

## 1. Objective

- 1) Drawing 3D color sphere

## 2. Data Structure & algorithm

- 1) Simply modify the *cgcirc* to *planet*

### 2) *create\_sphere\_vertices*

- i. Two iterations make the vertices of a sphere
- ii. inner loop for 36 vertices of latitudes, outer loop for 72 vertices of longitude,
- iii. To make the triangles of left plane indices have the order like

$[i], [i+1], [i + \text{the number of vertices in latitudes}(36) + 1]$

- iv. also to make the right triangles indices have the order like

$[i+1], [i + \text{nv} + 1], [i + \text{nv} + 2]$

- v. So, it makes the plane of longitude like the picture. It iterates for longitudes number times.

- vi. in `glDrawElement`, the number of totally drawn triangles is as followed

a.  $(\text{num of the vertices in longitude} + 1) * ((\text{num of the vertices in latitude}) * 2) + 1) * 3$

### 3) toggle the `texColor`

- i. The argument *b\_solid\_color* is easily passed by `glUniform1i` to fragment shader, and in fragment shader it decides to represent the color depending on *b\_solid\_color*, and its color follows the document of the assignment.

## 3. Advanced features

- 1) Because I have learned about shading in the last lecture of the CGP, I decide to add shading in the object.

### 2) shading

- i. phong - if press the s key, then you can see the sphere model applied phong shading.
  - a. light exists in front of the model and the right-top of the scene. That means light position is  $[3, 2, 2]$
- ii. cel – if press the + key in the phong, you can see the model applied cel shading and arrange the number of quantization.

## 4. Discussion

- 1) I didn't calculate the new normal vector, just add the normal vector as same as the position, because there is no fully understanding the last lecture in CGP. After the shading lecture in the CG class, I want to add more accurate normal vector to the model.

