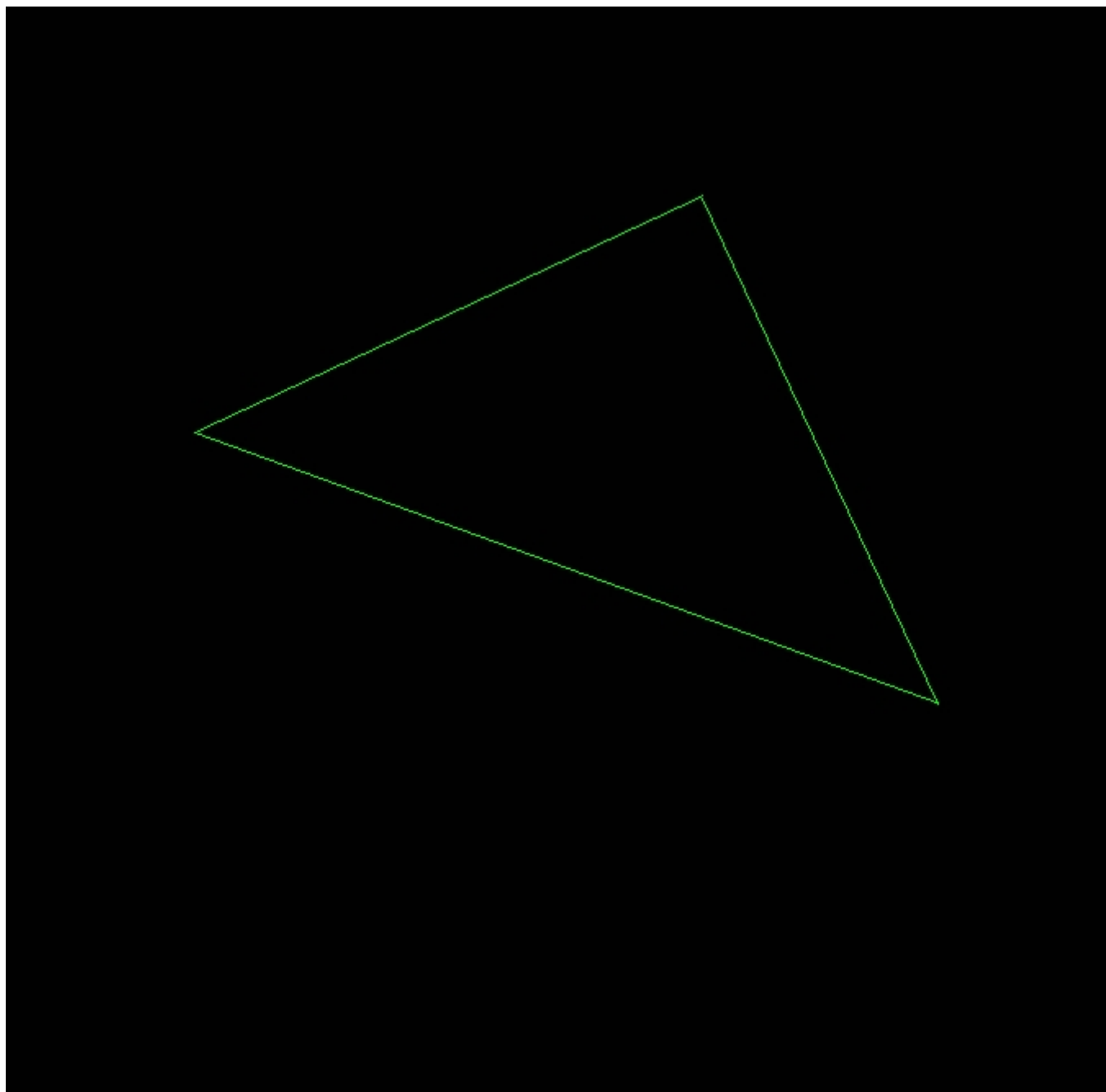


Games101 Report

张子聪 523030910065

