

Reinventing the Pixel:

Creating a Software Renderer

Kike Alcor @gamezer0 // Feb 2016

What is a Software Renderer?

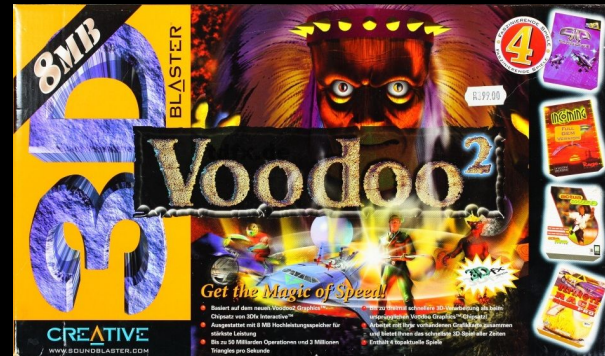
"In the context of computer graphics rendering, software rendering refers to a rendering process that is not dependant upon graphics hardware ASICs, such as a graphics card."

(Wikipedia)

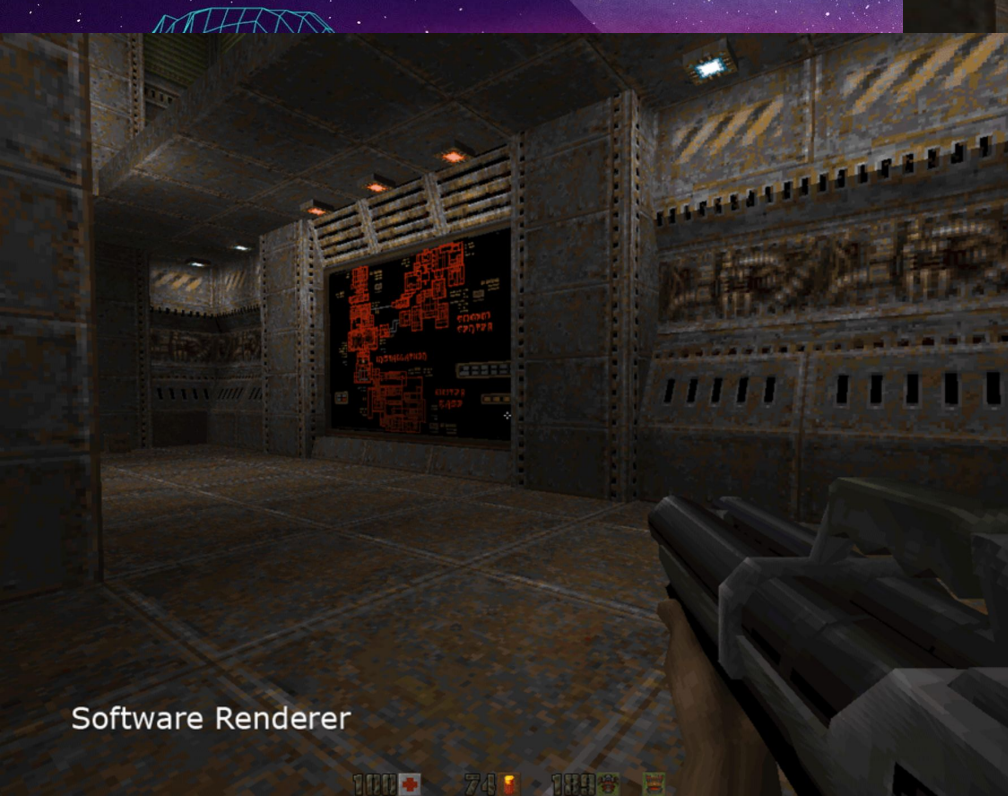
In other words, "DIY" in the CPU.

Benefits of GPU rendering

- Speed (highly parallelizable)
 - Resolution
 - Quality (Antialiasing)
 - Lighting
 - Post processing
 - ...
- Frees the CPU
 - Can do other stuff (Physics, AI,...)



Benefits of GPU rendering



Benefits of GPU rendering

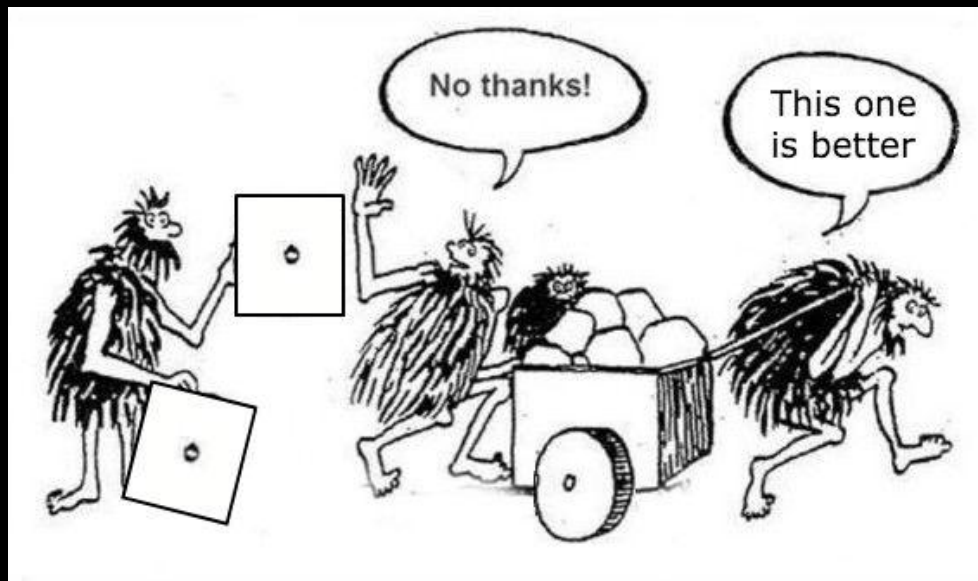


Benefits of GPU rendering



Benefits of CPU rendering

None!?



Motivation

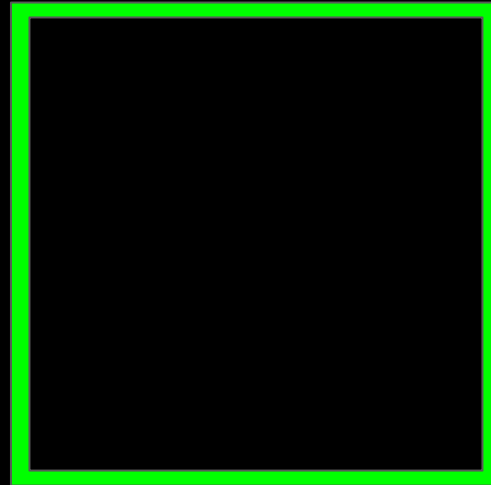
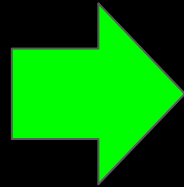
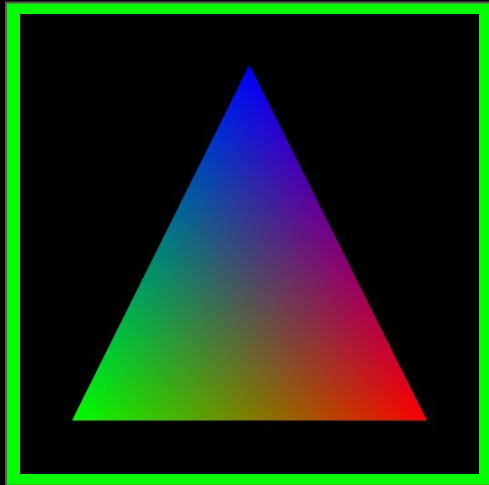
- Put into practice what I learnt about 3D.
- Understand part of what the GPU does.
- Feel like an old school developer.
- It's fun!

