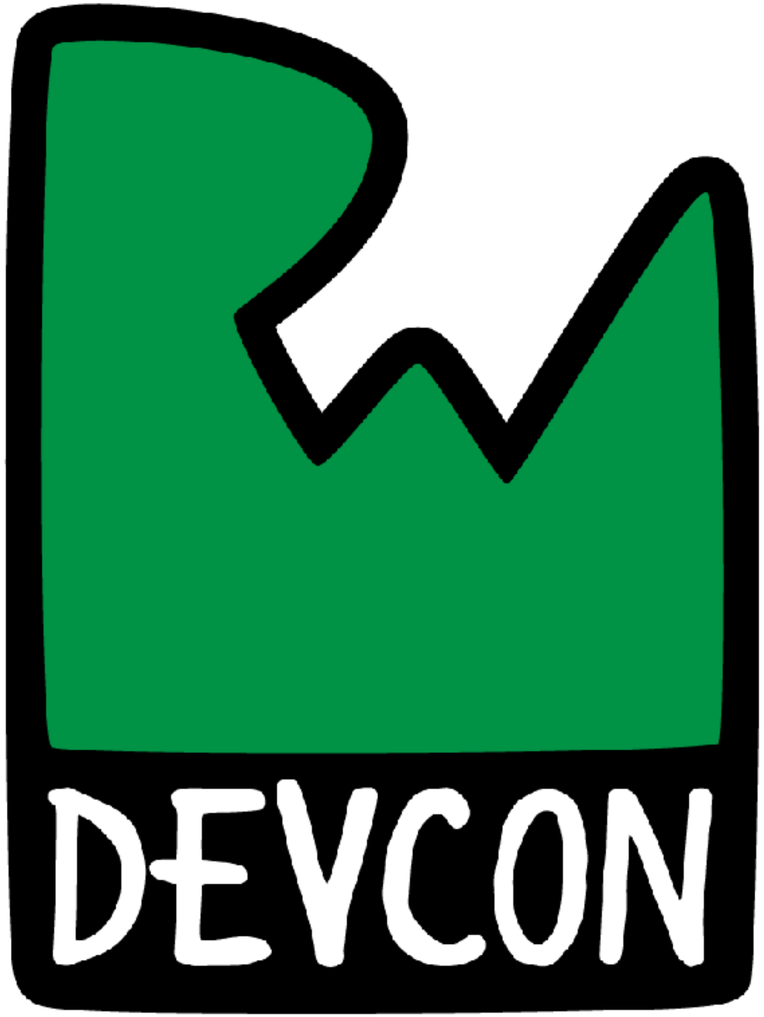
Session 16: Integrating Metal Shaders with SceneKit



Introduction

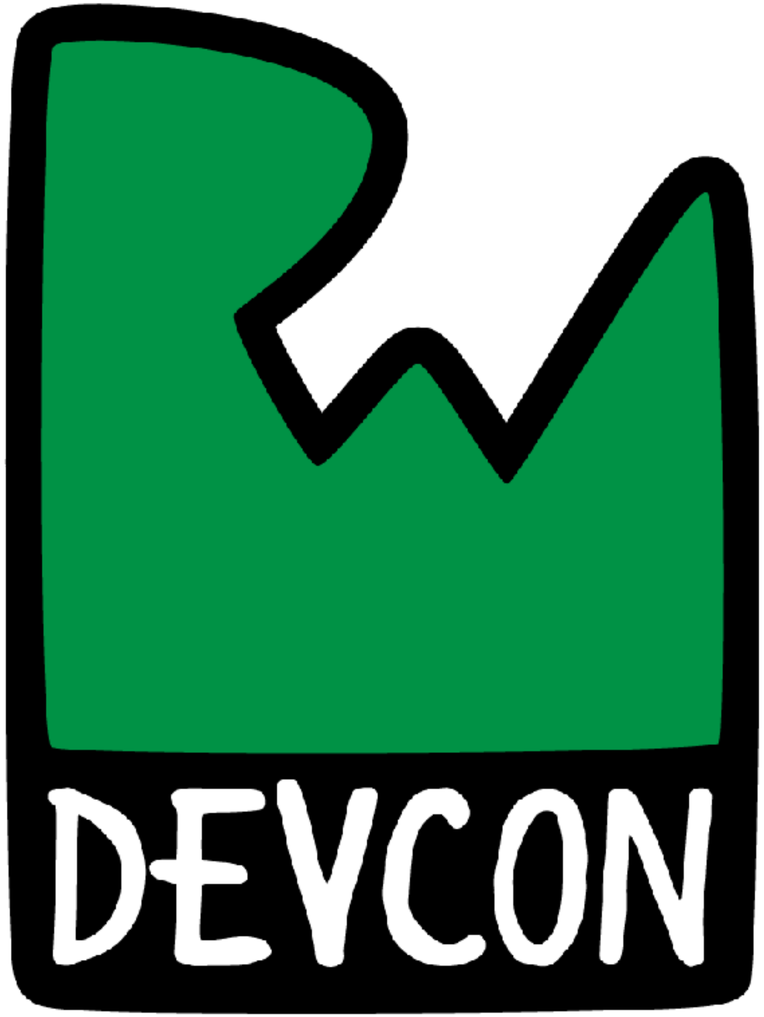
What We Are Doing Today

- Overview of SceneKit and the rendering pipeline
  - Simple Metal shader
- More Complex Metal Shader using code injections
  - Overview of Metal Shading Language and math concepts
- Metal Shader using SCNProgram

Warning!

Metal code does *not* work in the Simulator. In order to test your Projects you will need to build to a physical device!!

Session 16: Integrating Metal Shaders with SceneKit



Scene Kit and Metal

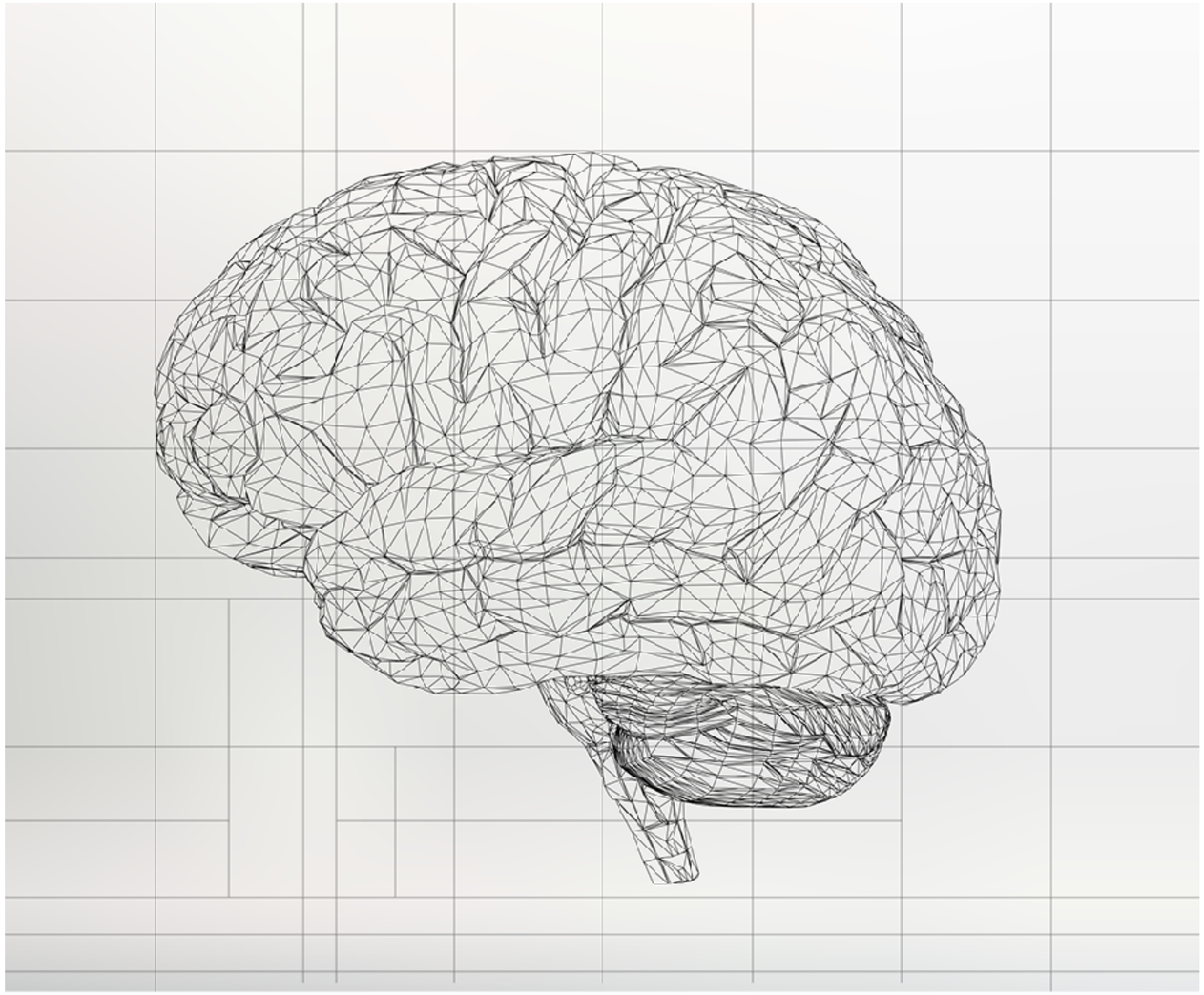
What is Metal?

- GPU programming framework introduced in 2014
- Used for graphics rendering and general purpose computational operations
- Goes “down to the Metal” by giving the programmer complete control of the rendering pipeline.

What is Metal?

