

# Raster Graphics

AP Computer Science A



Name: \_\_\_\_\_



# Contents

---

<b>1</b>	<b>Background</b>	<b>2</b>
1.1	Raster Graphics . . . . .	2
1.2	Indexed Colours & Colour Palettes . . . . .	2
1.3	Common Raster Graphic File Formats . . . . .	2
1.3.1	TIF/TIFF . . . . .	2
1.3.2	GIF . . . . .	3
1.3.3	PNG . . . . .	3
1.3.4	JPG/JPEG + EXIF/JFIF . . . . .	3
1.4	Our File Formats . . . . .	4
1.4.1	Computer Science Bit-Map File Format . . . . .	4
1.4.2	Computer Science Indexed Colour File Format . . . . .	4
<b>2</b>	<b>Activity #1</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Exercises . . . . .	5
2.3	Questions . . . . .	5
<b>3</b>	<b>Activity #2</b>	<b>6</b>
3.1	Introduction . . . . .	6
3.2	Exercises . . . . .	6
3.3	Questions . . . . .	6
<b>4</b>	<b>Final Analysis</b>	<b>7</b>
<b>5</b>	<b>Template Classes</b>	<b>8</b>

## Background

If you've used a computer, chances are you've taken, stored, transmitted, and edited digital image files. For over half a century, computer scientists and other researchers have developed techniques for the processing of these files. Despite this, every file format designed for storing digital image data can be placed in one of two categories: *raster* and *vector* file formats. Although vector formats are interesting, they are not nearly as common, mostly due to being unsuited for handling complex image data from digital cameras and other image capturing devices. As such, this case-study will focus on the handling of raster graphics.

## Raster Graphics





A *raster graphic* file format is any digital image file format that arranges its information according to a rectangular grid of pixels. Various file formats have been created that allow for varying degrees of image data compression, colour-depth, and the lossless preservation of image data. These types of image files will typically represent image data as a combination of an indexed palette of colours along with data representing which indexed colour should be reproduced for a given pixel. In memory, these images can then be stored as rectangular arrays of data corresponding to the indices within the colour palette and processed for display on a computer monitor or other display device.

A special type of raster image, called a *bitmap*, stores only binary image data. That is, a single bit is used to represent each pixel, with only two colours (typically black and white) available for display.

## Indexed Colours & Colour Palettes

Many raster image files store a *colour palette* alongside the image pixel data to enable more colour-depth. These colour palettes setup an indexed listing of colours that can then be referenced within the image data. The following example shows a 4-colour palette, the corresponding image data, and the corresponding image using this scheme.

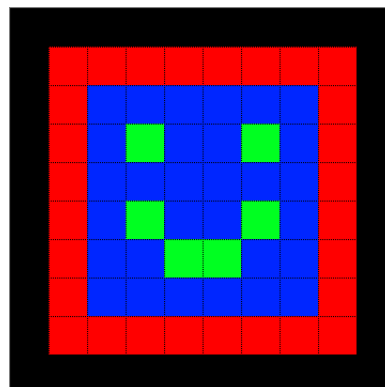
**Colour Palette**

0	
1	
2	
3	

**Image Data**

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	2	1	0
0	1	2	3	2	2	3	2	1	0
0	1	2	2	2	2	2	2	1	0
0	1	2	3	2	2	3	2	1	0
0	1	2	2	3	3	2	2	1	0
0	1	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0

**Displayed Image**



When combined with other data-compression techniques, the use of a colour palette can drastically reduce the size of digital image files containing a limited number of colours (256 or less), such as logos, icons, emoticons, etc.

## Common Raster Graphic File Formats

### 1.3.1 TIF/TIFF

The **Tagged Image File Format** is an extensible digital image format first introduced in the 1980s as an attempt to standardize the file format created by consumer-grade scanners. Initially designed to simply handle bitmap images, the file format has since expanded to include grayscale and color images, including the support for a colour palette scheme similar to that discussed previously.

Extensions to the TIFF file format to allow for CMYK color palettes, lossless compression, multiple layers, and the ability to hold vector-based clipping paths have all made the file format popular among digital artists and printers.

### 1.3.2 GIF

The **Graphics Interchange Format** was introduced by CompuServe (the first major American-based Internet Service Provider) as a colour digital image format replacement to their previous, black-and-white only image format. It has since become a popular image format for use on the Internet due to its relatively low file size, lossless compression, support for transparent pixels, and support for multiple-frame animations.

The GIF file format uses a colour palette scheme similar to the one discussed previously; however, each pixel is limited to 8-bits, resulting in a colour palette of no more than 256 colours. This makes the file format unsuitable for high-colour images.

### 1.3.3 PNG

The **Portable Network Graphics** image format was first designed as an alternative to the GIF file format due to GIF's use of patented compression techniques. Since its introduction, it has become a popular alternative for web-based graphics, especially for graphics requiring high colour resolution. It's ability to handle a 24-bit colour palette (32-bits with an alpha channel) as well as non-palettred "true-colour" images allows for a colour-depth far exceeding GIF's 256 colour palettes.