Demo



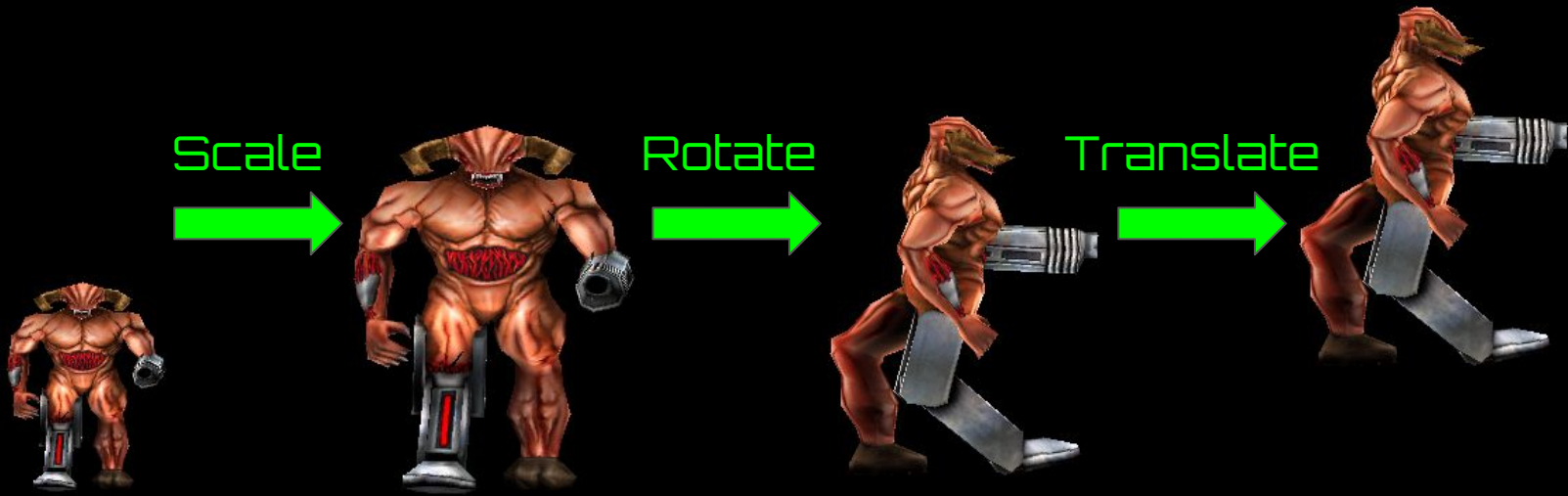
Rendering pipeline

1. Clear Frame Buffer & Depth Buffer



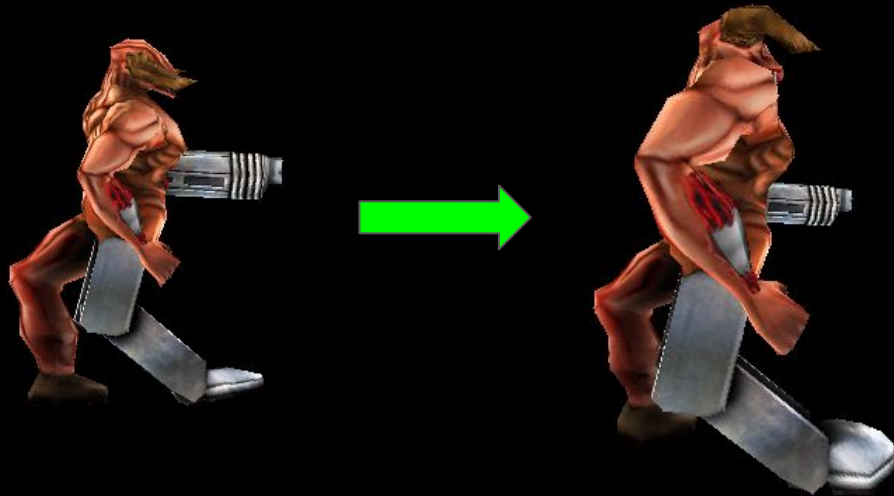
Rendering pipeline

2. Transform object



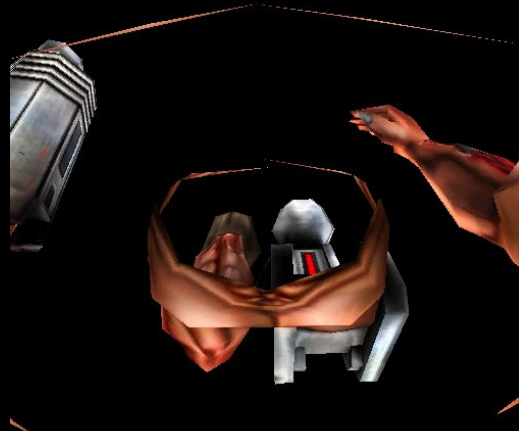
Rendering pipeline

3. Apply view + projection transform



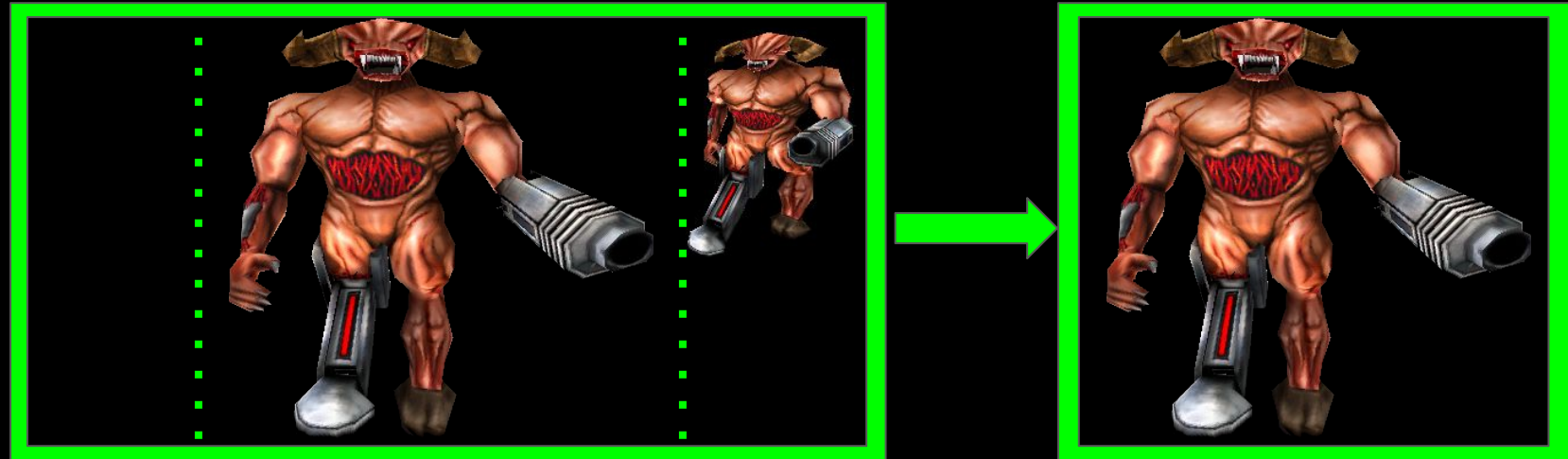
Rendering pipeline

4.1 Culling: backfaces



Rendering pipeline

4.2 Culling: frustum culling



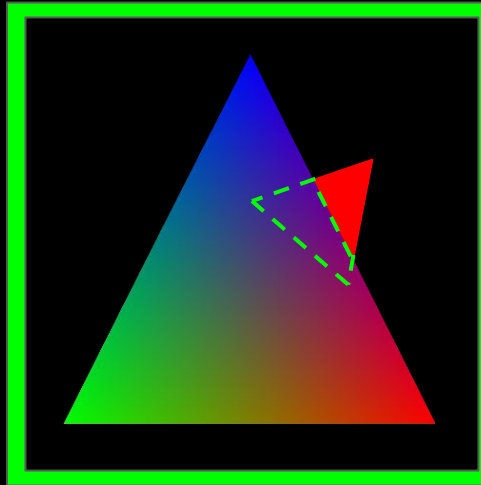
Rendering pipeline

4.3 Culling: frustum clipping



Rendering pipeline

5.1. Rasterize - Method 1



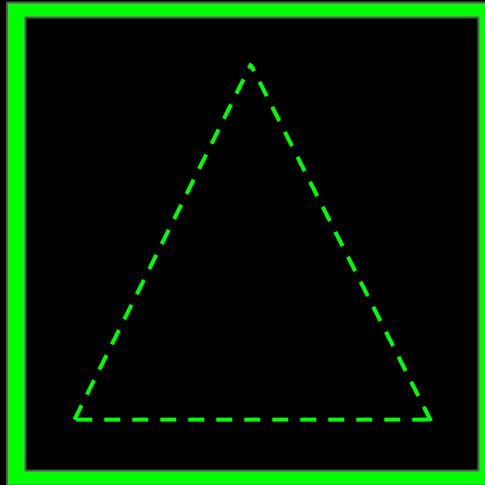
$z = 0$



$z = 1$

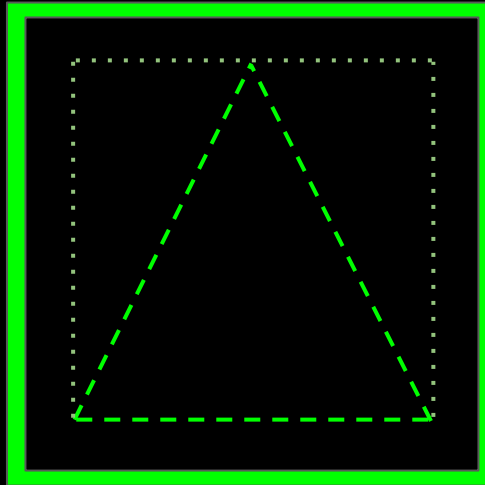
Rendering pipeline

5.1. Rasterize - Method 1



Rendering pipeline

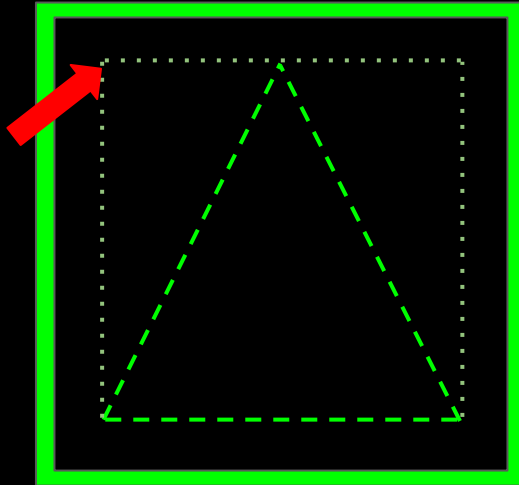
5.1. Rasterize - Method 1



Rendering pipeline

5.1. Rasterize - Method 1

■ $z = \emptyset$
■ $z = 1$



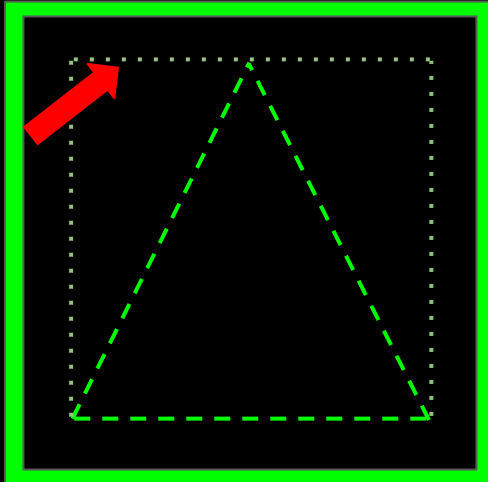
FB

ZB

Rendering pipeline

5.1. Rasterize - Method 1

■ $z = 0$
■ $z = 1$



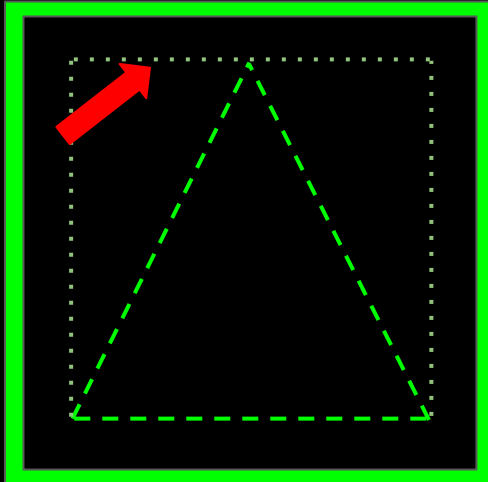
FB

ZB

Rendering pipeline

5.1. Rasterize - Method 1

■ $z = 0$
■ $z = 1$



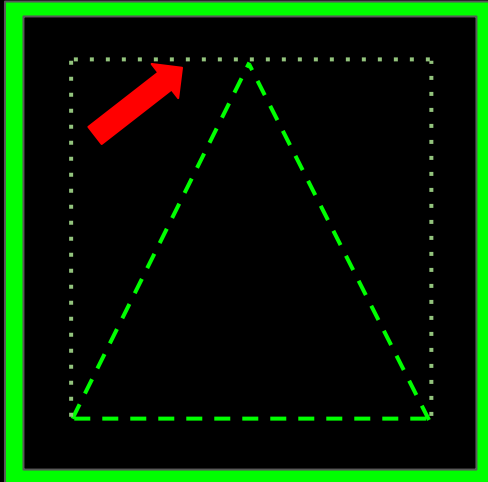
FB

ZB

Rendering pipeline

5.1. Rasterize - Method 1

■ $z = 0$
■ $z = 1$



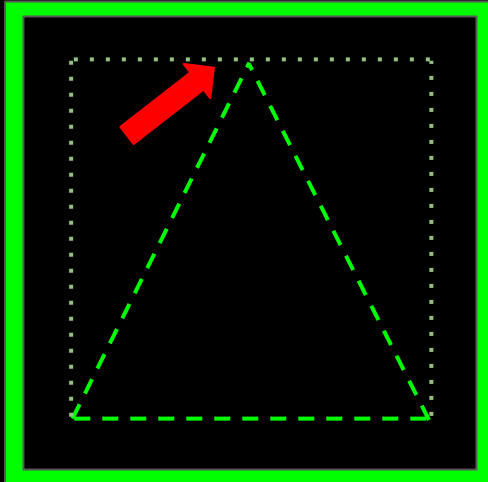
FB

ZB

Rendering pipeline

5.1. Rasterize - Method 1

■ $z = 0$
■ $z = 1$



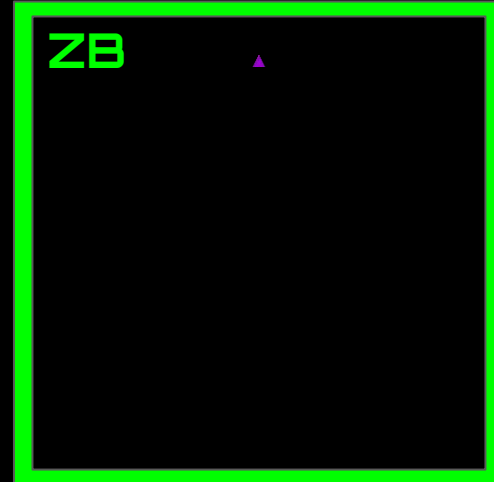
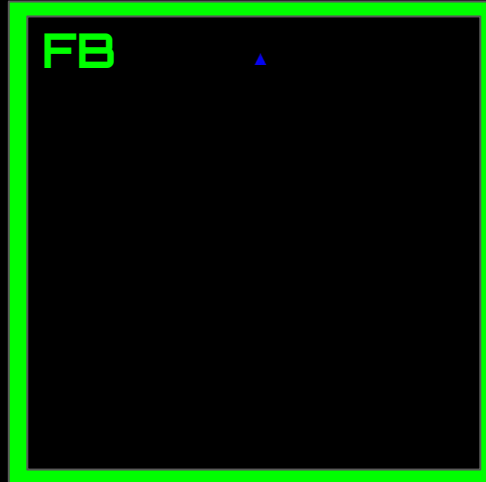
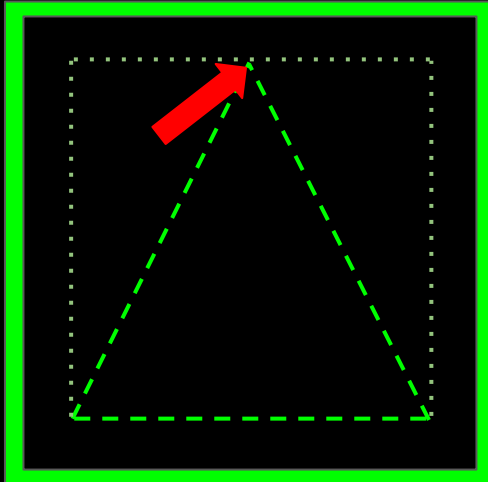
FB

ZB

Rendering pipeline

5.1. Rasterize - Method 1

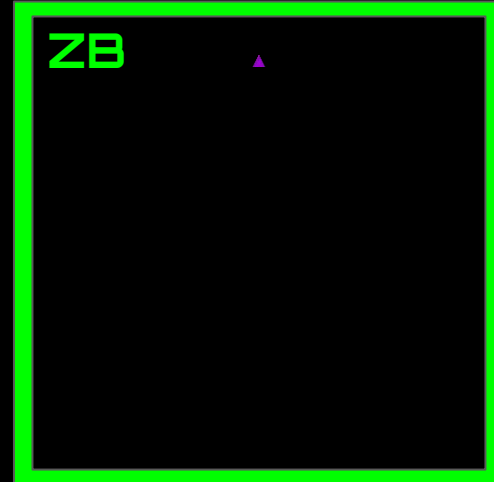
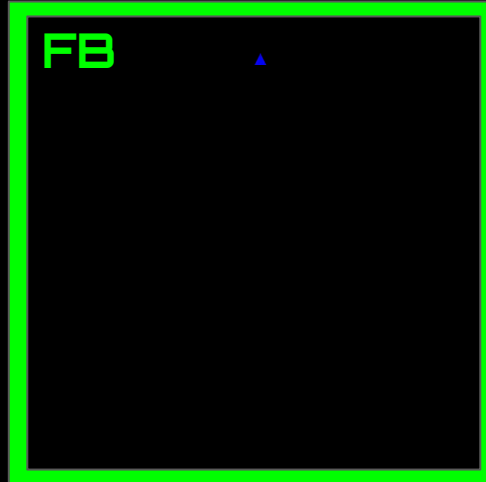
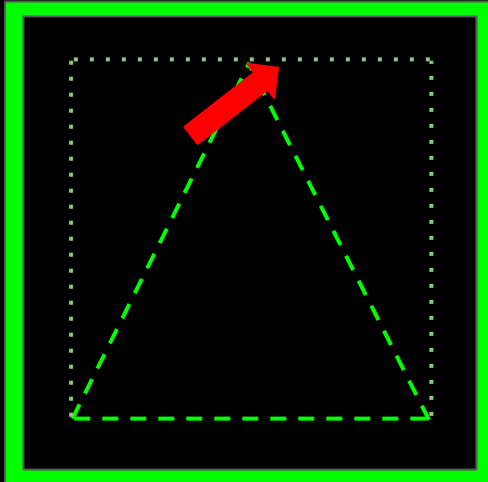
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