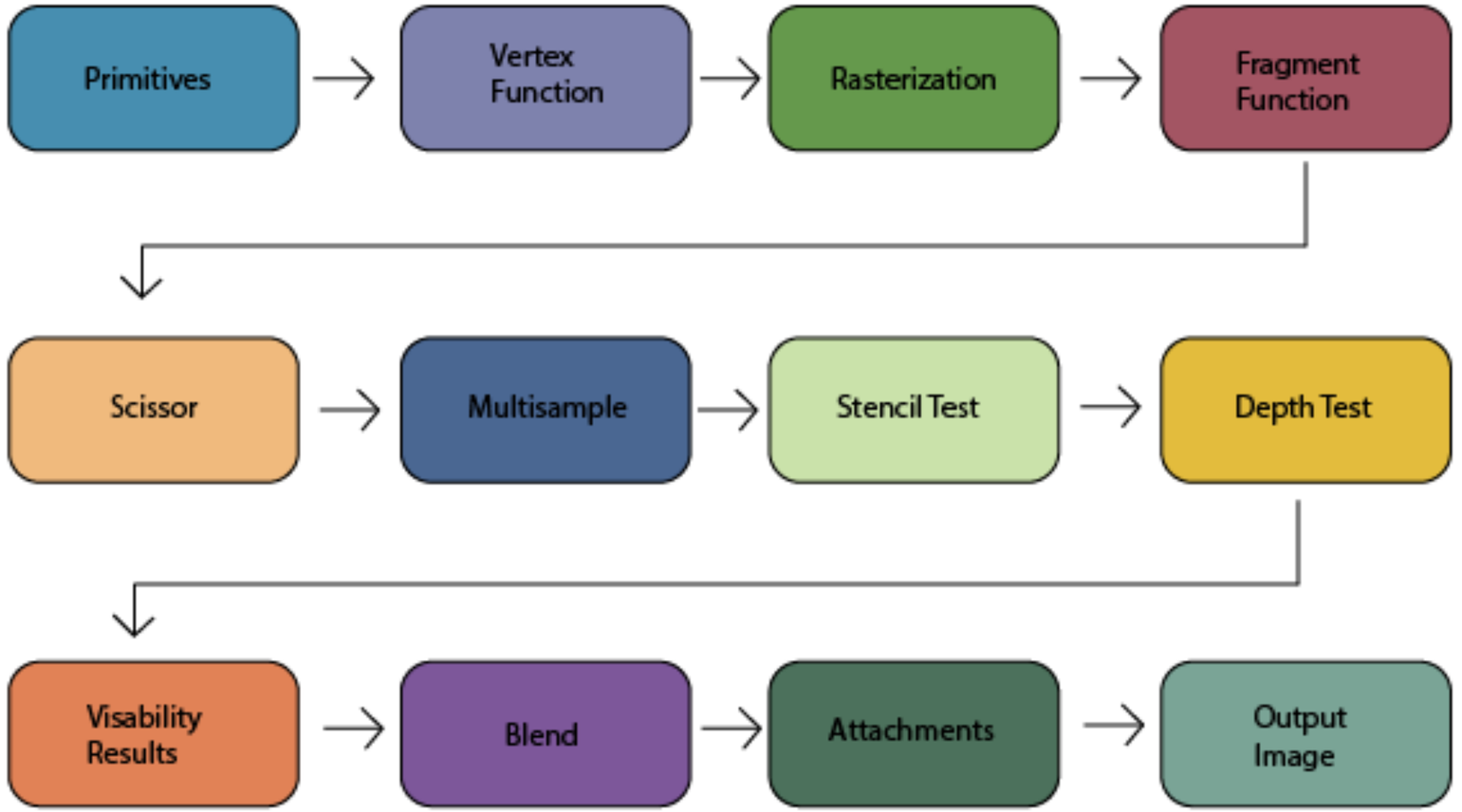
- But sometimes it’s not necessary to go completely down to the metal. There are a lot of common operations that have boiler plate code that can be abstracted away.
- What do you do if you don’t want to do all that work all the time but want to drop down every once and a while?

How Objects are Assembled



```
# Blender v2.65 (sub 0) OBJ File
# www.blender.org
o teapot.005
v -0.498530 0.712498 -0.039883
v -0.501666 0.699221 -0.063813
v -0.501255 0.717792 0.000000
v -0.624036 0.711938 -0.039883
v -0.526706 0.651362 -0.039883
v -0.508714 0.682112 -0.071712
v -0.622039 0.698704 -0.063813
v -0.624834 0.717232 0.000000
v -0.498530 0.712498 0.039883
v -0.638129 0.287158 0.000000
v -0.517593 0.664661 -0.063813
v -0.534329 0.646030 0.000000
v -0.614850 0.651067 -0.039883
v -0.616848 0.664299 -0.063813
v -0.619445 0.681503 -0.071790
v -0.741245 0.707456 -0.039883
v -0.744483 0.712577 0.000000
v -0.624036 0.711938 0.039883
v -0.501667 0.699221 0.063813
v -0.622039 0.698704 0.063813
v -0.712095 0.661370 -0.063813
v -0.733150 0.694655 -0.063813
v -0.741245 0.707456 0.039883
v -0.733150 0.694655 0.063813
v -0.631184 0.277569 -0.039883
v -0.526706 0.651362 0.039883
v -0.614053 0.645774 0.000000
```

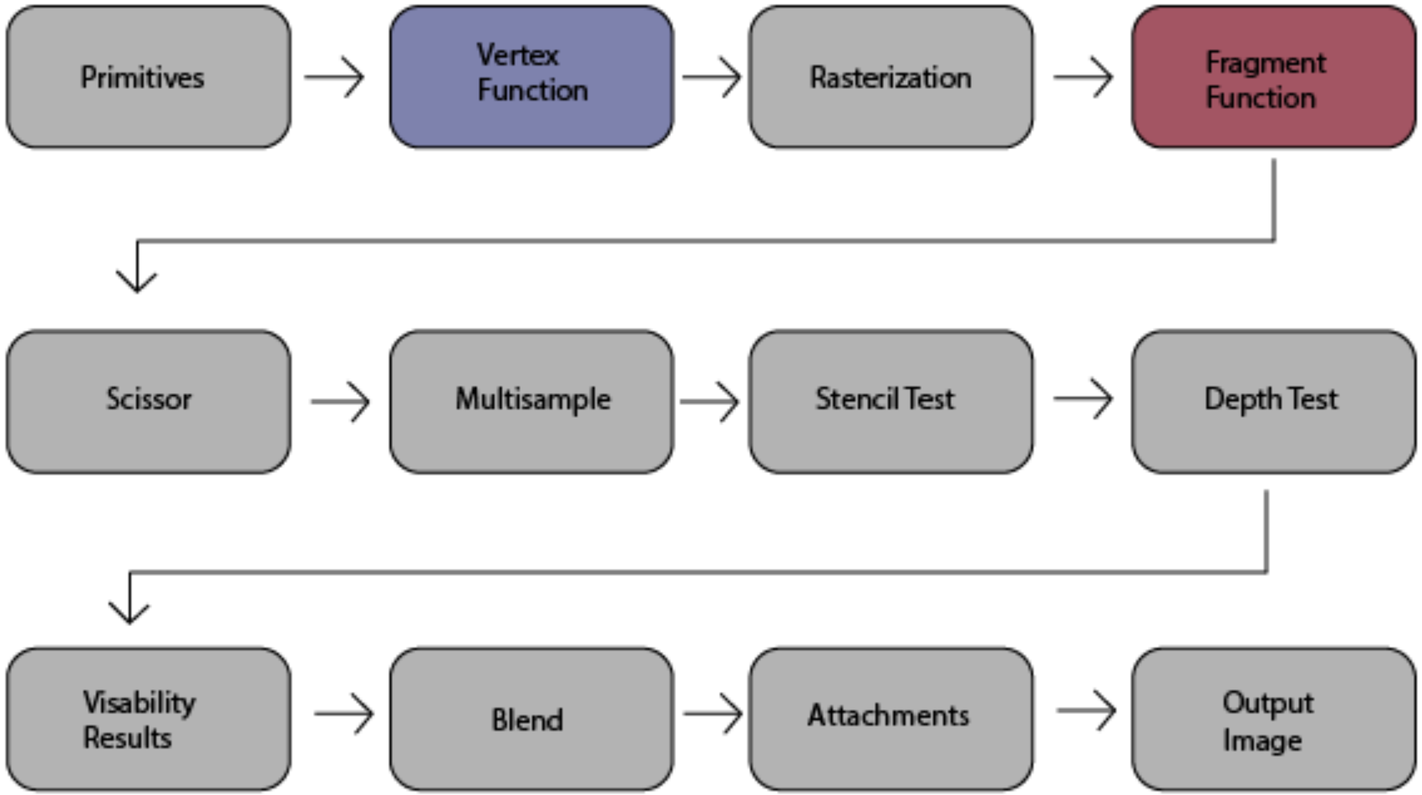
Rendering Pipeline



SceneKit

- Introduced in 2012. Was Mac only.
- Brought over to iOS in 2014
- 3D game framework similar to SceneKit and Unity.
- Wrapper around all the scary GPU functionality to make it much easier to do common things for 3D graphics and games.
- Intended as the starting point for 3D graphics rather than Metal/OpenGL.

