1) What are the main advantages and main disadvantages to generating images using the graphics pipeline?

The pipeline approach is fast, and it reduces the memory requirements of the program. However, the latency of the system needs to be able to handle increased throughput in calculating the pipeline's performance

(I was lost and confused on this question, so I did look it up: https://developer.nvidia.com/gpugems/gpugems/part-v-performance-and-practicalities/chapter-28-graphics-pipeline-performance)

2) Show that a rotation and a *uniform* scaling transformation (i.e. one where all the scale factors are identical) are commutative, i.e. that they can be applied in either order.

Show Steps

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \beta & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}=\begin{pmatrix} \beta\cos(\theta) & -\beta\sin(\theta) & 0 & 0 \\ \beta\sin(\theta) & \beta\cos(\theta) & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Steps

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \beta & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiply the rows of the first matrix by the columns of the second matrix

Show Steps

$$=\begin{pmatrix} \cos(\theta)\beta+(-\sin(\theta))\cdot 0+0\cdot 0+0\cdot 0 & \cos(\theta)\cdot 0+(-\sin(\theta))\beta+0\cdot 0+0\cdot 0 & \cos(\theta)\cdot 0+(-\sin(\theta))\cdot 0+0\cdot\beta+0\cdot 0 & \cos(\theta)\cdot\ 0+ \\ \sin(\theta)\beta+\cos(\theta)\cdot 0+0\cdot 0+0\cdot 0 & \sin(\theta)\cdot 0+\cos(\theta)\beta+0\cdot 0+0\cdot 0 & \sin(\theta)\cdot 0+\cos(\theta)\cdot 0+0\cdot\beta+0\cdot 0 & \sin(\theta)\ 0+ \\ 0\cdot\beta+0\cdot 0+1\cdot 0+0\cdot 0 & 0\cdot 0+0\cdot\beta+1\cdot 0+0\cdot 0 & 0\cdot 0+0\cdot 0+1\cdot\beta+0\cdot 0 & 0\cdot 0+0 \\ 0\cdot\beta+0\cdot 0+0\cdot 0+1\cdot 0 & 0\cdot 0+0\cdot\beta+0\cdot 0+1\cdot 0 & 0\cdot 0+0\cdot 0+0\cdot\beta+1\cdot 0 & 0\cdot 0+0 \end{pmatrix}$$

Simplify each element

$$=\begin{pmatrix} \beta\cos(\theta) & -\beta\sin(\theta) & 0 & 0 \\ \beta\sin(\theta) & \beta\cos(\theta) & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Show Steps

$$\begin{pmatrix} \beta & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \beta\cos(\theta) & -\beta\sin(\theta) & 0 & 0 \\ \beta\sin(\theta) & \beta\cos(\theta) & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Steps

$$\begin{pmatrix} \beta & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiply the rows of the first matrix by the columns of the second matrix

Show Steps

$$= \begin{pmatrix} \beta\cos(\theta)+0\cdot\sin(\theta)+0\cdot 0+0\cdot 0 & \beta(-\sin(\theta))+0\cdot\cos(\theta)+0\cdot 0+0\cdot 0 & \beta\cdot 0+0\cdot 0+0\cdot 1+0\cdot 0 & \beta \\ 0\cdot\cos(\theta)+\beta\sin(\theta)+0\cdot 0+0\cdot 0 & 0\cdot(-\sin(\theta))+\beta\cos(\theta)+0\cdot 0+0\cdot 0 & 0\cdot 0+\beta\cdot 0+0\cdot 1+0\cdot 0 & 0 \\ 0\cdot\cos(\theta)+0\cdot\sin(\theta)+\beta\cdot 0+0\cdot 0 & 0\cdot(-\sin(\theta))+0\cdot\cos(\theta)+\beta\cdot 0+0\cdot 0 & 0\cdot 0+0\cdot 0+\beta\cdot 1+0\cdot 0 & 0 \\ 0\cdot\cos(\theta)+0\cdot\sin(\theta)+0\cdot 0+1\cdot 0 & 0\cdot(-\sin(\theta))+0\cdot\cos(\theta)+0\cdot 0+1\cdot 0 & 0\cdot 0+0\cdot 0+0\cdot 1+1\cdot 0 & 0 \end{pmatrix}$$