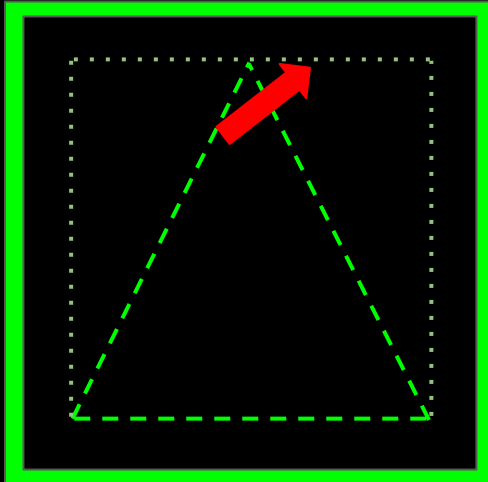
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

■ $z = 0$
■ $z = 1$



FB



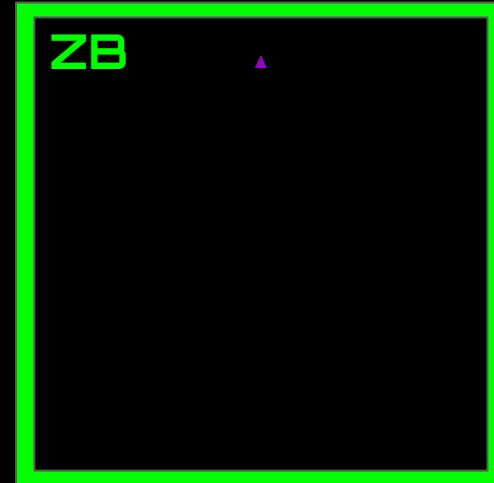
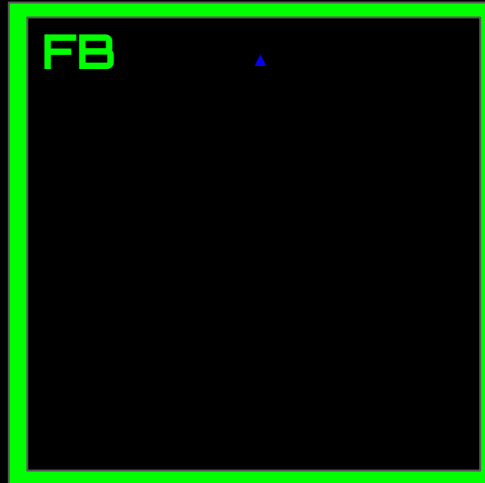
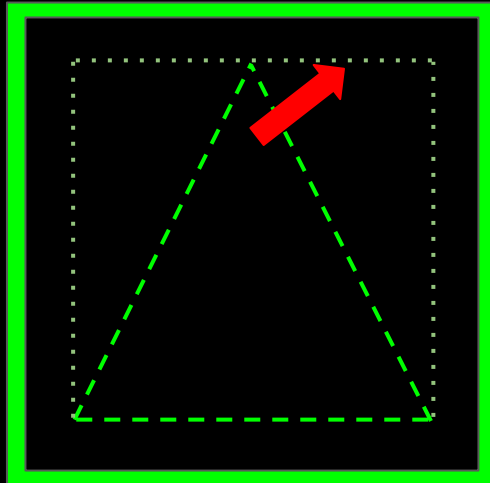
ZB



Rendering pipeline

5.1. Rasterize - Method 1

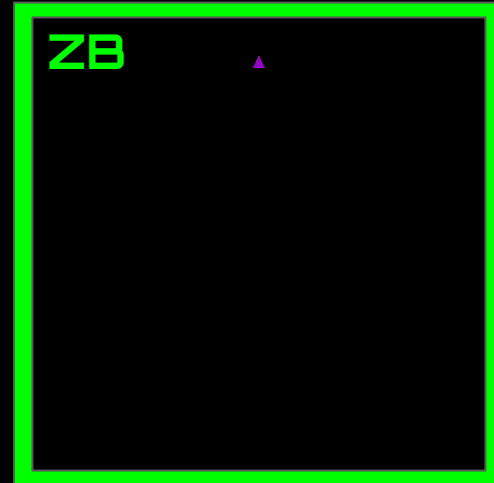
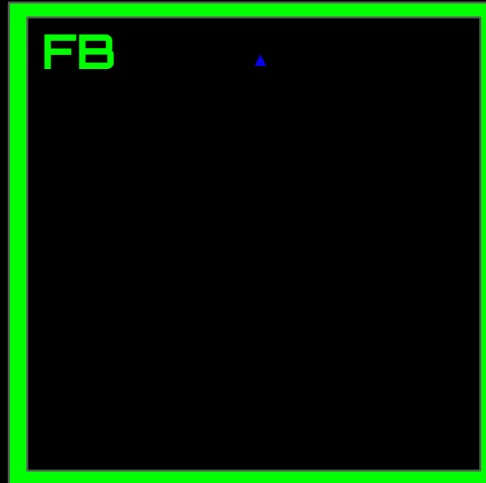
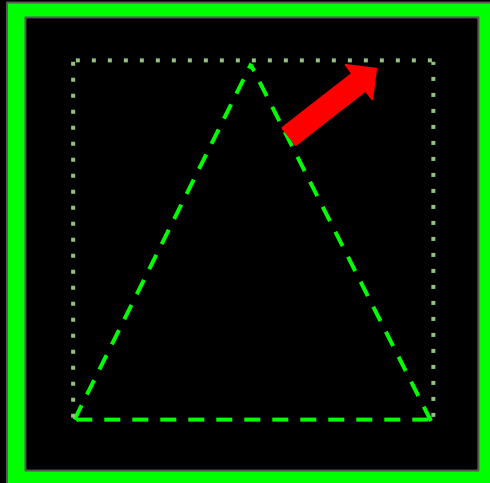
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

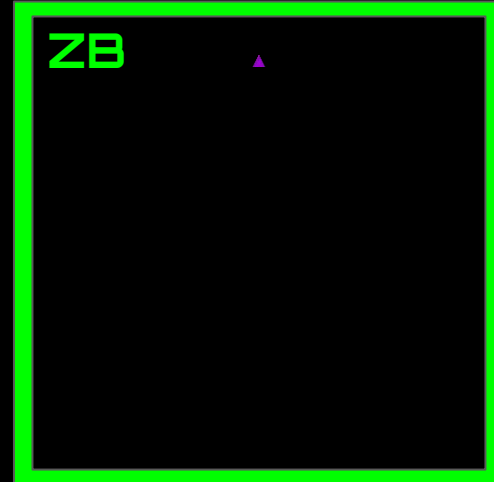
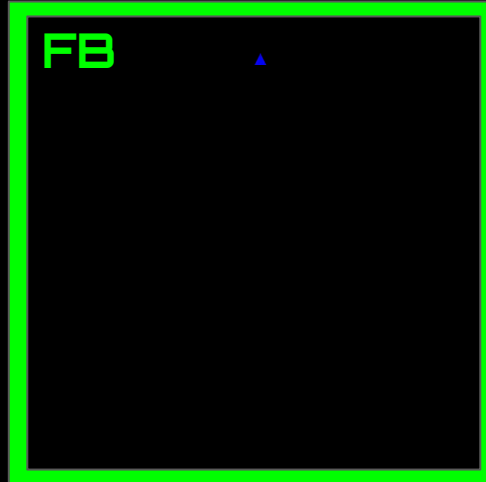
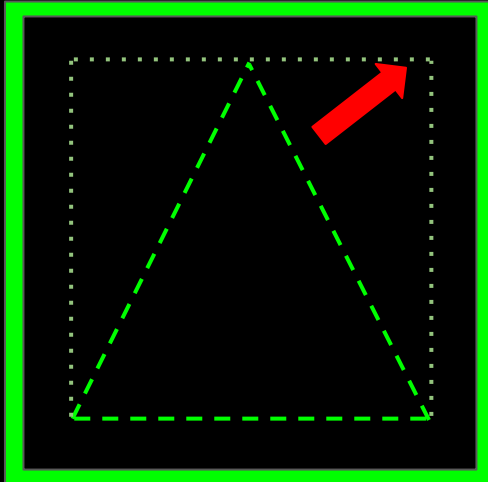
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

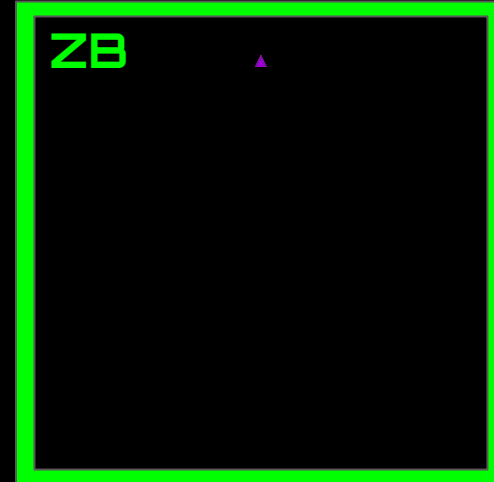
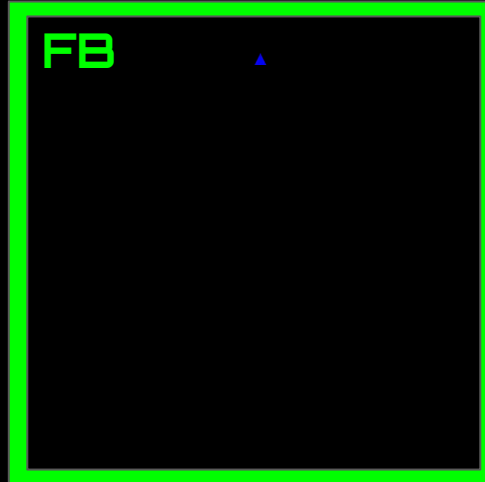
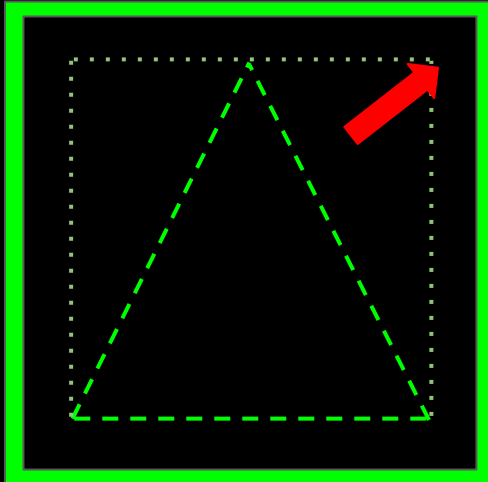
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

