Introduction to 2D and 3D Computer Graphics

Understanding Raster Architectures

• Introduction to raster graphics:
• Pixels & bitmaps
• Block moves & raster operations
• Drawing modes and transparency
• Pipeline

Discuss Programming Project, Selection of Groups, and Research Work
• Questions?
The objective for today’s lecture is to examine what is going on at a fairly low-level when we request a drawing operation to be performed.

Please note, you are definitely NOT restricted to just manipulating pixels at this low-level in your projects.

We will move rather quickly through the material and also cover material from packet #1 today.

*If time, a scientific visualization video will give us insights as to when “voxels” (low-level) versus object models (higher level should be used).*
Using Raster Graphics

Some Thoughts

- Like programming methodologies
  - raster graphics must be examined in relation to its underlying semantics
  - how do we justify what is “good” or “realistic”
  - the perceived quality of rendering can depend on our perception, the evaluation criteria, the viewing monitor (is it properly adjusted), the viewing-room illumination, depth of the pixmap
In addition,

- just because a graphics system performs an operation that is desired, it may not perform that operation in the fashion desired
- therefore, we must look beyond the surface and examine the underlying semantics behind each and every operation
- for example, are line segment endpoint’s included in the line segment drawn? (Sproull says :”yes”)

Using Raster Graphics
Some Thoughts
Using Raster Graphics

Pixels

- Pixels...
  - are the smallest addressable unit on a raster graphics device
  - can be thought of as an individually-addressable dot
  - stands for: picture element
  - are grouped into rectangular arrays creating raster output
- Monochrome is "1-bit per pixel"
- For devices supporting shades of gray or color...
  - pixels are considered to have depth
Using Raster Graphics

Pixels

- Pixels...
  - are the smallest modifiable element of a raster device
  - are device-dependent
  - have a 1-to-1 correspondence with the device coordinates of a raster device
  - contain one or more bits of information
  - can independently be assigned a value representing an intensity or a color
Using Raster Graphics
Pixels -- Shape

- When pixels are circular...
  ...no distortion occurs
When pixels are elliptical...
...distortion can occur (implicit display xform)
...functions that use device coordinates are affected
Using Raster Graphics

Pixels – Location relative to coordinates

Pixels on device coordinates

Pixels between device coordinates
Using Raster Graphics

*Pixels – Location relative to coordinates*

- For lines:

  Pixels on device coordinates

  Pixels between device coordinates
Using Raster Graphics

Pixels – Location relative to coordinates

• Guidelines for areas results in...
...the same number of pixels:

Pixels on device coordinates

Pixels between device coordinates
Using Raster Graphics

Pixels – Location relative to coordinates

• Not using these guidelines for areas results in... 
  ...different number of pixels:

Pixels on device coordinates

Pixels between device coordinates
Resolution...

...on a raster graphics device is a measure of the number of pixels per unit of distance along the X and Y dimensions

...is not necessarily the same in the X and Y dimensions

...can be, for example, 1024x768 (1K by 768), 2Kx2K, etc.
Using Raster Graphics

**Bitmaps**

- A bitmap is a region of computer memory containing pixels
- Bitmaps can be...
  - displayable bitmaps
  - nondisplayable bitmaps
  - application-created bitmaps
  - drawing bitmaps
Using Raster Graphics

**Bitmaps**

- Bitmaps...
  - are uniquely identified by an identifier
  - are therefore distinguishable
  - are controllable by the application

- Bitmaps can be...
  - created
  - deleted
  - selected to be displayed out of
  - selected to be drawn into
Using Raster Graphics

**Bitmaps - Displayable**

- Displayable bitmaps...
  - can be displayed on the display surface of a raster device
  - are predefined by the implementation
  - can also be created by the application
  - always exist
  - support color the same as the physical device:
    - direct
    - indexed
Nondisplayable bitmaps...
...cannot be displayed on the display surface of a device
...may need to be copied to a displayable bitmap to be seen
...are not predefined by the implementation
...can be created by the application

Two types of nondisplayable bitmaps exist...full depth, indexed
Full depth bitmaps have the...
...same number of bits per pixel as displayable bitmaps
...same color capabilities as displayable bitmaps (either direct color or indexed color)

Full depth bitmaps are useful when the implementation cannot create additional displayable bitmaps
Using Raster Graphics

**Bitmaps - Mapped nondisplayable**

- Mapped bitmaps have...
  - only one bit per pixel
  - a foreground value
  - a background value

- Mapped bitmaps are useful for storing graphics with only foreground and background information
For example, a mapped bitmap is an efficient way to store information such as bitmap character fonts, hatch styles, and black and white patterns.
The background color...