Rendering Pipeline

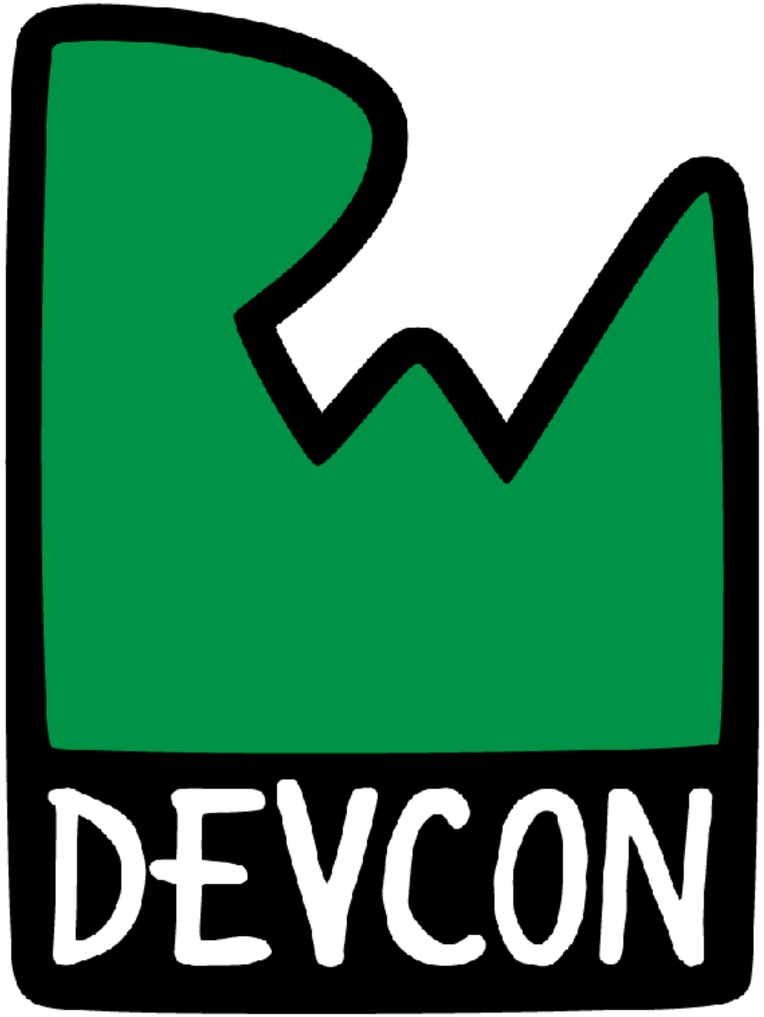


Demo 1

- Create a basic SceneKit box
- Use shader modifiers to inject some Metal code into the pipeline
- Shade the box a single color

Demo 1

Session 16: Integrating Metal Shaders with SceneKit



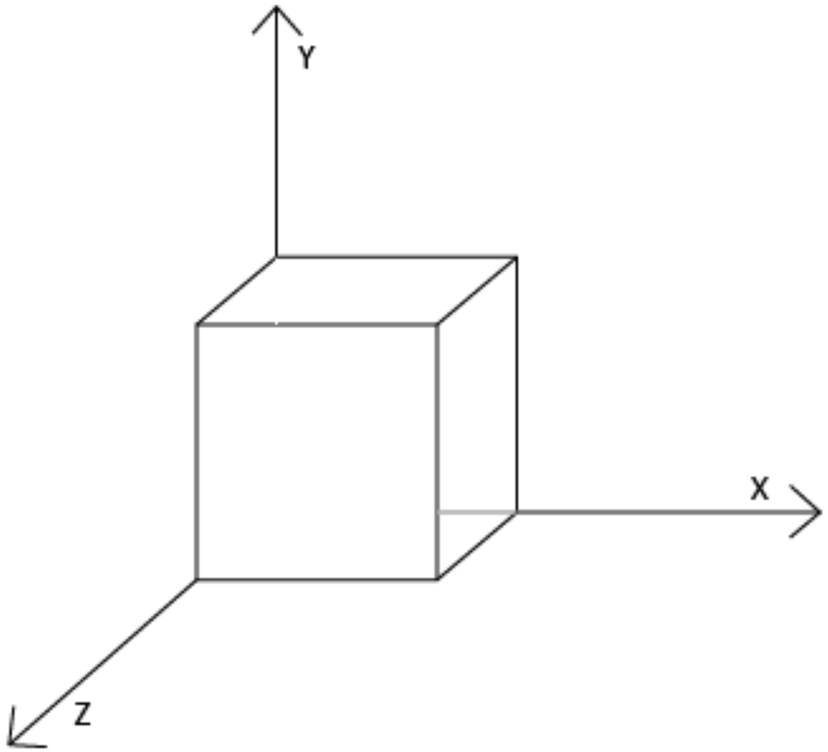
Metal Shading Language

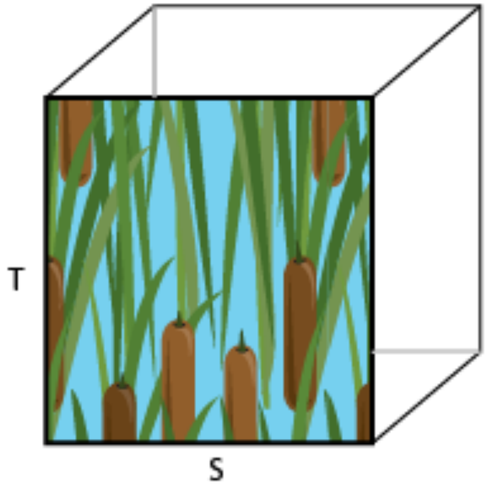
Demo 2

- Create a more complex shader modifier
- Inject a lighting model to create a triforce symbol on our box

