

# Handouts: Raster Graphics Hardware

Tidy up: Hardware & OpenGL  
Start: Software



Questions?



- A1:
  - random/new/used colors ...
  - ambiguity

# Handouts: Raster Graphics Hardware

## A quick OpenGL addition

---



- Drawing rectangles:
  - `glRect{sidf}( TYPE x1, y1, x2, y2);`
  - `glRect{sidf}v( TYPE *v1, *v2);`

## Creating fragments

---



- `glBegin (type)`
  - `GL_LINES, GL_POINTS, GL_TRIANGLES,`  
`GL_QUADS, GL_POLYGON, ... (p.43)`
  
- `glEnd ();`

# Handouts: Raster Graphics Hardware

## Specifying points

---



■ `glVertex{234}{sifd}( TYPE coords);`

## FYI, ...

---



```
glRect(x1,y1,x2,y2);
```

precisely equals

```
glBegin (GL_POLYGON);
```

```
    glVertex2(x1, y1);
```

```
    glVertex2(x2, y1);
```

```
    glVertex2(x2, y2);
```

```
    glVertex2(x1, y2);
```

```
glEnd();
```

# Handouts: Raster Graphics Hardware

## Back to Graphics Hardware

---



- Recap:
  - Framebuffer (Hardware, OpenGL)
  - CRTs
- Why 3 guns for color?
- What can the framebuffer look like?
- How does it get displayed?

## Color CRTs use three colors... Why? Why RGB?

---

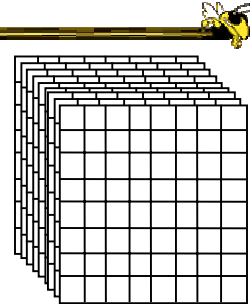


- "It's enough?"
  - Our eyes have 3 kinds of receptors
- Not really enough!
  - Colors that can't be displayed
  - Colors that can't be printed
- Talk about this more later (Color & Vision)

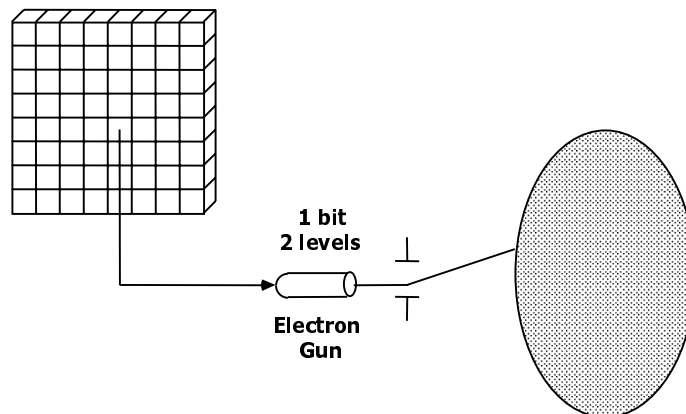
# Handouts: Raster Graphics Hardware

## Frame Buffers

- 2D array
  - each (x,y) location = a pixel
- *Bit Planes, Bit Depth*
  - number of bits in a pixel
- Typical frame buffers:
  - 640 x 480 x 8
  - 1280 x 1024 x 8
  - 1280 x 1024 x 24