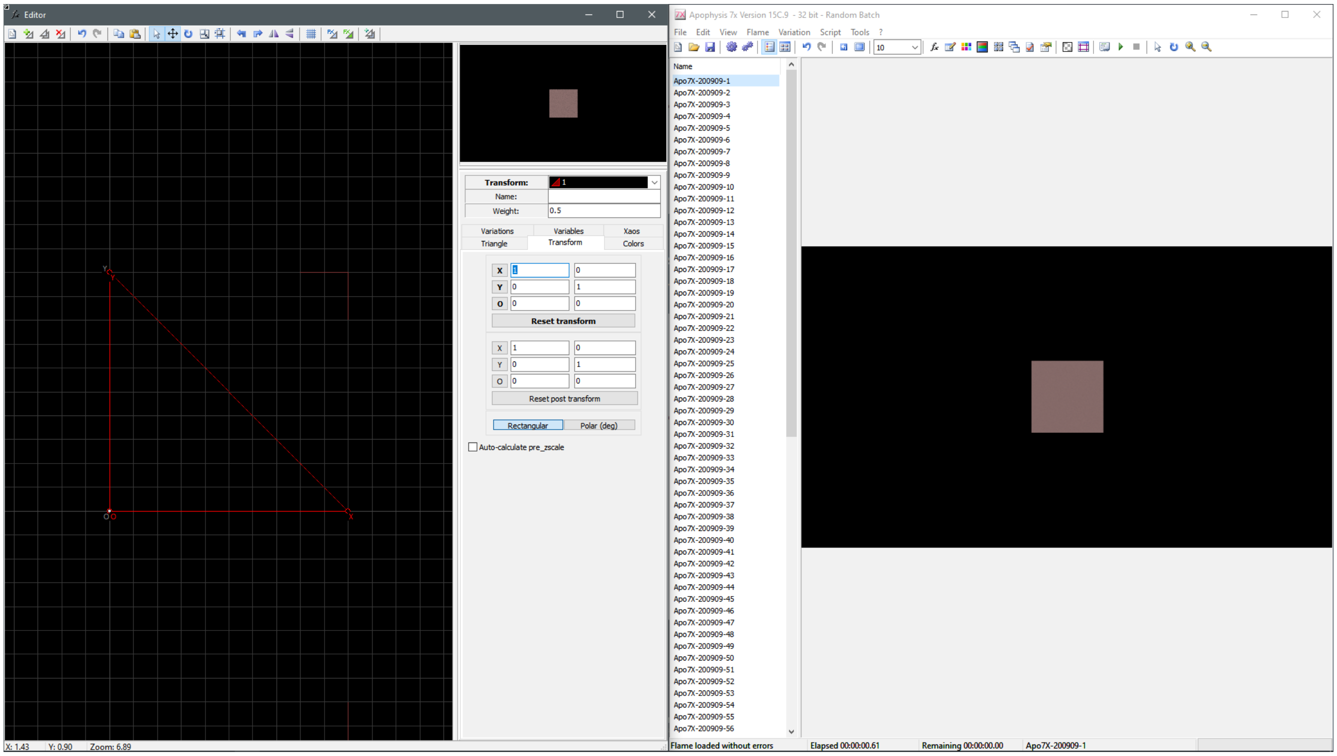
Simplify each element

$$= \begin{pmatrix} \beta\cos(\theta) & -\beta\sin(\theta) & 0 & 0 \\ \beta\sin(\theta) & \beta\cos(\theta) & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

