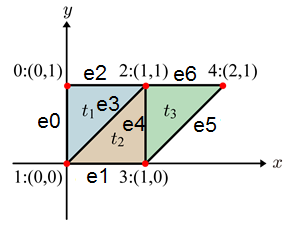
1. **Introduction**

This program is an application for drawing a sphere created using Visual Basic. The sphere is represented by many triangular polygons, in which two procedures named “GetPolyMesh” and “DrawSphere” are created to get every vertex’s coordinate, connect them with lines and draw the polygons on the screen. A function named “Normal” is also created to find the normal of each polygon to do the back-face culling on the sphere. There is also a class named “ClassSphere.vb” which is specifically created to store the data structures and some procedures that are mainly used in the program.

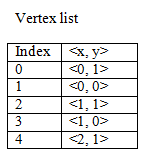
Generally, there are some data structures that can be used to represent a polygon mesh (e.g., list of vertex coordinates, list of vertex indices, list of edge indices, triangle strip). The polygon mesh in this program will be represented by a list of edge indices.

1. **Basic Theory**

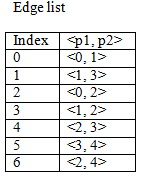
A polygon mesh is a set of connected polygonally bounded planar surfaces. Polygon meshes can also be used to represent objects with curved surfaces (e.g. Sphere). Polygon meshes are usually triangular or quadrilateral.

One of the data structures that can be used to represent polygon meshes is a list of edge indices. For example :

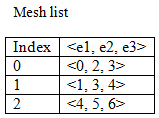
* Each vertex is stored in a list/array of vertices.



* Each edge is stored in a list/array of edges.

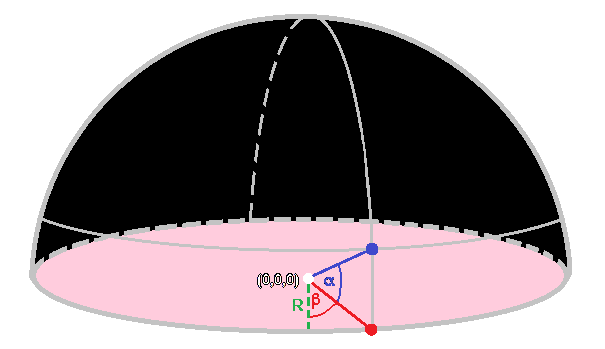


* Each polygon stores three integers indicating the indices of the edges list.



The sphere is divided vertically by a number of longitudes (M) and horizontally by a number of latitudes (N). In finding each vertex’s coordinates on the screen, the upper half part of the sphere is done first; the lower half of the sphere done next.

In the upper half part, to find each vertex’s coordinates, the vertical & horizontal angles for each vertex need to be known. The value of the vertical angle (α) is calculated by dividing 90 degrees with the number of latitudes. The value of the horizontal angle (β) is calculated by dividing 360 degrees with the number of longitudes.

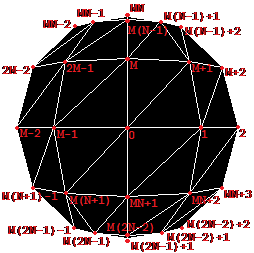


Assuming that the sphere is centered at (0, 0, 0), the formulas to find the coordinates for all vertices on the upper half part of the sphere are :

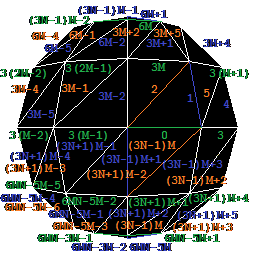
* x = R \* cos(α) \* sin(β)
* y = R \* sin(α)
* z = R \* cos(α) \* cos(β)

In the lower half part, the x and z coordinates of each vertex are the same as the ones on the upper half. The only difference is the y-coordinate of each vertex on the lower half part is the negative of the ones on the upper half.

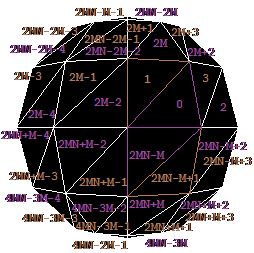
* Number of vertices on the sphere is represented by :



* Number of edges on the sphere is represented by :



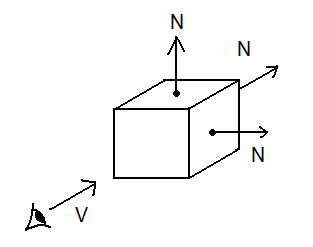
* Number of polygons on the sphere is represented by :



The polygon mesh is created by:

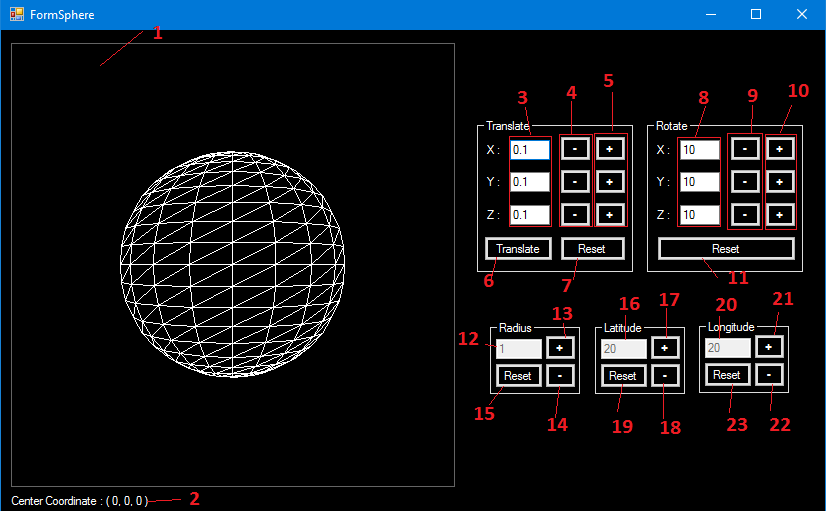
* Taking the coordinates of all vertices (X, Y, Z, W) and insert them to the array of vertices.
* Starting from the first vertex, find a pattern to get the two other correlated vertices to create a polygon mesh.
* Set three edges to connect each of the three correlated vertices consecutively.
* For each edge, insert the indices of the two vertices on each end of the edge, to the array of vertex indices.
* For each polygon, insert the indices of its three edges, to the array of edge indices.

These processes are repeated for each subsequent vertex and will stop when the last edge index is inserted to the array.

Back-face culling is a preprocessing to eliminate faces which are not visible. For every surface, if a surface’s normal is away from the viewer () then that surface is a back surface.

1. **User Manual**

The interface of this application is shown in the picture below.



These are descriptions for the interface of the application :

1. Screen, used to show the visual/graphical figure of the sphere.
2. Center coordinate, used to show the center of the sphere.
3. Translation textboxes, used to set the value of translation on each axis.
4. Negative translation button, used to translate the sphere by negative value on the translation textboxes at the corresponding axes.
5. Positive translation button, used to translate the sphere by positive value on the translation textboxes at the corresponding axes.
6. Translate button, used to translate the sphere on all axes by value on translation textboxes.
7. Reset translation button, used to reset all translation done on the sphere.
8. Rotation textboxes, used to set the value of rotation on each axis.
9. Negative rotation button, used to rotate the sphere by negative value on the rotation textboxes at the corresponding axes.
10. Positive rotation button, used to rotate the sphere by positive value on the rotation textboxes at the corresponding axes.
11. Reset rotation button, used to reset all rotation done on the sphere.
12. Radius textbox, used to show the radius of the sphere.
13. Increase radius button, used to increase the radius of the sphere.
14. Decrease radius button, used to decrease the radius of the sphere.
15. Reset radius button, used to reset the radius of the sphere to 1.
16. Latitude textbox, used to show the number of the “latitude” edges of the sphere.
17. Increase latitude button, used to increase the number of the “latitude” edges of the sphere.
18. Decrease latitude button, used to decrease the number of the “latitude” edges of the sphere.
19. Reset latitude button, used to reset the number of the “latitude” edges of the sphere to 20.
20. Longitude textbox, used to show the number of the “longitude” edges of the sphere.
21. Increase longitude button, used to increase the number of the “longitude” edges of the sphere.
22. Decrease longitude button, used decrease the number of the “longitude” edges of the sphere.
23. Reset longitude button, used to reset the number of the “longitude” edges of the sphere to 20.