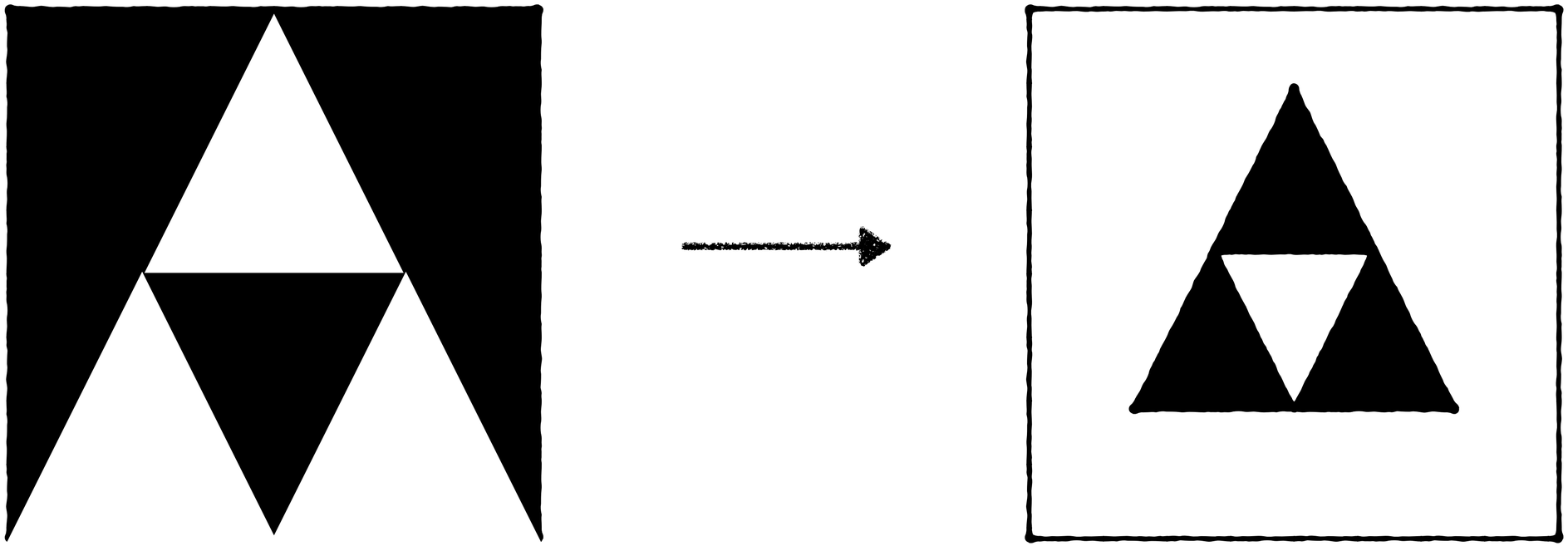
Demo 2

Texel Coordinates

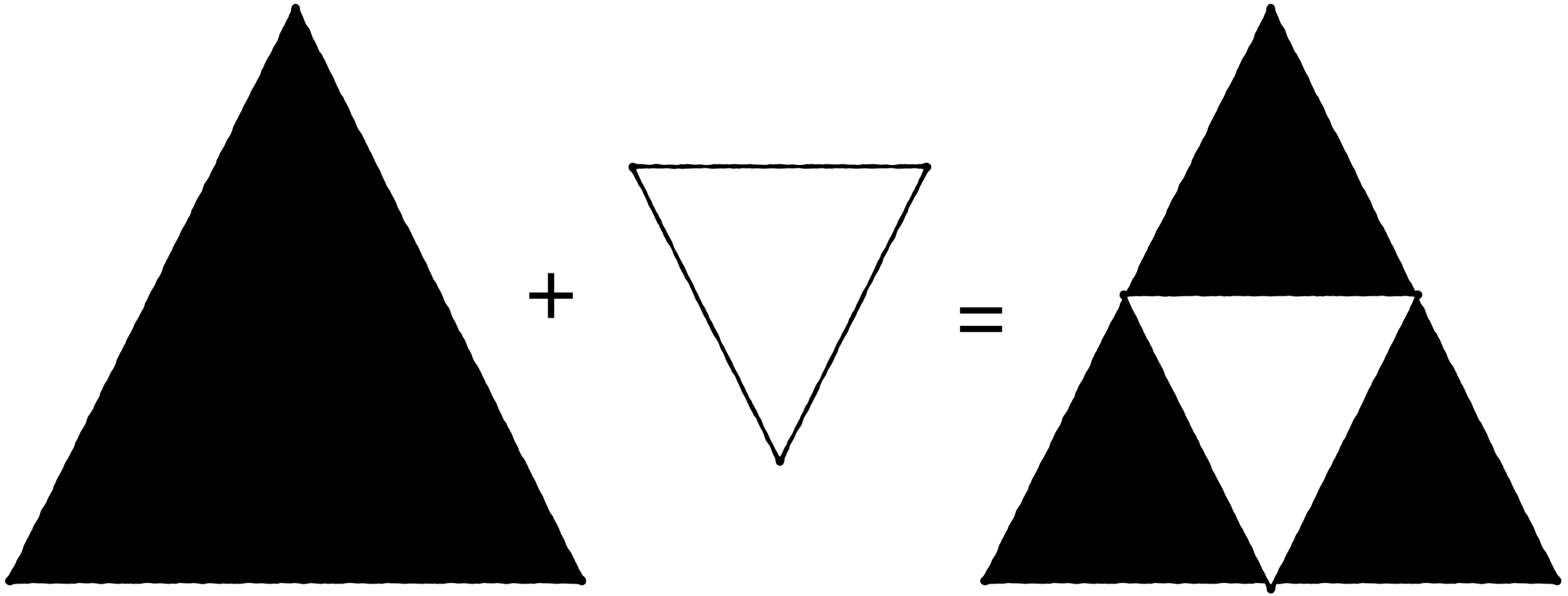




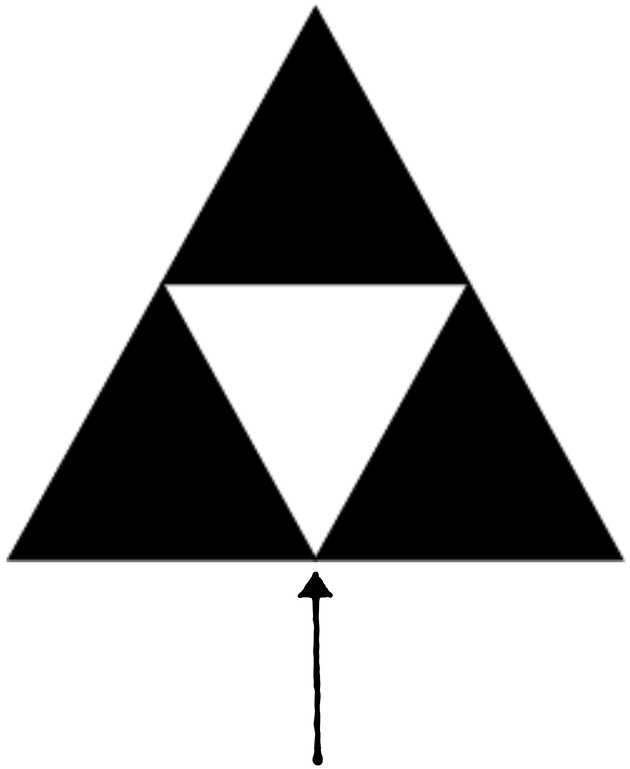
Invert and Scale



Calculate Triangle



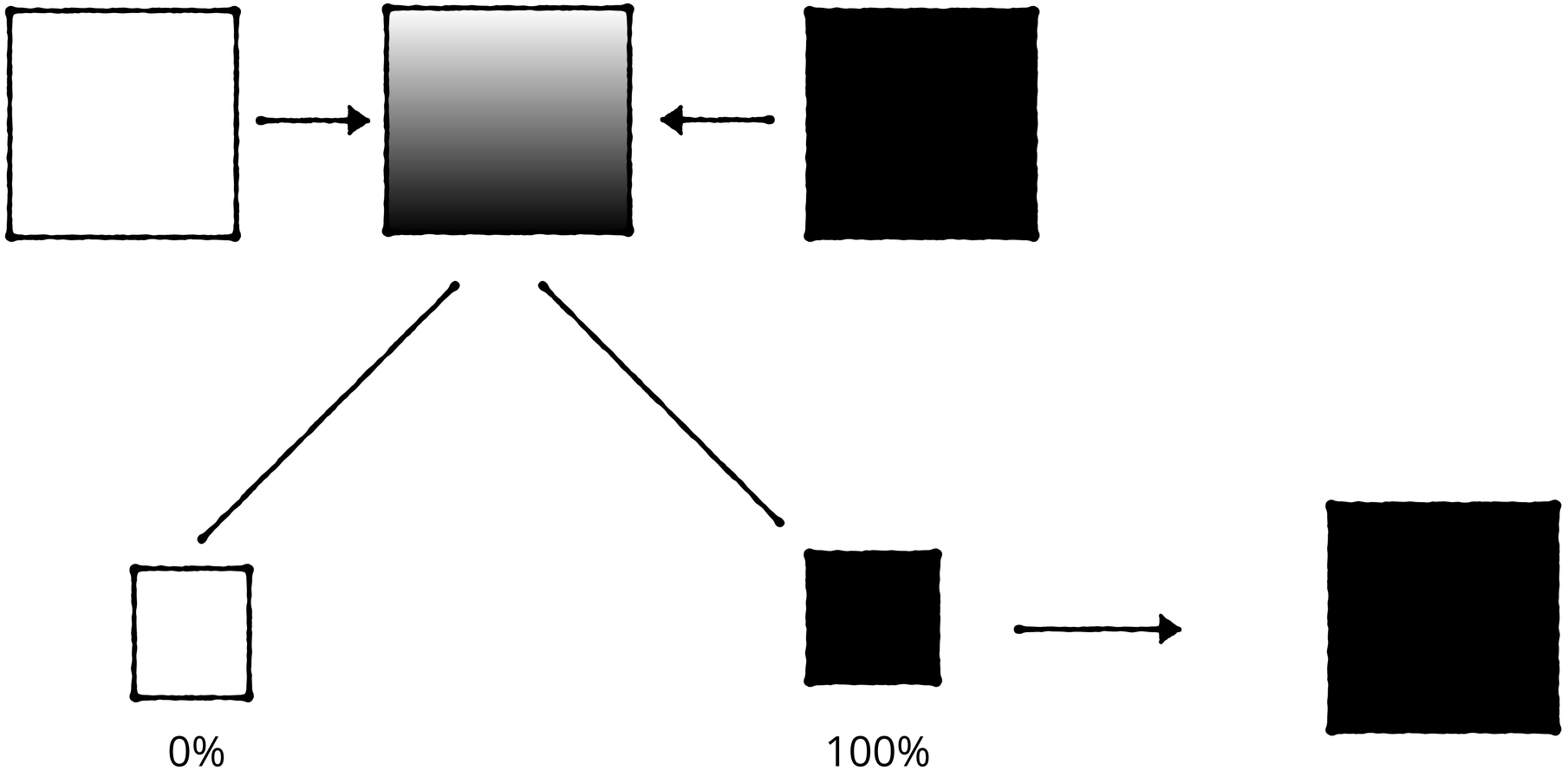
Fill Color



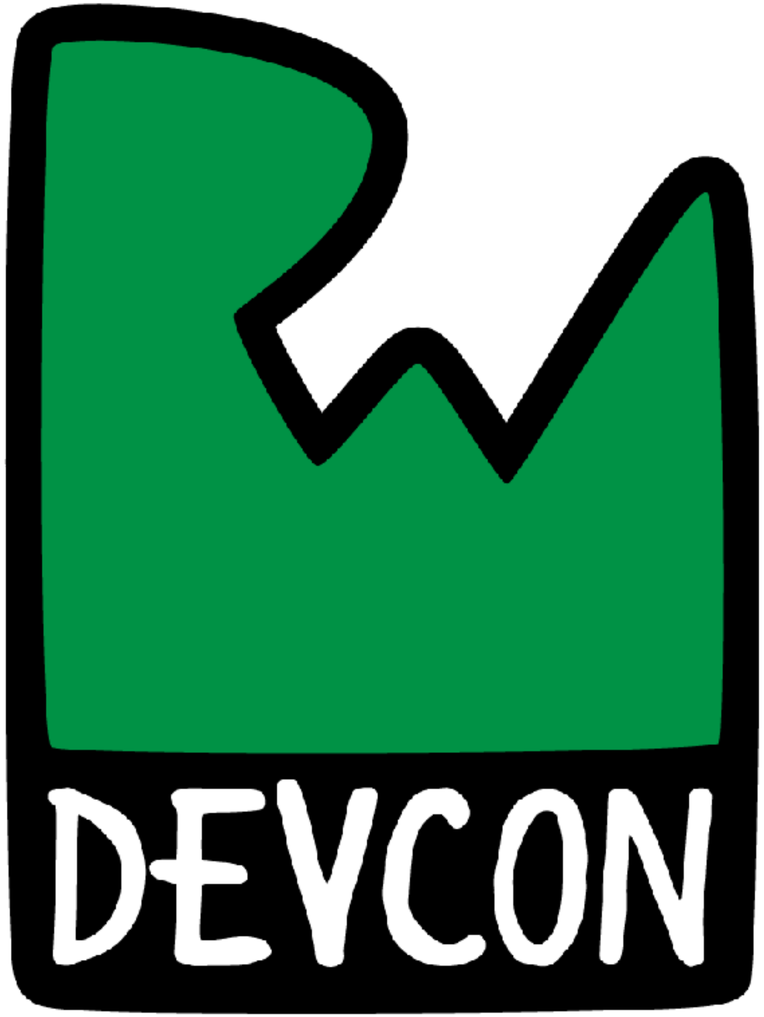
If the point is located on an outer triangle, color is black

Else the color is white

Mix the Colors



Session 16: Integrating Metal Shaders with SceneKit



SCNPROGRAM

SCNProgram

- Replaces SceneKit's built in material or geometry rendering
- You're responsible for both the vertex and fragment shaders

