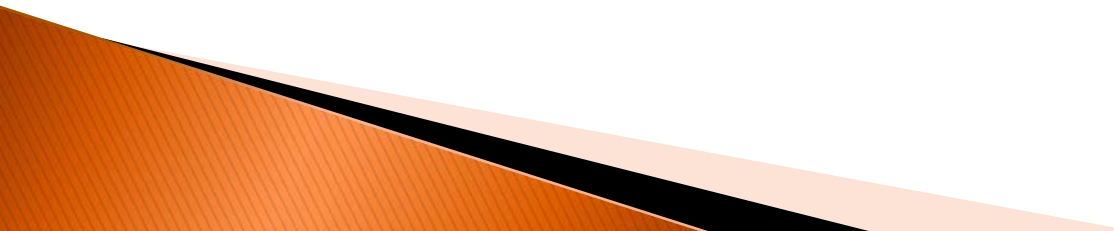


CSI 3202 Micro-Computer Graphics Buffers

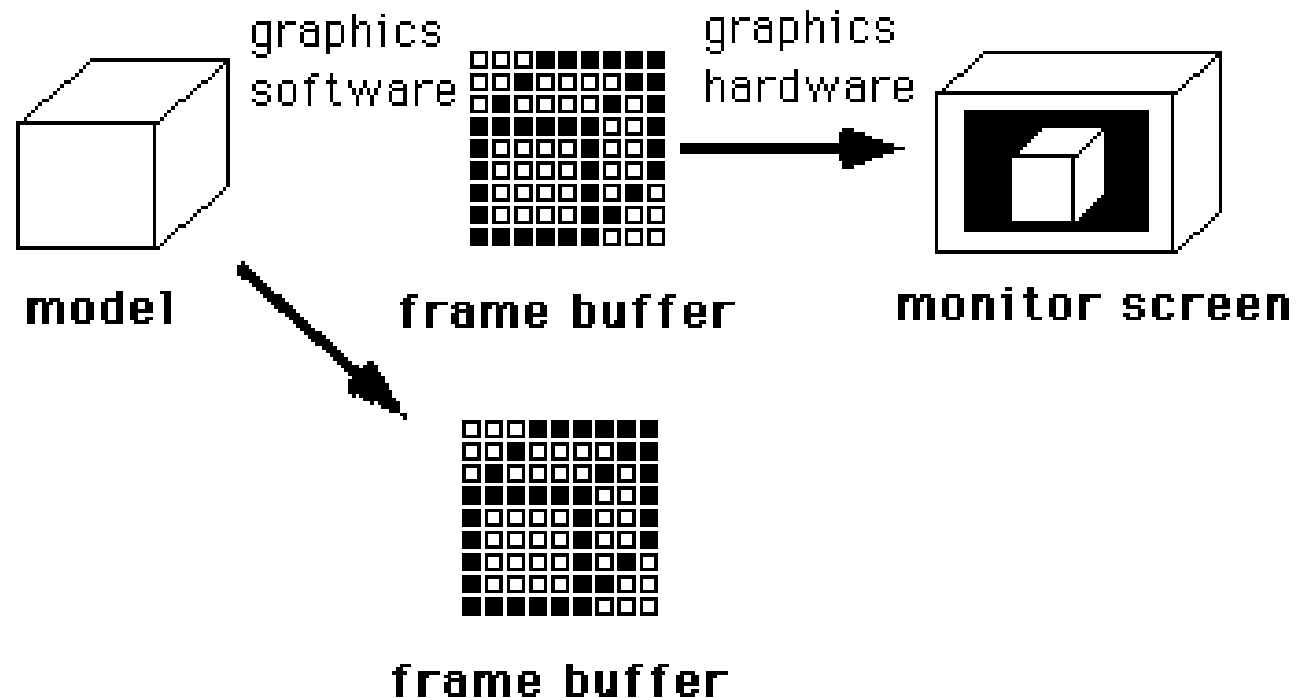
Presenter: Girendra Persaud
University of Guyana