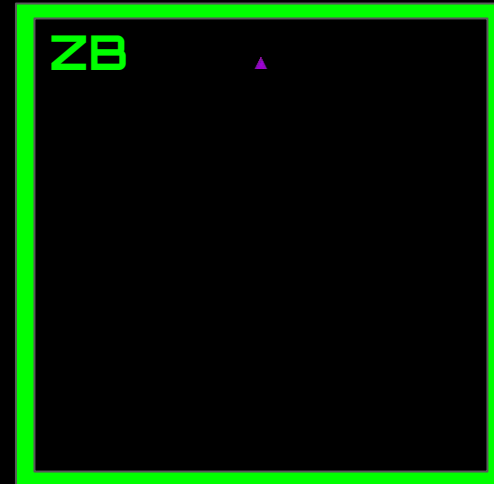
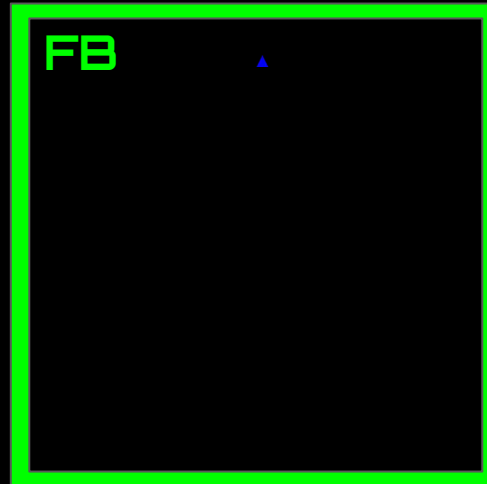
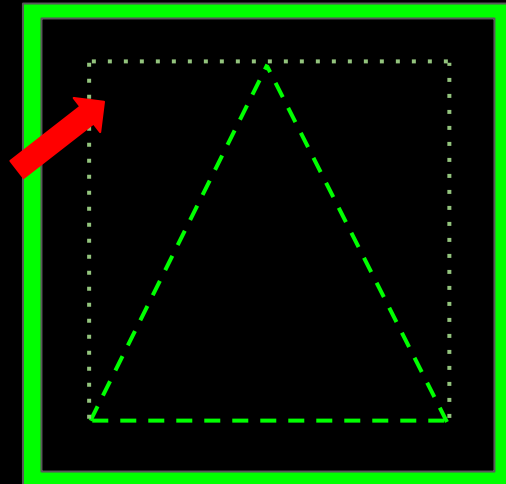
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

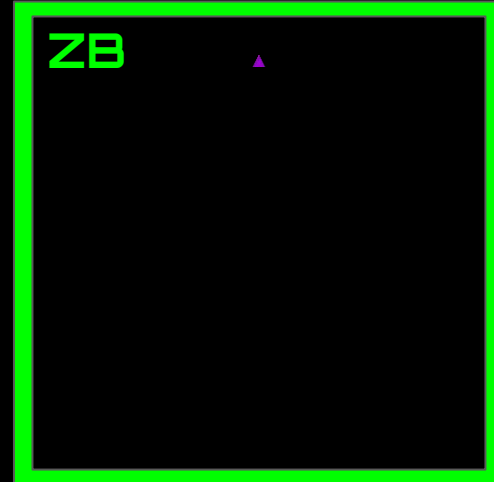
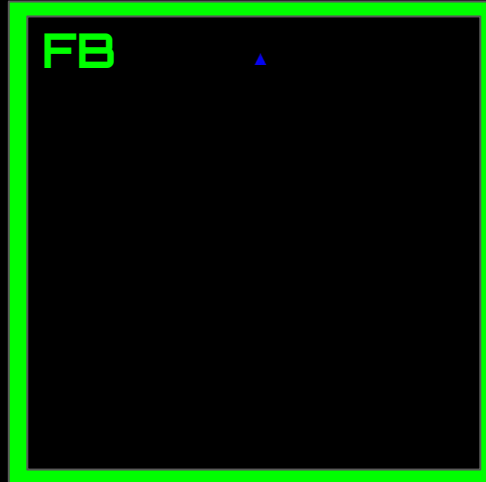
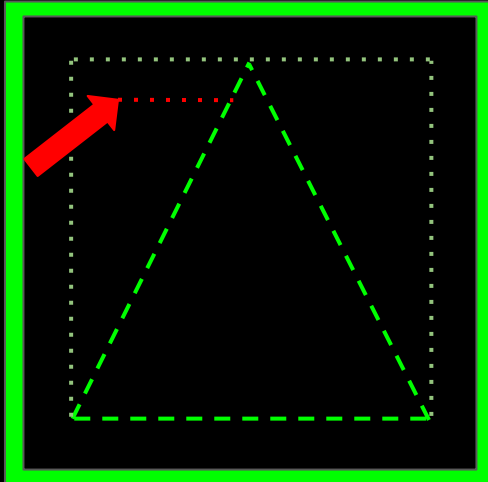
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

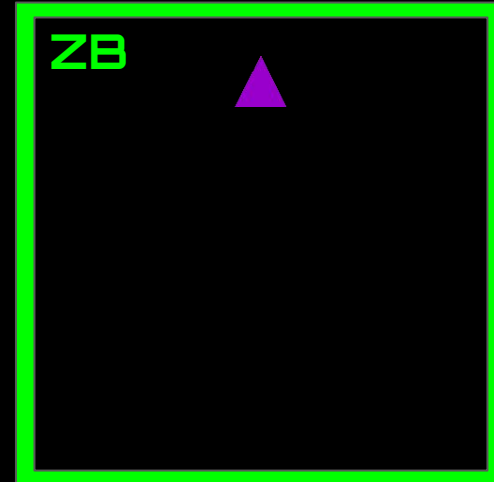
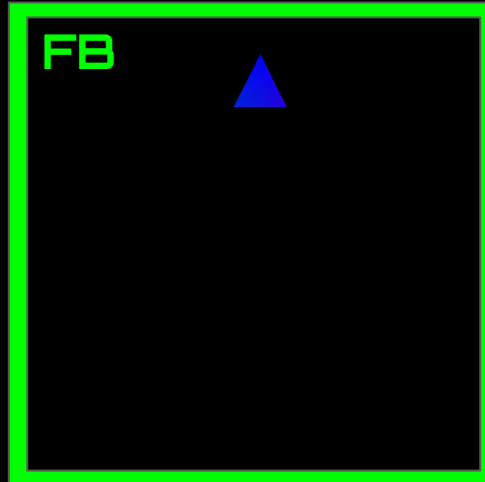
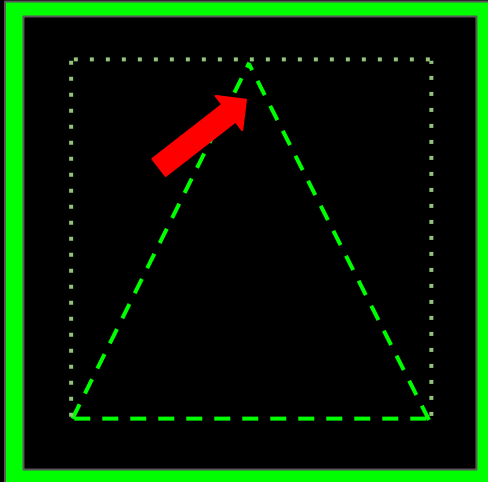
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

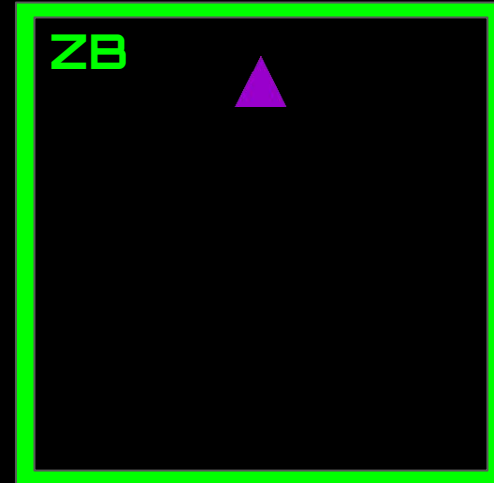
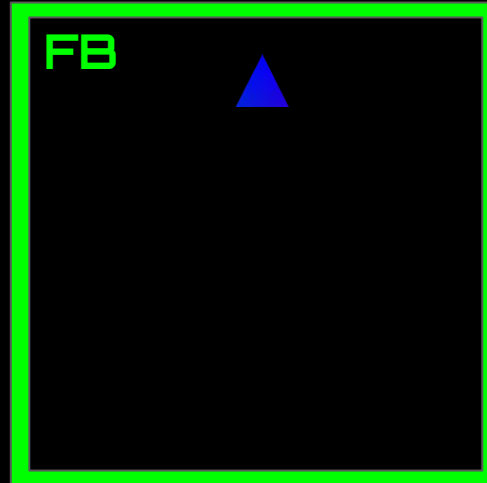
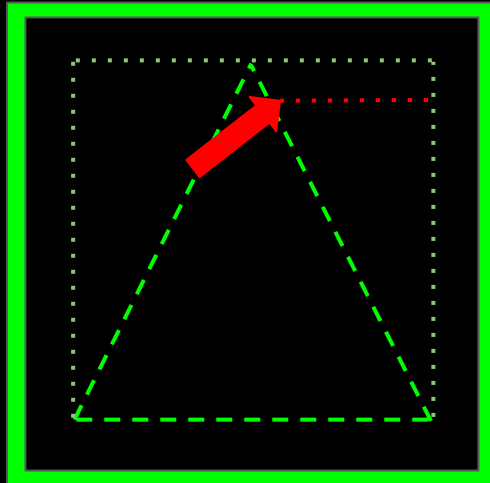
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

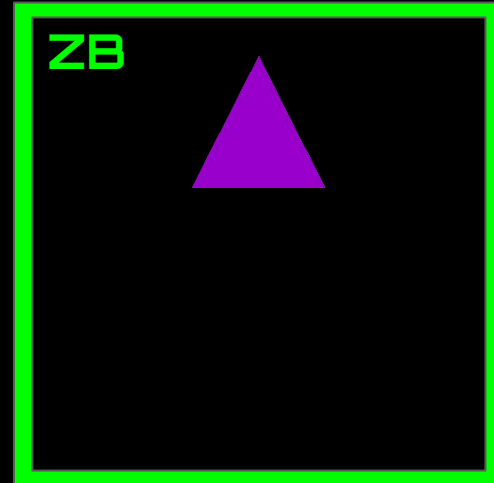
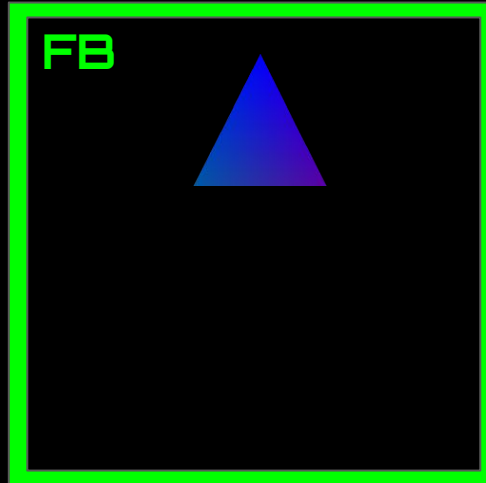
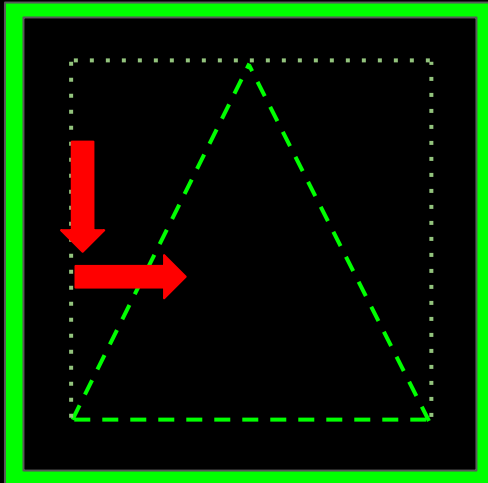
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

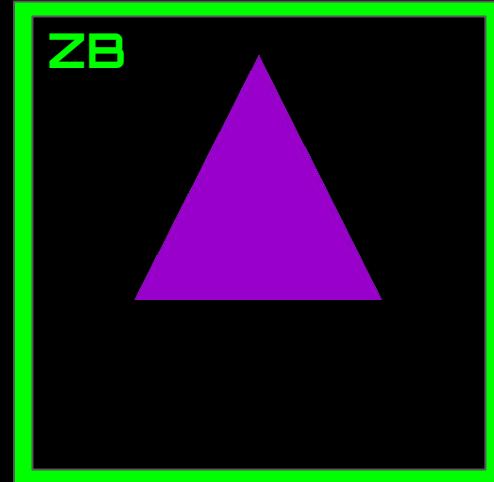
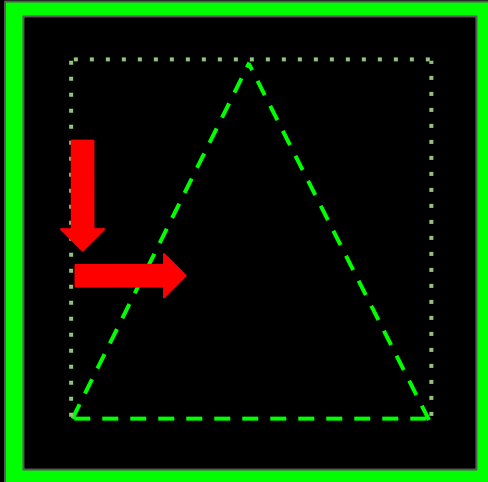
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