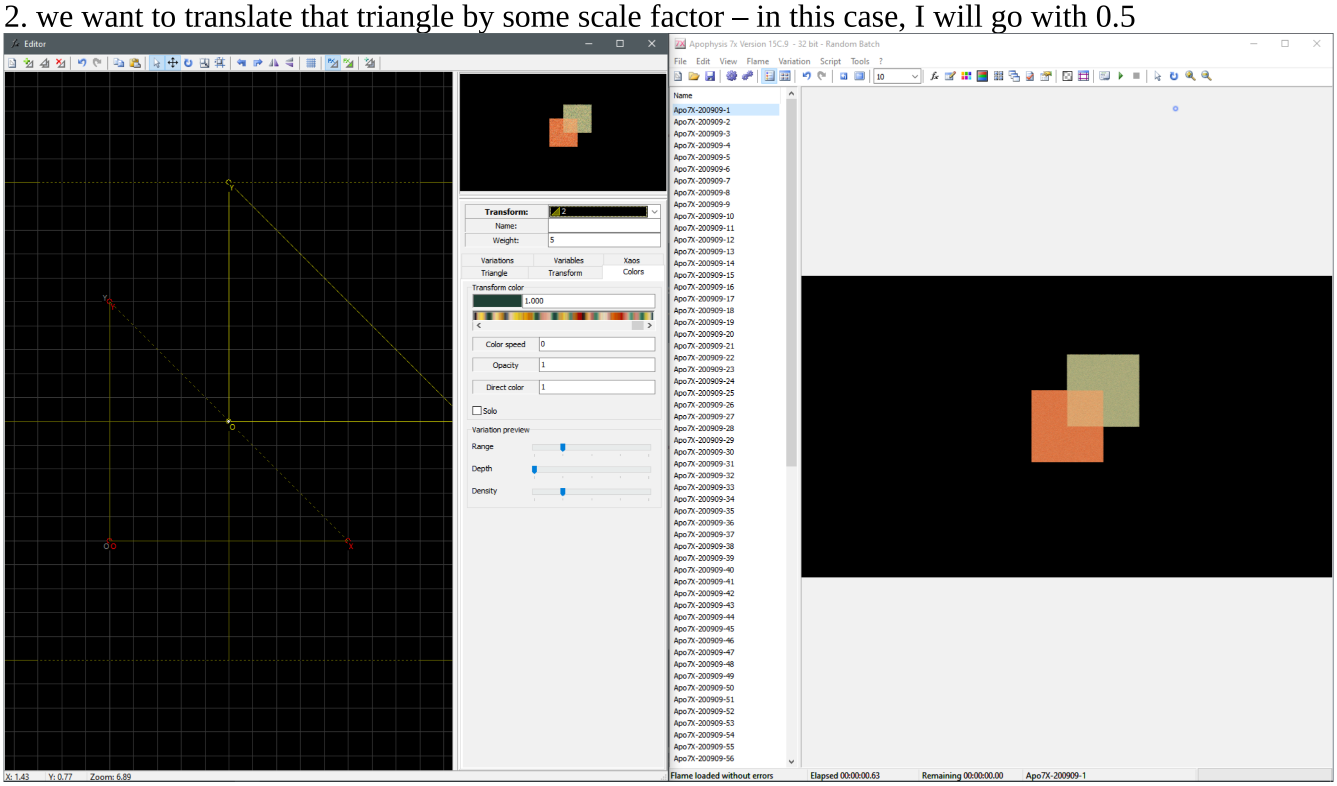
The two functions generate the exact same answer.

3. Given the following set of vertices that define a square,
- Sketch a set of *intermediate* transformations to produce the following end result with proper size, orientation, and location. Give the final product of your particular transformation sequence using T(dx, dy) for a translation by *dx* and *dy*, R(θ) for a rotation (about the z axis) by angle θ, and S(sx, sy) for a scaling by *sx* and *sy*.

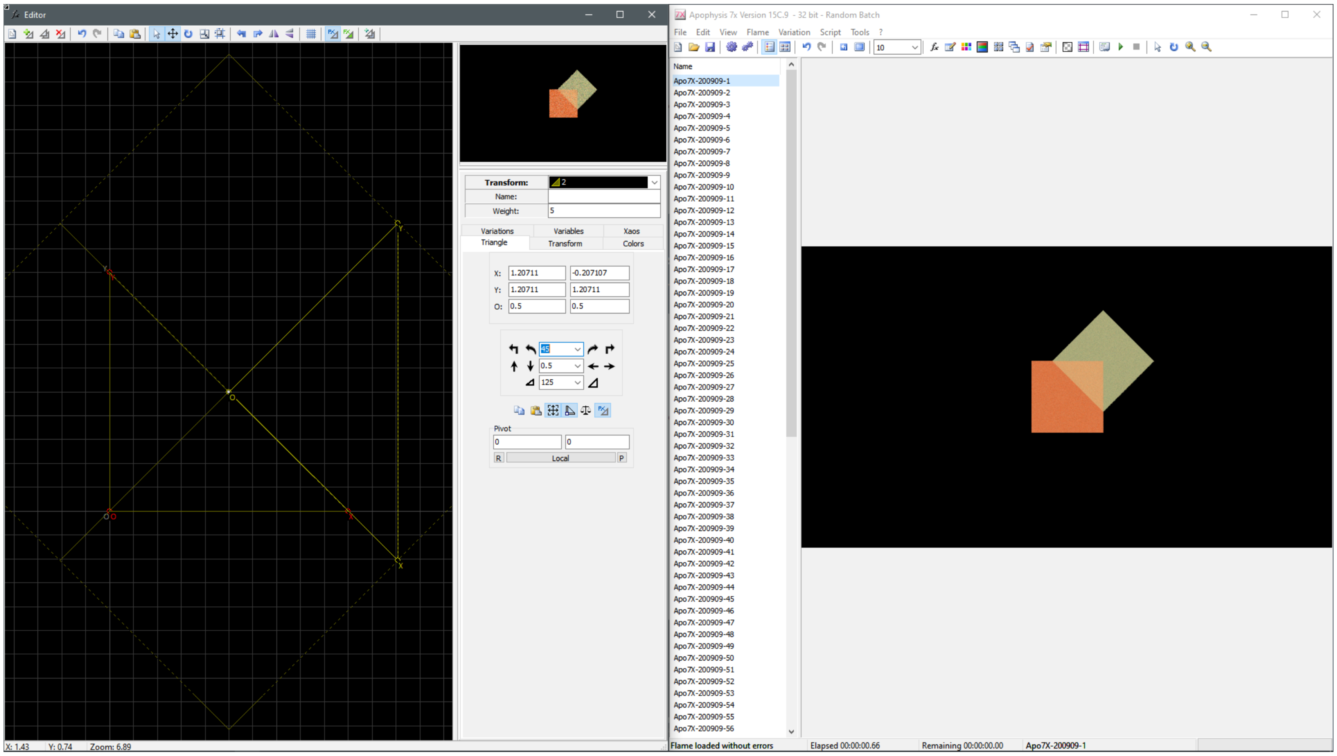
I'm not at all sure what this is asking me to do, so I put it in terms I could understand and modeled this in the fractal software Apophysis 7x.



