GLSL
really supports basic data types

vec2 vec3 vec4 also ivec, uvec
all names always available
V. xy2w
V. rgba
V. s t p q (not u, v

Swizzling

variable qualifiers
in smooth, non-perspective, flat
uniform
out
const

lots of useful functions
normalize (vec)

how do you transform normals?
transform "not quite
adjoint (transform) = inverse transpose
for rotations, this is
the matrix

gl_Normal
Stripe

if \( u > 0.5 \)  \( \Rightarrow \) step, mix

step edge will alias

1. Blur

\[ \text{step} \rightarrow \text{smooth step} \]

\[ \frac{dF}{dx} \]

\[ \text{fwidth} = \sqrt{dx^2 + dy^2} \]

really abs

gives a way to peek at "next" fragment/pixel
Using Textures

1. Texture Setup
   - Name parameters - repeat, filters
   - Image

2. Texture "Unit" ← for multitexture

3. Bind unit ← Texture

Units appear in GLSL as samplers

Multiple textures?
- Day vs. night (blend between)
- Clouds over planet
  - Make clouds move by changing
- Dirt over pattern
FANCY SHADING

1. Making something look not flat
   a. normal mapping
   b. bump mapping

   ideas
   use normals from this

   trick: how to combine with existing normals
   just add?
   compute based on desired height?

2. Displacement Mapping
   - really move xyz
   - really recompute normals

   hard to do - since fragments move

   difference - at edges

3. Parallax hack
   View
   View-Dependent Texture Mapping

   different colors

Environment Maps