Lab1

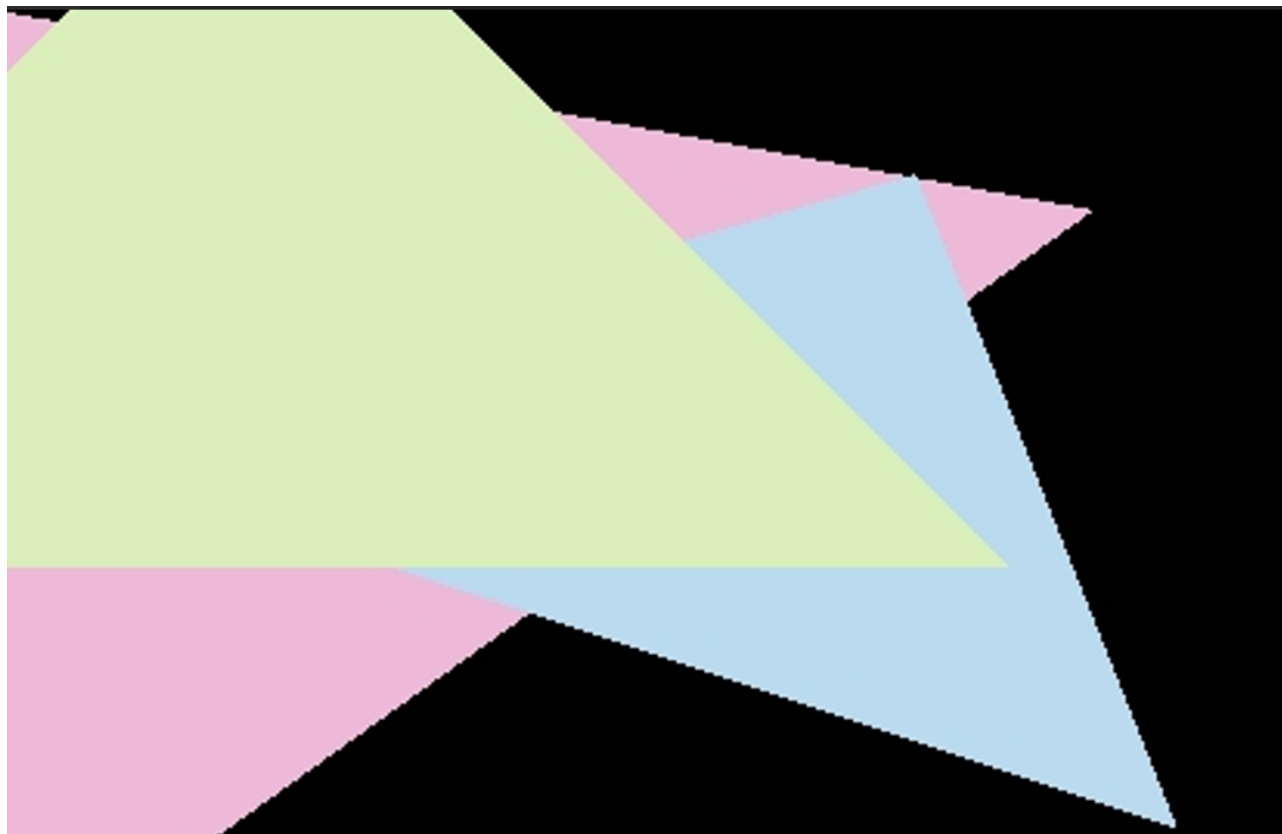


Lab2

