The Framebuffers

**The Framebuffer Uses RGB Colors**

- Red
- Yellow
- Green
- Magenta
- White
- Cyan
- Blue

**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)

The following tables show the total colors for different bit settings:

<table>
<thead>
<tr>
<th># Bits/pixel</th>
<th>Total colors:</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>$2^{24} = 16.7$ M</td>
</tr>
<tr>
<td>30</td>
<td>$2^{30} = 1$ B</td>
</tr>
<tr>
<td>36</td>
<td>$2^{36} = 69$ B</td>
</tr>
</tbody>
</table>
The Framebuffer: Floating Point Color Storage

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

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 display all of those possible colors.

The Framebuffer

- Alpha values
  - Transparency per pixel
    - $\alpha = 0.$ is invisible
    - $\alpha = 1.$ is opaque
  - Represented in 8-32 bits (integer or floating point)
  - Alpha blending equation:

$$\text{Color} = \alpha C_1 + (1 - \alpha) C_2$$

$0.0 \leq \alpha \leq 1.0$

Note: this is really blending, not transparency!

The Framebuffer

- Z-buffer
  - Used for hidden surface removal
  - Holds pixel depth
  - Typically 32 bits deep
  - Integer or floating point

# Bits / Z

Total Z Values:

32 $2^{32} = 4B$

glutSwapBuffers()

// swap the double-buffered framebuffers:

```cpp
glutSwapBuffers();
```

```cpp
glutInitDisplayMode( GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH );
```

```cpp
glDrawBuffer( GL_BACK );
```

You draw into here

This is called the **update**

The monitor displays from here

"swap buffers" changes the role of the two framebuffers

This is called the **refresh**

You draw into here

The monitor displays from here
The viewer sees the contents of this framebuffer.

- **N refreshes/second** (N is between 50 and 100)
- 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.