Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

Module 8B
Stencils

Stencil Buffer

- Stencil buffer provides extra storage at a pixel
  - Separate from framebuffer (color) and depth buffer
  - Generally 8 bits
- Stenciling compares a value with a value stored in the stencil buffer
  - Can then modify the value of the stencil buffer
- Applications
  - Stenciling - only allow drawing in designated regions
  - Rendering silhouettes
  - Highlighting interference of polygonal solids
  - Preventing additional drawing of overlaying objects
    - Useful for transparency and shadow regions
  - Stippling
  - Shadow volumes
- **Stencil testing can be used to improve visual quality and add unique effects**

Stencil Testing

- Per-pixel test
  - Like depth buffering
  - Occurs during span/fragment drawing
  - Provides extra control of pixel update
- Tests against value from stencil buffer
  - Rejects fragment if stencil test fails
  - Distinct stencil operations performed when
    - Stencil test fails
    - Depth test fails
    - Depth test passes
- Supported by both OpenGL, DirectX, Vulkan, Metal
- Supported in most consumer graphics cards
  - Many of today's 32-bit graphics modes have 24-bit depth and 8-bit stencil packed in same memory word
    - If using depth testing, stenciling adds minimal (or no) cost

OpenGL Stencil Usage

- Use SDL to request stencil buffer
  - SDL_GL_SetAttribute(**SDL_GL_STENCIL_SIZE, 8**);
  - Requests stencil buffer
- Implementations may support from zero to 32 bits of stencil
  - 8, 4, or 1 bit are common possibilities
  - Query the state using SDL_GL_GetAttribute
  - **SDL_GL_GetAttribute(SDL_GL_STENCIL_SIZE, int* value)**
    - Populates ‘value’ with number of bits allocated to the stencil buffer
- Clear using glClear
  - **glClear(… | GL_STENCIL_BUFFER_BIT)**
  - Often clear when clearing framebuffer and depth buffer
- Enable and disable
  - glEnable(GL_STENCIL)
  - glDisable(GL_STENCIL)

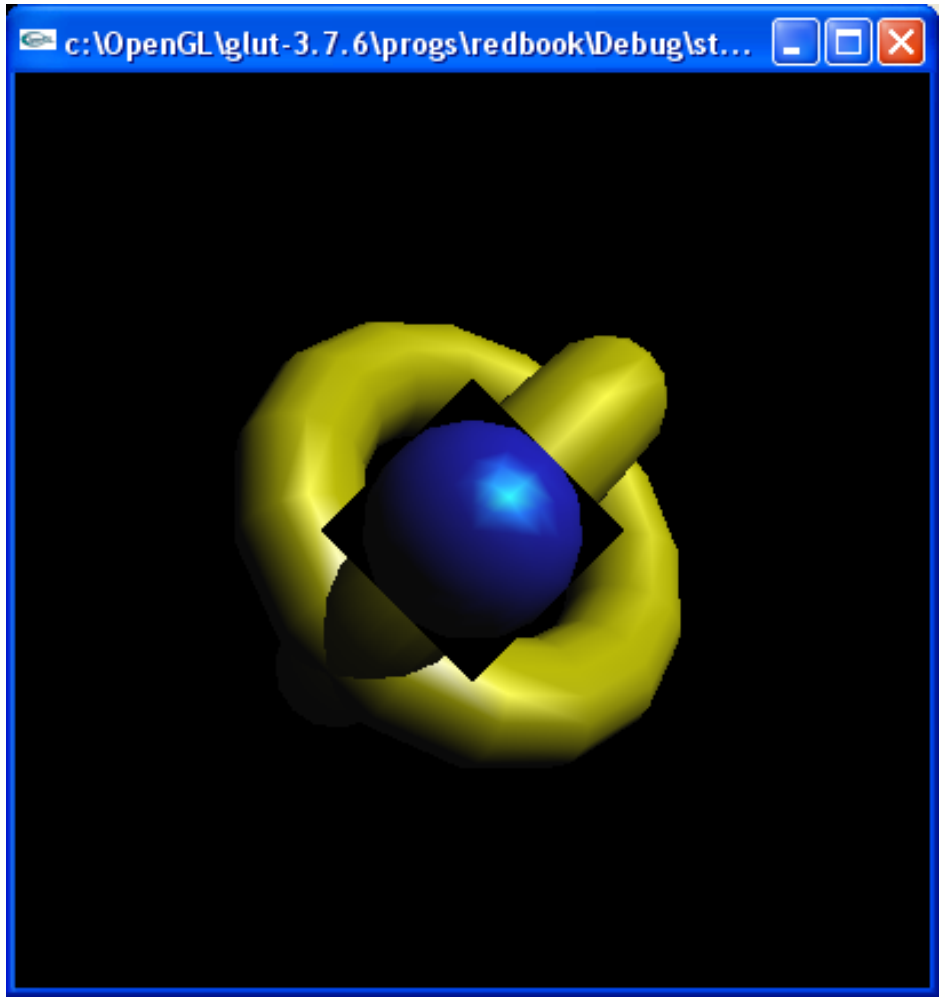
OpenGL Stencil Usage (cont.)

- Set the stencil mask
  - **glStencilMask(GLuint mask)**
  - Bit mask to control writing bits of stencil values; ‘1’ - writeable; ‘0’ - write protected
  - Also applies to the clear
  - Default is all 1’s
- Specify the stencil test
  - glStencilFunc(GLenum func, GLint ref, GLuint mask)
  - Sets the comparison function
    - GL_NEVER, GL_ALWAYS, GL_LESS, GL_LEQUAL, GL_EQUAL, GL_GEQUAL, GL_GREATER, GL_NOTEQUAL
    - Reference value is compared to the stencil buffer using the comparison function
      - Comparison only applies to bits that are set to 1 in the mask
  - Enable the test
    - **glEnable(GL_STENCIL_TEST)**

OpenGL Stencil Usage (cont.)

- Specify how the data in the stencil buffer is modified when a fragment passes or fails the stencil test
  - **glStencilOp(GLenum stencilFail, GLenum zfail, GLenum zpass)**
  - Set for each of the three tests
    - If stencil test passes, then either zfail or zpass is used
  - GL_KEEP – keep current value
  - GL_ZERO – set value to 0
  - GL_REPLACE – replace with the reference value
  - GL_INCR – increment value
    - Clamped to maximum value the stencil buffer can hold
  - GL_DECR – decrement value
    - Clamped to 0
  - GL_INVERT – replace with bitwise inversion of current value

Stencil Example



OpenGL Redbook: stencil.c
Two torus are drawn. A sphere is drawn in a diamond shaped stencil mask. The stencil mask prevents drawing a torus in diamond shape.