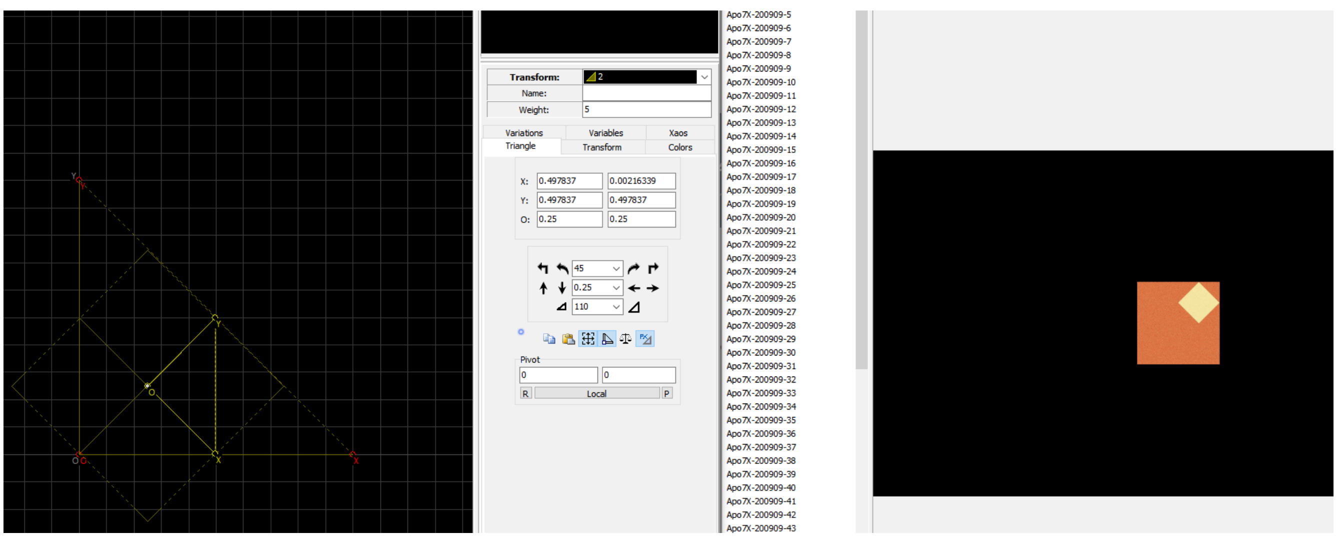
1. we have a square whose x and y dimensions are all 1
2. we want to translate that triangle by some scale factor – in this case, I will go with 0.5



(the initial square is the orange one; the translated one is the green one)
3. we want to rotate that square by some rotation – I n this case, I will go with 45 degrees



4. we want to scale the square by some factor. I didn't know how to figure this out, so I just fidgeted with it and hoped it was some nice multiple of 5. Scaling it down by 10% of its current size 11 times got a good visual fit. Since I did it by hand, and since this works out to a little more than pi/10, I'm just assuming that the scale factor is supposed to be pi/10.



R(θ) = 45
S(sx, sy) = S(pi/10, pi/10) I should have stopped while I was ahead
T(dx, dy) = T(0.5,0.5)

1) Rotate 45 degrees clockwise
2) Scale sx = 0.25, sy = 0.5 what does this even mean
3) Translate dx = 0 (do not change), dy = -0.5