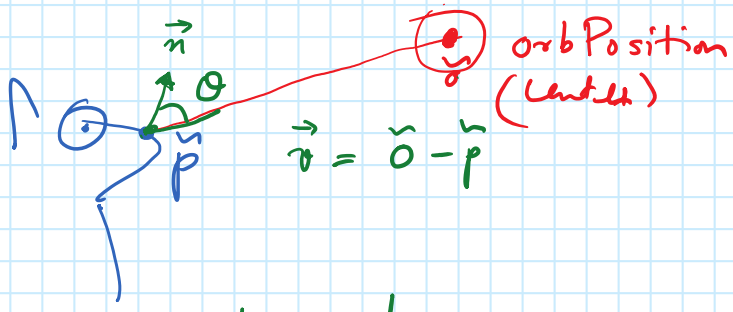
September 20, 2021 2:58 PM

§ A1

1b)



vertex color

$vColor = \max(\cos\theta, 0)$

§ Recap

$$\tilde{p} = \vec{v} + \tilde{o}$$

$$= \vec{b}_0 v_0 + \vec{b}_1 v_1 + \tilde{o}$$

$$= \begin{pmatrix} \vec{b}_0 & \vec{b}_1 & \tilde{o} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ 1 \end{pmatrix}$$

homogeneous coordinate of a point

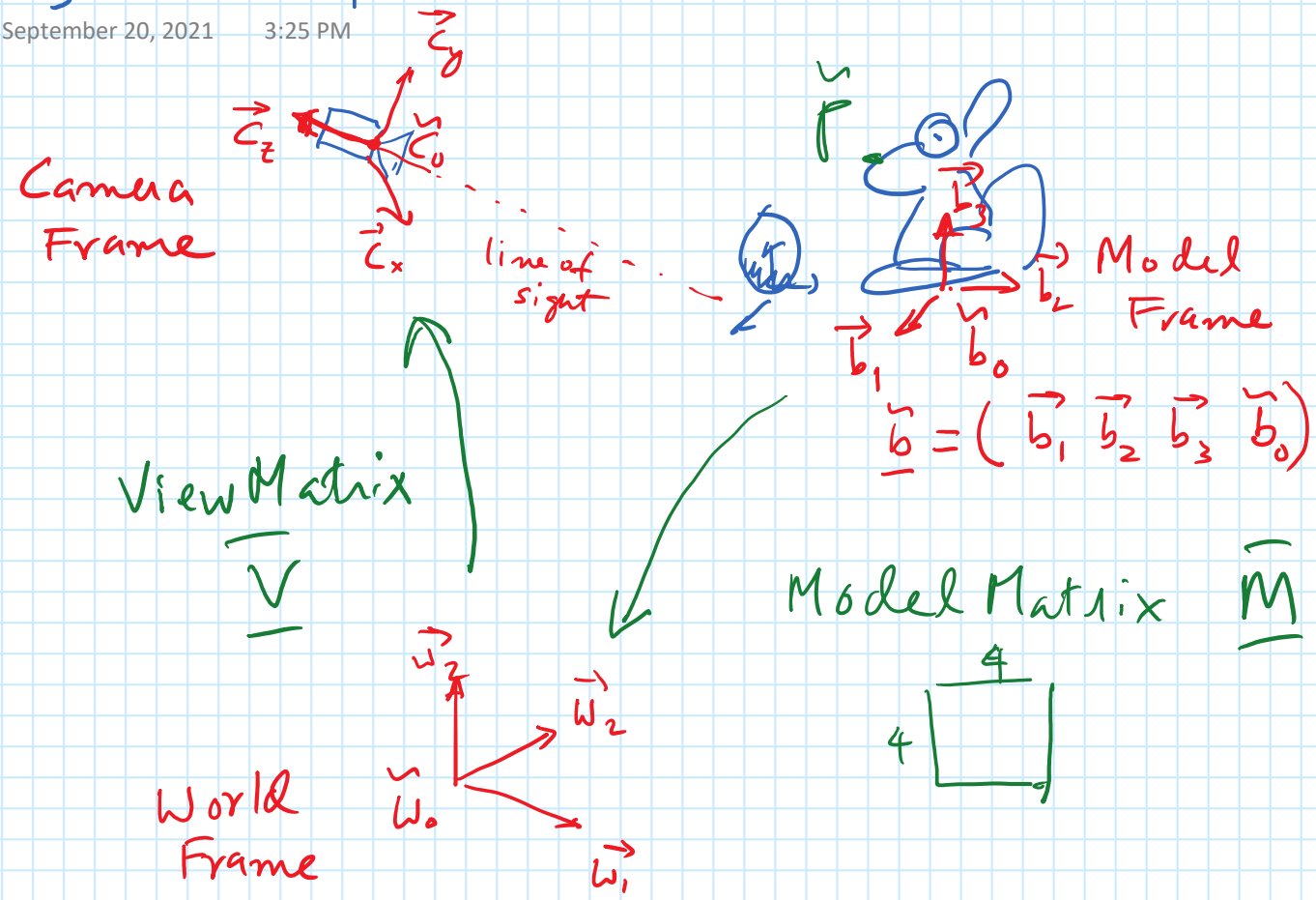
A coordinate frame

Notation:

Frame $\underset{\sim}{b} = \left(\vec{b}_0 \;\; \vec{b}_1 \;\; \tilde{o}\right)$