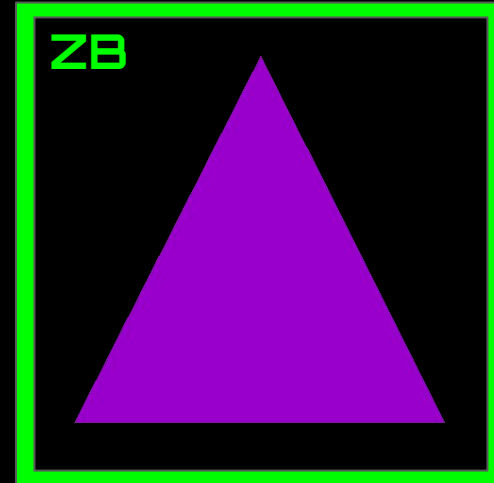
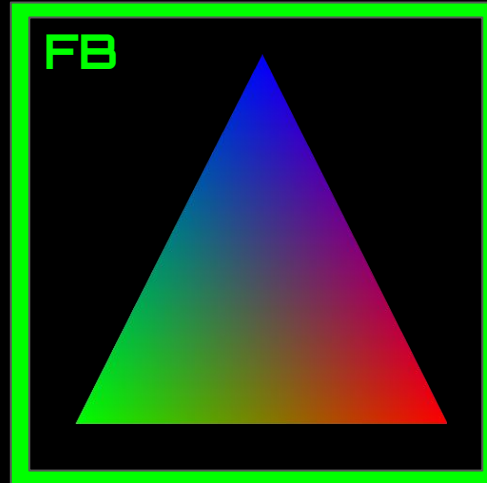
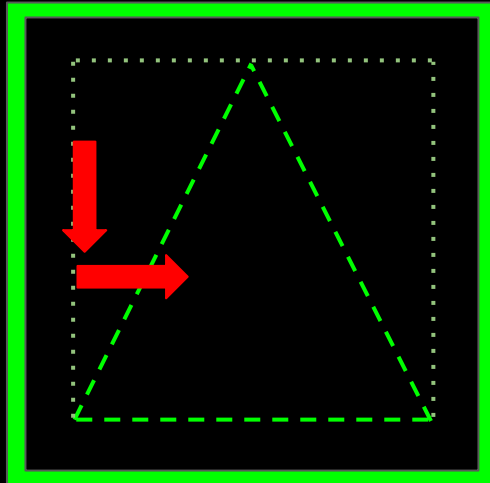
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

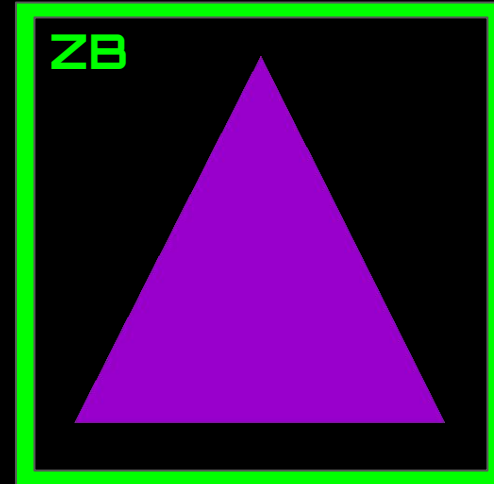
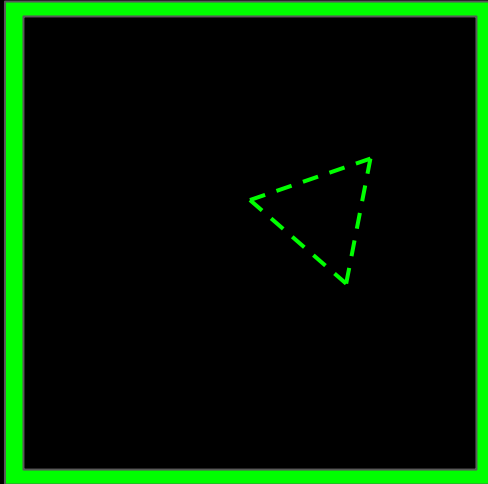
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

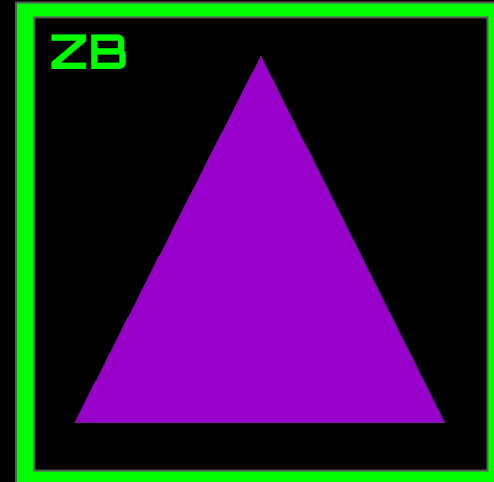
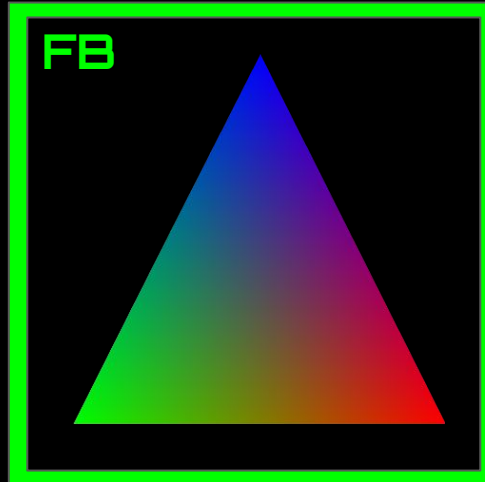
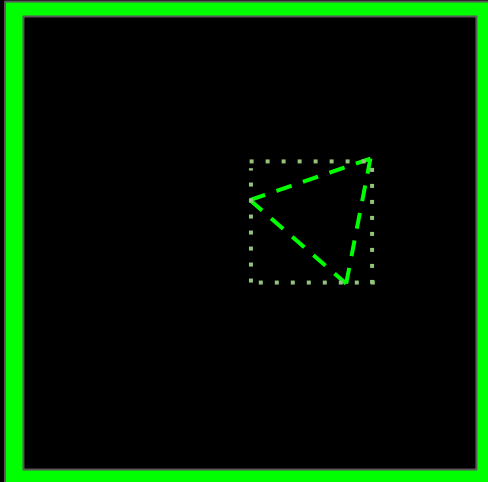
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

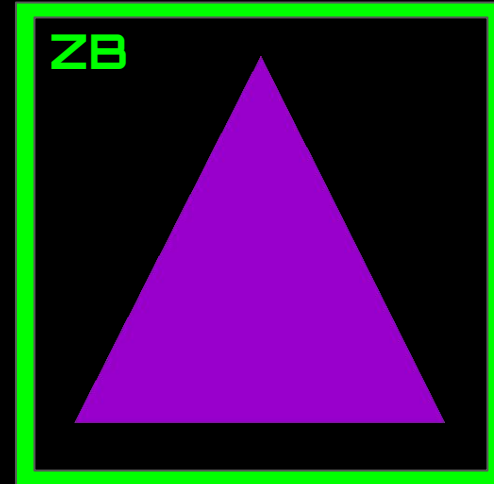
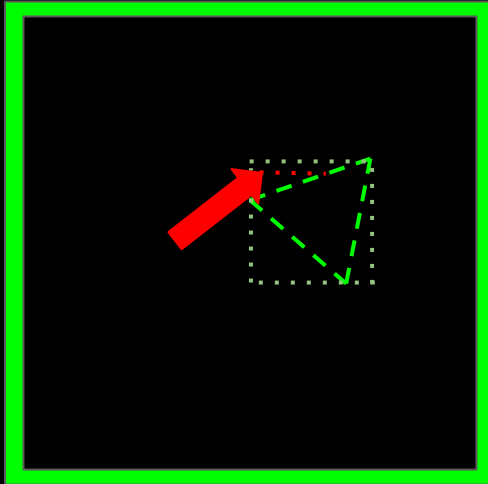
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

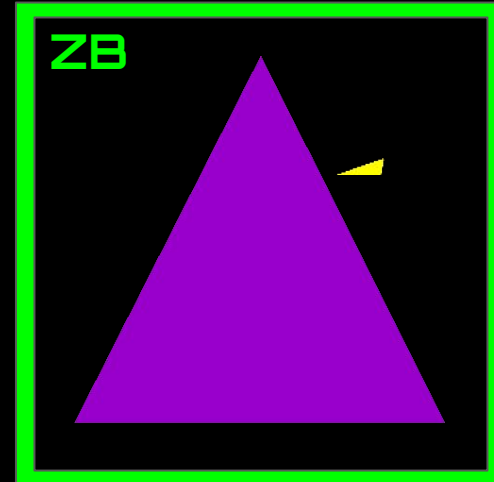
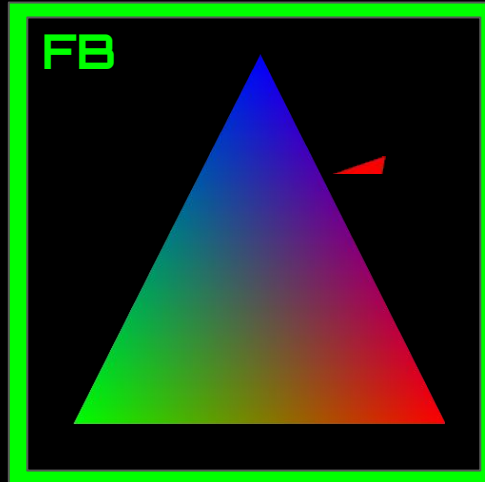
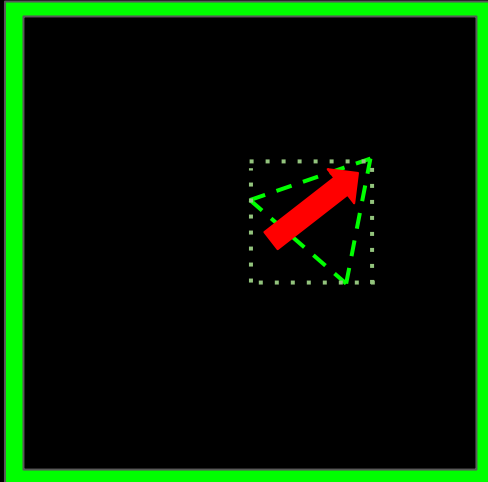
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

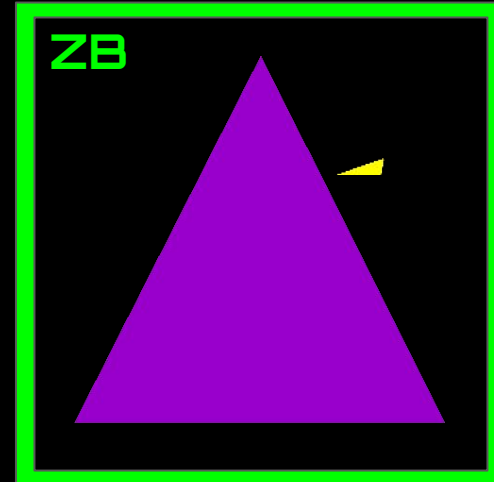
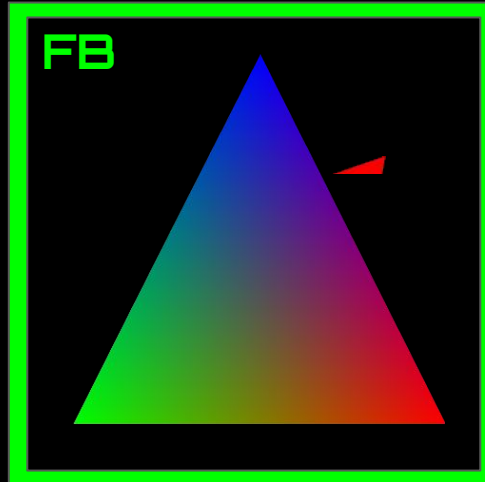
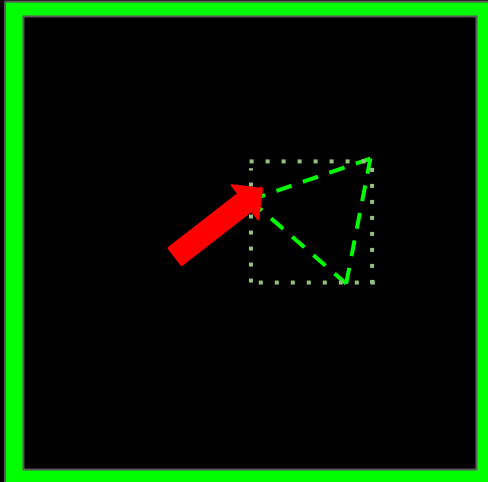
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