Outline

- ▶ What is a CG buffer?
 - ▶ Frame buffer
 - ▶ Color buffers
 - ▶ Depth buffer
 - ▶ Stencil buffer
 - ▶ Accumulation buffer
 - ▶ Clearing buffers
 - ▶ Questions?
 - ▶ Review Questions
- 

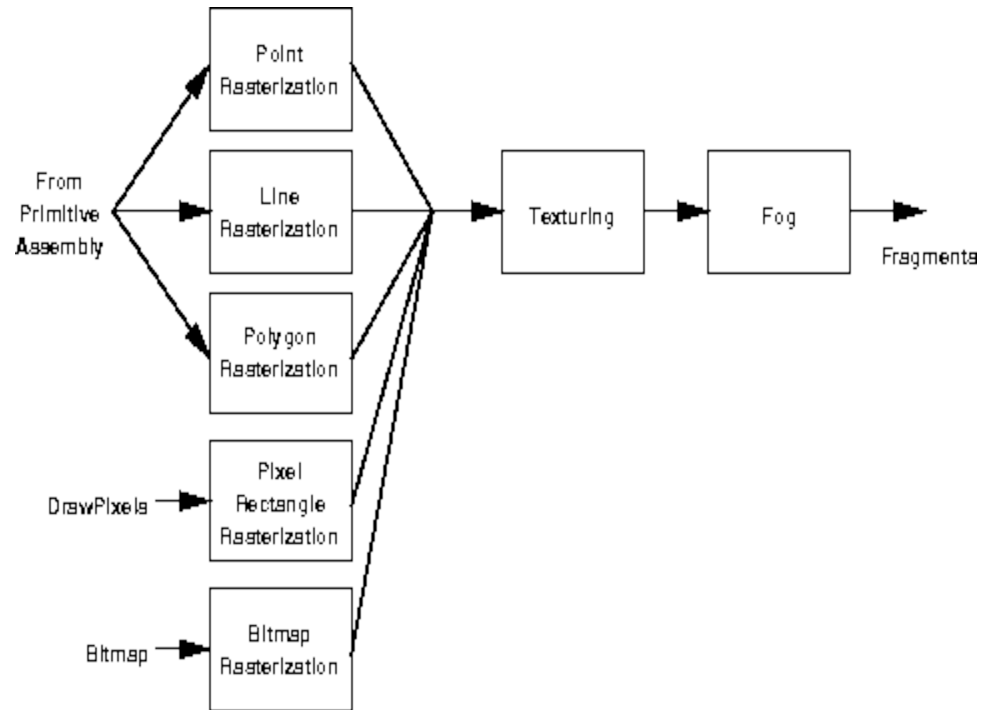
What is a CG buffer?

- ▶ Whenever data is stored uniformly for each pixel, such storage for all the pixels is called a *buffer* – these buffers make up the final image that is displayed on the screen




Frame buffer

- ▶ The frame buffer is a combination of all the buffers to form the final image to be displayed on the screen



After the rasterization stage (including texturing and fog), the data are not yet pixels, but are fragments. Each fragment has coordinate data which corresponds to a pixel, as well as color and depth values. Then each fragment undergoes a series of tests and operations, like blending etc.

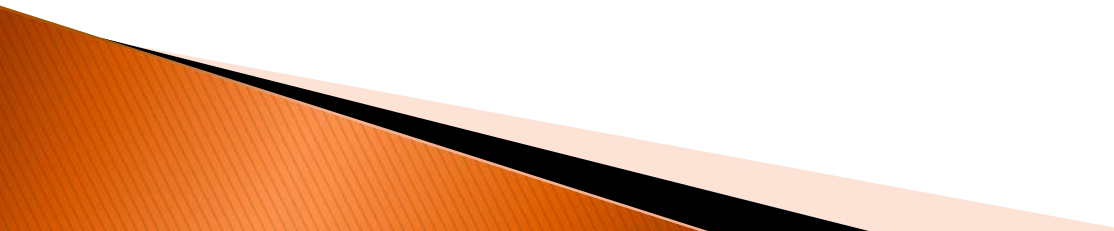
Frame buffer

- ▶ If the tests and operations are survived, the fragment values are ready to become pixels
 - ▶ To draw these pixels, you need to know what color they are, – the information that's stored in the color buffer.
- 

