James’ Guide to Software Rendering

Over the years I have tried to make a whole host of 3d apps and, because I’m an engineer at heart, I wondered how 3d graphics work. I’ve even made a few basic software renderers over the years. At the time I never really understood why the various tricks that created the pretty shapes on the screen worked. That was when I was a teenager.

For some reason I’ve come back to this over the last year. This time round I spent a while really digging into the theory. This is my effort to explain what I’ve learned.

|  |
| --- |
| **Buyer Beware**  I am not a mathematician. I am a hobby programmer whose main job is being a doctor. Take what you read below with appropriate caution! |

# My Assumptions

1. You are a programmer, preferably one who is familiar with C/C++
2. You remember your GCSE (high school?) maths; preferably a bit of A-level maths too
3. You have a bit of experience with simple 3d graphics, probably with OpenGL or Direct X

If that’s not you, you might find this a bit difficult. Sorry, I’m not sure what I can do about that.

# The Articles