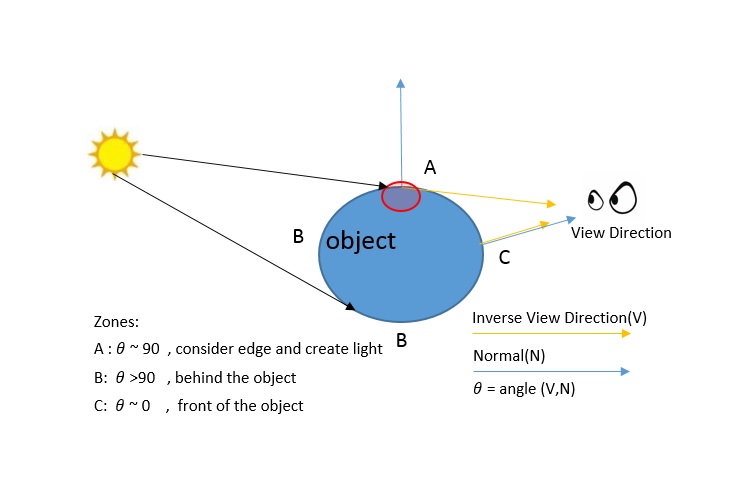
Rim Lighting

Using the following material create a basic rim lighting shader:



* In the image above, we don’t use light direction to get the rim lighting but I just added it to see the concept.
* We know that the dot product of two orthogonal vectors is *0.0* because they form a *90*degrees angle and cosine of *90*is equal to *0.0*. The problem is that we actually need that dot product to be *1.0*instead of *0.0* when we create the rim effect (to brighten the contour). So we can subtract the dot product result from *1.0*:
* The only thing left to discuss about Rim lighting implementation is the contribution of light on the surface of the object. There are a couple of ways to deal with this. We can use an **if-else-if** statemant or a **pow** function or a **smoothstep** function. The last two can create a nice cutoff and give some really nice results.

