Chapter 13

Blending

Rendering Pipeline

Geometric Content → Model/View Transform. → Lighting → Perspective Transform. → Clipping

Scan Conversion → Texturing → Depth Test → Blending → Frame-buffer

Copyright Alla Sheffer
UBC 2012
Blending

- How might you combine multiple elements?
  - New color $A$, old color $B$

**Alpha Blending (OpenGL)**

- Parameters:
  - $s = \text{source color}$
  - $d = \text{destination color}$
  - $a_s = \text{source blend factor}$
  - $d' = a_s s + (1-a_s)d$

- Where
  - “Source” means “color/alpha of currently rendered primitive”
  - “Destination” means framebuffer value
Over operator

- \( d' = \alpha_s \cdot s + (1-\alpha_s) \cdot d \)
- Examples: \( a_A = 1 \) \( a_B = 0.4 \)

A over B:
\[ d' = 1 \cdot C_A + (1-1) \cdot C_B \]

B over A:
\[ d' = 0.4 \cdot C_B + (0.6) \cdot C_A \]
OpenGL Blending

In OpenGL:

- Enable blending
  - glEnable( GL_BLEND )
- Specify alpha channel for colors
  - glColor4f( r, g, b, alpha )
- Specify blending function
  - E.g: glBlendFunc( GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA )
    - $C = \text{alpha}_\text{new} \cdot C_{\text{new}} + (1 - \text{alpha}_\text{new}) \cdot C_{\text{old}}$
  - Other options available

Caveats:

- Note: alpha blending is an order-dependent operation!
  - It matters which object is drawn first AND
    - Which surface is in front
- For 3D scenes, this makes it necessary to keep track of rendering order explicitly
  - E.g. always draw “back” surface first
Double Buffering

- Framebuffer:
  - Piece of memory where the final image is written
  - Problem:
    - The display needs to read the contents, cyclically, while the GPU is already working on the next frame
    - Could result in display of partially rendered images on screen
  - Solution:
    - Have TWO buffers
      - Currently displayed (front buffer)
      - Render target for the next frame (back buffer)
## Double Buffering

- **Front/back buffer:**
  - Each buffer has both color channels and a depth channel
    - Important for advanced rendering algorithms
    - Doubles memory requirements!

- **Switching buffers:**
  - At end of rendering one frame, simply exchange the pointers to the front and back buffer
  - GLUT toolkit: glutSwapBuffers() function