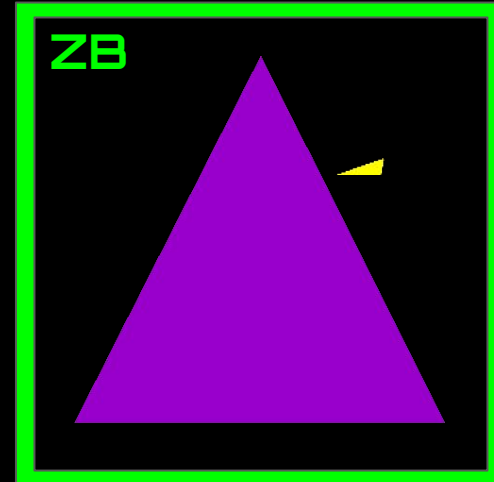
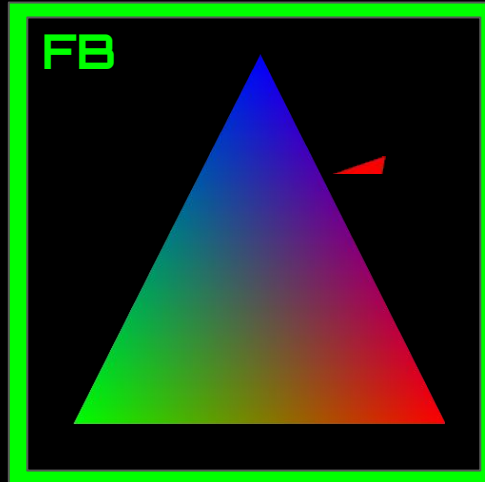
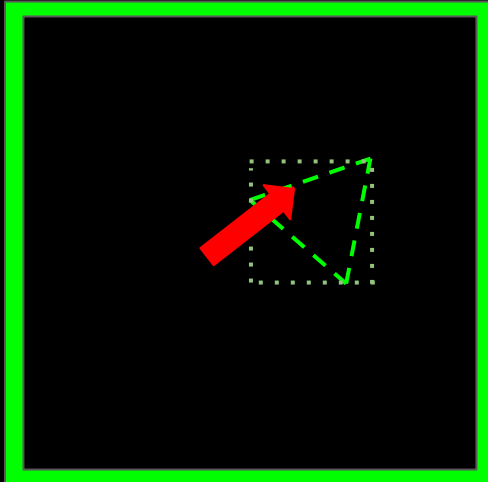
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

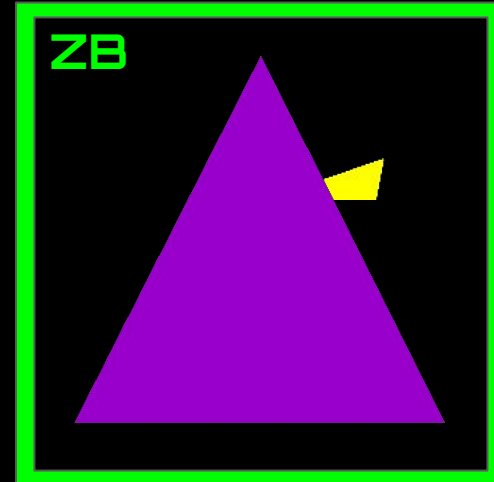
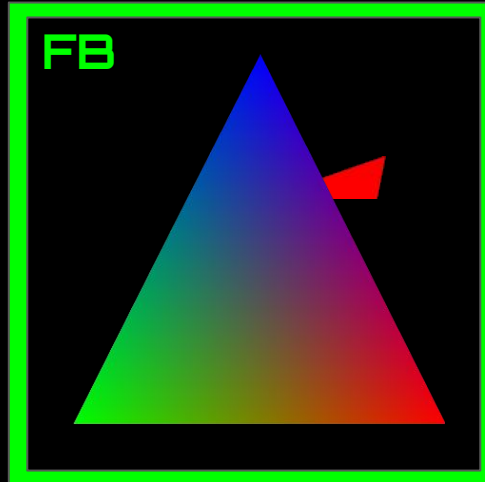
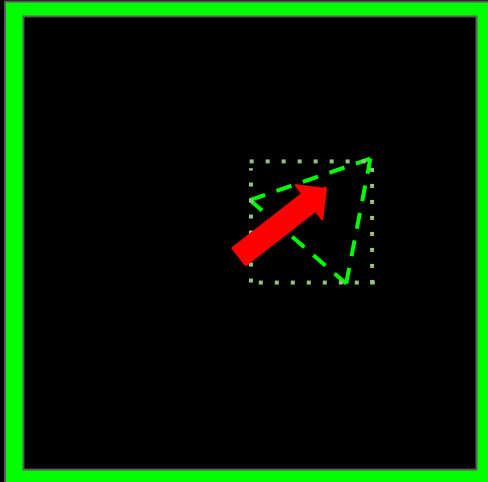
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

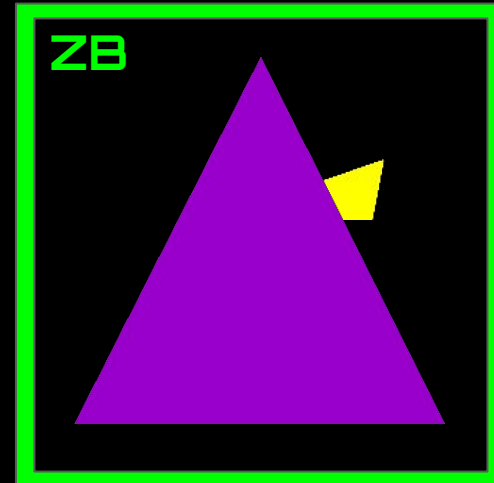
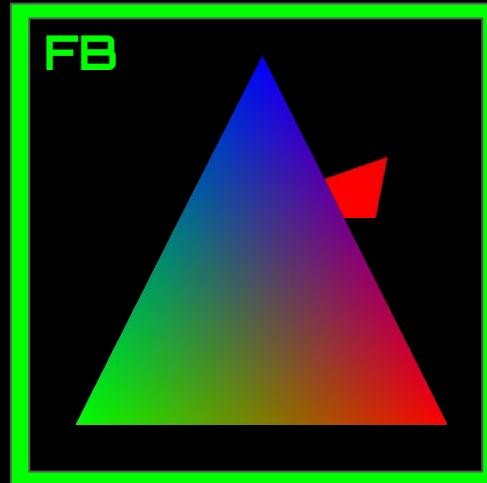
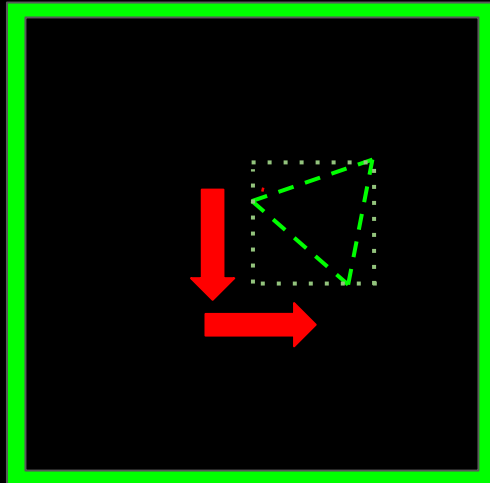
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

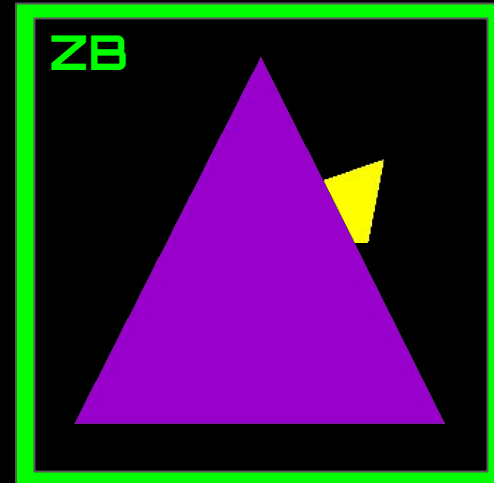
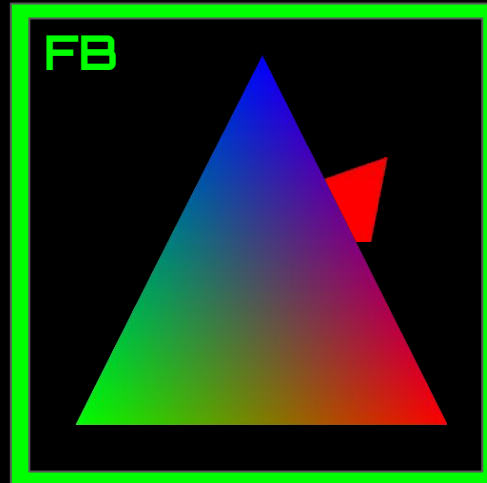
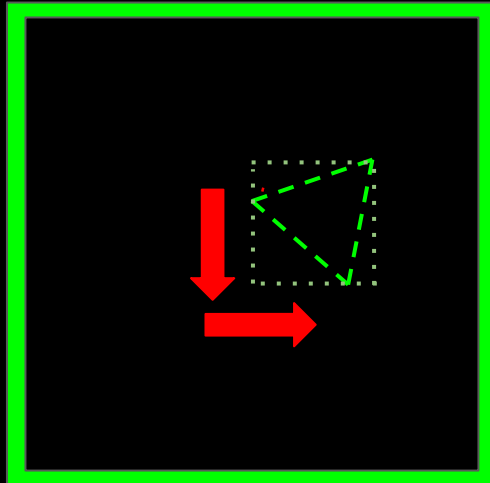
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.1. Rasterize - Method 1

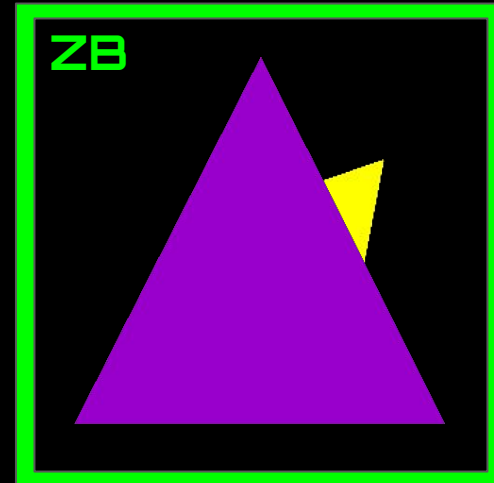
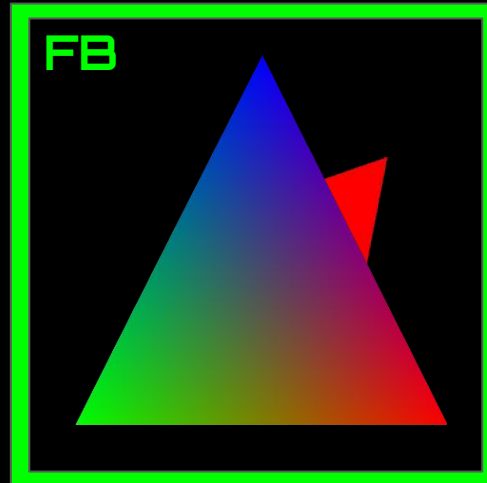
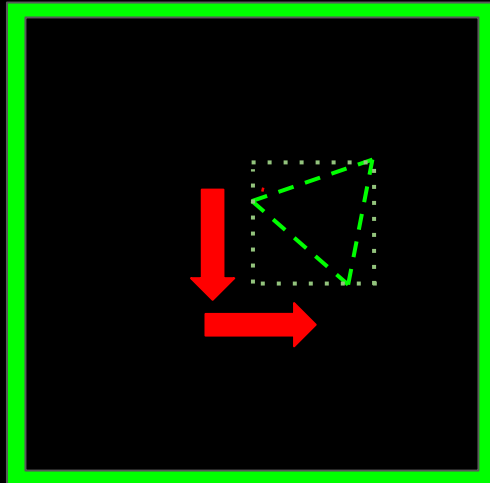
■ $z = 0$
■ $z = 1$