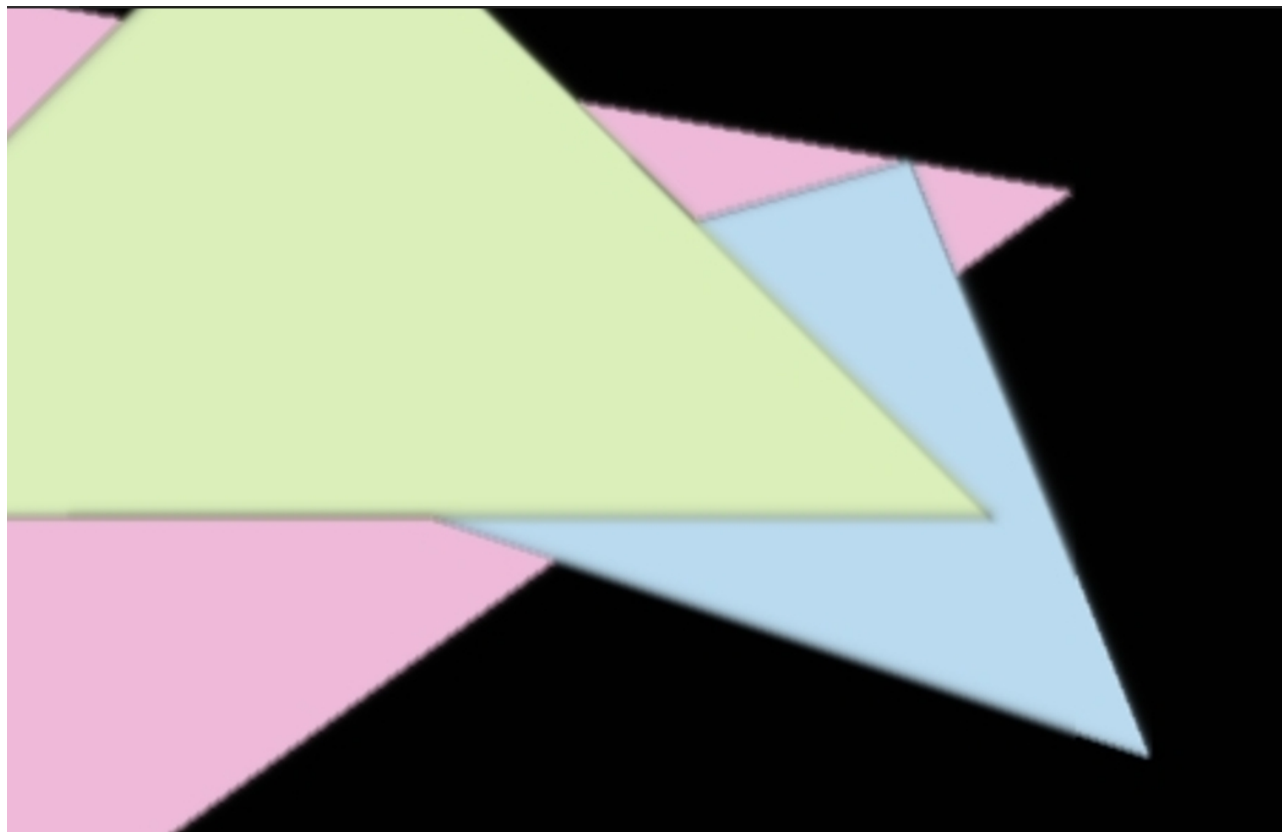
在进行光栅化的基础上，选择了 **MSAA** 和 **FXAA** 两种反走样方式来进行抗锯齿处理