Features

1. Change or set the value of the translation

To change or set the value of the translation, the user needs to fill in the “Translation” textbox of each axis with the desired value of the translation.

1. Translate the sphere

* To translate the sphere by the negative value on the “Translation” textboxes, the user needs to click the “Negative translation” button, according to which axis the user wants to be translated. The minimum center coordinates of the sphere on all axes is -2.
* To translate the sphere by the positive value on the “Translation” textboxes, the user needs to click the “Positive translation” button, according to which axis the user wants to be translated. The maximum center coordinates of the sphere on all axes is 2.
* To translate the sphere on all axes by the value on the “Translation” textboxes, the user needs to click the “Translate” button.

1. Reset all the translations done on the sphere

To reset all the translations done on the sphere, the user needs to click the “Reset translation” button.

1. Change or set the value of the rotation

To change or set the value of the rotation, the user needs to fill in the “Rotation” textbox of each axis with the desired value of the rotation.

1. Rotate the sphere

* To rotate the sphere by the negative value on the “Rotation” textboxes, the user needs to click the “Negative rotation” button, according to which axis the user wants to be rotated.
* To rotate the sphere by the positive value on the “Rotation” textboxes, the user needs to click the “Positive rotation” button, according to which axis the user wants to be rotated.

1. Reset all the rotations done on the sphere

To reset all the rotations done on the sphere, the user needs to click the “Reset rotation” button.

1. Increase/Decrease the radius of the sphere

* To increase the radius of the sphere, the user needs to click the “Increase radius” button. The maximum radius is 1.8.
* To decrease the radius of the sphere, the user needs to click the “Decrease radius” button. The minimum radius is 0.2.

1. Reset the radius of the sphere to default

To reset the radius of the sphere to default, the user needs to click the “Reset radius” button. The default radius is 1.

1. Increase/Decrease the number of the “latitude” edges of the sphere

* To increase the number of the “latitude” edges of the sphere, the user needs to click the “Increase latitude” button.
* To decrease the number of the “latitude” edges of the sphere, the user needs to click the “Decrease latitude” button. The minimum number of the “latitude” edges is 2.

1. Reset the number of the “latitude” edges of the sphere to default

To reset the number of the “latitude” edges of the sphere to default, the user needs to click the “Reset latitude” button. The default number of the “latitude” edges of the sphere is 20.

1. Increase/Decrease the number of the “longitude” edges of the sphere

* To increase the number of the “longitude” edges of the sphere, the user needs to click the “Increase longitude” button.
* To decrease the number of the “longitude” edges of the sphere, the user needs to click the “Decrease longitude” button. The minimum number of the “longitude” edges is 2.

1. Reset the number of the “longitude” edges of the sphere to default

To reset the number of the “longitude” edges of the sphere to default, the user needs to click the “Reset longitude” button. The default number of the “longitude” edges of the sphere is 20.

1. **Data structure and global variables**

Classes of the sphere used in this program

The sphere in this program is represented by a triangular polygon mesh. The triangular polygon mesh is represented by a list of edge indices. This program use Class for the data structure. There are 6 classes used to represent the sphere, which are TPoint, TArrPoint, TLine, TArrLine, TPoly and TArrPoly.

1. TPoint class can be defined as follows :

**Class TPoint**

X : double

Y : double

Z : double

W : double

Class TPoint is used to store the coordinates of a vertex of the sphere.

Properties : - X, used to store the x-coordinate of the vertex.

- Y, used to store the y-coordinate of the vertex.

- Z, used to store the z-coordinate of the vertex.

- W, used to store the w-coordinate of the vertex.

1. TArrPoint class can be defined as follows :

**Class TArrPoint**

N : integer

Elmt() : TPoint

Class TArrPoint is used to store all vertices of the sphere.

Properties : - N, used to store the current size of the set of the vertices.

- Elmt, an array of TPoint, used to store the set of the vertices.

1. TLine class can be defined as follows :

**Class TLine**

P1 : integer

P2 : integer

Class TLine is used to store the vertex indices of an edge of the sphere.

Properties : - P1, used to store the index of the first vertex of the edge.

- P2, used to store the index of the second vertex of the edge.

1. TArrLine class can be defined as follows :

**Class TArrLine**

N : integer

Elmt() : TLine

Class TArrLine is used to store the vertex indices of all edges of the sphere.

Properties : - N, used to store the current size of the set of vertex indices of the edges.

- Elmt, an array of TLine, used to store the set of vertex indices of the edges.

1. TPoly class can be defined as follows :

**Class TPoly**

E1 : integer

E2 : integer

E3 : integer

Class TPoly is used to store the edge indices of a polygon of the sphere.

Properties : - E1, used to store the index of the first edge of the polygon.

- E2, used to store the index of the second edge of the polygon.

- E3, used to store the index of the third edge of the polygon.

1. TArrPoly class can be defined as follows :

**Class TArrPoly**

N : integer

Elmt() : TPoly

Class TArrPoly is used to store the edge indices of all polygons of the sphere.

Properties : - N, used to store the current size of the set of edge indices the polygons.

- Elmt, an array of TPoly, used to store the set of edge indices of the polygons.

Additional classes used in this program

1. TRtt class can be defined as follows :

**Class TRtt**

Axis : integer

Deg : double

Class TRtt is used to store the information of a rotation.

Properties : - Axis, used to store the axis of a rotation.

- Deg, used to store the degree of a rotation.

1. TArrRtt class can be defined as follows :

**Class TArrRtt**

N : integer

Elmt() : TRtt

Class TArrRtt is used to store information of all rotations that have been done on the sphere.

Properties : - N, used to store the current size of the set of the rotations.

- Elmt, an array of TRtt, used to store the set of the rotations.

The camera and transformation matrices used in this program

The view direction of the camera is “towards z-negative”. There are 7 transformation matrices used in the program, which are Translation, Rotation-X, Rotation-Y, Rotation-Z, Perspective, View-Transformation (Vt) and Screen-Transformation (St).

1. Translation matrix : 2. Rotation-X matrix :
2. Rotation-Y matrix : 4. Rotation-Z matrix :
3. Vt matrix : 6. St matrix :

7. Perspective matrix :

Global variables used in this program

* Lat : integer

Used to store the number of the “latitude” edges of the sphere.

* Lon : integer

Used to store the number of the “longitude” edges of the sphere.

* URad : integer

Used to know whether the radius of the sphere will be increased or decreased.

* ULat : integer

Used to know whether the number of the “latitude” edges of the sphere will be increased or decreased.

* ULon : integer

Used to know whether the number of the “longitude” edges of the sphere will be increased or decreased.

* Rot : integer

Used to know which axis that will be rotated next and whether it will be rotated clockwise or counter clockwise.

* Trans : integer

Used to know which axis that will be translated next and whether it will be translated with positive or negative value.

* disx : integer

Used to store the recent translation value that has been done on the x-axis.

* disy : integer

Used to store the recent translation value that has been done on the y-axis.

* disz : integer

Used to store the recent translation value that has been done on the z-axis.

* Tr(3,3) : double

Used to store the value on the translation matrix.

* APoint : TArrPoint

Used to store all vertices of the sphere.

* Vs : TArrPoint

Used to store all vertex coordinates of the sphere after they have been multiplied with Perspective, Vt and St matrix sequentially.

* ALine : TArrLine

Used to store the vertex indices of all edges of the sphere.

* APoly : TArrPoly

Used to store the edge indices of all polygons of the sphere.

* ARtt : TArrRtt

Used to store information of all rotations that have been done on the sphere.

