

11/22

GLSL

really supports basic data types

vec2 vec3 vec4 also ivec, uvec
all names always available

v. xyzw

v. rgba

v. stpq

(not u, v)

swizzling

variable qualifiers

in

uniform

out

const

smooth, non-perspective, flat

lots of useful functions
normalize (vec)

how do you transform normals?

transform ← not quite

adjoint (transform) \equiv inverse transpose

for rotations, this is
the matrix

gl-Normal

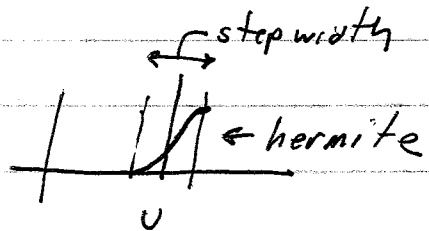
11/22-2

Stripe

if $(v > .5)$ \Rightarrow step, mix
step edge will alias

(i) Blur

step \rightarrow smooth step



how wide?

fixed amount?

1 pixel?

dF/dx

$$fwidth = \sqrt{dx^2 + dy^2} \quad \text{really abs}$$

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

11/22 - 3

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

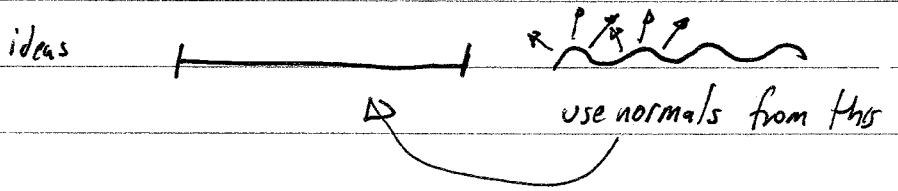
dirt over pattern

11/22 - 4

FANCY SHADING

① Making something look not-flat

- ① normal mapping
- bump mapping



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

② Displacement Mapping

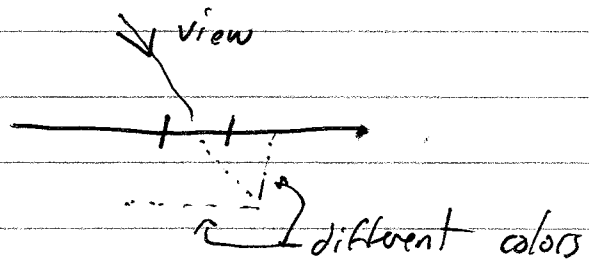
- really move xyz
- really recompute normals

hard to do - since fragments move

difference - at edges

③ Parallax hack

View-Dependent
Texture Mapping



Environment Maps