效果展示：

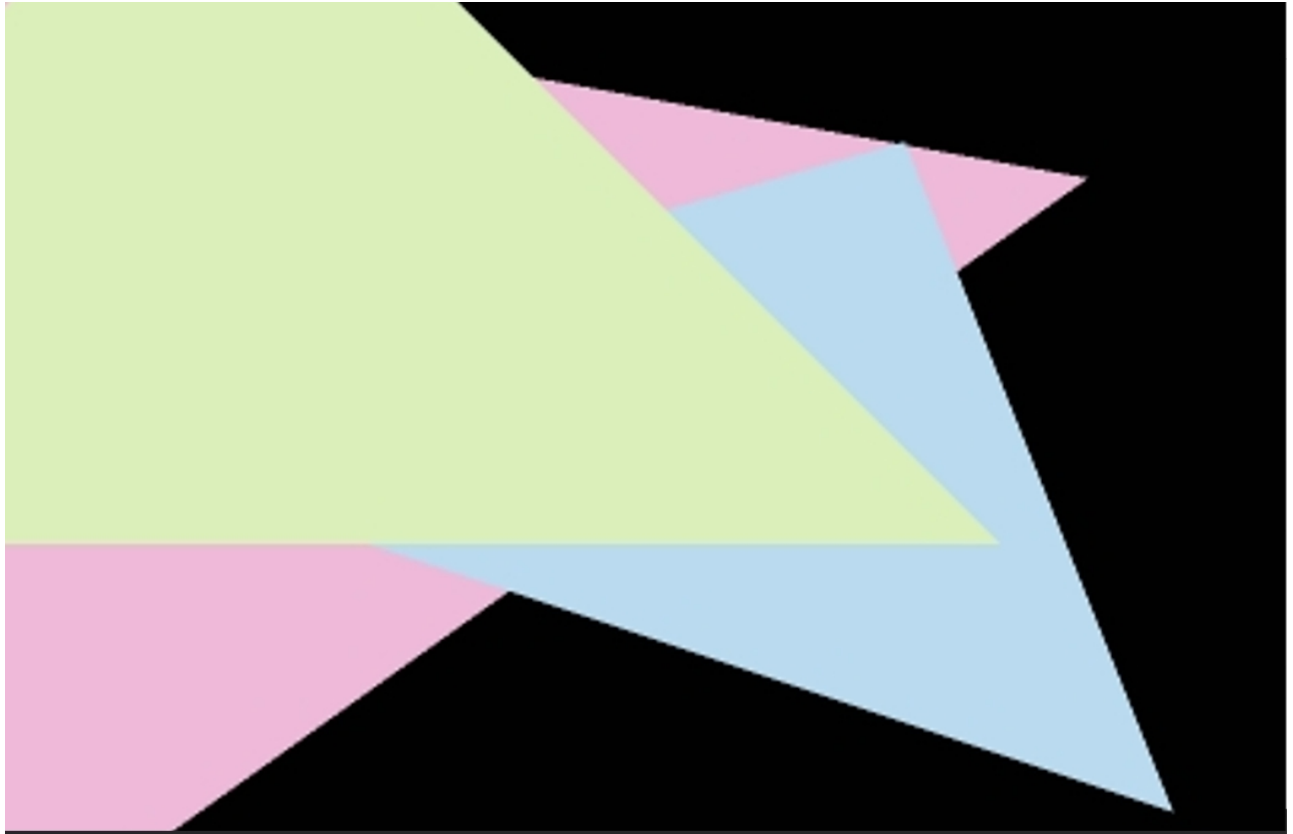
1. 未进行抗锯齿处理的情况 (NoAA) :



2. 使用 **FXAA** 方法进行抗锯齿处理 :



3. 使用 **MSAA** 的采样方法进行抗锯齿处理：



通过上面的图片可以看出

- 在不进行抗锯齿处理时（NoAA）在边界粉色和蓝色三角形上具有很明显的锯齿状纹路。
- **FXAA** 方法在处理边界上呈现出一种钝化模糊的感觉，以此来消除锯齿，效果比较明显
- **MSAA** 的采样方法能够很好的在原图的基础上消除锯齿，并且不会产生**FXAA**处理一样的模糊感

渲染时间：

不进行反走样（NoAA）的渲染平均用时：324.47ms

进行FXAA反走样的渲染平均用时：560.68ms

进行MSAA反走样的渲染平均用时：1.44s

这个结果和预期一致，NoAA因为不进行抗锯齿处理所以用时最低；**FXAA** 的方法是在原来渲染的基础上只处理了边界的锯齿，所以用时也不会很大；而**MSAA**的方法因为增大了采样量，在我的代码实现中对于每一个点，都采了周围的四个点进行处理，所以用时也近乎不进行反走样渲染的4倍。

所以如果对于渲染效果要求不高，可以接受**FXAA**处理的边界钝化感，那么**FXAA**其实是一种性能比较优越的反走样方法。

关于走样的分析

走样指的是用同一种采样频率，对不同频率的信号进行采样得到了同样的结果。这种采样可以是对空间中物体的采样，对时间的采样等。

体现在光栅器上主要是指光栅器对平面的采样点间隔比较大，可以看成一种比较低频的采样频率，这种采样不能很好的处理图形的边界情况，不能精准的采集到边界形状，所以产生了走样的现象，表现为图形的边界呈现锯齿状。

解决走样问题的方法主要有两种：

1. 提高采样频率，这也是最根本的方法，当采样频率足够高，采样点足够多的时候，自然能够更精准的采集到图形边界的信息。
2. 采用反走样的方法，例如本lab中使用的 **MSAA** 和 **FXAA** 方法等，通过卷积等操作处理掉高频信号之后再采样。

Lab3

1. Normal shader:



2. Blinn-Phong shader:



3. Texture shader:



4. Bump shader:



5. Displacement shader:

