



GALGOTIAS  
UNIVERSITY

**School of Computing  
Science and Engineering**

Program: B.C.A.

Course Code: BCAS3003

Course Name: Computer Graphics

## **Course Prerequisites**

- Knowledge of Mathematics**
- Fundamental knowledge of Computer**

# Syllabus

## Unit 3 – Attributes of Output Primitives

**(8 hours)**

- Line Attributes**
- Curve Attributes**
- Color and Gray-Scale levels**
- Area-Fill Attributes**
- Character Attributes**
- Bundled attributes**
- Inquiry functions.**

## **Recommended Books**

### **Text books**

- ❑ D. Hearn, P. Baker, "Computer Graphics - C Version", 2nd Edition, Pearson Education, 1997

### **Reference Book**

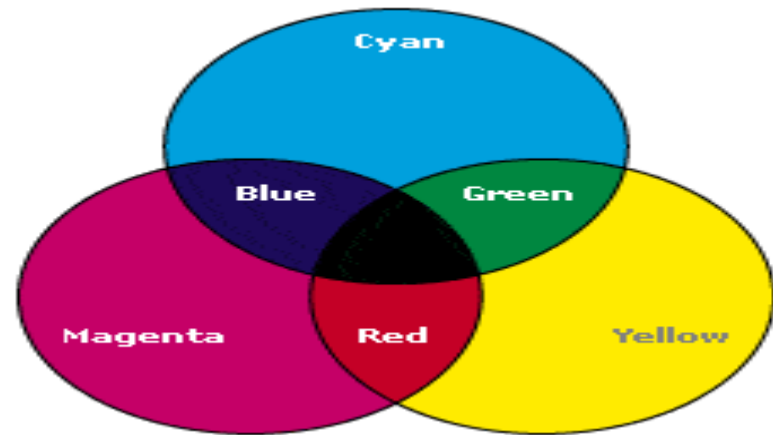
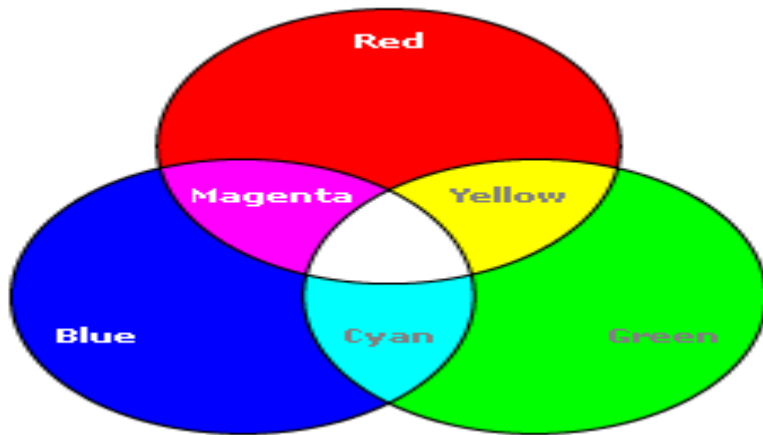
- ❑ Heam Donald, Pauline Baker M: "Computer Graphics", PHI 2nd Edn. 1995.
- ❑ Harrington S: "Computer Graphics - A Programming Approach", 2nd Edn. Mc GrawHill.
- ❑ Shalini Govil-Pai, Principles of Computer Graphics, Springer, 2004

### **Additional online materials**

- ❑ Coursera - <https://www.coursera.org/learn/fundamentals-of-graphic-design>
- ❑ <https://www.youtube.com/watch?v=fwzYuhduME4&list=PLE4D97E3B8DB8A590>
- ❑ NPTEL - <https://nptel.ac.in/courses/106/106/106106090/>
- ❑ <https://www.coursera.org/learn/research-methods>
- ❑ <https://www.coursera.org/browse/physical-science-and-engineering/research-methods>

## Color Models

- ❑ There are several established color models used in computer graphics, but the two most common are the RGB model (Red-Green-Blue) for computer display and the CMYK model (Cyan-Magenta-Yellow-black) for printing.
- ❑ **RGB Color Model:** Additive color model; For computer displays; Uses light to display color; Colors result from transmitted light,  $\text{Red} + \text{Green} + \text{Blue} = \text{White}$
- ❑ **CMYK Color Model:** Subtractive color model; For printed material; Uses ink to display color; Colors result from reflected light;  $\text{Cyan} + \text{Magenta} + \text{Yellow} = \text{Black}$



## **Attributes of Output Primitives**

- Any parameter that affects the way a primitive is to be displayed is referred to as an attribute parameter.
- Example attribute parameters are color, size etc.
- A line drawing function for example could contain parameter to set color, width and other properties.
- Line Attributes, Curve Attributes, Color and Grayscale Levels, Area Fill Attributes, Character Attributes, Bundled Attributes

## Area-Fill Attributes

- Options for filling a defined region include a choice between a solid color or a pattern fill and choices for particular colors and patterns