1. **How the features work**
2. Translate the sphere

**Definitions :**

Trans, posneg, dis : integer

disx, disy, disz : double

Tr(3,3) : double

APoint : TArrPoint

**Algorithm :**

If Trans > 0 Then

posneg = 1

Else 'Trans <= 0

posneg = -1

End If

If Trans = 1 Or Trans = -1 Or Trans = 4 Then

dis = Val(TxtTx.Text) \* posneg

If Math.Abs(disx + dis) <= 20 Then

disx = disx + dis

End If

End If

If Trans = 2 Or Trans = -2 Or Trans = 4 Then

dis = Val(TxtTy.Text) \* posneg

If Math.Abs(disy + dis) <= 2 Then

disy = disy + dis

End If

End If

If Trans = 3 Or Trans = -3 Or Trans = 4 Then

dis = Val(TxtTz.Text) \* posneg

If Math.Abs(disz + dis) <= 2 Then

disz = disz + dis

End If

End If

InitMatrix(Tr)

Tr(3, 0) = disx

Tr(3, 1) = disy

Tr(3, 2) = disz

For i = 0 To APoint.N - 1

APoint.Elmt(i) = MultiplyMatrix(APoint.Elmt(i), Tr)

Next

When the user clicks the “Positive translation” button, the value of Trans will be assigned to 1 for x-axis, 2 for y-axis and 3 for z-axis.

When the user clicks the “Negative translation” button, the value of Trans will be assigned to -1 for x-axis, -2 for y-axis and -3 for z-axis.

When the user clicks the “Translate” button, the value of Trans will be assigned to 4.

disx will store the recent translation value that has been done on the x-axis.

disy will store the recent translation value that has been done on the y-axis.

disz will store the recent translation value that has been done on the z-axis.

The translation value will depend on the value on the “Translation” textboxes, which will be added to disx/disy/disz according to the value of Trans.

Tr is a 4 x 4 translation matrix.

The procedure InitMatrix(Tr) is used to initialize Tr to an identity matrix.

Then the value of Tr(3,0) will be assigned with value of disx, the value of Tr(3,1) will be assigned with value of disy and the value of Tr(3,2) will be assigned with value of disz.

APoint is an array of point that store all vertices’ coordinates on the sphere.

The procedure MultiplyMatrix(A,B) is used to multiply matrix A (1x4) with matrix B (4x4).

All the coordinates of all vertices on the sphere will then be multiplied with matrix Tr and be translated accordingly.

1. Rotate the sphere

**Definitions :**

Rot, posneg, Ɵ : integer

Rt(3,3) : double

APoint : TArrPoint

Rtt : TRtt

ARtt : TArrRtt

**Algorithm :**

If Rot > 0 Then

posneg = 1

Else 'Rot <= 0

posneg = -1

End If

If Rot = 1 Or Rot = -1 Then

Ɵ = Val(TxtRx.Text) \* posneg

ElseIf Rot = 2 Or Rot = -2 Then

Ɵ = Val(TxtRy.Text) \* posneg

ElseIf Rot = 3 Or Rot = -3 Then

Ɵ = Val(TxtRz.Text) \* posneg

End If

If Rot <> 0 Then

Rtt.SetRot(Abs(Rot), Ɵ)

ARtt.InsLast(Rtt)

End If

If ARtt.N >= 0 Then

For i = 0 To APoint.N - 1

For j = 0 To ARtt.N - 1

InitMatrix(Rt)

If ARtt.Elmt(j).Axis = 1 Then

SetColMatrix(Rt, 1, 0, Cos(Ɵ), -Sin(Ɵ), 0)

SetColMatrix(Rt, 2, 0, Sin(Ɵ), Cos(Ɵ), 0)

ElseIf ARtt.Elmt(j).Axis = 2 Then

When the user clicks the “Positive rotation” button, the value of Rot will be assigned to 1 for x-axis, 2 for y-axis and 3 for z-axis.

SetColMatrix(Rt, 0, Cos(Ɵ), 0, Sin(Ɵ), 0)

SetColMatrix(Rt, 2, -Sin(Ɵ), 0, Cos(Ɵ), 0)

ElseIf ARtt.Elmt(j).Axis = 3 Then

SetColMatrix(Rt, 0, Cos(Ɵ), -Sin(Ɵ), 0, 0)

SetColMatrix(Rt, 1, Sin(Ɵ), Cos(Ɵ), 0, 0)

End If

APoint.Elmt(i) = MultiplyMatrix(APoint.Elmt(i), Rt)

Next

Next

End If

When the user clicks the “Negative rotation” button, the value of Rot will be assigned to -1 for x-axis, -2 for y-axis and -3 for z-axis.

Ɵ will store the rotation degree depending on the value on the “Rotation” textboxes.

Rtt will save the absolute value of Rot and the value of Ɵ.

ARtt will save all rotations that have been done on the sphere.

Rt is a 4 x 4 rotation matrix.

The procedure SetColMatrix(Rt,Col,0,1,2,3) is used to assign the value of Rt matrix on row 0/1/2/3 on column Col.

The value of Rt matrix will be set depending on the rotation matrix for each corresponding axis, according to the value on each element of ARtt.

All the coordinates of all vertices on the sphere will then be multiplied with matrix Rt and be rotated accordingly.

1. Perform a back-face culling on the sphere

**Definitions :**

Pt1, Pt2 : TPoint

Vs : TArrPoint

L : TLine

APoly : TArrPoly

**Algorithm :**

For i = 0 To APoly.N - 1

If Normal(i) < 0 Then

For j = 0 To 2

If j = 0 Then

L = ALine.Elmt(APoly.Elmt(i).E1)

ElseIf j = 1 Then

L = ALine.Elmt(APoly.Elmt(i).E2)

Else 'j = 2

L = ALine.Elmt(APoly.Elmt(i).E3)

End If

Pt1 = Vs.Elmt(L.P1)

Pt2 = Vs.Elmt(L.P2)

Drawline(Pt1.X, Pt1.Y, Pt2.X, Pt2.Y, Color.White)

Next

End If

Next

APoly is an array that store the edge indices of all polygons of the sphere.

The function Normal(i) will return the result of multiplication between the normal of polygon APoly.Elmt(i) and the view direction.

Vs is an array of point that is used to store all vertex coordinates of the sphere after they have been multiplied with Perspective, Vt and St matrix sequentially.

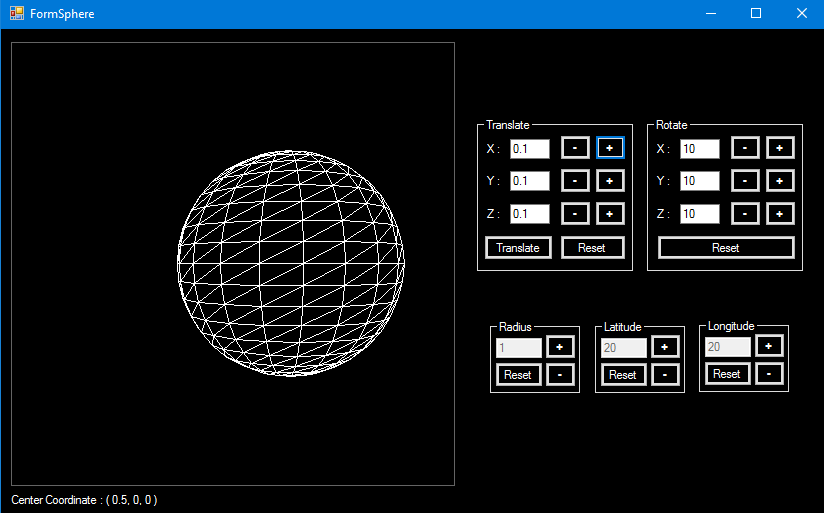
The procedure Drawline(Pt1.X, Pt1.Y, Pt2.X, Pt2.Y, Color.White) is used to connect 2 points with a line and draw it on the screen with white as the color.

For every polygon, if Normal(i) < 0 then for every edge of the polygon, draw the edge on the screen.

1. **Evaluation of the Main Features**
2. Translate the sphere on positive x-axis

****

For this case, the center coordinate of the sphere on x-axis is 0 and then increased by 0.5.