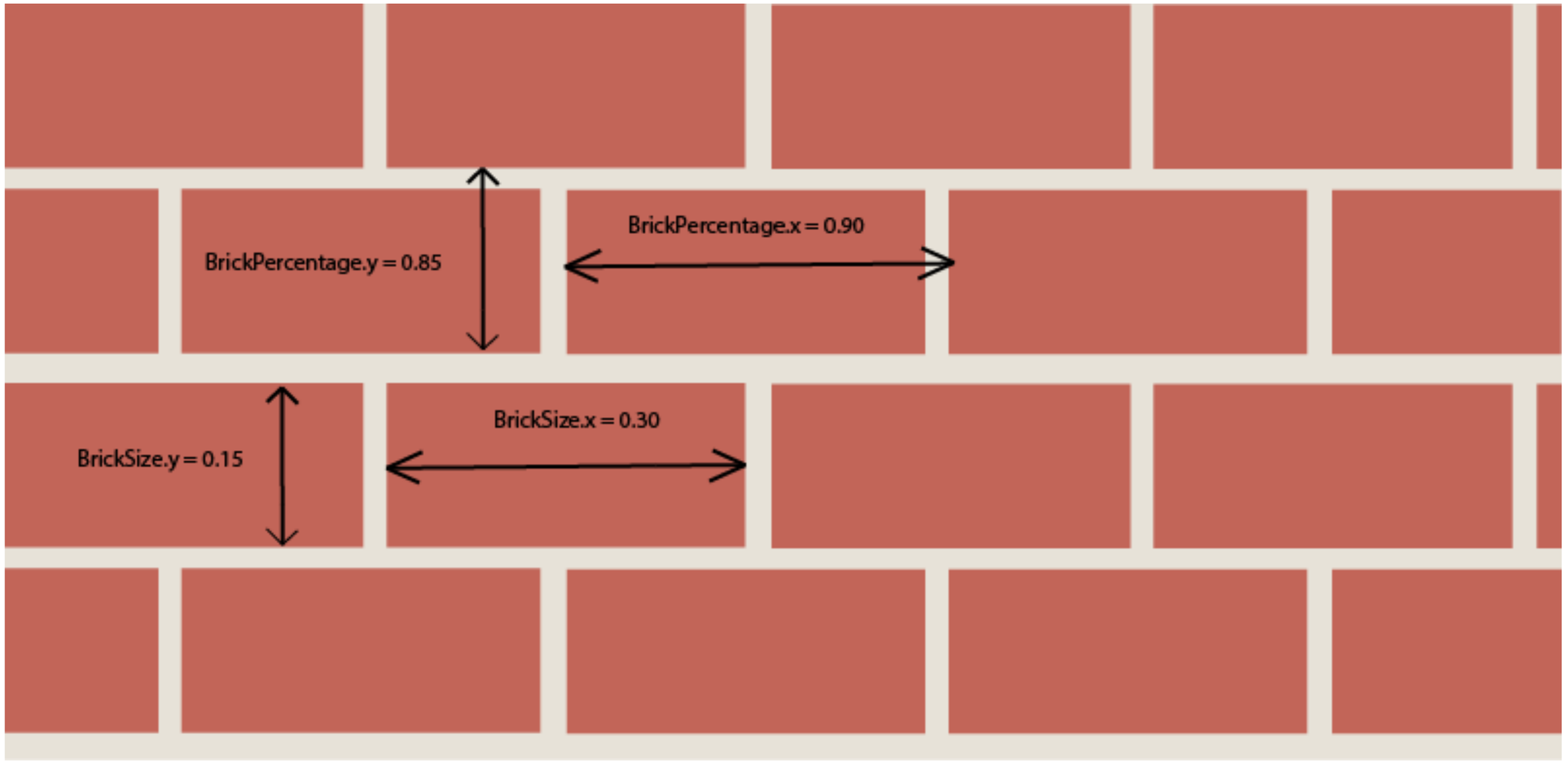
Vertex Shader

- Per vertex data
- Any modifications involving geometry (transforms, translation, rotations, scales…)
- More efficient to calculate values per vertex that can be linearly interpolated rather than fragment, such as lighting. However, can be less accurate

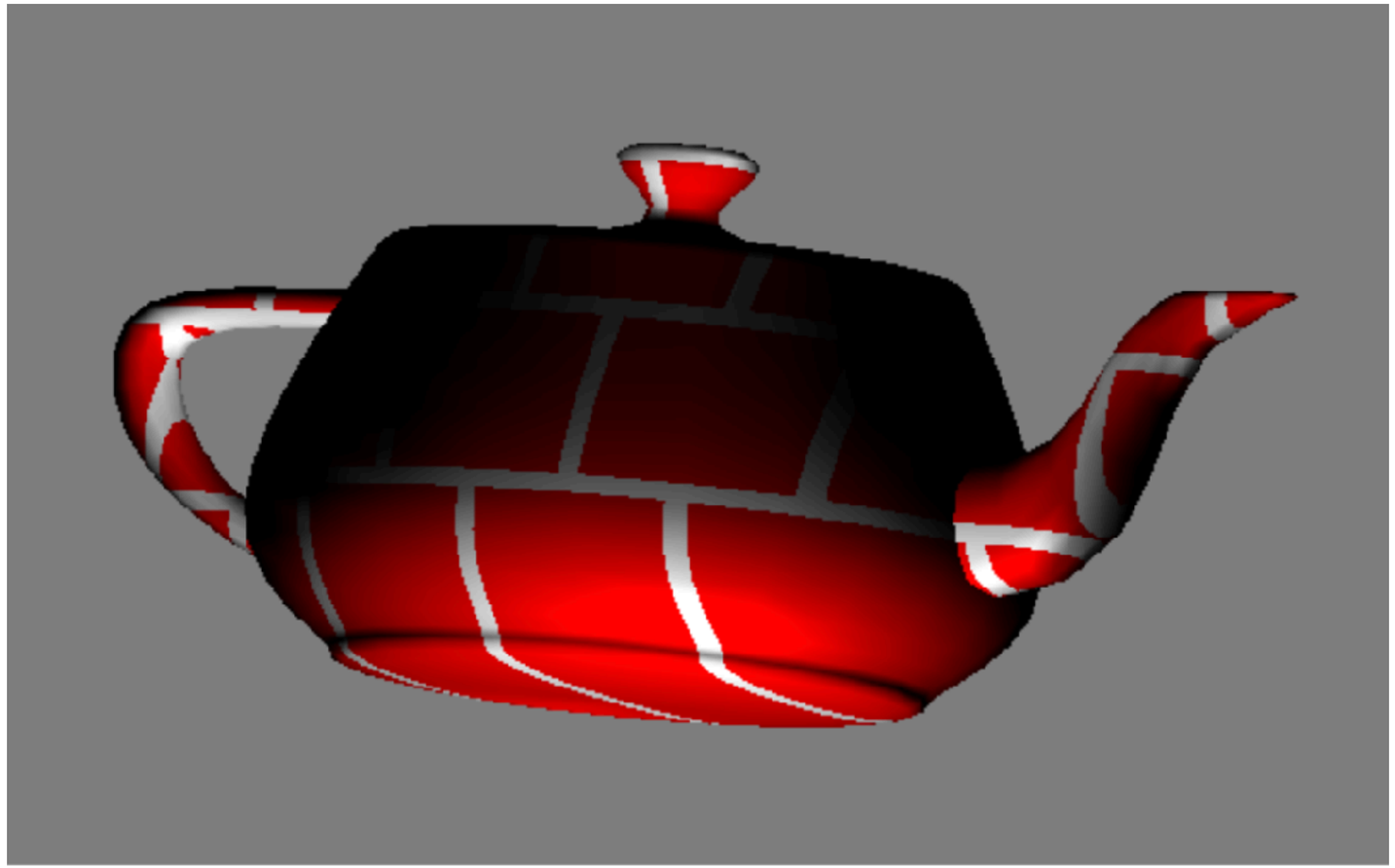
Fragment Shader

- Per pixel/fragment processing
- Lighting calculations, color manipulation, special effects like blurs
- Image processing and machine vision
- The more versatile of the two shaders
- Last step before each pixel is rendered to the screen

