**Math**

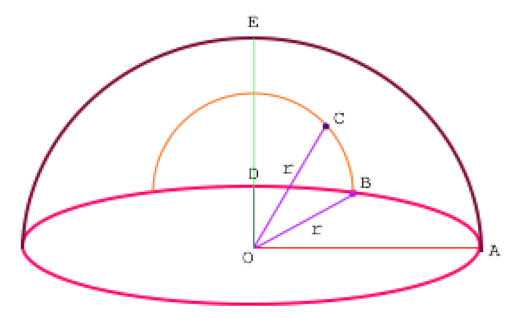
The goal of this document is to explain the math I used for the camera and projection for my program.

**Camera definition**

* float:
  + yaw
    - rotation parallel to the game space
  + pitch
    - rotation perpendicular to the game space
  + aspect ratio
  + fov
  + rotate speed
  + move speed
  + zoom
* matrix:
  + view
  + proj
* vector:
  + target
  + position
  + facing
  + right
  + up

**Basic Idea**

3D view space. Camera above world looking down at tilt angle (camera.pitch). Camera moves in a circle around a target, which is where the player is and what defines the rotation angle (camera.yaw).



* OA = x-axis
* OD = z-axis
* OE = y-axis
* ∠AOB = rotation (camera yaw)
* ∠BOC = tilt (camera pitch)
* r = radius (camera distance)
* C = camera
* O = target

**Rotate and Tilt camera**

The position of a point on the sphere with radius 1 given the rotation and tilt angles is as follows:

* x = cos(rotation) \* cos(tilt)
* y = sin(tilt)
* z = sin(rotation) \* cos(tilt)

The result is scaled by the radius and displaced by the position of the target. From these values, the position of the camera is updated. Conveniently, the facing vector

**View and Projection matrix**

After updating the facing vector, the right and up vectors can be computed.