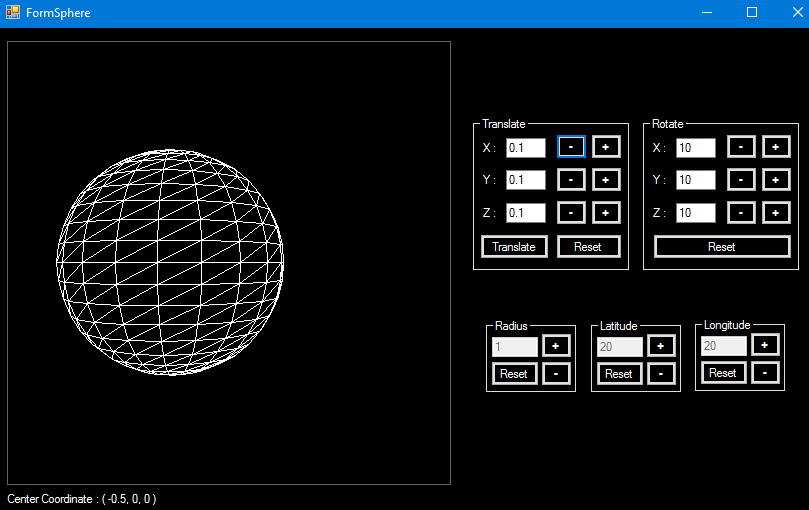
****

The center coordinate of the sphere on x-axis has been increased; now the coordinate is 0.5.

1. Translate the sphere on negative x-axis



For this case, the center coordinate of the sphere on x-axis is 0 and then decreased by 0.5.

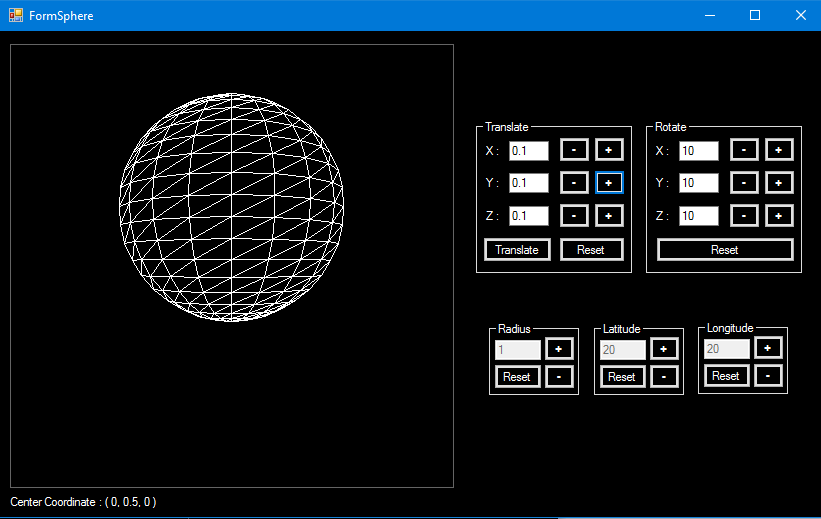


The center coordinate of the sphere on x-axis has been decreased; now the coordinate is -0.5.

1. Translate the sphere on positive y-axis



For this case, the center coordinate of the sphere on y-axis is 0 and then increased by 0.5.

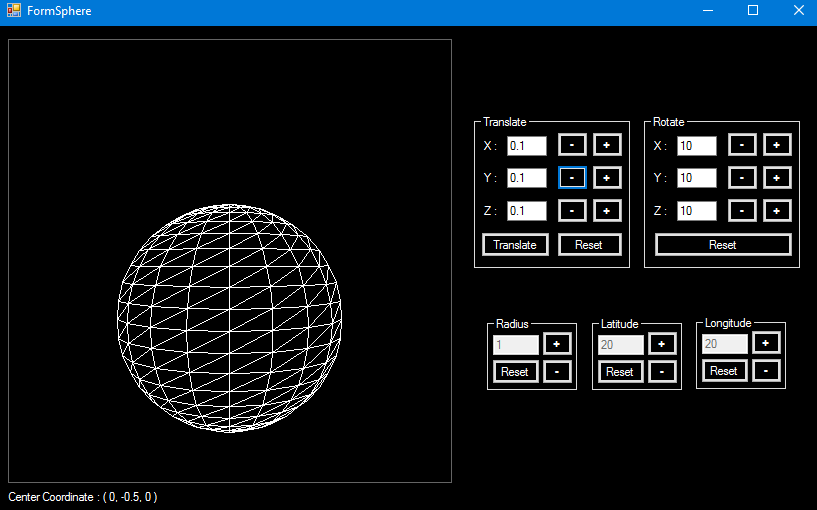


The center coordinate of the sphere on y-axis has been increased; now the coordinate is 0.5.

1. Translate the sphere on negative y-axis



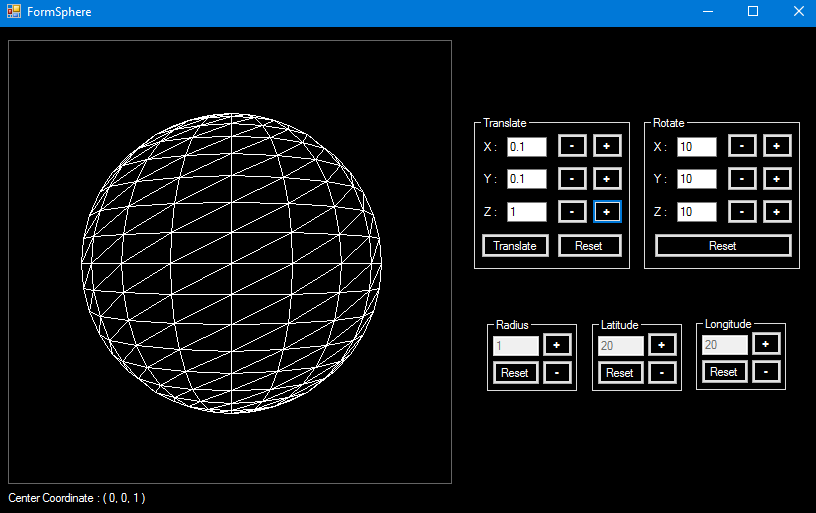
For this case, the center coordinate of the sphere on y-axis is 0 and then decreased by 0.5.



The center coordinate of the sphere on y-axis has been decreased; now the coordinate is -0.5.

1. Translate the sphere on positive z-axis 

For this case, the center coordinate of the sphere on z-axis is 0 and then increased by 1.

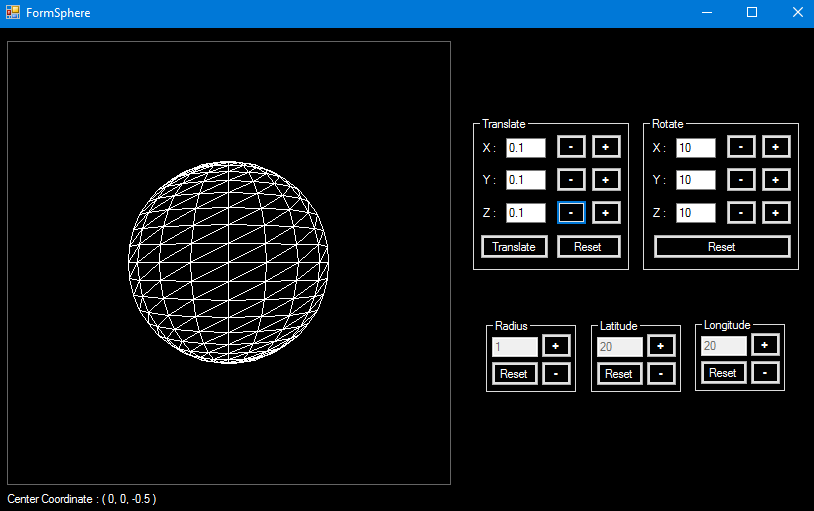


The center coordinate of the sphere on z-axis has been increased; now the coordinate is 1.

