ECM3423: Computer Graphics
Worksheet 5 - Continue on Models & Normals

Getting the new code

For this worksheet, I have uploaded a new version of the codebase onto the VLE, `ecm3423_ws5.py`. The aim of this workshop is to write a class to generate a sphere mesh, with the number of segments as a parameter, and calculate the normals.

If you run this code, you will see a sphere mesh. The following keys control the rendering:

- 0: Switch the rendering between wireframe and polygon fill.
- 1: Render colour using the colour array, set to white at the moment for the bunny, and each vertex to a different colour for the square. Note the interpolation between vertices on the square..
- 2: Render colour according to the vertices x, y, z coordinates (will work out of the box for both).
- 3: Render colour according to the normal array: this is set to [0,0,1] for the all vertices on the square, your aim for this workshop is to set this up for the bunny.
- 4: Set all colour to black. This is the initial state at the start of the program.

Generate a model of a sphere

In the second part of this workshop, you are asked to create a new model class (inheriting from `BaseModel`), that will draw a sphere with the number of segments horizontally and vertically as parameters. You need to ensure as in task 1 that all normals are correctly computed.

- Start by creating the vertex array, which should be simple, and then define the index array (the more tricky part).
- Start by building the top and bottom end of the sphere, they are special cases and it will get you started (look at the picture below)
- we want to draw a unit sphere, so the top of the sphere will have the coordinates $[0, 1, 0]$ and the bottom $[0, -1, 0]$.
- start with a small number of segments, to make debugging easier, but not too small (5 horizontal and 5 vertical can be a good start)
- watch out for the *ordering* of your indices in your triangles: they impact which triangles are visible and which are not. This is called the *winding order*. If some triangles only appear when rotating the sphere, try inverting the order of the indices (eg, $(1, 3, 2)$ instead of $(1, 2, 3)$).

Hint: The result of this task should look similar to this, for 10 horizontal and vertical segments (Left: rendered using mode 3, where colours indicate normals; right: rendered in wireframe to show details of triangle layout at the top, rendered using mode 2 where colours indicate position):