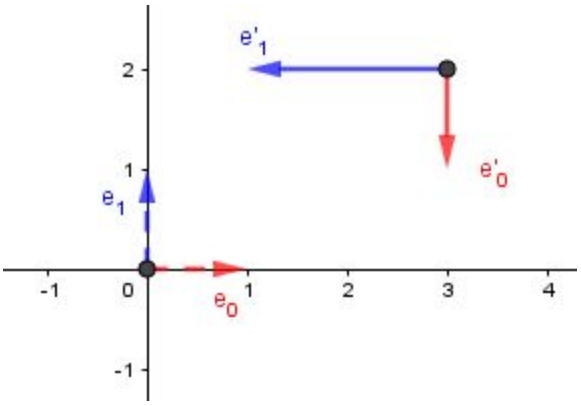
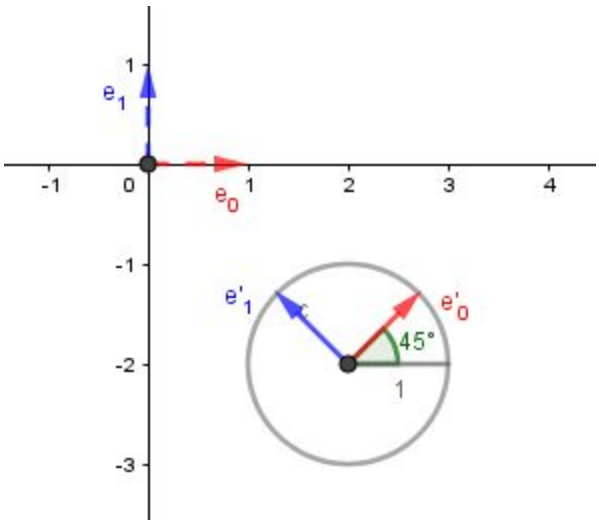
Computer Graphics

Basic I Math Tasks

1. Find the cosine of the angle between vectors $u = (\sqrt{3},\ 1)$ and $v = (\sqrt{3},\ 3)$ using the dot product.

2. Find the matrices that do the following affine transformations.





3. Show that scale is a linear transformation.

4. Show that translation is not a linear transformation.

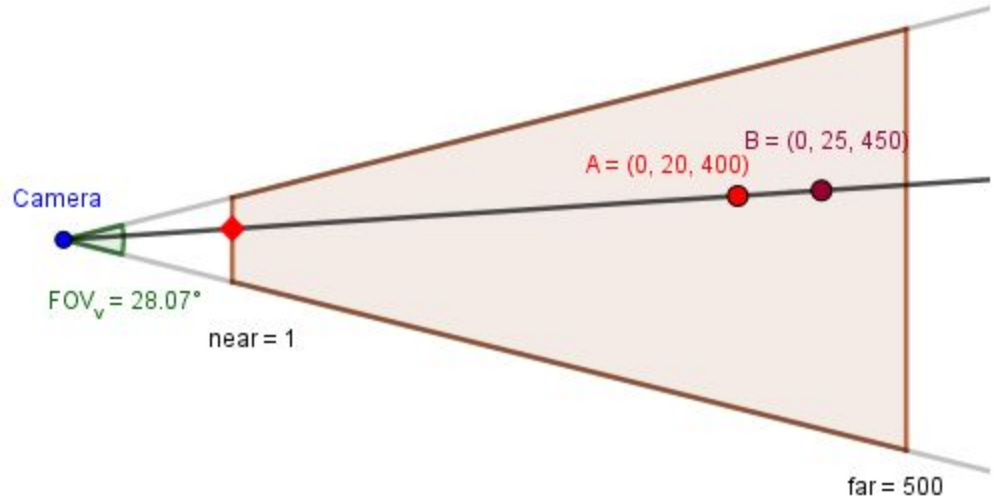
5. Invert the following augmented transformation matrix that has an orthonormal linear transf. part.

$$\begin{pmatrix} 0.5 & 0 & 0.87 & 5 \\ 0 & 1 & 0 & 6 \\ -0.87 & 0 & 0.5 & 7 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} =$$

6. Show that a perspective projection with FOV 60 is exactly the same as with FOV 420.

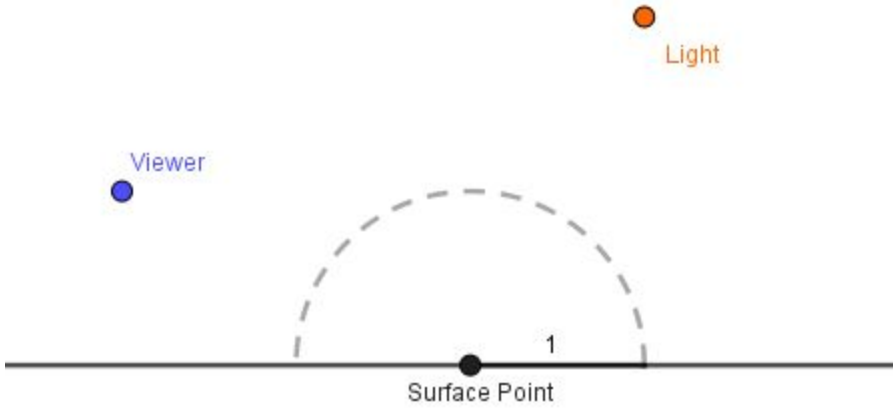
7. Show that and explain why the *lookAt* command fails when the *up* vector is collinear with the direction the camera is looking at.

8. Show and explain what is happening in the render with the projections of fragments A and B when the MAX_INT value is 256 and buffers use integers for value storage.

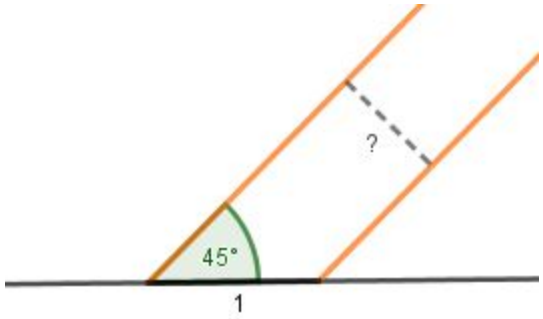


$$P = \begin{pmatrix} \frac{1}{ar \cdot \tan(\frac{FOV_v}{2})} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\frac{FOV_v}{2})} & 0 & 0 \\ 0 & 0 & \frac{near+far}{near-far} & \frac{2 \cdot near \cdot far}{near-far} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

9. Draw the vectors n, l, v and r from the Phong's lighting model on the following diagram.



10. Calculate the percentage of light reaching this surface using a cosine.



11. Show that the Phong's and Blinn-Phong's specular terms do not give the same result given the same input parameters.

12. Show why with per-fragment shading you need to normalize the normal before outputting it from the vertex shader (after you move it to the camera space with the *normalMatrix*).
Assume the following *modelView* matrix:

$$MV = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

13. Explain why with per-fragment shading you need to normalize the normal when you receive it in the fragment shader.

14. Draw an illustration of, write the formula of and explain the bilinear filtering (interpolation).

15. Given an uncompressed RGBA texture with 8 bits per channel and the resolution of 4096×4096**, how much memory is it going to take on the GPU? How much do its mipmaps take?**

16. Write the general blending formula for the premultiplied alpha blending.

17. Show that the conventional alpha blending causes alpha bleeding with bilinear filtering.