Brick Shader



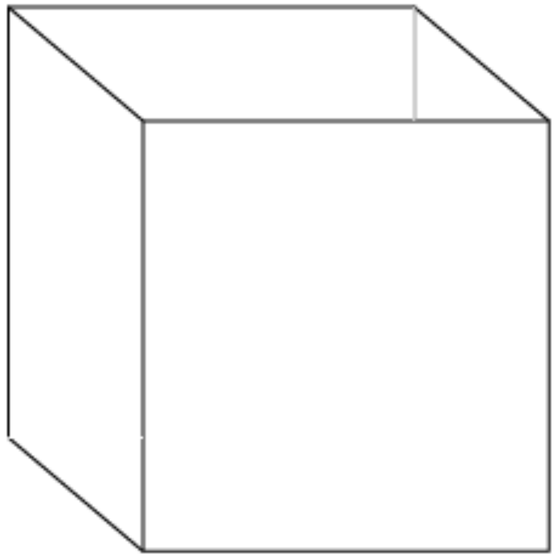
Brick Shader



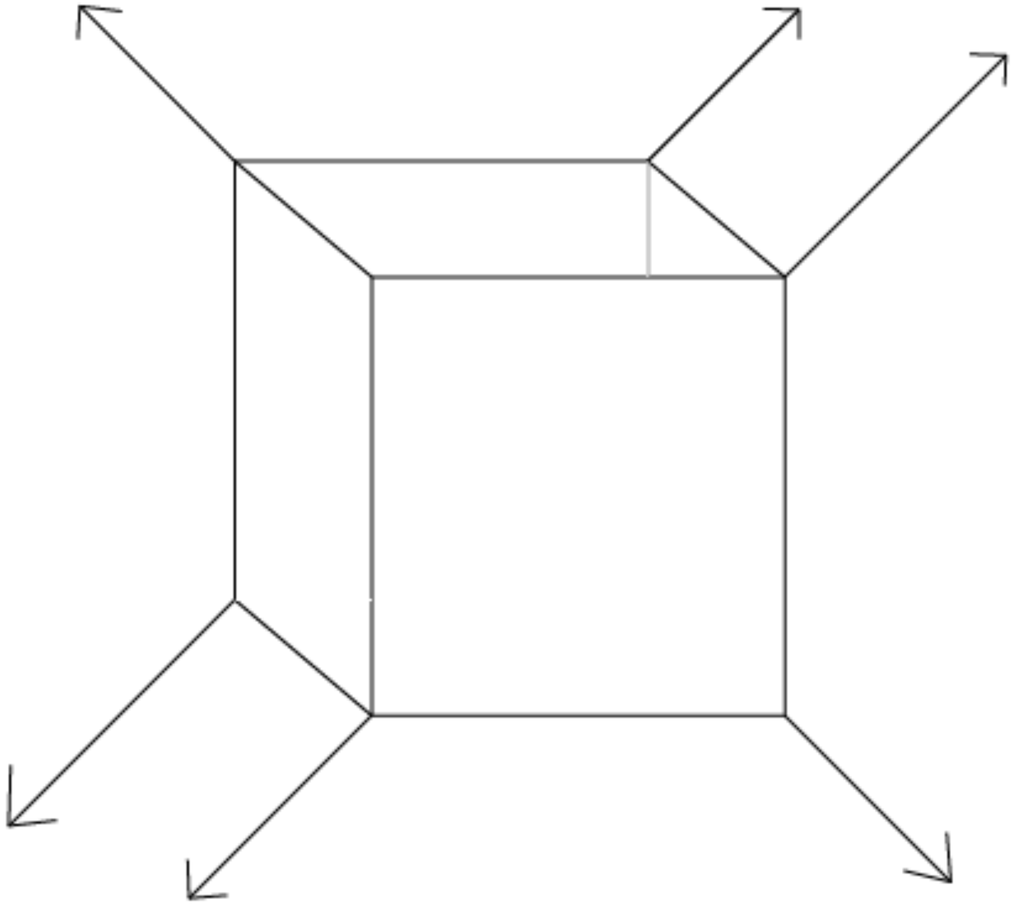
Brick Shader Parameters

- A light source and position
- Diffuse and Specular Reflection Characteristics
- Brick Pattern Sizes and Ratios
- Brick and Mortar Colors