...specifies the color that will be saved as the background value when writing into a mapped bitmap

...saves all other colors as the foreground value when writing into a mapped bitmap

...specifies the color that the background value will assume when reading from a mapped bitmap
Using Raster Graphics

Bitmaps - Mapped nondisplayable

- The foreground color...
  - ...specifies the color that the foreground value will assume when reading from a mapped bitmap
- Colors can be lost when stored in a mapped bitmap
The drawing bitmap...
...is the bitmap into which pixels are written
...can be either displayable or nondisplayable
...cannot be seen unless it is also selected to be the display bitmap as well

The drawing bitmap may be different than the bitmap being displayed
Raster operation functions... (rasterops)
  ...manipulate rectangular regions of pixels

Raster operation functions are...
  ...Bitblt functions
  ...Pixel array functions

Bitblt functions move and combine rectangular regions of pixels between bitmaps
Using Raster Graphics

Raster Operations

- Pixel array functions move or combine rectangular regions of pixels between a bitmap and an array

- RASTER OPERATION FUNCTIONS OPERATE ON PIXEL DATA. THEY DO NOT MANIPULATE GRAPHIC OBJECTS. THERE IS NO INHERENT STRUCTURE ASSOCIATED WITH THE DATA ON WHICH THEY OPERATE
Using Raster Graphics

Raster Operations -- Bitblts

- Bitblt functions... (we called them pixblt)
  ...combine rectangular regions of pixels from...
  ...one or two source bitmaps to...
  ...a destination bitmap

- The source bitmap can be any bitmap...
  ...displayable
  ...full depth nondisplayable
  ...mapped nondisplayable
Using Raster Graphics

Raster Operations - Source Destination Bitblts

- Two-operand bitblt has...
  ...one source bitmap
  ...one destination bitmap

- The pixels from the source bitmap are moved to and combined with pixels in the destination bitmap starting at the destination origin point
Using Raster Graphics
Raster Operations - Source Destination Bitblts

- Three-operand bitblt has...
  - one source bitmap
  - one pattern bitmap
  - one destination bitmap
Using Raster Graphics

Raster Operations - Source Destination Bitblts

Drawing Mode = OR; Transparency = OPAQUE

source origin  Source bitmap  destination origin  Drawing Bitmap before bitblt

Drawing Bitmap after bitblt
Using Raster Graphics

Raster Operations - Three Operand Bitblts

Drawing Mode = OR; Transparency = OPAQUE
Drawing modes specify how pixels representing a graphic object are combined with pixels in the drawing bitmap during rendering.

Drawing modes...

...work with a source pixel (value of the pixel representing the graphic object)

...destination pixel (value of the pixel in the drawing bitmap before combination with the graphic object)
Drawing mode combines the source and destination pixels and places the result in the drawing bitmap.

Three common drawing mode classes are...

- BOOLEANOP
- ADDITIVEOP
- COMPARATIVEOP
Using Raster Graphics

Drawing Modes - BOOLEANOP

• 0) $d' = 0$
• 1) $d' = s \text{ AND } d$
• 2) $d' = s \text{ AND } \neg d$
• 3) $d' = s$
• 4) $d' = \neg s \text{ AND } d$
• 5) $d' = d$
• 6) $d' = s \text{ XOR } d$
• 7) $d' = s \text{ OR } d$

s = source pixel
$d = \text{destination pixel before operation}$
$d' = \text{destination pixel after operation}$
8) $d' = \neg (s \lor d)$
9) $d' = \neg (s \oplus d)$
10) $d' = \neg d$
11) $d' = s \lor (\neg d)$
12) $d' = \neg s$
13) $d' = (\neg s) \lor d$
14) $d' = \neg (s \land d)$
15) $d' = 1$
Using Raster Graphics

Drawing Modes - BOOLEANOP

Graphic object written to the Drawing Bitmap

Existing picture in the Drawing Bitmap

Result when Drawing Mode is set to OR

Result when Drawing Mode is set to AND
Using Raster Graphics

Transparency Examples

Source for pixels

Drawing Bitmap before raster operation

Drawing Bitmap after raster operation
Using Raster Graphics

Transparency Examples

Using Opaque Transparency
Using Raster Graphics

*Transparency Examples*

Using Transparent Transparency

- Fill Bitmap
- Auxiliary Colour (white)
- Fill Bitmap Region
- Fill Reference Point
- Drawing Bitmap
**Components common with 2D and 3D graphic object pipelines**

- Raster Pipeline

- Components common with 2D and 3D graphic object pipelines

- Drawing Surface Clipping
- Apply
- Render Physically
- Colour Table
- Apply
- Implicit Display Transformation
- Echo
- Apply
- Display Surface

Graphic Objects
Raster Pipeline

- Drawing bitmap connection
- Display bitmap connection
- Raster operation source bitmap connection

Graphic Objects:
- Drawing Surface Clipping
- Render Physically
- Apply
- Drawing Bitmap
- Non Displayable Bitmap
- Non Displayable Bitmap
- Colour Table
- Apply
- Apply Implicit Display Transformation
- Display Bitmap
- Echo
- Display Surface
- Display Bitmap
- Display Bitmap
Raster Pipeline

Example: bitmap connections

- Drawing bitmap connection
- Display bitmap connection
- Raster operation source bitmap connection

Graphic Objects

PIXEL ARRAY Function

BITBLT Functions

Displayable Bitmap

Colour Table

Apply Implicit Display Transformation

Display Bitmap

Echo

Source Bitmap(s)

Non Displayable Bitmap

Non Displayable Bitmap