Computer Graphics Framebuffers

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License
The Framebuffers

Video Driver

Update

Refresh

Front

Back

Double-buffered Color Framebuffers

Depth-Buffer
The Framebuffer Uses RGB Colors
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>

<table>
<thead>
<tr>
<th># Bits/pixel</th>
<th>Total colors:</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>$2^{24} = 16.7 \text{ M}$</td>
</tr>
<tr>
<td>30</td>
<td>$2^{30} = 1 \text{ B}$</td>
</tr>
<tr>
<td>36</td>
<td>$2^{36} = 69 \text{ B}$</td>
</tr>
</tbody>
</table>

Oregon State University
Computer Graphics
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

\[
\text{Total Z Values: } 32, \quad 2^{32} = 4 \text{ B}
\]
glutSwapBuffers( )

// swap the double-buffered framebuffers:

    glutSwapBuffers();

    glutInitDisplayMode( GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH );

    glDrawBuffer( GL_BACK );

You draw into here

This is called the *update*

"swap buffers" changes the role of the two framebuffers

The monitor displays from here

This is called the *refresh*

You draw into here

The monitor displays from here
The Video Driver

The viewer sees the contents of this framebuffer.
The Video Driver

- \( N \) 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.