

Problem::

Function that generates a half circle given a number of points and radius.

Answer::

To make a function that generates a half sphere you need a std vector of type glm vec4 that takes in two arguments number of points 'np' and radius. you need to declare a std vector of type glm vec 4 named circleVerts. then you to declare a for loop that has the condition that Float i =0 and i<np i++ in side of the for loop create a local variable float angle that is assigned the value of $\text{glm::pi<float>}() / ((\text{float})\text{np}-1)$; then make another float theta that is assigned the value of $i * \text{angle}$; then you want to push back the circle verts .

```
std::vector<glm::vec4> RenderingGeometryApp::genHalfCircle(int np, double radius)
{
    //1st two arguments int np(Number of Points); double radius;
    //2nd declare number of points;
    //3rd declare local variable that will represent an vertex's position.
    std::vector<glm::vec4> CircleVerts;

    for (float i=0; i<np; i++)
    {
        //calculate (angle or theta) in for loop.
        //angle is equals the answer of (3.14/number of points)
        float angle = glm::pi<float>() / ((float) np - 1);
        float theta = i * angle;

        //push back each vertice in the vertex _points->
        //that shows each generated portion of the half circle
        CircleVerts.push_back(glm::vec4(glm::cos(theta)*radius, glm::sin(theta)*radius, 0, 1));
    }
    return CircleVerts;
}
```