Data in Buffers

- ▶ Different buffers might contain different amounts of data per pixel
- ▶ In a given buffer, each pixel is assigned the same amount of data (1 bit, 4 bits, 16bits...)
- ▶ A buffer that stores a single bit of information about pixels is called a bitplane

CG Buffers

- ▶ An OpenGL system can manipulate the following buffers:
 - Color buffers: front-left, front-right, back-left, back-right, and any number of auxiliary color buffers
 - Depth buffer (z-buffer)
 - Stencil buffer
 - Accumulation buffer
- 

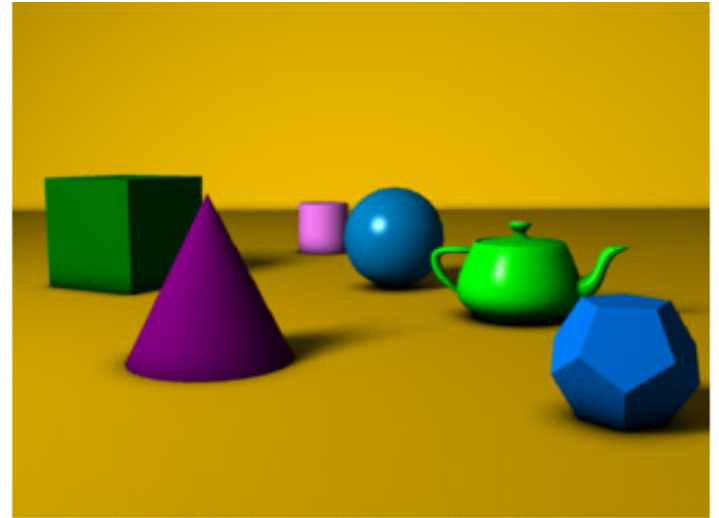
Color buffers

- ▶ Contain either color-index or RGB color data and may also contain alpha values
- ▶ stereoscopic viewing has **left** and **right** color buffers for the left and right stereo images.
- ▶ If stereo isn't supported, only the **left** buffers are used
- ▶ Similarly, double-buffered systems have **front** and **back** buffers, and a single-buffered system has the **front** buffers only

Depth buffer

- ▶ stores a depth value for each pixel
- ▶ depth is usually measured in terms of distance to the eye, so pixels with larger depth-buffer values are overwritten by pixels with smaller values
- ▶ Default value = Far or 1
- ▶ AKA *z-buffer*

Depth Buffer

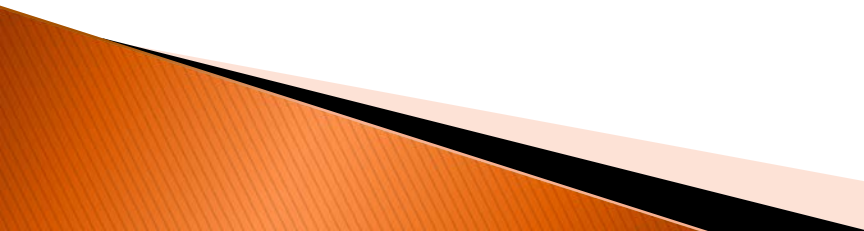


A simple three dimensional scene

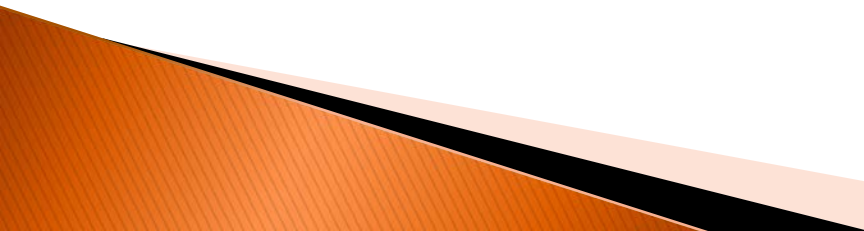


Z-buffer representation

Stencil buffer

- ▶ An extra buffer in addition to the color buffer and depth buffer
 - ▶ Used to restrict drawing to certain portions of the screen, just as a cardboard stencil can be used with a can of spray paint to make fairly precise painted images
- 