Despite the advantages of the PNG file format, it's greater colour-depth combined with lossless compression results in file-sizes generally larger than that of GIF and the popular JPG/EXIF file formats. Additionally, the PNG file format does not support animations, although a related format, the **Multiple-image Network Graphics** format, was later created to support this feature.

### 1.3.4 JPG/JPEG + EXIF/JFIF

The **Joint Photographic Experts Group** created a compression and decompression technique for the sharing of digital images, particularly for photographs or other scenes depicting a large colour-depth and variation. Because it is a *lossy* compression technique, it is not suitable for the sharing of images requiring precise high-resolution. The techniques used in compression also do not handle line-drawings or other images representing stark colour transitions, such as images of vertical or horizontal textures of different colours. Despite these limitations, the compression technique specified usually results in much smaller images, making JPEG a popular choice for sharing high-colour resolution images on the Internet.

It should be noted that although images using this technique are usually labeled using the extension `jpg`, it is *not* a file format. File formats have been developed which use the JPEG compression technique. Two popular ones are the **Exchangeable Image File Format**, often used by digital image capturing equipment such as digital cameras and camcorders, and **JPEG File Interchange Format**, often used by popular image editors, such as Adobe's Photoshop®.

## Our File Formats

Because traditional image file formats use various forms of compression in order to save hard drive space and decrease transmission times as well as support a number of more advanced features (such as animations and alpha channels), their implementation is beyond the scope of this case study. To that end, the following two file formats have been designed in order to give the student a feel for how to handle image data without being bogged down with the individual details that go into advanced image file formats. They may be considered highly simplified versions of some of the more popular raster image file formats.

### 1.4.1 Computer Science Bit-Map File Format

In order to distinguish a traditional bit-mapped image from an indexed colour one for this case study, the first four bytes of each bitmap image will contain the ASCII encoded characters: 'CSBM' (Decimal: 67 83 66 77). This is known as the file's *signature* and indicates to a program attempting to process the file exactly what type of file is being read.

The following two bytes of the file will contain the width and height, respectively.

**Note:** Because the width and height are each restricted to a single byte of information, the maximum size of an image in this format is  $255 \times 255$ .

Each of the remaining bits represents the image data as either a 0 to indicate a black pixel or a 1 to indicate a white one. Any bits exceeding  $width \times height$  should be ignored.

'C'	'S'	'B'	'M'	<i>width</i>	<i>height</i>	<i>image data</i>
-----	-----	-----	-----	--------------	---------------	-------------------

### 1.4.2 Computer Science Indexed Colour File Format

The signature for the Computer Science Indexed Colour File Format is 'CSIC' (Decimal: 67 83 73 67).

As with the Computer Science Bit-Map File Format, the following two bytes of the file will contain the width and height, respectively.

The next byte will contain the number of elements in the colour palette,  $p$ . What follows is a group of  $3p$  bytes, wherein each set of 3 bytes contains the values for *red*, *blue*, and *green* making up each color.

**Note:** For the purposes of this case-study (and most colour indexed file formats),  $p$  will be a power of 2.

Finally, each remaining group of  $\lg p$  bits will represent a single pixel of the image. Any bits exceeding  $width \times height \times \lg p$  should be ignored.

'C'	'S'	'I'	'C'	<i>width</i>	<i>height</i>	<i>palette count</i>	<i>palette colours</i>	<i>image data</i>
-----	-----	-----	-----	--------------	---------------	----------------------	------------------------	-------------------

## Activity #1

---

### Introduction

### Exercises

1. Implement the `readImageFile()` method with the following requirements:
  - (a) Instantiates and returns an `Image` class object, populating all of its relevant instance variables, as indicated below.
  - (b) The `width` and `height` of the `Image` should be read from the image file.
  - (c) If the image type is `CSBM`, a colour palette should be created containing only two entries, one for `Color.BLACK` and one for `Color.WHITE`.
  - (d) If the image is type `CSIC`, a colour palette should be read from the file containing the appropriate number of colours.
  - (e) The `pixels` attribute of the `Image` should be created as a two-dimensional array of appropriate values.  
**Note:** `pixels.length` should be equal to the *width* of image and `pixels[i].length` should be equal to the *height* for each  $i < width$ .
2. Implement the `writeImageFile()` method with the following requirements:
  - (a) Reads relevant information from the given `Image` object.
  - (b) Selects which file format to use (`CSBM` or `CSIC`).
  - (c) Write to a file following the guidelines of the appropriate file format from the Background.

### Questions

## Activity #2

---

### Introduction

### Exercises

1. Implement the `listColourPresets()` method. This method should populate the `colourPresetsList` with each of the colour palettes in the "ColourPresets" file.
2. Implement the `selectColourPreset()` method. This method should update the currently displayed `Image` to use the selected preset colour palette.
3. Implement the `saveColourPreset()` method. This method should append the current palette to the "Colour-Presets" file.

### Questions



## Final Analysis

---

## Template Classes

---