Rendering pipeline

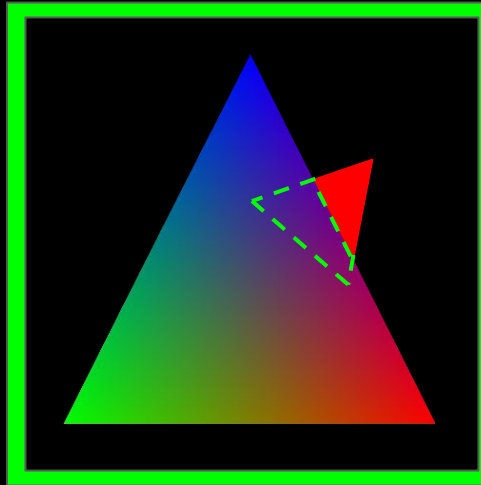
5.1. Rasterize - Method 1

■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2



$z = 0$

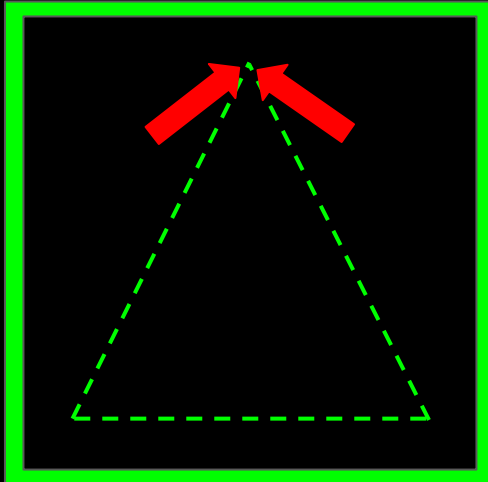


$z = 1$

Rendering pipeline

5.2. Rasterize - Method 2

■ $z = 0$
■ $z = 1$



FB



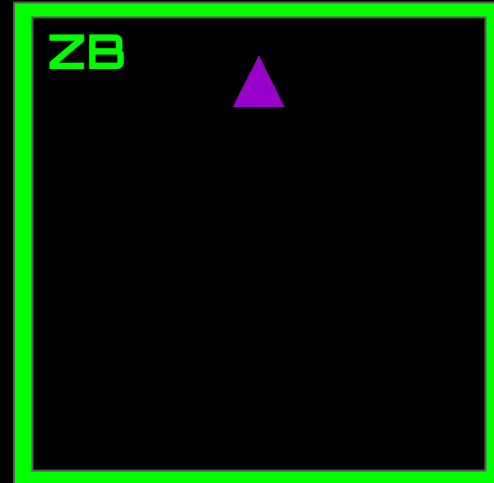
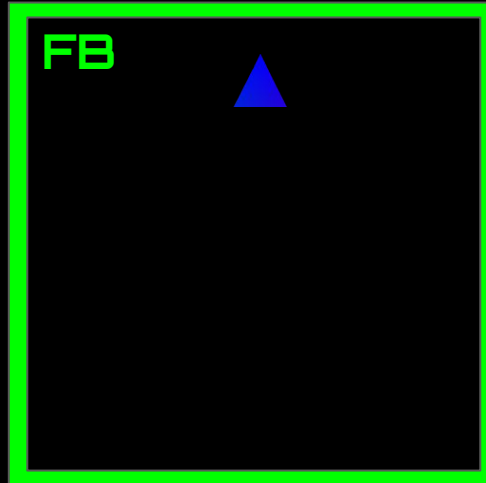
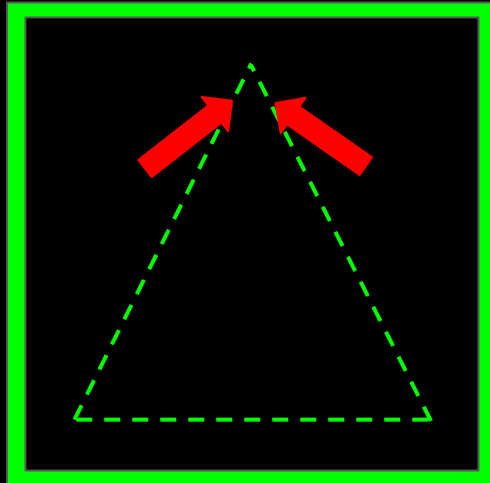
ZB



Rendering pipeline

5.2. Rasterize - Method 2

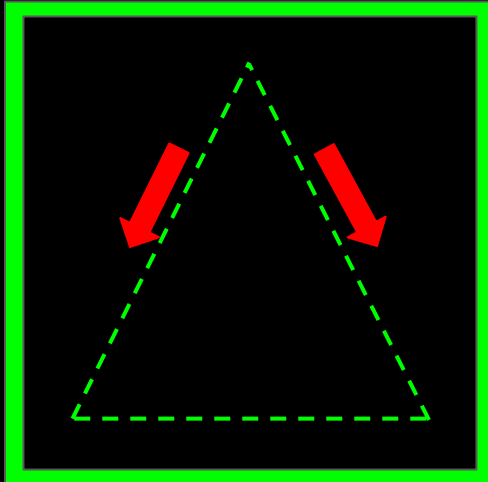
■ $z = 0$
■ $z = 1$



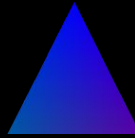
Rendering pipeline

5.2. Rasterize - Method 2

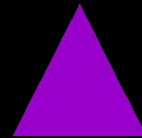
■ $z = 0$
■ $z = 1$



FB



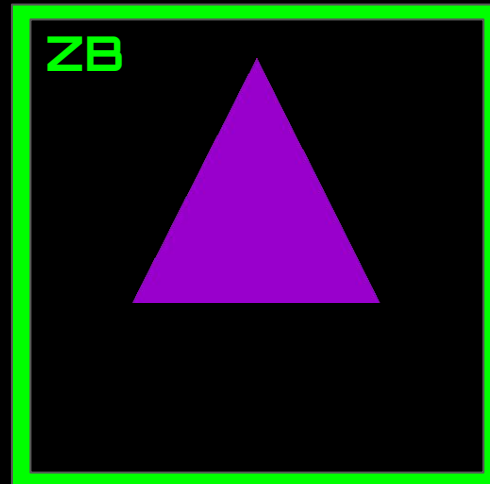
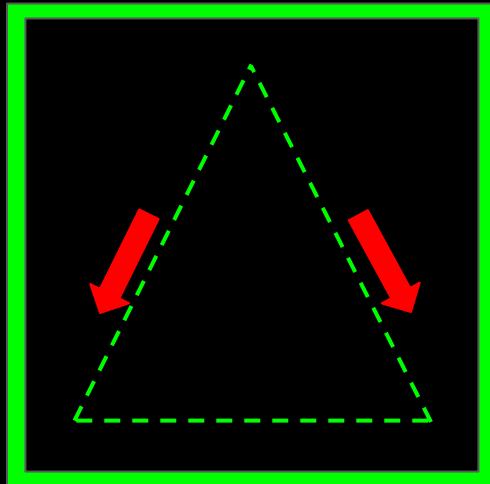
ZB



Rendering pipeline

5.2. Rasterize - Method 2

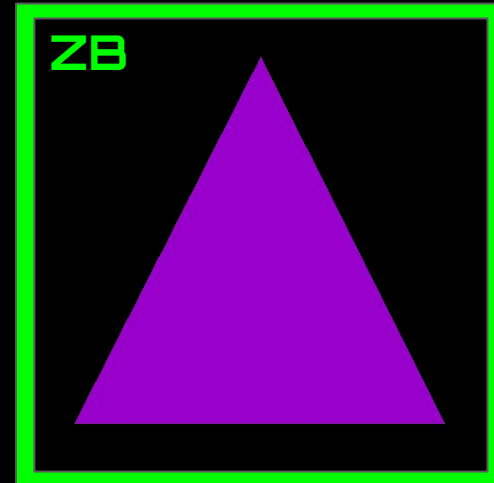
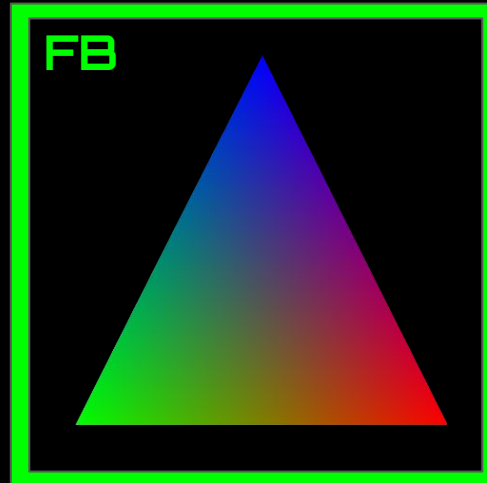
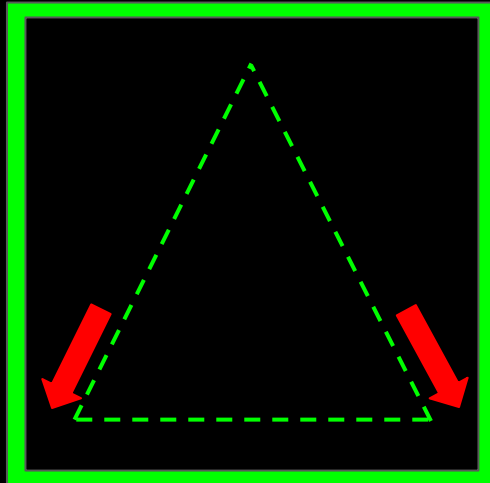
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

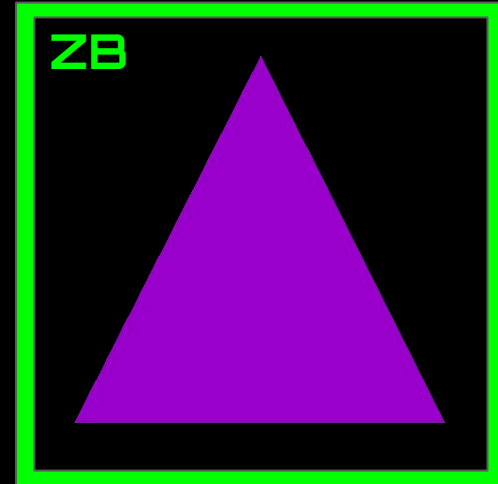
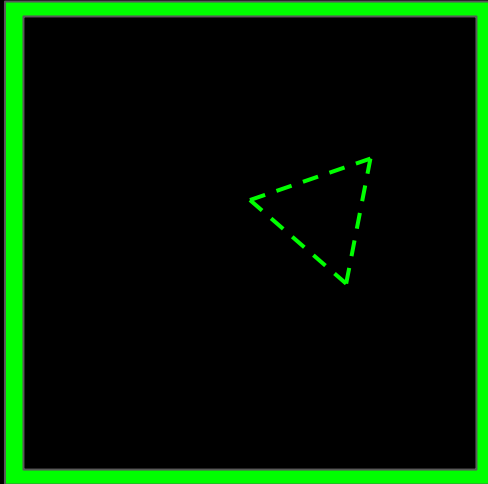
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

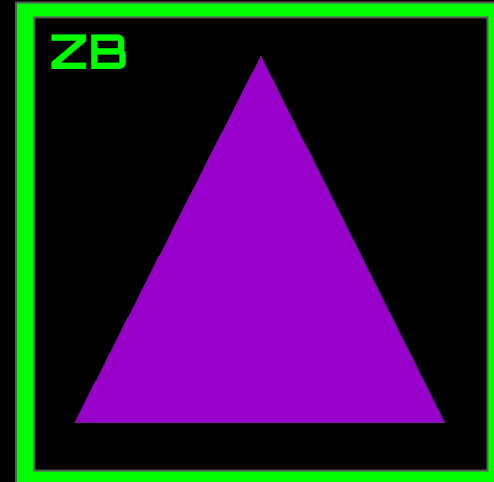
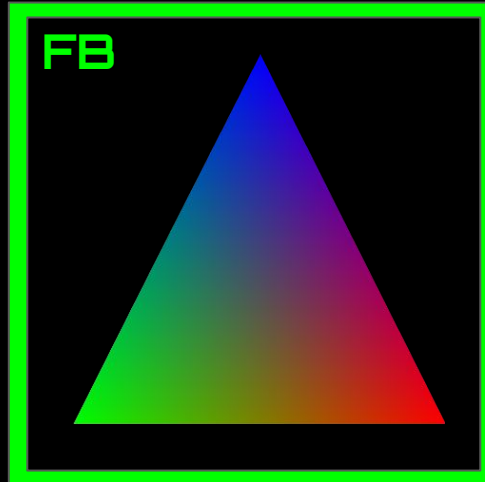
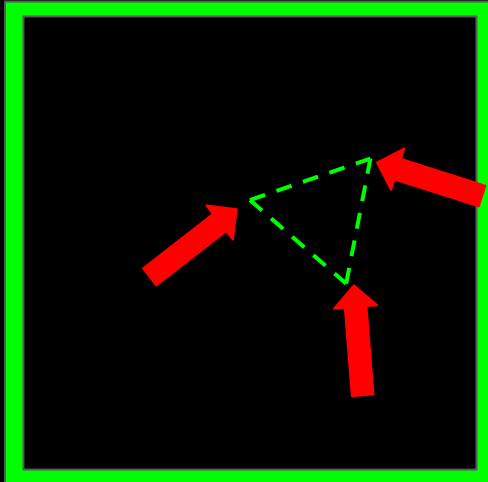
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

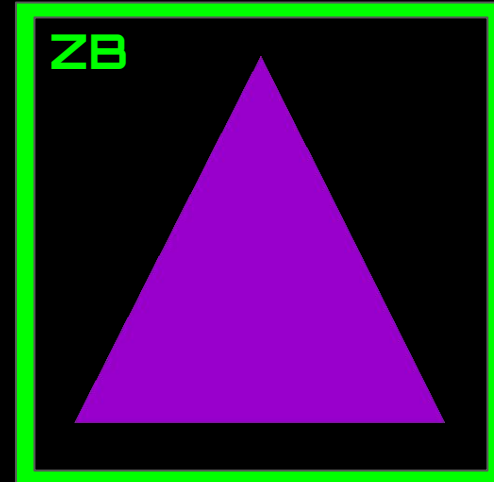
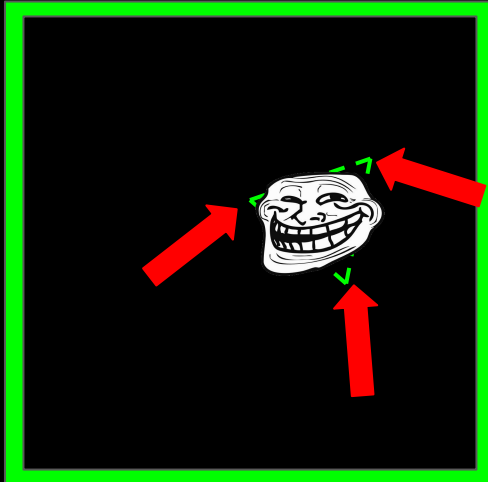
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