§ Three important frames

September 20, 2021 3:25 PM



model view matrix

$$\bar{P}_c = (\ )\,\underline{V}\,\underline{M}\,\bar{P}$$

gl_position — Projection (later) — position

Three.js support

- Recall: Built-in uniforms and attributes
  https://threejs.org/docs/#api/en/renderers/webgl/WebGLProgram

Vertex shader (unconditional):

```
// = object.matrixWorld
uniform mat4 modelMatrix;

// = camera.matrixWorldInverse * object.matrixWorld
uniform mat4 modelViewMatrix;

// = camera.projectionMatrix
uniform mat4 projectionMatrix;

// = camera.matrixWorldInverse
uniform mat4 viewMatrix;

// = inverse transpose of modelViewMatrix
uniform mat3 normalMatrix;

// = camera position in world space
uniform vec3 cameraPosition;
```

```
// default vertex attributes provided by Geometry and BufferGeo
attribute vec3 position;
attribute vec3 normal;
attribute vec2 uv;
```

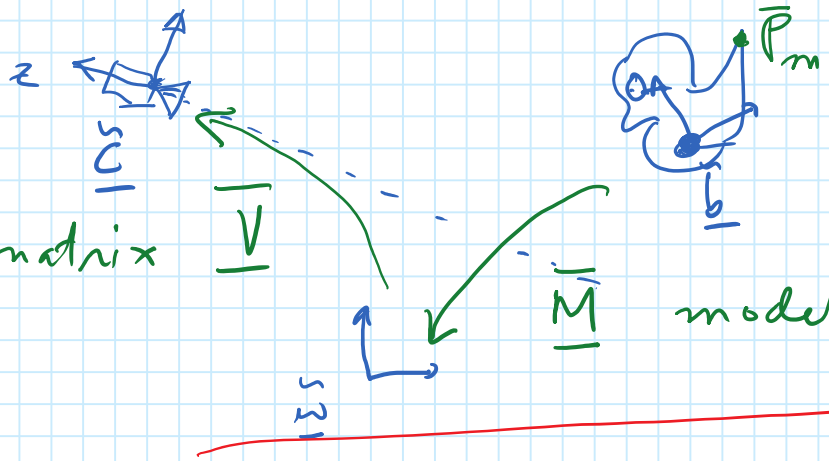
Fragment shader:

```
uniform mat4 viewMatrix;
uniform vec3 cameraPosition;
```

For next class

- Try the transforms app by Eric Haines that I showed in class, it's available here:
  https://www.realtimerendering.com/udacity/transforms.html
- Review Chapter 5 of textbook

§ Review 3 frames



$$\bar{P}_w = \bar{M}\,\bar{P}_m$$

$$\underline{w}\,\bar{P}_w = \underline{w}\,\bar{M}\,\bar{P}_m = \underline{b}\,\bar{P}_m = \bar{P}$$

$$\underline{w}\,\bar{M} = \underline{b} = (\vec{b}_1\ \vec{b}_2\ \vec{b}_3\ \underline{b}_0)$$

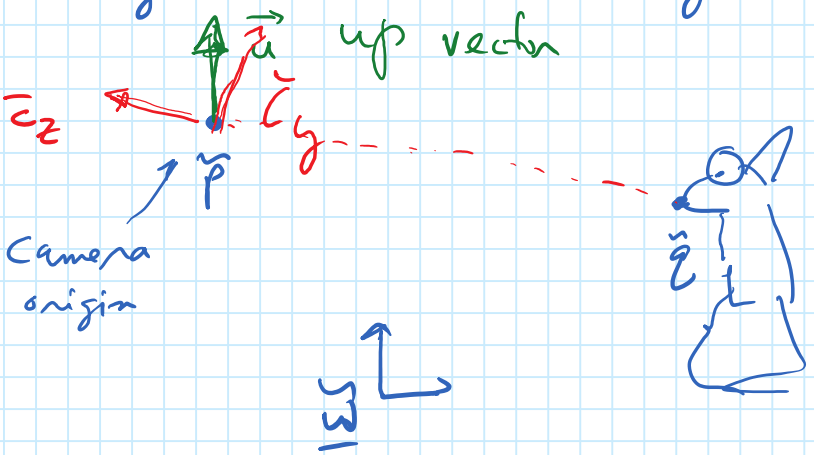
§ Anatomy of a transformation matrix

$$\bar{M} = \left[\begin{array}{ccc|c} x & x & x & x \\ x & x & x & y \\ x & x & x & z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

coordinates of $\vec{b}_1$ in frame $\underline{w}$

coordinates of $\underline{b}_0$ in frame $\underline{w}$

§ Finding the $\bar{V}$ matrix using "lookAt"



Camera matrix

$$\bar{\bar{C}} = \left[\begin{array}{c|c|c|c} \bar{c}_x & \bar{c}_y & \bar{c}_z & \bar{c}_0 \end{array}\right]$$

all coords in world frame

$$\bar{c}_0 = \bar{p}$$

$$\bar{c}_z = \text{normalize}(\bar{p} - \bar{q})$$

$$\bar{c}_x = \text{normalize}(\bar{u} \times \bar{c}_z)$$

$$\bar{c}_y = \bar{c}_z \times \bar{c}_x$$

Issues with Textbook's "lookAt"

- Book description in 5.2.3 has a bug, fixed in online Errata (make this and other corrections in your textbook copy)
  - z = normalize(p - q)
    x = normalize(u × z)
    y = (z × x)
- The book's "lookAt" should be called "eye" matrix (E in textbood, or camera C in our notes).
- It is the inverse of Three.js's camera.lookAt() method (which produces the view Matrix, V in our notes)
- The author is aware of these issues, will fix it in future editions

7