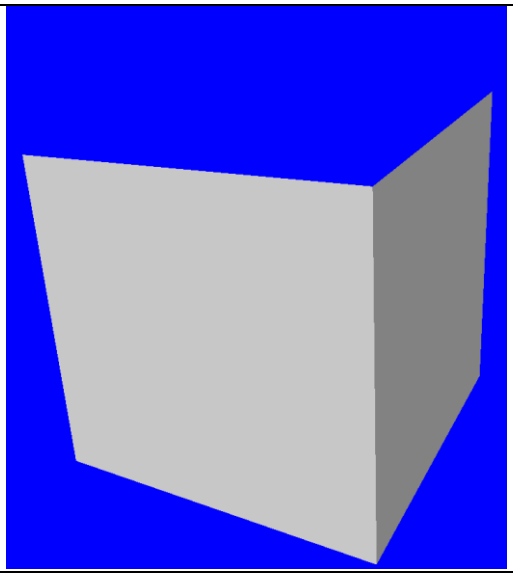
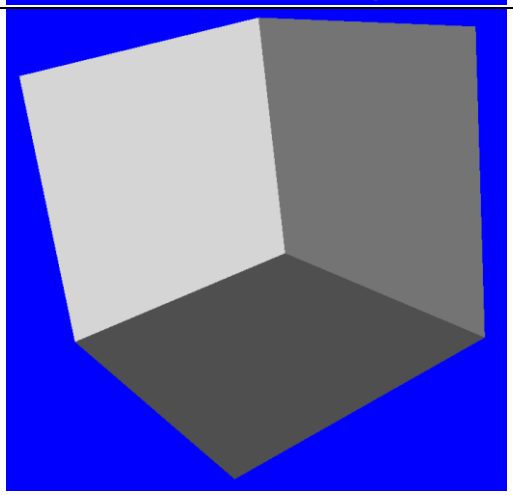
3D Cube



To obtain the shape of a 3D cube, 8 coordinates corners of the 3D cube was identified and set as vertices where between each corners of the 3D cube must be of equal length. For the coordinates I was using, the distance between each corner of the cube, also known as length, breadth, or height of the cube, is 2 unit.

Vertices values:

- 0 = -1.0 -1.0 1.0
- 1 = -1.0 -1.0 -1.0
- 2 = 1.0 -1.0 -1.0
- 3 = 1.0 -1.0 1.0
- 4 = -1.0 1.0 1.0
- 5 = 1.0 1.0 1.0
- 6 = 1.0 1.0 -1.0
- 7 = -1.0 1.0 -1.0

3D Cube No.	Results	Notes
3D Cube 1		In 3DCube1.wrl, coordIndex: #bottom square 0, 1, 2, 3, -1, #top square 4, 5, 6, 7, -1, #left side 0, 4, 7, 1, -1, #right side 3, 2, 6, 5, -1, #front 0, 3, 5, 4, -1, #back 2, 1, 7, 6, -1
3D Cube 2		In 3DCube2.wrl, coordIndex: #bottom square 0, 1, 2, 3, -1, #top square 7, 6, 5, 4, -1, #left side 0, 4, 7, 1, -1, #right side 3, 2, 6, 5, -1, #front 0, 3, 5, 4, -1, #back 2, 1, 7, 6, -1
3D Cube 3		In 3DCube3.wrl, coordIndex: #bottom square 3, 2, 1, 0, -1, #top square 7, 6, 5, 4, -1, #left side 1, 7, 4, 0, -1, #right side 5, 6, 2, 3, -1, #front 4, 5, 3, 0, -1, #back 6, 7, 1, 2, -1

3D Cube 1 shows the original cube where all coordinate index are placed in counter clockwise order. 3D Cube 2's have its top square's coordinate index placed in

clockwise order, therefore, the top surface of the cube is not visible. 3D Cube 3 have all its coordinate index placed in clockwise order. This resulted in the inner surface of the cube to be visible while the external surface of the cube not visible.