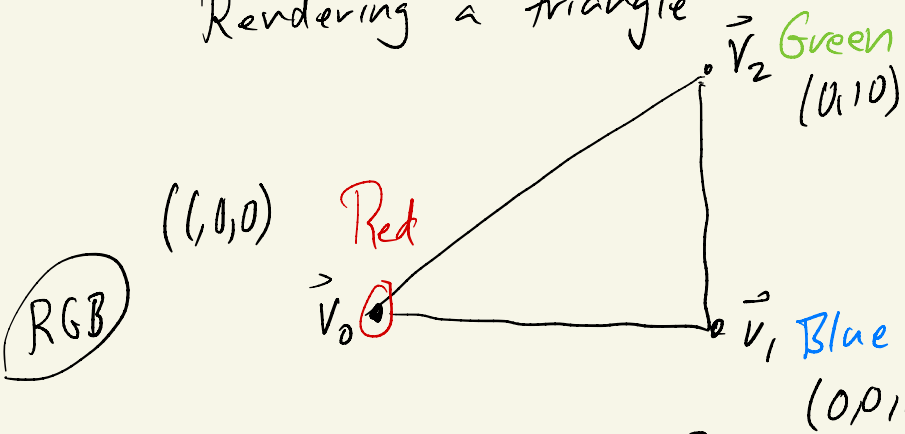
Labs APM B349 or B432

- We hope that one works at 6:00

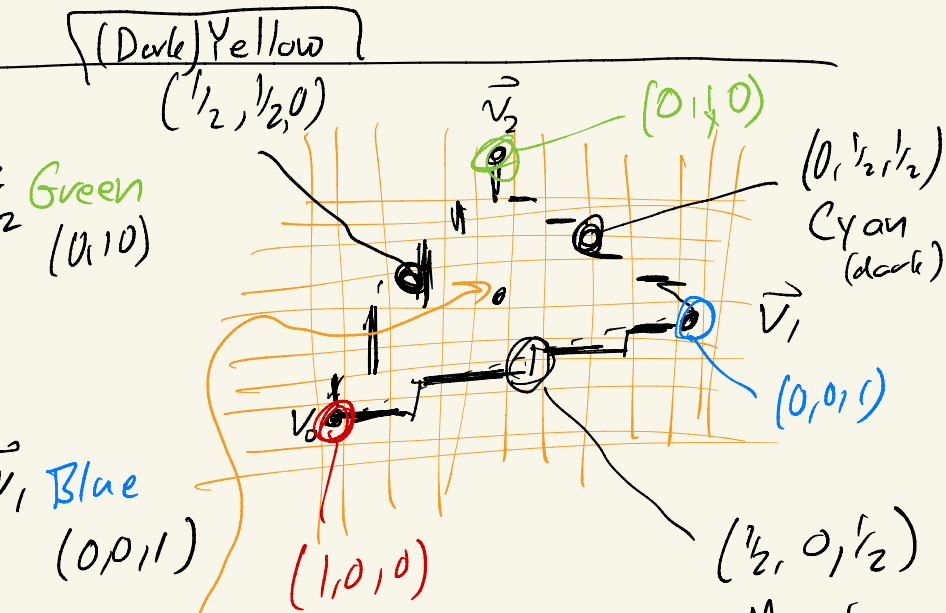
cloudlabs.ucsd.edu – is now working.

Rendering a triangle



$\vec{v}_0, \vec{v}_1, \vec{v}_2$ in $\mathbb{R}^3$

Viewer in $\mathbb{R}^3$.



Smooth Shading

(1/3, 1/3, 1/3) Gray

Flat shading - Doesn't average, it uses the value from the last vertex in the triangle.

In the example, the whole triangle will be green.

Hidden surface algorithms -

Back faces

Culling triangles outside the field of view

Also often triangles that are too far away or too close.

Real hidden surface purpose -

Not see triangle occluded by closer triangles.

