**GAME ENGINE FOUNDATIONS**

**Lab Two**

In this lab, we are going to draw our first 3D object: a cube. However, we will draw only four sides of it, so no top and no bottom for now.

Spend a few minutes and find out the coordinations of the vertices that you will need. Let the dimension of each face to be 0.9 units.

You’ll need 6 vertices for every face and in total, 24 vertices. In this case, some of your vertices will overlap. Be careful when you are calculating the coordination of the vertices, otherwise you will end up with a distorted geometry. Let the bottom of the cube to be located on (0,0,0).

Apply a solid and uniques color to every face of the cube to make it distinguishable.

Now, place the camera on these locations: (1, 1, 0), (-1, 1, 0), (-1, -1, 0), (1, -1, 0). For each location, convince yourself that what is rendered is correct. Otherwise, go back a few steps and check your work. Have you properly calculated the up vector? What parameters do you use for the direction of the camera in each location?

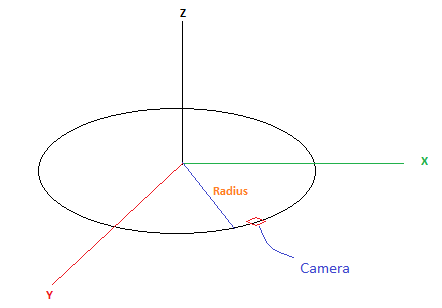
Finally, we are going to animate the camera over time around our geometry on a circular path and on X-Y plane (look at the illustration on the next page). To do this, we will need to parametrize the location of the camera and change it in every frame, as follow:

1. Define a variable as a float, say alpha and initialize it to zero.
2. in the idle function increase “alpha” and call to display function to re-render the scene. NOTE: Use “glutPostRedisplay()” to call your display function.
3. In your display function, set X and Y position of the camera to be as follow:

X= R \* cos(alpha)

Y= R \* sin(alpha)

Where: R is the radius of your circular path from the origin.



Animating your camera on circular path on X-Y plane

|  |  |
| --- | --- |
|  |  |
| Parameters for camera matrix | Parameters for projection matrix |