1. Translate the sphere on negative z-axis



For this case, the center coordinate of the sphere on z-axis is 0 and then decreased by 0.5.

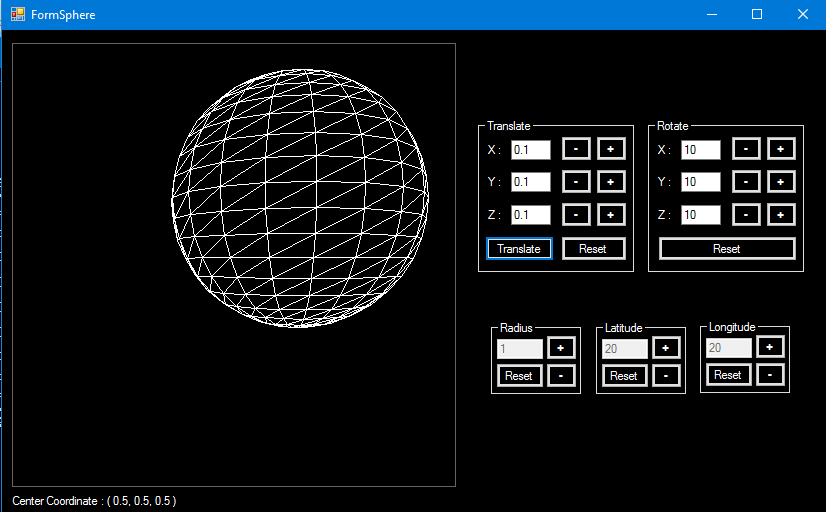


The center coordinate of the sphere on z-axis has been decreased; now the coordinate is -0.5.

1. Translate the sphere on all axes

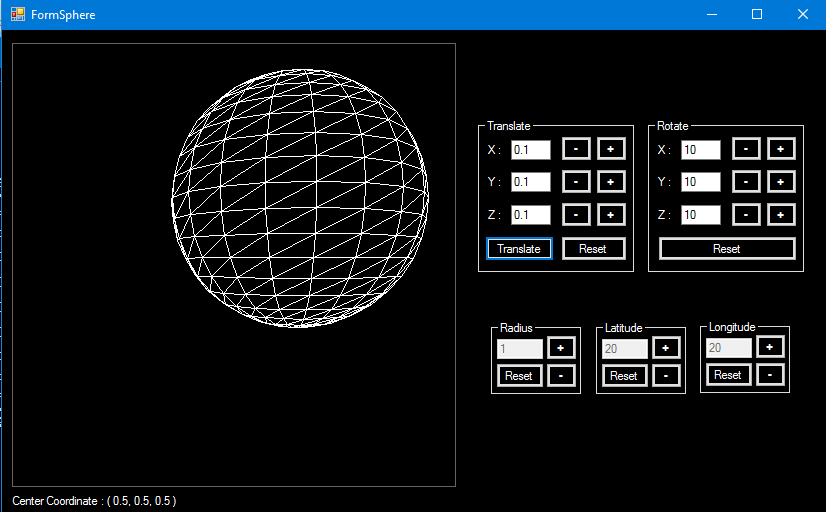


For this case, the center coordinate of the sphere on all axes are 0 and increased by 0.5.



The center coordinate of the sphere on all axes have been increased; now the center coordinate is (0.5, 0.5, 0.5).

1. Reset the translation on the sphere



For this case, the center coordinate of the sphere on all axes are 0.5.

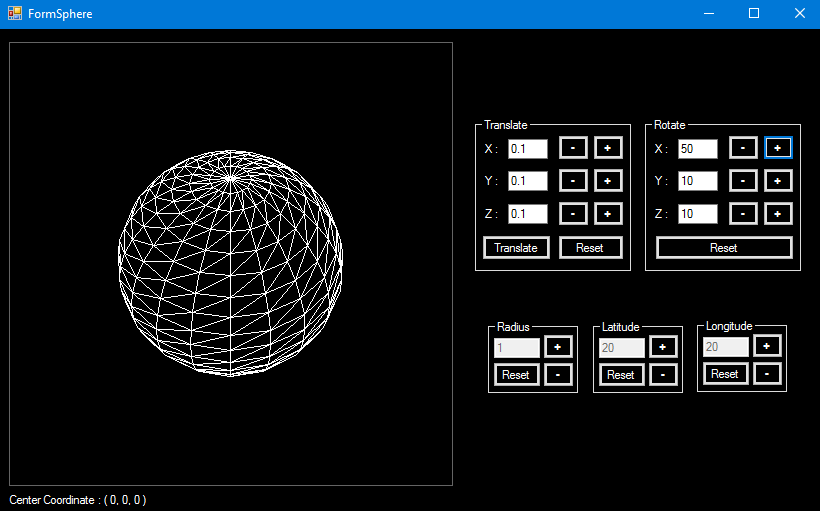


All translations done on the sphere have been reset; now the center coordinate of the sphere is (0, 0, 0).

1. Rotate the sphere on x-axis (Clockwise)



For this case, the sphere still has not rotated yet and then it is rotated clockwise on x-axis by 50 degrees.

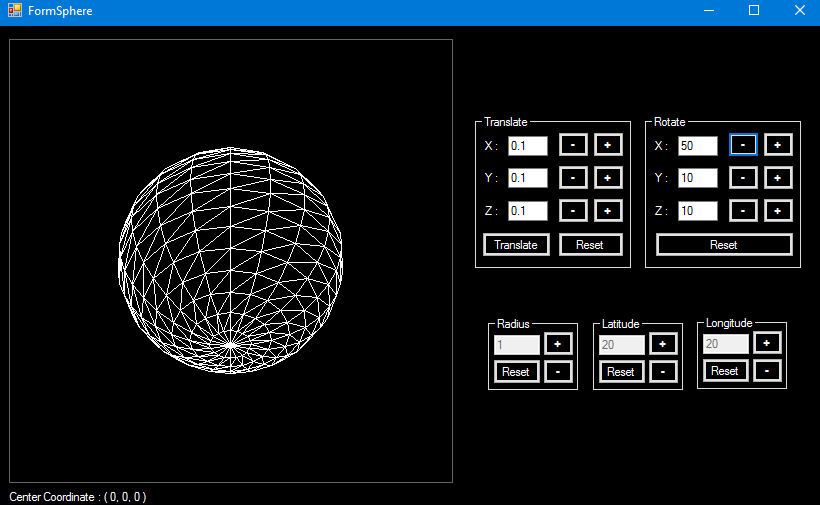


The sphere has been rotated clockwise on the x-axis by 50 degrees.

1. Rotate the sphere on x-axis (Counter Clockwise)

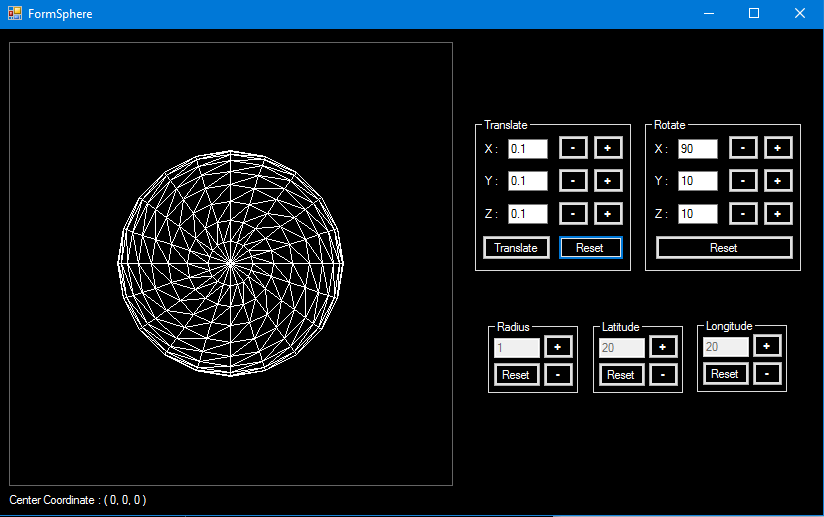


For this case, the sphere still has not rotated yet and then it is rotated counter clockwise on x-axis by 50 degrees.

