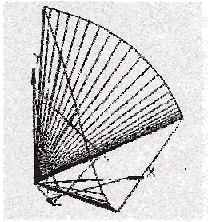
Exercise –
Lighting - Shading Models - Multipass Rendering

Discussion – CIE Chart

6 f): „Bestimmen sie die Valenz C123, die aus der Mischung der 3 Valenzen C1, C2 und C3 hervorgeht."

=> Use XYZ-Space

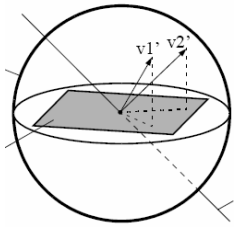


$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

Discussion - Trackball

- Rotation angle depends on trackball radius:



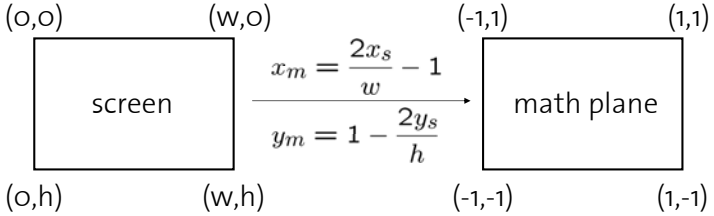
$$z = \sqrt{r^2 - x^2 - y^2}$$

z: projected, not scaled

$$\phi = 2arcsin\left(\frac{\sqrt{(x_{new} - x_{old})^2 + (y_{new} - y_{old})^2 + (z_{new} - z_{old})^2}}{2r}\right)$$

Discussion - Trackball

- Rotation direction up-down



- Rotation direction left-right:

$v_{new} \times v_{old} = -(v_{old} \times v_{new})$

Vector product not commutative

Question 1 - Lighting Model

- „Geben Sie zu den unten aufgeführten Bildern an, mit welchem Beleuchtungsmodell sie erstellt wurden. Begründen Sie Ihre Antwort.“

1/3

Light Simulation in OpenGL

- Constant Shading
  - One color (and one normal) per primitive – Flat Shading
- Gouraud Shading
  - Computed at vertices
  - Linear interpolation of vertex intensities
- Phong Shading
  - Linear interpolation of vertex normals

Question 2 - Light Properties

- **Ambient** is color of the object from all the undirected light in a scene.
- **Diffuse** is the base color of the object under current lighting. There must be a light shining on the object to get a diffuse contribution.
- **Specular** is the contribution of the shiny highlights on the object.
- **Emission** is the contribution added in if the object emits light (i.e. glows)

Practical Exercise – Multipass-Rendering

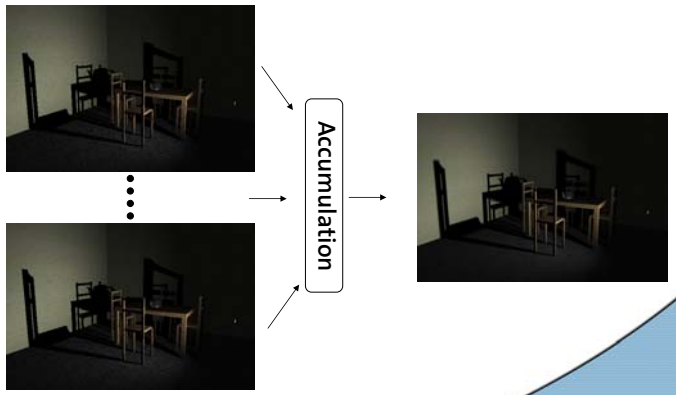
Slides copied from Tim Weyrich

Multipass – Rendering

- Full-scene antialiasing
- Shadows
- Advanced shading (e.g., bump-mapping)

- OpenGL: accumulation buffer
  `glAccum()`
- Programmable HW: Fragment shaders

Multipass – Rendering



Aufgabe 3

- OpenGL material properties
  - Specify parameters of the shading model (e.g. Phong exponent `GL_SHININESS`)
  - `glMaterialf()`
  - `glMaterialf (GL_FRONT, GL_SHININESS, fShininess);`

Aufgabe 4

- Spotlights
  - `glLightfv()`
  - `GL_SPOT_DIRECTION`
  - `GL_SPOT_CUTOFF`
  - `GL_SPOT_EXPONENT`
  - `GL_*_ATTENUATION`
  - `glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, constant_attenuation);`
- Text: „Setzen Sie nun in der Funktion setLighting die vorgegebenen Variabeln auf sinnvolle Werte."
  - E-m@ail

Aufgabe 5

- Shadows
  - Not directly supported by OpenGL
- Algorithm for shadowing planes
  1. Draw scene without shadows
  2. Generate shadow projection matrices
  3. Render objects using shadow projection matrices
- Tricks
  - Axis aligned shadow receivers
  - Before projection: Shift origin to light source

  => simple „identity matrices"

Aufgabe 6

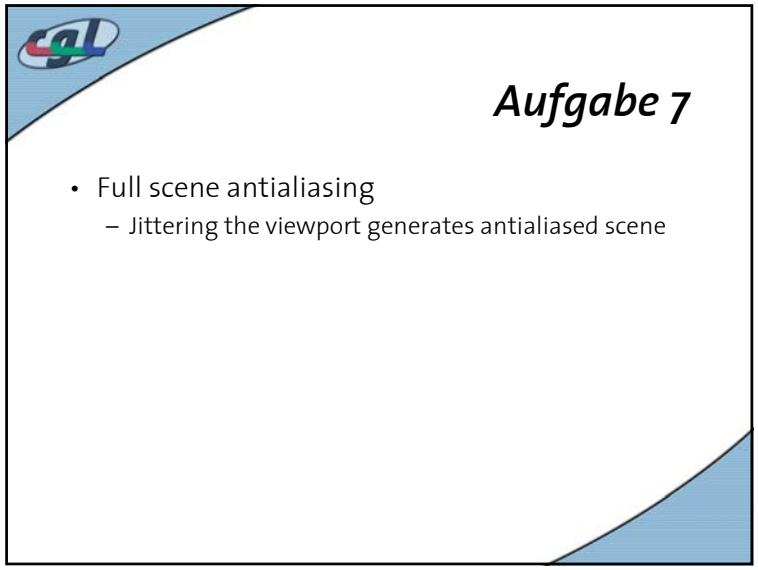
- Multipass – Shadows
  - Multiple spotlights
  - buildLightSource(...): 2-D array of lights

```
for(i = 0; i < nbr_y; i++) {
   for(j = 0; j < nbr_z; j++) {
      ...
}}
```

  - Render multiple images with jittered light position

Aufgabe 6

- Multipass – Accumulation buffer
  - Accumulating images generates soft shadows
  - `glAccum(GL_ACCUM, 1.0/(float)AL_NBR_OF_LIGHTS);`
  - `glAccum(GL_RETURN, 1.0);`


Aufgabe 7
• Full scene antialiasing
– Jittering the viewport generates antialiased scene