# Pixel Tracer

## Presentation of the project

Proposed by : Mr. Halim Djerroud
Prepared by:

- Mr. Halim Djerroud
- Mr. Fabien Calcado

**Asma Gabis**
asma.gabis@efrei.fr

**L1, L1-BN, L1-INT, L1-R, L1-BDX - 2022/2023**

01

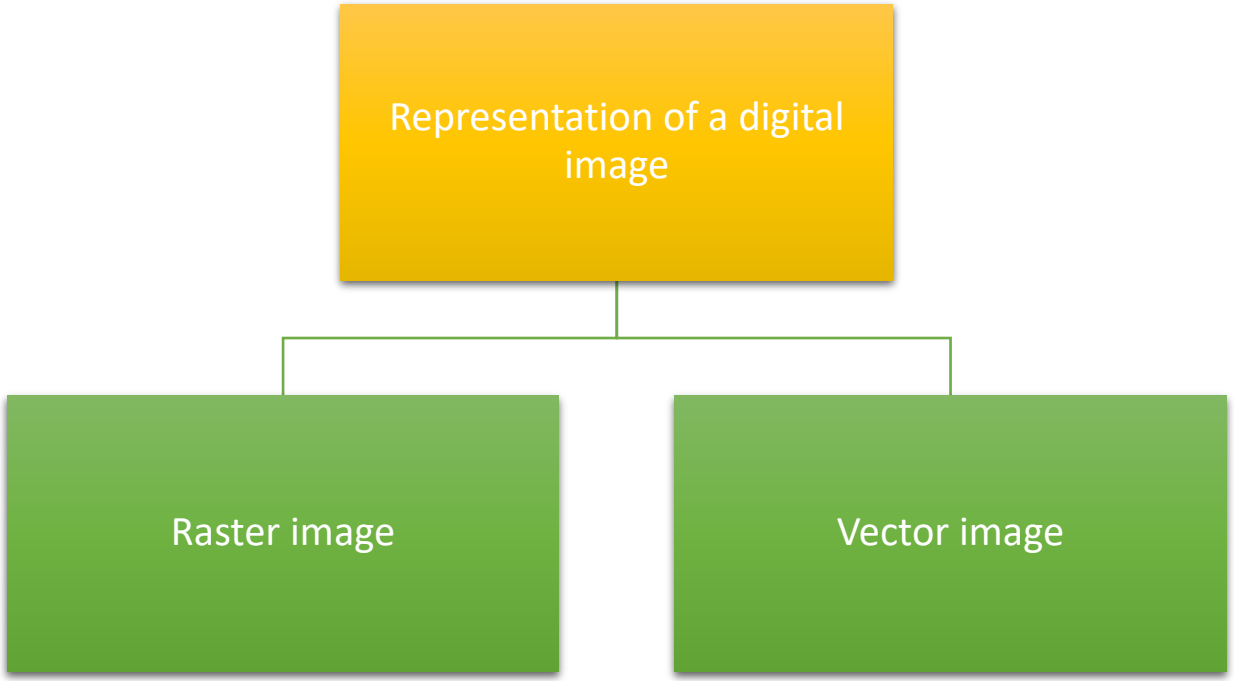
# Digital image

01

# Digital image

## Definition

- Real image obtained from digital camera, scanner, ...
- Computer generated image



## Raster image

- The image data is represented as a matrix of multi-dimensional points called (pixel in the 2D case).
- Examples of raster file formats: jpeg, gif, png, bmp
- Examples of software for creating raster images: Photoshop, After Effects, Gimp

### Benefits

- Possibility to modify the image pixel by pixel
- Create important color shades (gradients, shadows, . . .)
- Have textures effects

### Disadvantages

- large file (even compressed)
- Pixelation (enlargement)

## Vector image

- the image data is represented from a mathematical point of view.
- Examples of vector file formats: svg, ai, eps, pdf
- Examples of software for creating vector images: Illustrator, CorelDRAW, Inkscape, librecad, Dia Diagram Editor

**Benefits**

- light file, no quality loss when resizing

**Disadvantages**

- Allows to represent only simple geometrical shapes (lines, arcs, circles, . . .) so not adapted to photography.

