Computer Graphics Framebuffers

Mike Bailey
mjb@cs.oregonstate.edu

The Framebuffers

- Update
- Refresh
- Video Driver

The Viewer sees the contents of the front framebuffer.

The Video Driver

- N refreshes/second (N is between 50 and 120, 60 is common)
- The framebuffer contains the R,G,B that define the color at each pixel
- Because of the double-buffering, Refresh is asynchronous from Update, that is, the monitor gets refreshed at N (60) frames per second, no matter how fast or slowly you update the back buffer.

The Framebuffer Uses RGB Colors

- Red
- Yellow
- White
- Green
- Cyan
- Blue
- Magenta
## The Framebuffer: Integer Color Storage

<table>
<thead>
<tr>
<th># Bits/color</th>
<th># Intensities per color</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$2^8 = 256$</td>
</tr>
<tr>
<td>10</td>
<td>$2^{10} = 1024$</td>
</tr>
<tr>
<td>12</td>
<td>$2^{12} = 4096$</td>
</tr>
</tbody>
</table>

- **Typical**
- **High Dynamic Range (HDR)**

### # Bits/pixel Total colors:
- 24: $2^{24}$ = 16.7 M
- 30: $2^{30}$ = 1 B
- 36: $2^{36}$ = 69 B

## Why so many bits?

Many modern algorithms do arithmetic on the framebuffer color components or treat the framebuffer color components as data. They need the extra precision during the arithmetic.

However, the display system cannot produce all of those possible colors.

## The Framebuffer: Floating Point Color Storage

- 16- or 32-bit floating point for each color component

**Why many bits?**

Many modern algorithms do arithmetic on the framebuffer color components or treat the framebuffer color components as data. They need the extra precision during the arithmetic.

However, the display system cannot produce all of those possible colors.

### Why do things in front look like they are really in front?

**Solution #1**: Sort your polygons in 3D by depth and draw them back-to-front. In this case 1-2-3-4-5-6 becomes 5-6-2-4-1-3.

This is called the **Painter's Algorithm**. Once upon a time, we had to do things this way. It sucked even more than it sounds.

**Solution #2**: Add an extension to the framebuffer to store the depth of each pixel. This is called a **Depth-buffer or Z-buffer**. Only allow pixel stores to take place when the depth of the incoming pixel is closer to the viewer than the pixel that is already there.
Incoming RGBZ from the application

Existing RGBZ in the framebuffer

Compare $Z_{\text{incoming}}$ closer to the viewer than $Z_{\text{existing}}$?

- No
- Yes

Do nothing

Allow $Z_{\text{incoming}}$ to overwrite $Z_{\text{existing}}$?

- No
- Yes

Why do things in front look like they are really in front?

With Depth Buffer

Without Depth Buffer