### Fill Styles

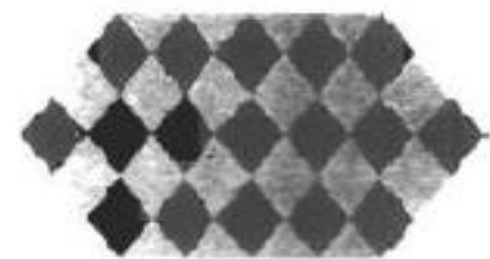
- Areas are displayed with three basic fill styles: hollow with a color border, filled with a solid color, or filled with a specified pattern or design. A basic fill style is selected in a PHIGS program with the function **setInteriorStyle (fs)**



Hollow



Solid



Patterned

Diagonal  
Hatch FillDiagonal  
Cross-Hatch Fill

# Area-Fill Attributes

## Fill Styles

- Values for the fill-style parameter `fs` include hollow, solid, and pattern.
- Another value for fill style is hatch, which is used to fill an area with selected hatching patterns parallel lines or crossed lines
- The color for a solid interior or a hollow area outline is chosen with where fill color parameter `fc` is set to the desired color code

**`setInteriorColourIndex (fc)`**



# Area-Fill Attributes

## Pattern Fill

- ❑ We select fill patterns with `setInteriorStyleIndex` (`pi`) where pattern index parameter `pi` specifies a table position
- ❑ For example, the following set of statements would fill the area defined in the `fillArea` command with the second pattern type stored in the pattern table:

```
SetInteriorStyle( pattern)
SetInteriorStyleIndex(2);
Fill area (n, points)
```

<i>Index (pi)</i>	<i>Pattern (cp)</i>
1	$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
2	$\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$

## Character Attributes

- The appearance of displayed character is controlled by attributes such as font, size, color and orientation.
- Attributes can be set both for entire character strings (text) and for individual characters defined as marker symbols.

## Text Attributes

- ❑ The choice of font or type face is set of characters with a particular design style as courier, Helvetica, times roman, and various symbol groups.
- ❑ The characters in a selected font also be displayed with styles. (solid, dotted, double) in bold face in italics, and in outline or shadow styles.
- ❑ A particular font and associated style is selected in a PHIGS program by setting an integer code for the text font parameter `tf` in the function

`setTextFont (tf)`

- ❑ Control of text color (or intensity) is managed from an application program with **`setTextColourIndex (tc)`**

Where text color parameter `tc` specifies an allowable color code.

## Text Attributes

- ❑ Text size can be adjusted without changing the width to height ratio of characters with

`setCharacterHeight(ch)`

Where parameter `ch` is assigned a real value greater than 0 to set the coordinate height of capital letters

- ❑ The width only of text can be set with function.

`setCharacterExpansionFactor(cw)`

Where the character width parameter `cw` is set to a positive real value that scales the body width of character

Height 1

Height 2

Height 3

width 0.5

width 1.0

width 2.0

## Text Attributes

- ❑ Spacing between characters is controlled separately with set

CharacterSpacing (cs)

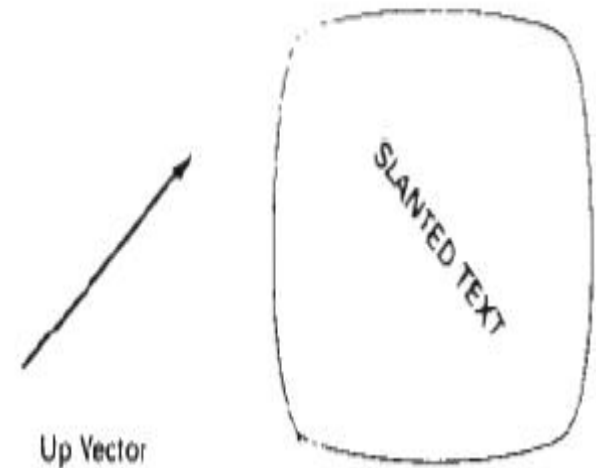
Where the character-spacing parameter cs can be assigned any real value.

- ❑ The orientation for a displayed character string is set according to the direction of the character up vector

setCharacterUpVector (upvect)

Parameter upvect in this function is assigned two values that specify the x and y vector components. For example, with upvect = (1, 1), the direction of the up vector is  $45^\circ$  and text would be displayed as shown in Figure.

Spacing 0.0  
Spacing 0.5  
Spacing 1.0



## Text Attributes

- Another handy attribute for character strings is alignment. This attribute specifies how text is to be positioned with respect to the \$start coordinates. Alignment attributes are set with

`setTextAlignment (h,v)`

where parameters `h` and `v` control horizontal and vertical alignment. Horizontal alignment is set by assigning `h` a value of `left`, `center`, or `right`. Vertical alignment is set by assigning `v` a value of `top`, `cap`, `half`, `base` or `bottom`.

- A precision specification for text display is given with `setTextPrecision (tpr)` `tpr` is assigned one of values `string`, `char` or `stroke`.

## Questions

- Explain Color and Gray scale levels.
- Justify color and grey scale system.



Thank You