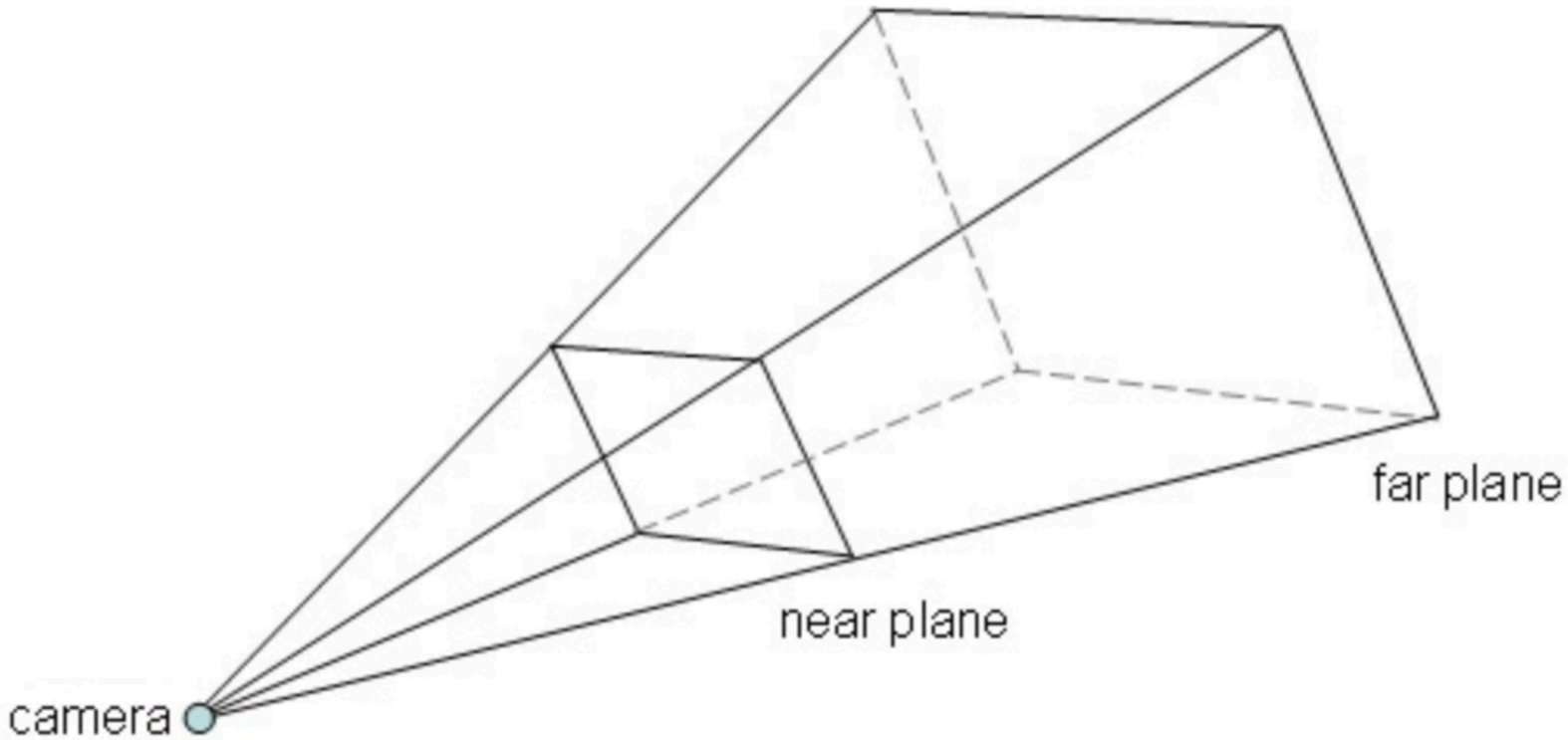
Vertex Constants



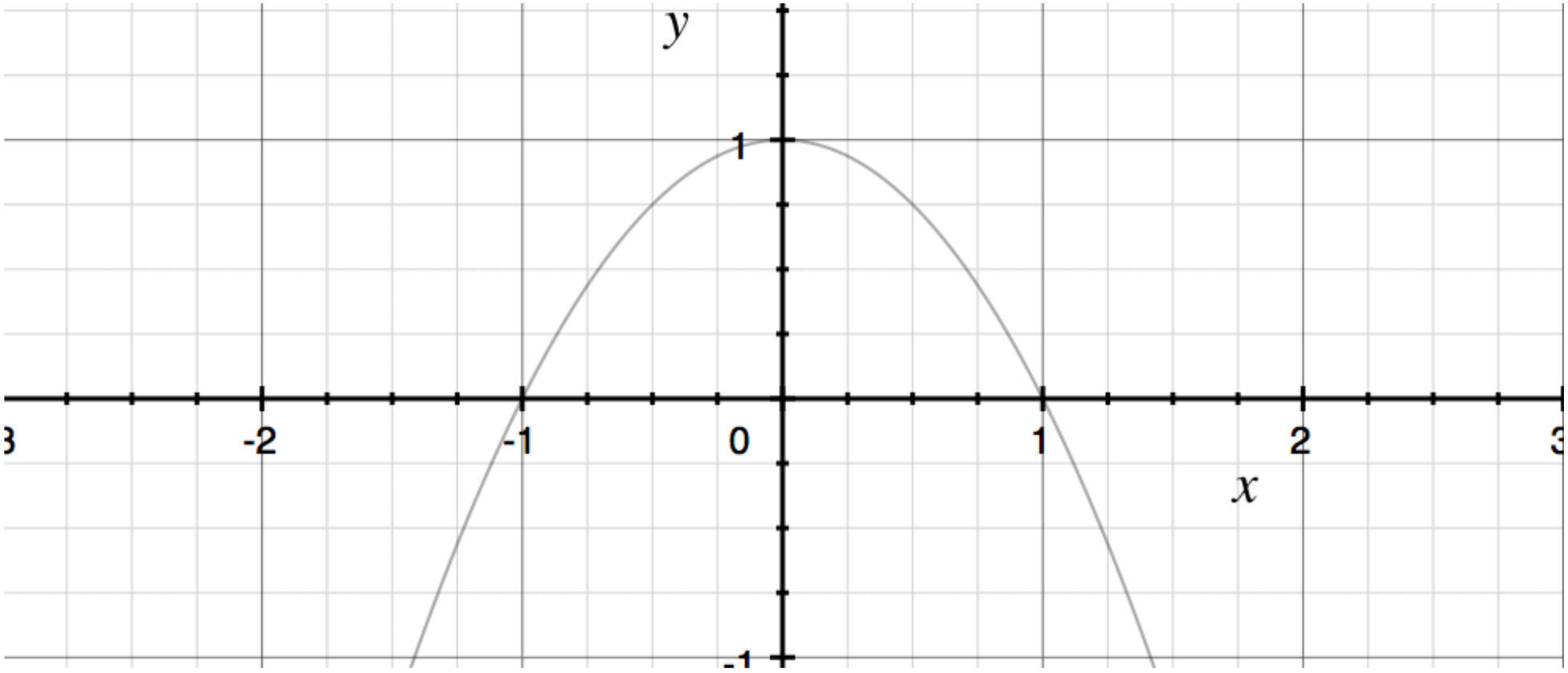
Vertex Constants



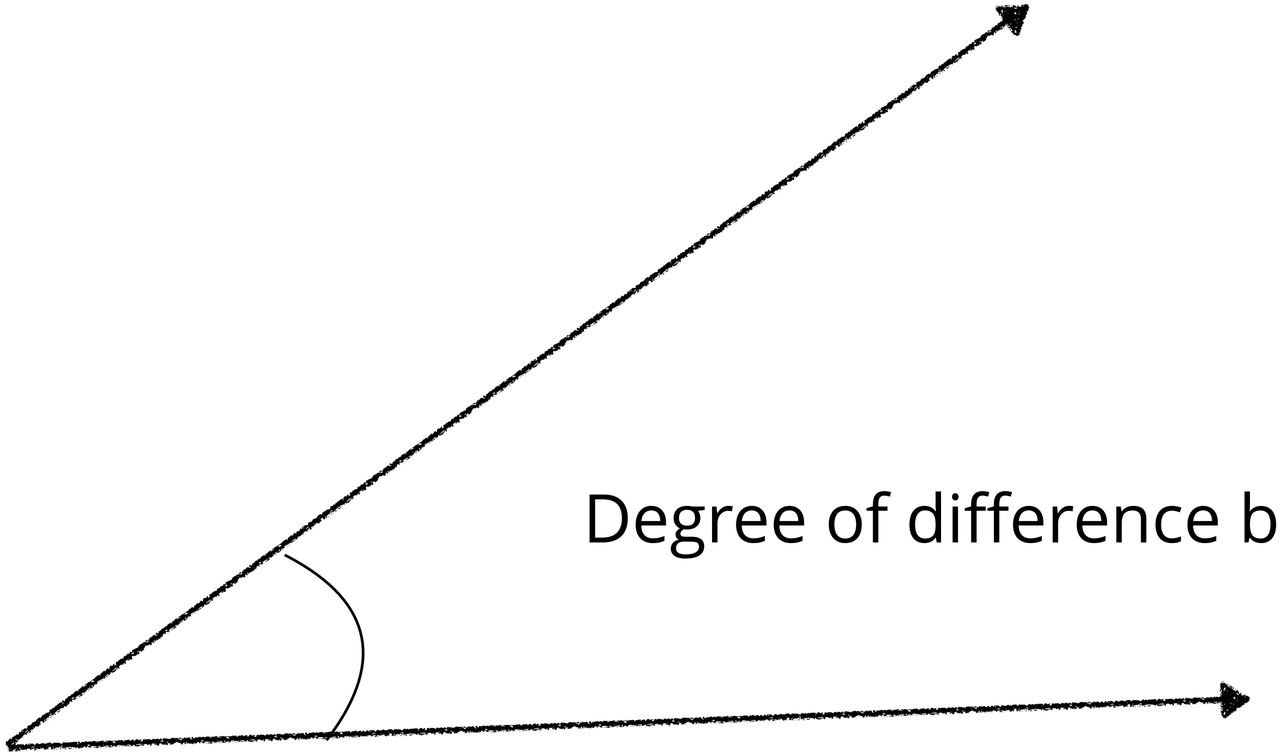
Vertex Constants



Specular Highlight

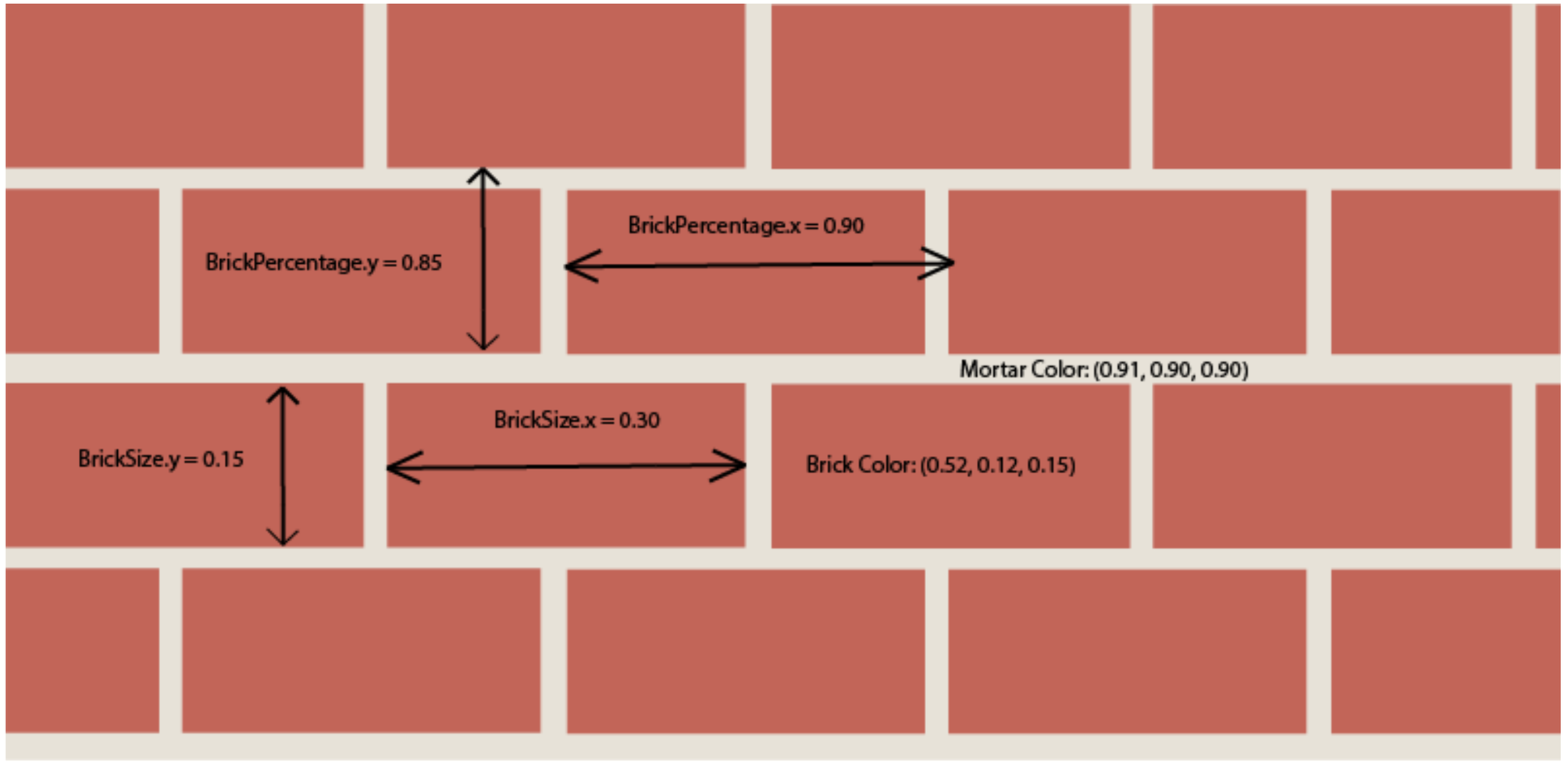


Dot Product



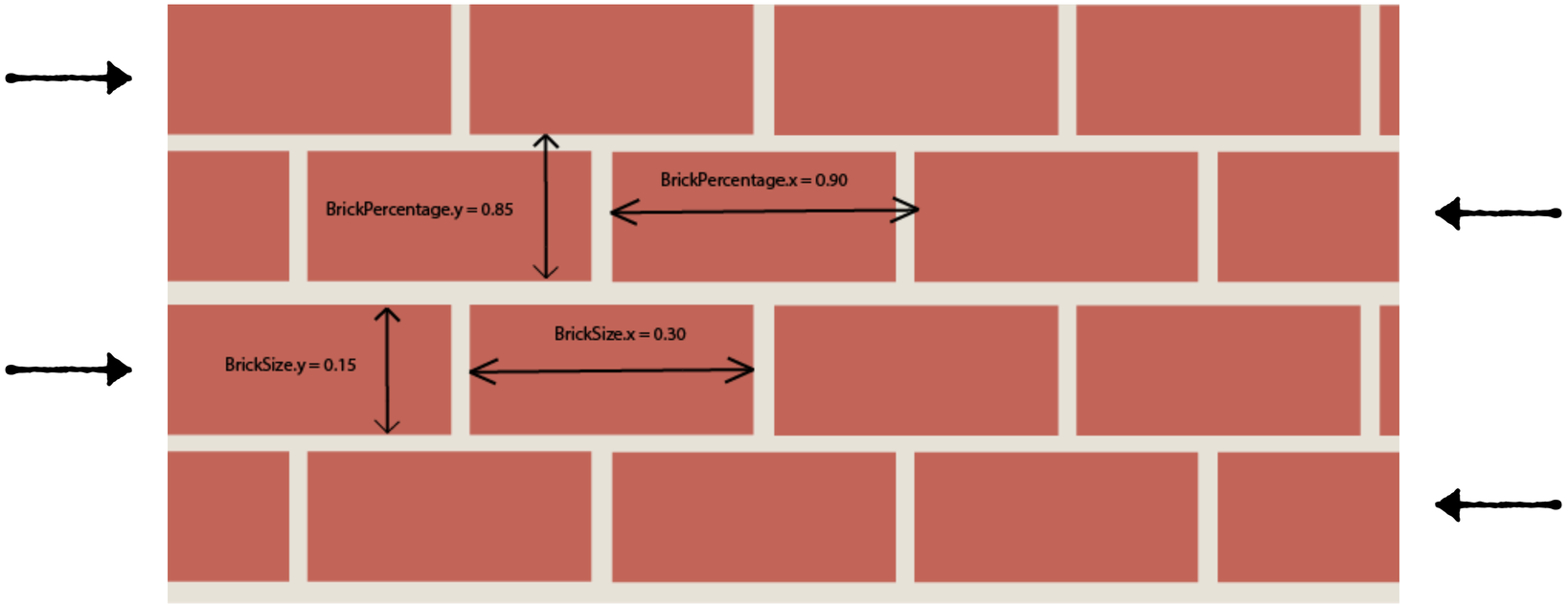
Degree of difference between two vectors

Fragment Constants



Brick Positions

If these rows, do nothing.



If these rows, offset by half brick.