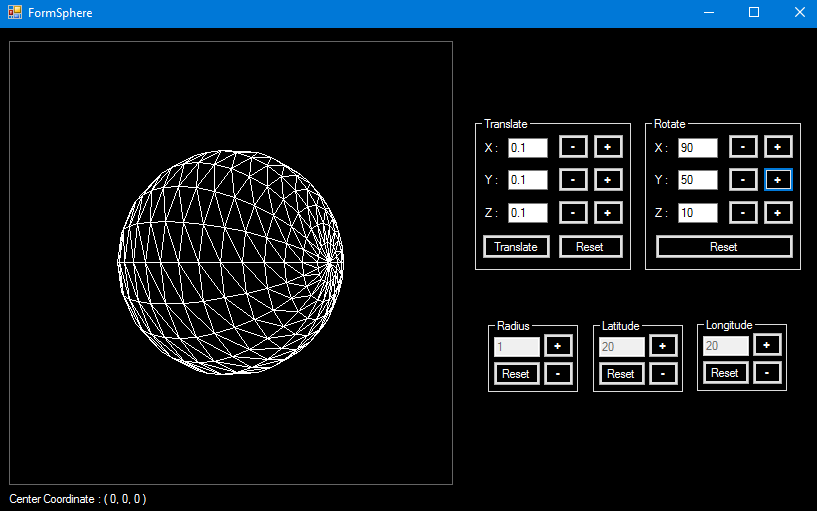


The sphere has been rotated counter clockwise on the x-axis by 50 degrees.

1. Rotate the sphere on y-axis (Clockwise)

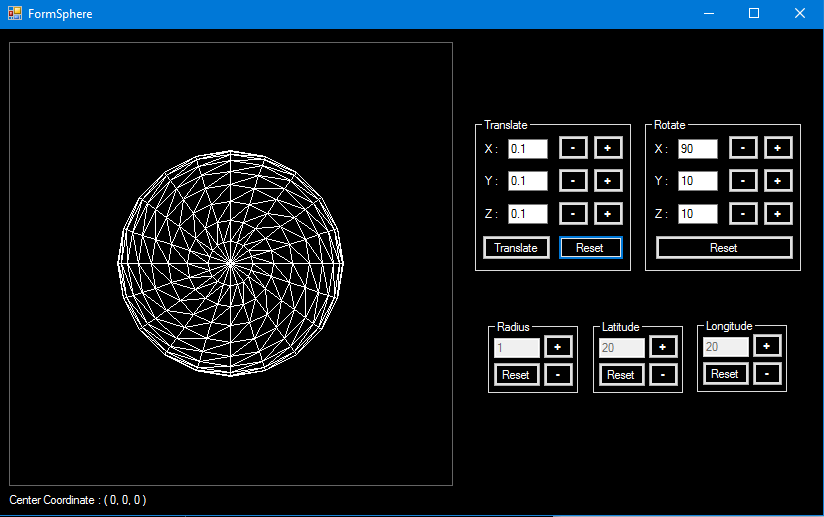


For this case, the sphere has been rotated clockwise on x-axis by 90 degree and then it is rotated clockwise on y-axis by 50 degrees.

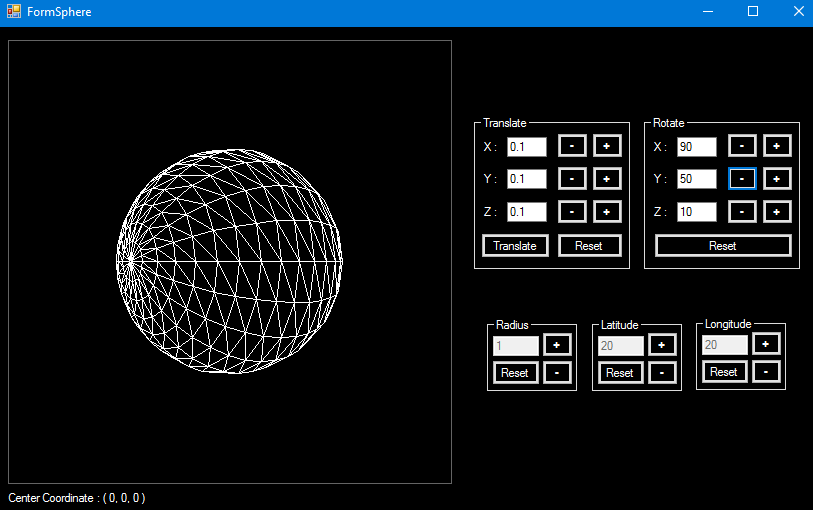


The sphere has been rotated clockwise on the y-axis by 50 degrees.

1. Rotate the sphere on y-axis (Counter Clockwise)



For this case, the sphere has been rotated clockwise on x-axis by 90 degrees and then it is rotated counter clockwise on y-axis by 50 degrees.

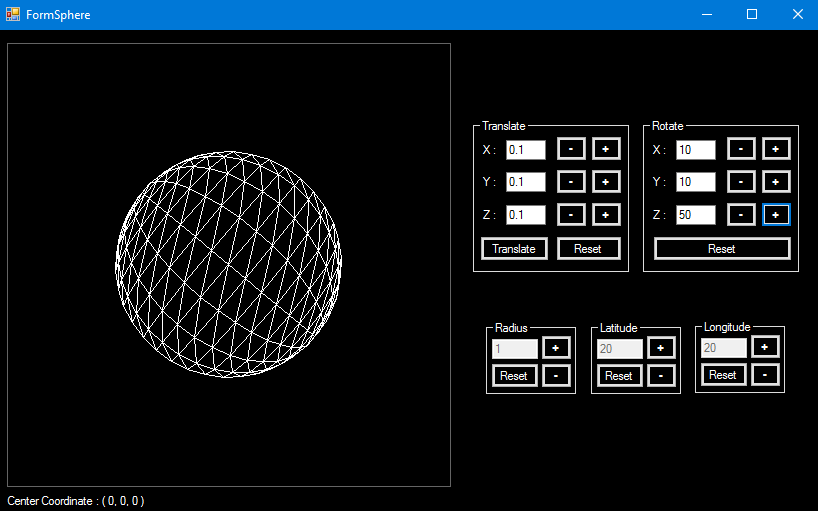


The sphere has been rotated counter clockwise on the y-axis by 50 degrees.

1. Rotate the sphere on z-axis (Clockwise)



For this case, the sphere still has not rotated yet and then it is rotated clockwise on z-axis by 50 degrees.

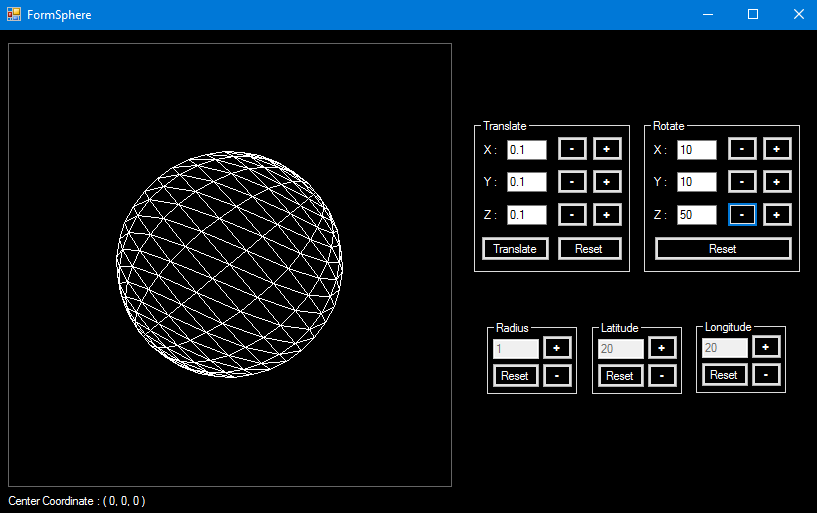


The sphere has been rotated clockwise on the z-axis by 50 degrees.