Stencil buffer

- ▶ For example, if you want to draw an image as it would appear through an odd-shaped windshield, you can store an image of the windshield's shape in the stencil buffer, and then draw the entire scene
 - ▶ The stencil buffer prevents anything that wouldn't be visible through the windshield from being drawn
- 

Stencil Buffer



Stencil Buffer Commands

- ▶ `glEnable(GL_STENCIL_TEST);` // by default not enabled
- ▶ `glStencilMask(stencilMask);` // allow writing to stencil buffer, by default (0xFF) no mask.
- ▶ `glClearStencil(clearStencilValue);` // clear stencil value, by default = 0
- ▶ `glStencilFunc(func, ref, mask);` // by default GL_ALWAYS, 0, 0xFF, always pass stencil test
- ▶ `glStencilOp(fail, zfail, zpass);` // by default GL_KEEP, GL_KEEP, GL_KEEP, dont change stencil buffer
- ▶ `glClear(GL_STENCIL_BUFFER_BIT);` // clear stencil buffer, fill with (clearStencilValue & stencilMask)

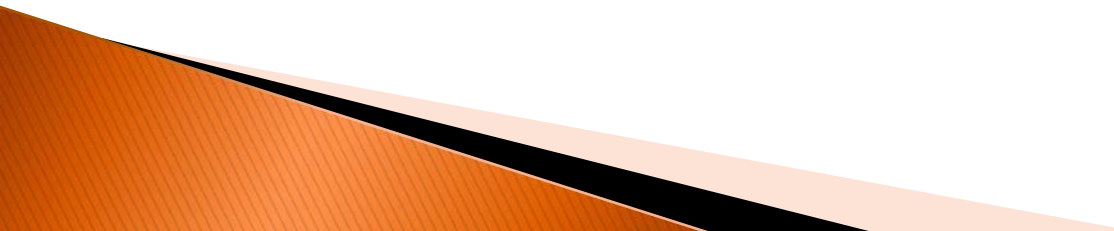
Accumulation Buffer

- ▶ The accumulation buffer holds RGBA color data just like the color buffers do in RGBA mode
- ▶ It's typically used for accumulating a series of images into a final, composite image.
 - you can perform operations like scene antialiasing
- ▶ accumulation operations are always performed in rectangular blocks, which are usually transfers of data to or from a color buffer

Clearing buffers

- ▶ In graphics programs, clearing the screen (or any of the buffers) is typically one of the most expensive operations you can perform
- ▶ The following commands set the clearing values for each buffer.
 - void **glClearColor**(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha);
 - void **glClearIndex**(GLfloat index);
 - void **glClearDepth**(GLclampd depth);
 - void **glClearStencil**(GLint s);
 - void **glClearAccum**(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha);
- ▶ The default depth-clearing value is 1.0; all the other default clearing values are 0. The values set with the clear commands remain in effect until they're changed by another call to the same command
- ▶ To clear all buffers – **glClear()**.

Review

- ▶ What is a CG buffer?
 - ▶ Frame buffer
 - ▶ Color buffers
 - ▶ Depth buffer
 - ▶ Stencil buffer
 - ▶ Accumulation buffer
 - ▶ Clearing buffers
- 

Questions?

Resources

- ▶ <http://glprogramming.com/red/chapter10.html>

Review Question

- ▶ Why is the default clear value of the depth buffer is 1 and not 0 or -1 ?
- ▶ Is the Frame buffer the same as the accumulation buffer?
- ▶ Data in the _____ buffer is changed when we do screen antialiasing