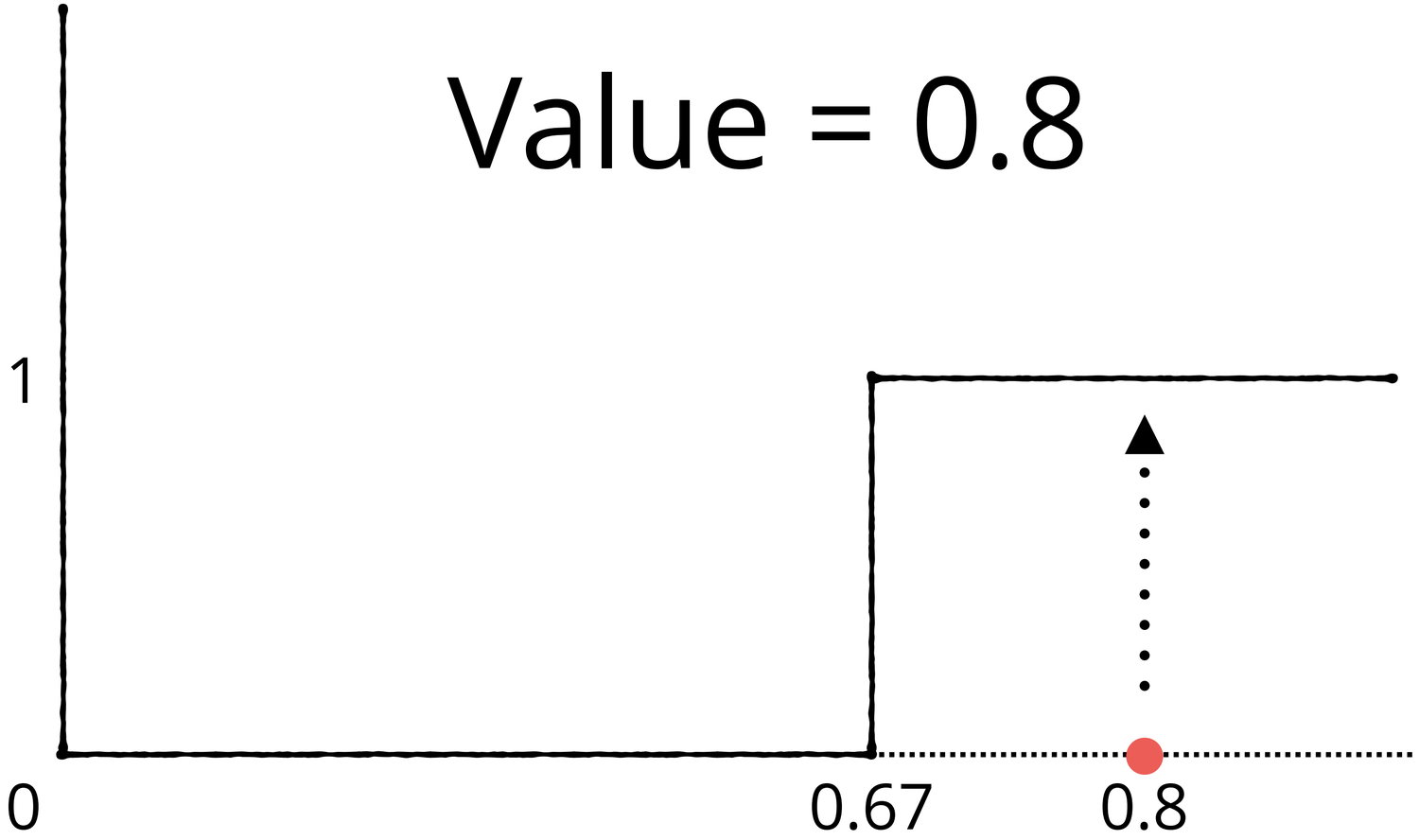
Fract

3.14159

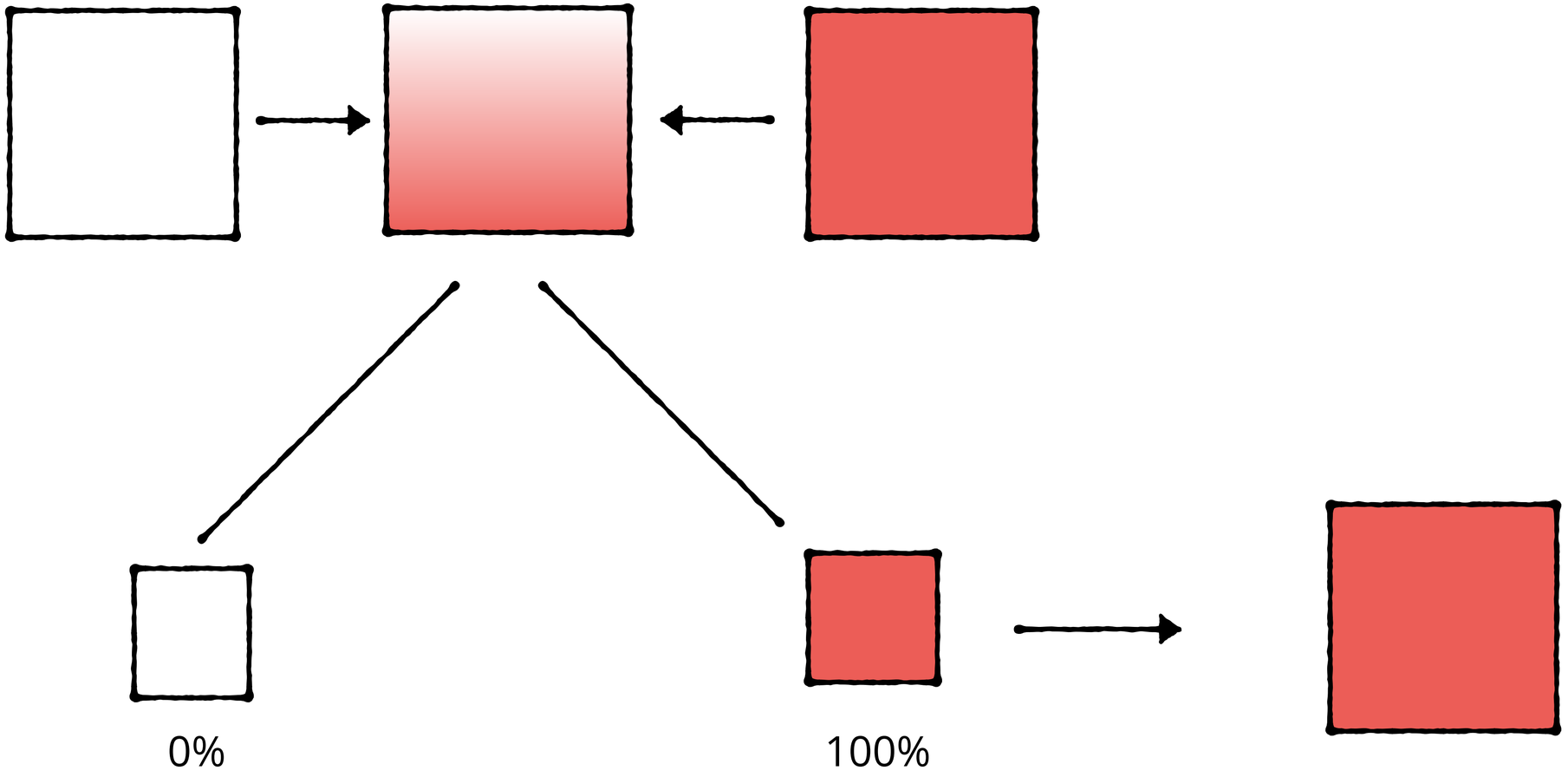
Step Function

Threshold = 0.67

Value = 0.8

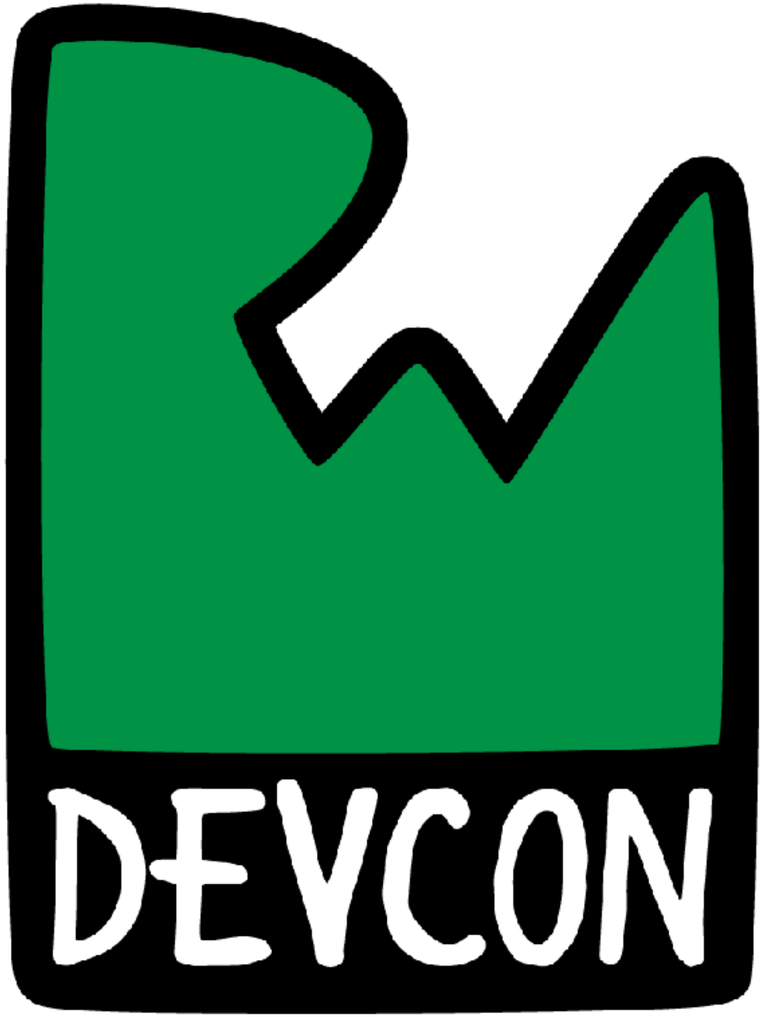


Mix the Colors



Demo 3

Session 16: Integrating Metal Shaders with SceneKit



Conclusion

What You Learned

- **Demo 1**: How to work with SceneKit and what you get from it
- **Demo 2**: Overview of Graphics Math
- **Demo 3**: Full Metal shader integrated into SceneKit

Where To Go From Here?

- 3D Apple Games by Tutorial
- OpenGL Shading Language (the Orange Book)
- 3D Math Primer for Graphics and Game Development
- Computer Graphics: Principles and Practice
- Twitter: @redqueencoder