TinyRenderer翻译

课0：

这是一个小型光栅化渲染器，你可以在这里找到它的源代码。

我的课程源码是开源的，阅读Wiki并尝试自己实现。只有你自己趟过所有tiny的坑之后，你才会明白它是怎么运行的。

如果你有任何问题，请不要犹豫，通过这个邮件和我联系(dmitry.sokolov@univ-lorraine.fr)

如果你是一个教授并且想将这个素材用于你的课堂教学，不需要作者授权，你可以任意使用它，只需要通过上述的邮件地址告知我一声，以便帮助我改进这门课程

在本系列文章中，我将通过写一个OpenGL的精简版本来给你展示它是怎么工作的。

令人称奇的是，我经常能遇见无法克服学习OpenGL/DirectX初期困难的人，为此，我准备了一个简短的课程，在学习本课之后你应该会得到一个有不错渲染效果的渲染器。

在这门课你不需要使用任何第三方库(尤其是图形管线方面的)就能渲染出如下图所示的图形。

提示：我们将使用软渲染渲染出一份渲染素材，而它将基本上重现OpenGL库的效果。我不是想取展示怎么给OpenGL写应用而是告诉你OpenGL是怎么工作的。我确信，重点不在于怎么去表现3D效果而是去明白这其中的原理。

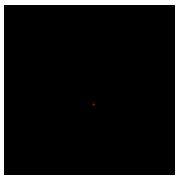
我会用大约500行代码去实现它，你大概需要10–20个工程时去制作这个渲染器。我们用包含多边形和图片素材的测试文件作为输入，渲染模型作为输出。这个项目不使用管线接口，最后生成的是一张图片。

既然目的是尽量减少外部依赖，我只会给你一堂课的时间去学会使用TGA文件。它是支持RGB/RGBA/以及二值化格式的最简单文件类型之一。所以开始本堂课之前，我们将学会怎么使用这种图片类型。学会之后，将它做成可调用的函数(包括加载和保存)以便在每一堂课开始之前，你能够正确设置每一个像素的颜色。

由于没有现成能画线和三角形的函数，我们需要手动去实现它们。我会提供给你，将和你一起写的源码。但是我不推荐你去使用，这样做没有意义。完整源码能够在github上获得，或者你可以直接在这里找到。

```
#include "tgaimage.h"
const TGAColor white = TGAColor(255, 255, 255, 255);
const TGAColor red   = TGAColor(255, 0,   0,   255);
int main(int argc, char** argv) {
        TGAImage image(100, 100, TGAImage::RGB);
        image.set(52, 41, red);
        image.flip_vertically(); // i want to have the origin at the left bottom corner of the image
        image.write_tga_file("output.tga");
        return 0;
}
```

output.tga应该看起来像这样



Teaser：一些渲染后的例子