(1) Geometric analysis

(2) Painter's Algorithm

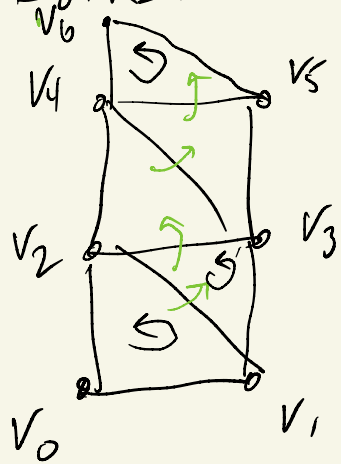
(3) Depth Buffer

- Pixels store a depth or distance value
  - Typically use 32 bit integer for the depth value.
  - Uses a nonlinear function of distance.
  - Thus there are minimum/maximum distances

Transparent / Translucent triangles

- Use a hybrid of depth buffer + painters algorithm.

GL_TRIANGLE_STRIP



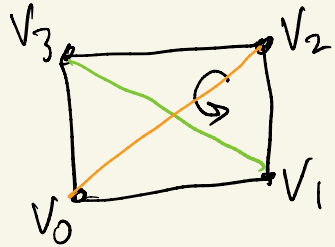
Vertex order

$V_0, V_1, V_2, V_3, V_4, V_5 \ldots$

Triangles have vertices

V_i, V_{i+1}, V_{i+2}

(Can think of it as alternating CCW, CW)



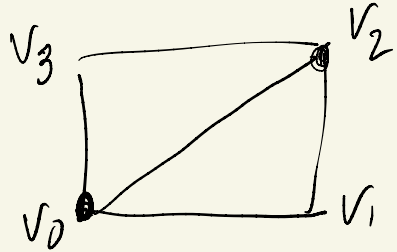
What vertex is needed to render this as a triangle strip?

V_0, V_1, V_3, V_2

Not this. $V_1, V_0, V_2, \ldots$ - Starts off clockwise ✗

$\vec{V}_2, \vec{V}_3, \vec{V}_1, \vec{V}_0$ - this order does work! ✓

V_3, V_0, V_2, V_1 - Also works, but  are the triangles!

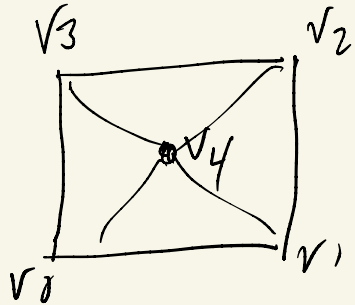


Render this as a triangle fan.

V_0, V_1, V_2, V_3 — Works! ✓

Triangles V_0, V_i, V_{i+1}

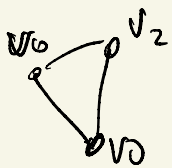
V_2, V_3, V_0, V_1 — Also works. Same triangles

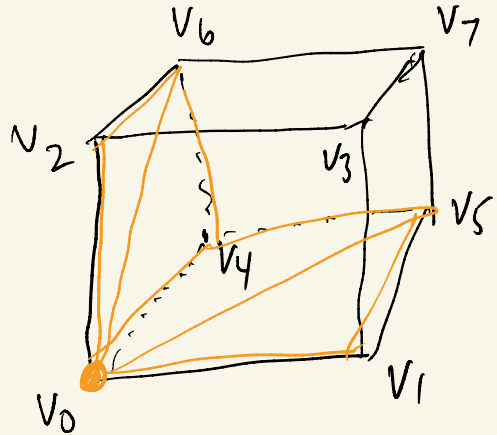


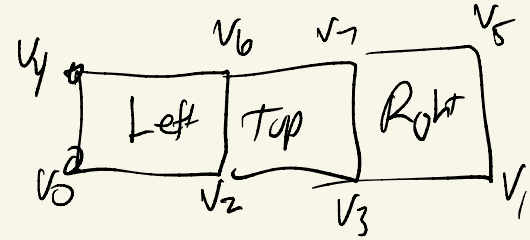
$\vec{V}_4, \vec{V}_0, \vec{V}_1, \vec{V}_2, \vec{V}_3$

$$\vec{V}_4 = \frac{V_0 + V_1 + V_2 + V_3}{2}$$

Cube







Render as 2 triangle fans

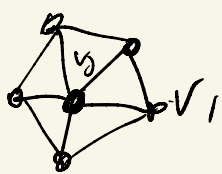
$v_0, v_2, v_6, v_4, v_5, v_1, v_3, v_2$

or

Render as 2 triangle strips.

left-top-right $v_4, v_0, v_6, v_2, v_7, v_3, v_5, v_1$

& front-bottom-back.



$v_0, v_1, \ldots v_1$