1. Rotate the sphere on z-axis (Counter Clockwise)

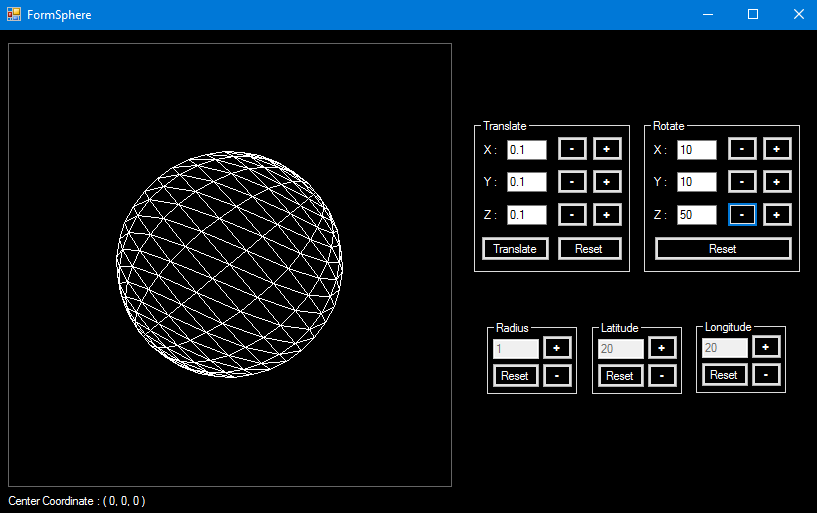


For this case, the sphere still has not rotated yet and then it is rotated counter clockwise on z-axis by 50 degrees.



The sphere has been rotated counter clockwise on the z-axis by 50 degrees.

1. Reset the rotation on the sphere



For this case, the sphere has been rotated counter clockwise on the z-axis by 50 degrees.

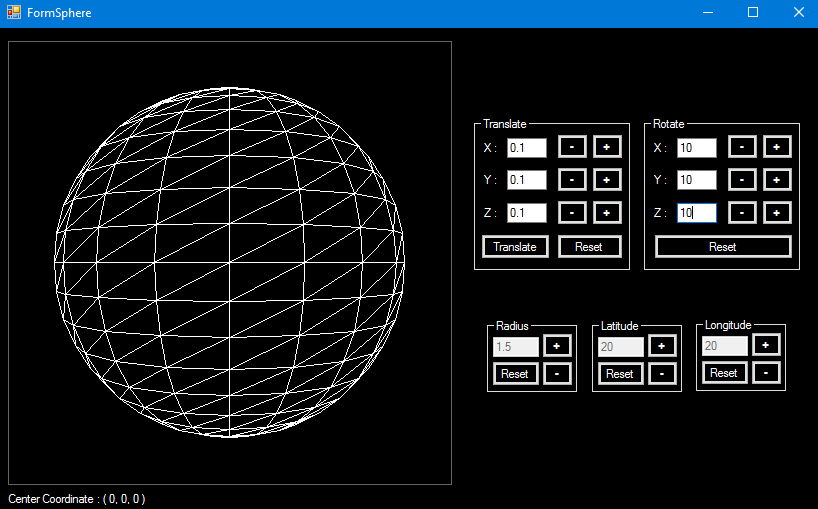


The rotation position on the sphere has been reset to default.

1. Increase radius of the sphere



For this case, the radius of the sphere is 1 and then increased by 0.5.

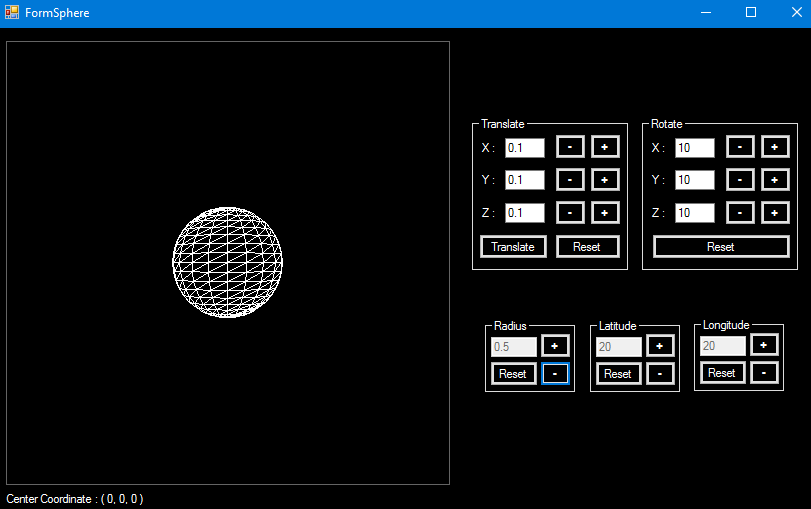


The radius of the sphere has been increased by 0.5; now the radius is 1.5.

1. Decrease radius of the sphere

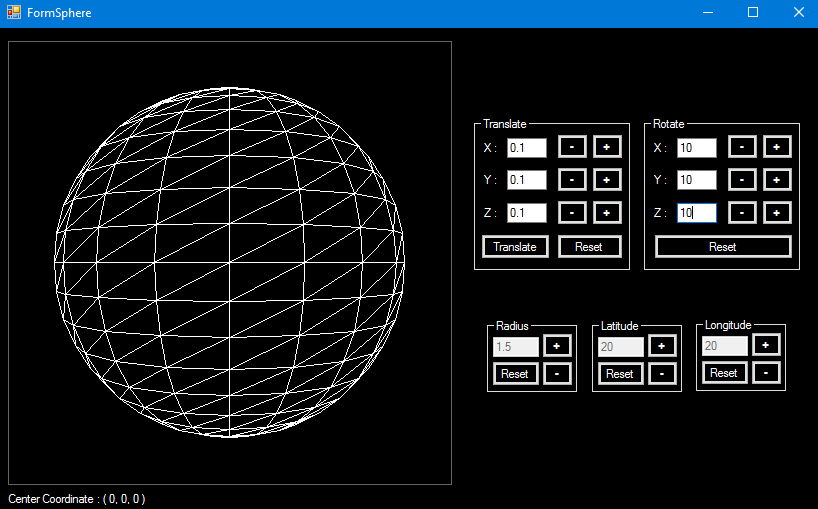


For this case, the radius of the sphere is 1 and then decreased by 0.5.



The radius of the sphere has been decreased by 0.5; now the radius is 0.5.

1. Reset the radius of the sphere



For this case, the radius of the sphere has been increased by 0.5, so the radius is 1.5.

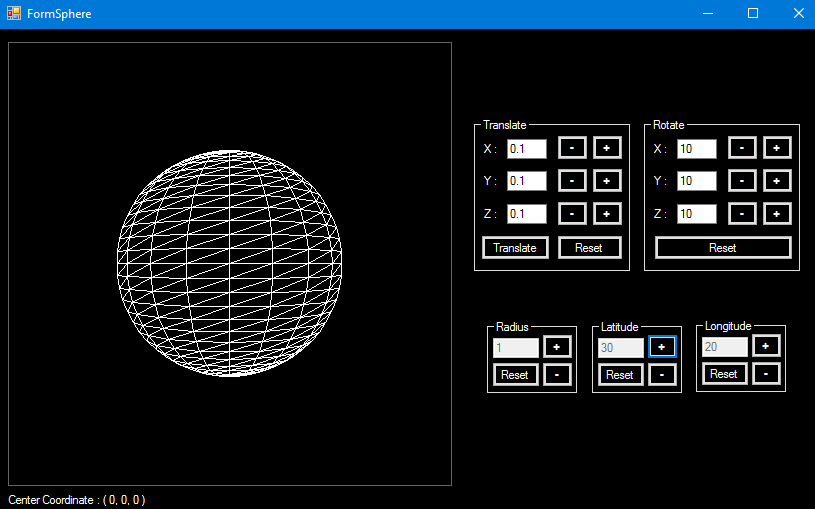


The radius of the sphere has been reset; now the radius is 1.

1. Increase Latitude on the sphere



For this case, the latitude on the sphere is 20 and then increased by 10.