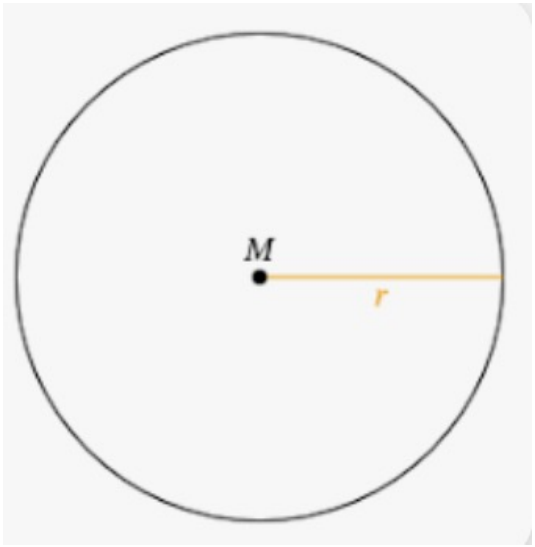
02

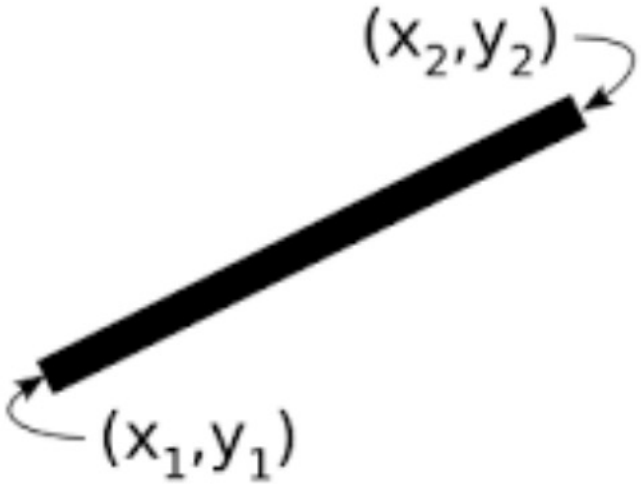
# Vector drawing

02

# Vector drawing

Store only the characteristics of the object



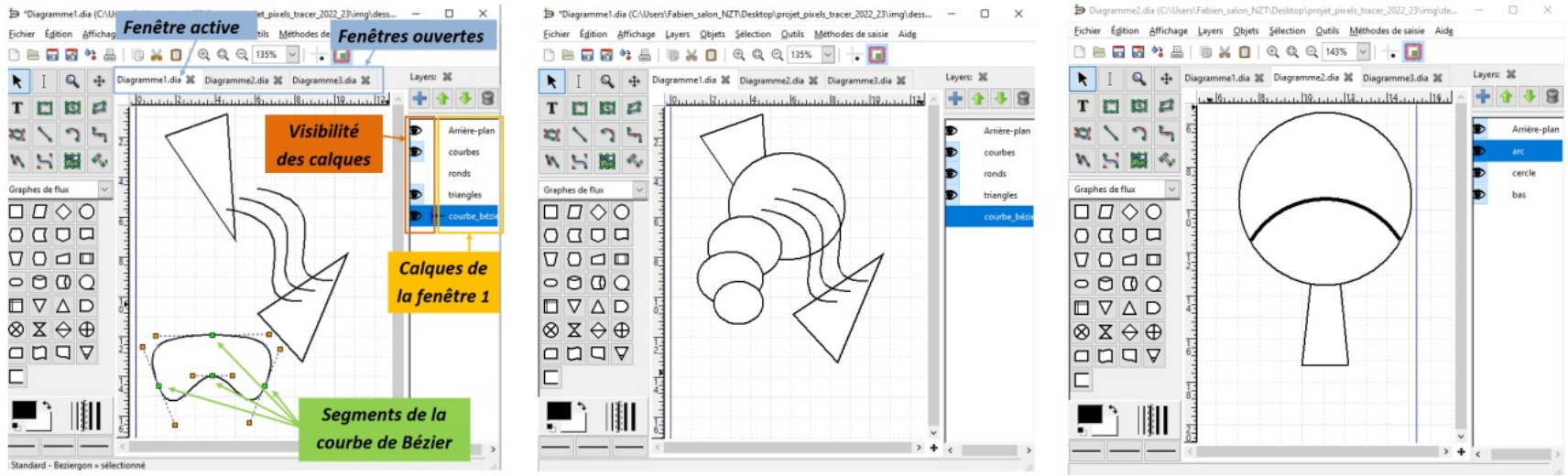

```
                . . . . . . . . . . . . .
(px,py) . . . . . . . . . .(px,py+longeur)
   . . # # # # # # # # # . .
   . . #                 # . .
   . . #                 # . .
   . . #                 # . .
   . . # # # # # # # # # . .
(px+largeur,py) . . . . . . . (px+largeur,py+longeur)
```

```
          . . . . . . . . .
(px,py) . . . .(px,py+longeur)
   . . # # # # # . .
   . . #       # . .
   . . #       # . .
   . . #       # . .
   . . # # # # # . .
(px+longeur,py) . . . .(px+longeur,py+longeur)
          . . . . . . . . .
```