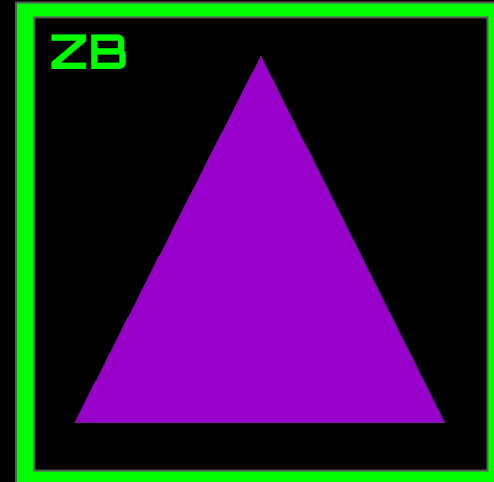
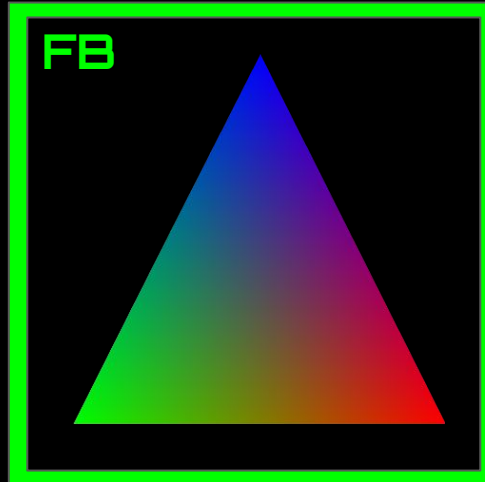
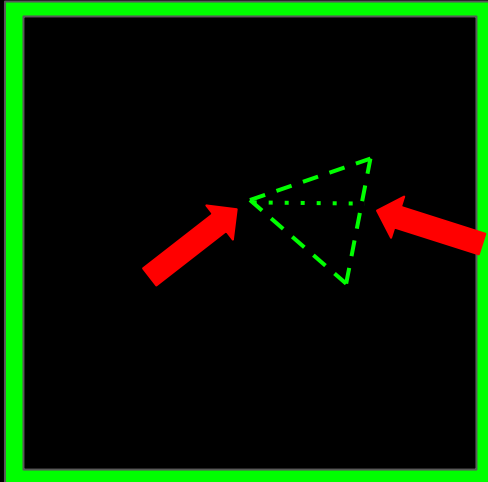
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

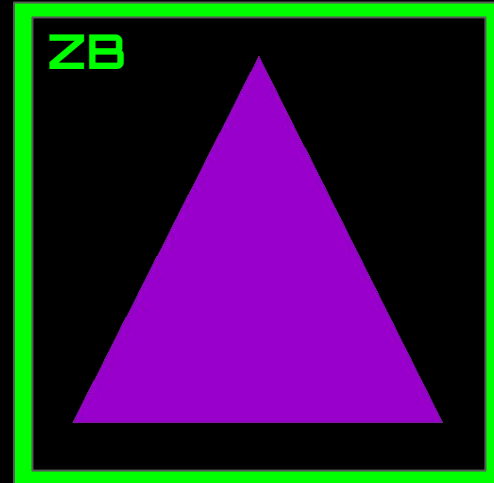
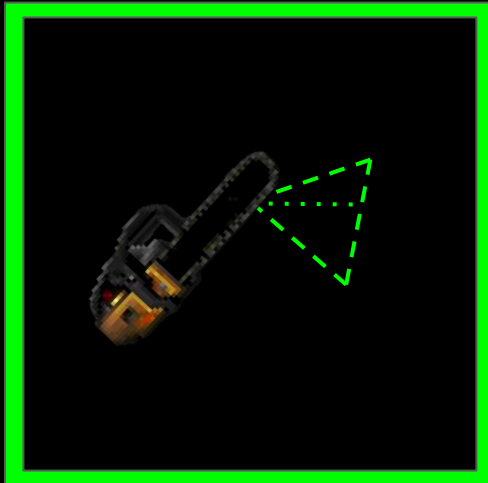
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

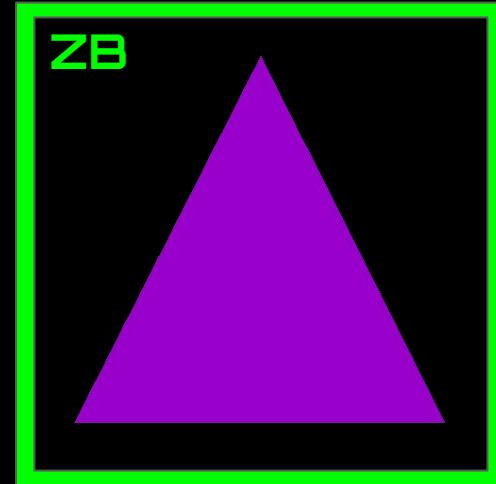
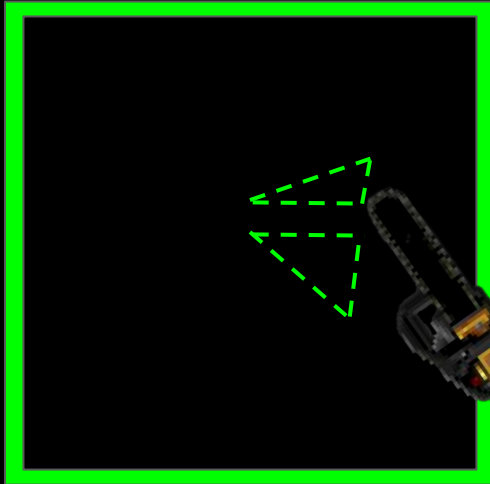
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

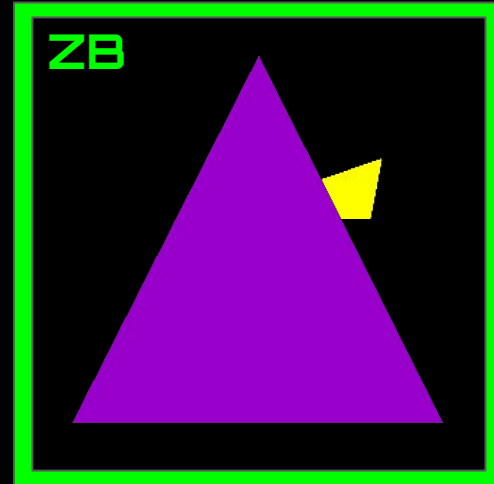
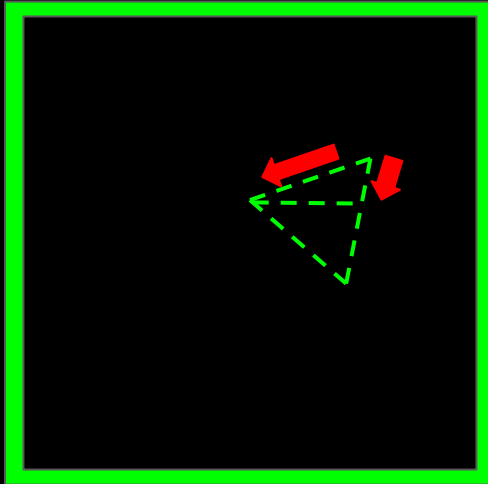
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

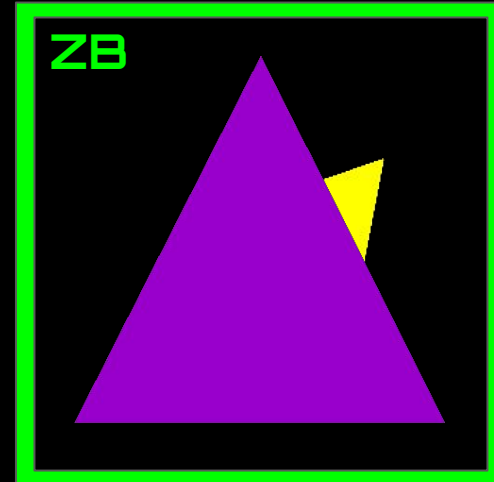
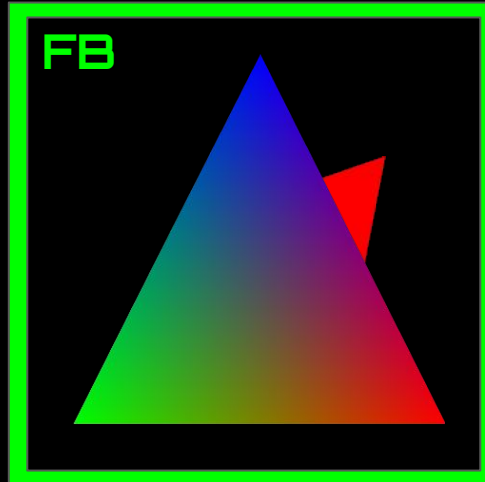
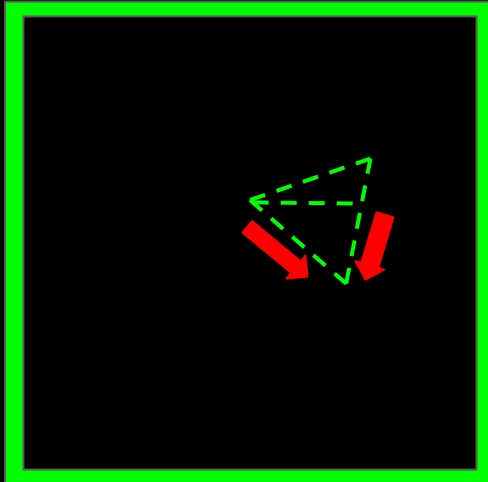
■ $z = 0$
■ $z = 1$



Rendering pipeline

5.2. Rasterize - Method 2

■ $z = 0$
■ $z = 1$



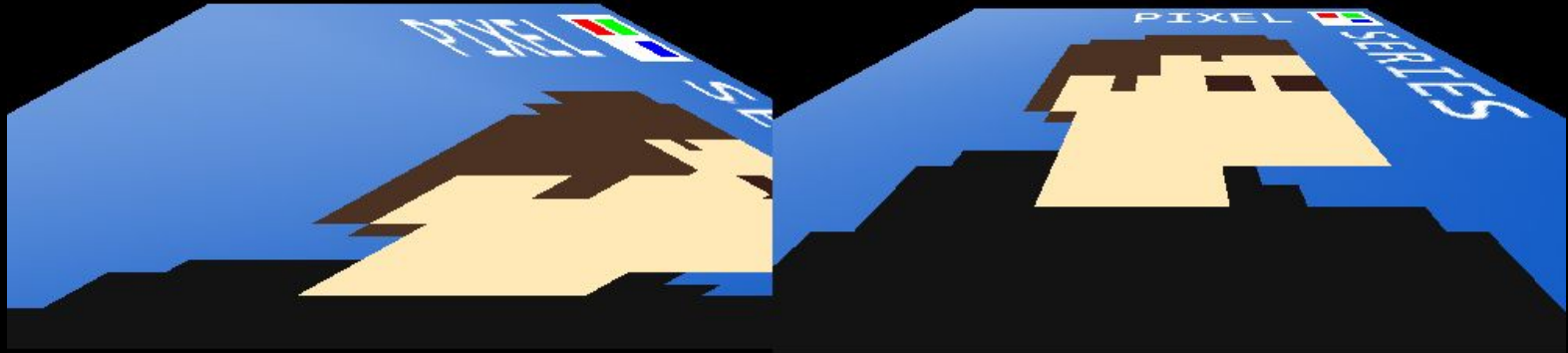
Rendering pipeline

5.3. Which method is better?

- Method 1 (CheckPixelInTriangle): Easier to implement, easier to parallelize
- Method 2 (Scanlines): A bit more complex to get right, higher performance when single threading (+20%)

Rendering pipeline

5.4. Perspective correct texture mapping



More info: https://en.wikipedia.org/wiki/Texture_mapping#Perspective_correctness

Want to know more?

My Software Renderer repo:

bitbucket.org/gamezer0/softwarerenderer

Amazing collection of related articles:

fgiesen.wordpress.com/2013/02/17/optimizing-sw-occlusion-culling-index/

FIN

THANKS!