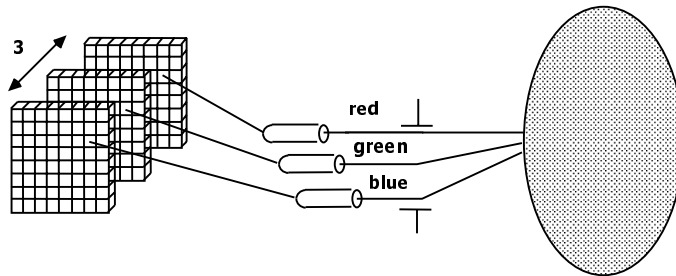


## 1-Bit = Monochrome Display (Bit-map Display)



# Handouts: Raster Graphics Hardware

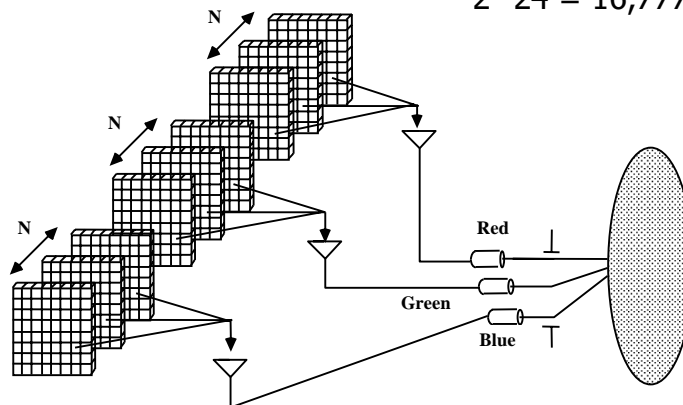
## 3-Bit Color Display



COLOR: black red green blue yellow cyan magenta white

R	0	1	0	0	1	0	1	1
G	0	0	1	0	1	1	0	1
B	0	0	0	1	0	1	1	1

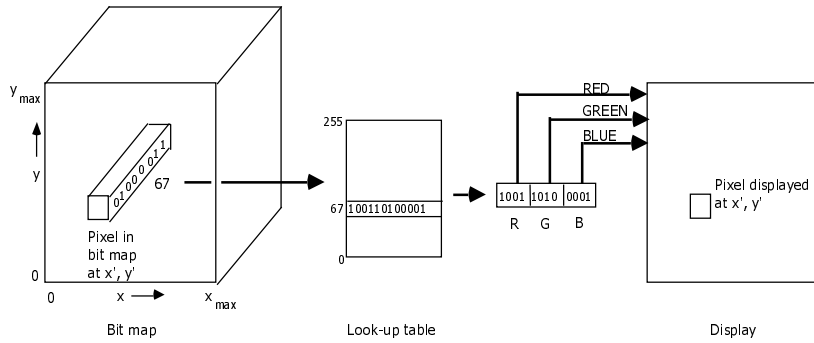
## True Color Display 24 bitplanes, 8 bits R/G/B



$$2^{24} = 16,777,216$$

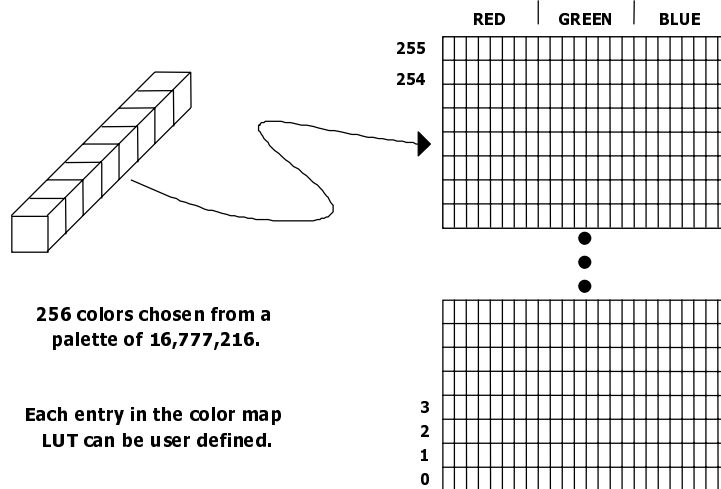
# Handouts: Raster Graphics Hardware

## Color Map Look-Up Tables



**Fig. 4.LUT Video look-up table organization. A pixel with value 67 (binary 01000011) is displayed on the screen with the red electron gun at 9/15 of maximum, green at 10/15, and blue at 1/15. This look-up table is shown with 12 bits per entry. Up to 24 bits per entry are common.**

## Pseudo Color: $2^8 \times 24$ Color Map LUT



256 colors chosen from a palette of 16,777,216.

Each entry in the color map LUT can be user defined.

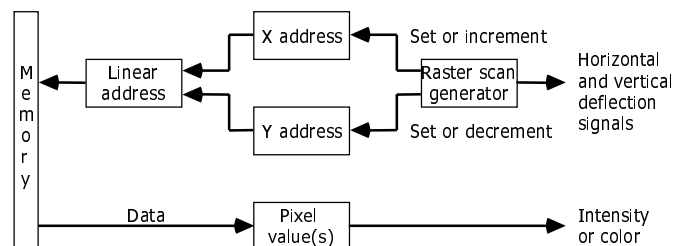
# Handouts: Raster Graphics Hardware

## Display Processor

- Specialized hardware
  - i.e. scan converts primitives into frame buffer
- Fundamental difference between graphics systems
  - work done by display processor vs. CPU

## Video Controller

- Cycles through frame buffer
  - FB contents used to control the electron beam intensity (color)



# Handouts: Raster Graphics Hardware

## Input Hardware: Logical Devices

---



- Locator
  - position and/or orientation
- Keyboard
  - characters and strings
- Valuator
  - single values in the space of real numbers
- Choice
  - select from a set of actions or choices

## Physical Device Examples

---



- Locator Devices
  - Tablet, mouse, trackball, touch panel, Light pen
- Keyboard devices
  - Alphanumeric keyboard (coded or unencoded)
- Valuator Devices
  - Rotary dials (bounded or unbounded), sliders
- Choice Devices
  - Function keys

# Handouts: Raster Graphics Hardware

## GLUT Devices

---



### ■ Locator Devices: "rodents"

```
void glutMotionFunc(void (*func)(int x, int y));  
void glutPassiveMotionFunc(void (*func)(int x, int y));  
void glutMouseFunc(void (*func)(int button, int state,  
                             int x, int y));  
void glutEntryFunc(void (*func)(int state));
```

### ■ Keyboard devices

```
void glutKeyboardFunc(void (*func)(unsigned char key,  
                                  int x, int y));  
int glutGetModifiers(void);
```

### ■ Choice Devices

```
void glutSpecialFunc(void (*func)(int key, int x, int y));
```

## Other GLUT supported devices

---



### ■ Locator Devices

- SpaceBall, Tablet

### ■ Valuator Devices

- Dials

### ■ Choice Devices

- ButtonBox

### ■ Checking devices:

- `int glutDeviceGet(GLenum info);`