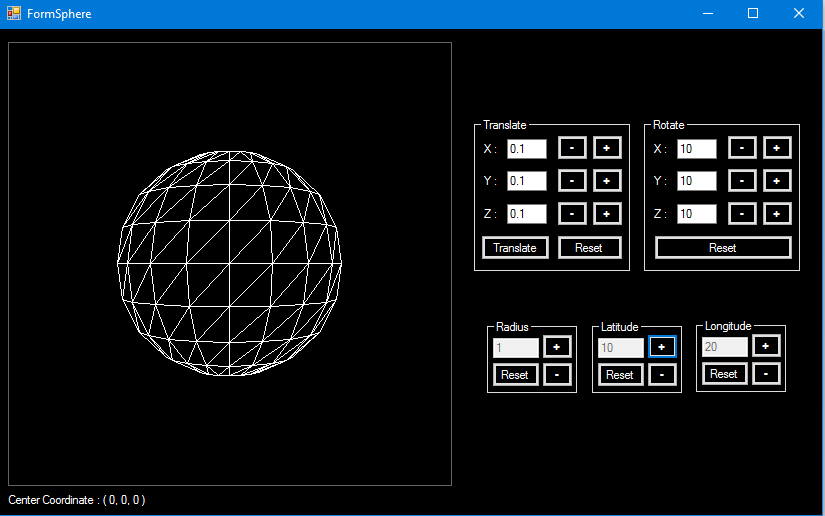


The latitude on the sphere has been increased by 10; now the latitude is 30.

1. Decrease latitude on the sphere

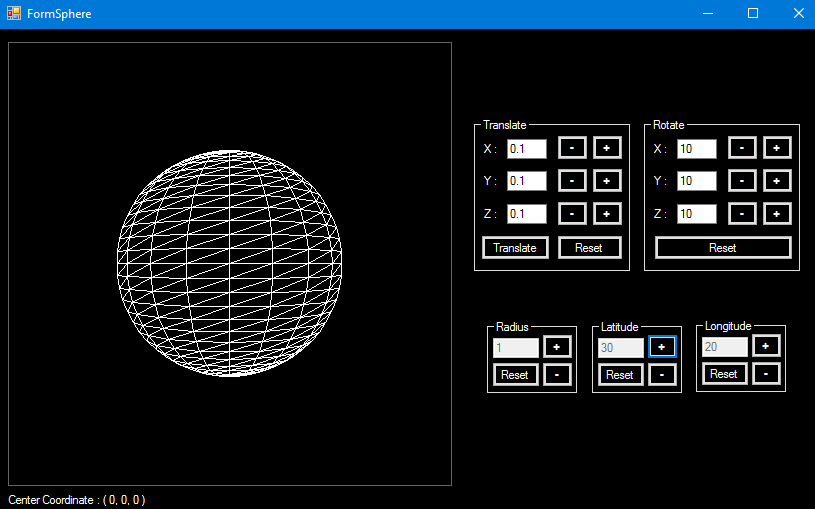


For this case, the latitude on the sphere is 20 and then decreased by 10.



The latitude on the sphere has been decreased by 10; now the latitude is 10.

1. Reset the latitude on the sphere



For this case, the latitude on the sphere has been increased by 10, so the latitude is 30.

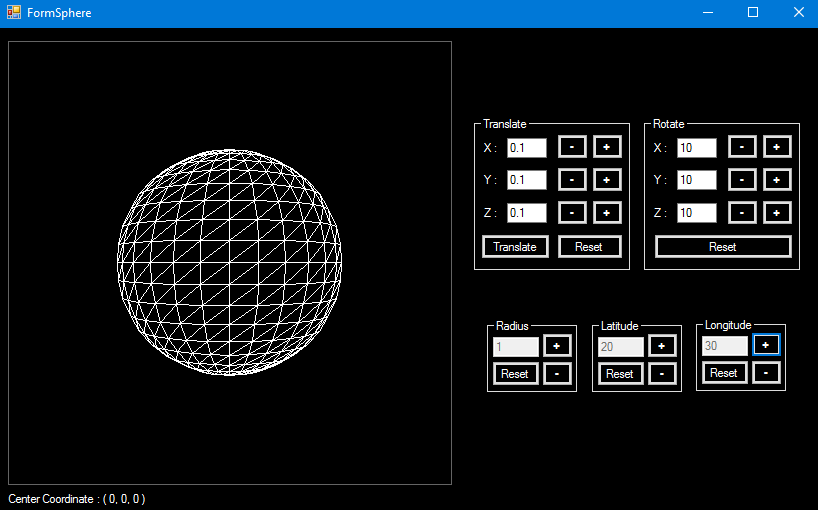


The latitude on the sphere has been reset; now the latitude is 20.

1. Increase longitude on the sphere



For this case, the longitude on the sphere is 20 and then increased by 10.

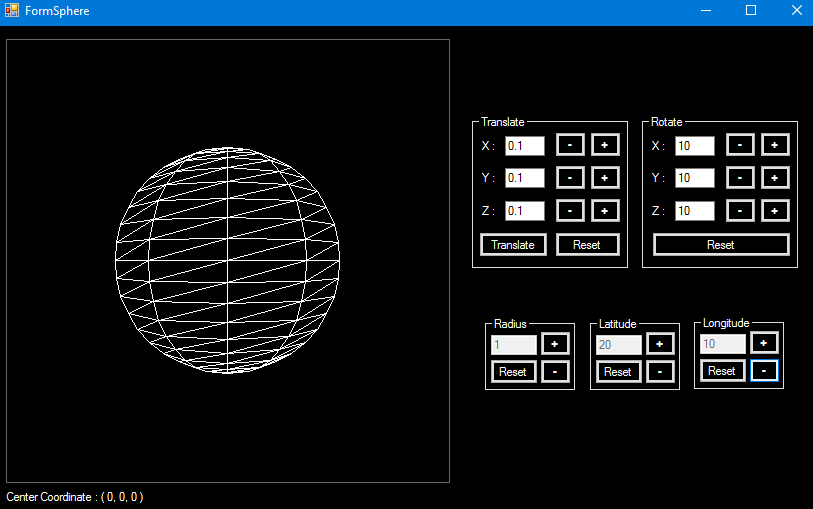


The longitude on the sphere has been increased by 10; now the longitude is 30.

1. Decrease longitude on the sphere

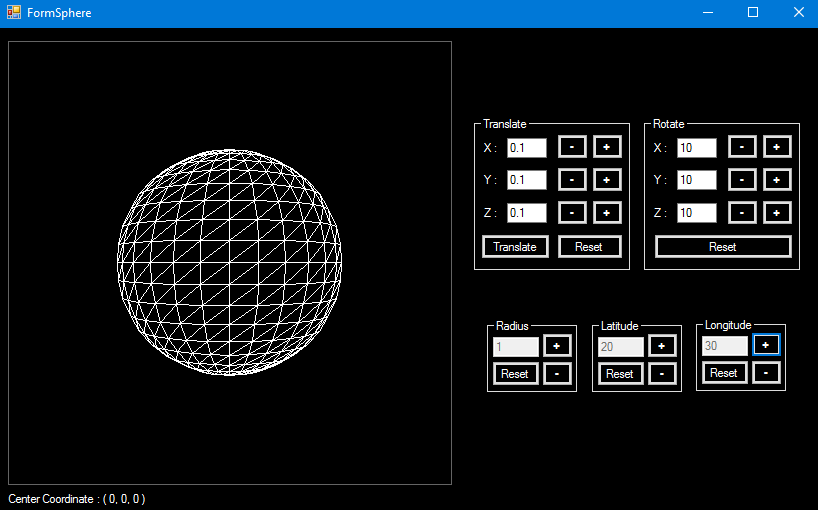


For this case, the longitude on the sphere is 20 and then decreased by 10.



The longitude on the sphere has been decreased by 10; now the longitude is 10.

1. Reset the longitude on the sphere



For this case, the longitude on the sphere has been increased by 10, so the longitude is 30.



The longitude on the sphere has been reset; now the longitude is 20.