# Vector drawing

For more complex images, vector drawing software relies on the :

- Windows
- Drawing areas
- Layers
- Geometric objects




(a) Contenu fenêtre 1 avec calque ronds non visible

(b) Contenu fenêtre 1 avec calque courbe de Bézier non visible et calque ronds visible

(c) Contenu fenêtre 2 et ses différents calques

FIGURE 3 – Logiciel Dia Diagram Editor avec son système de calques par fenêtre

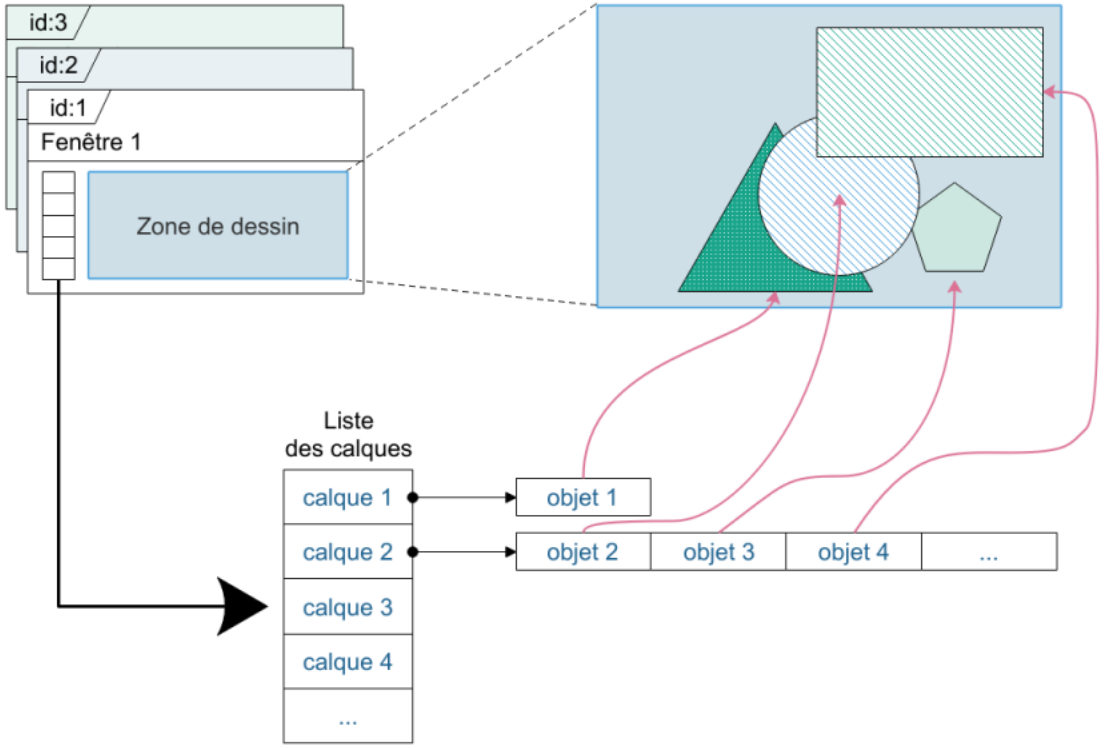
03

# Objective of the project

03

## Representation in memory

Think about the data structures to be used



## Actions to be implemented

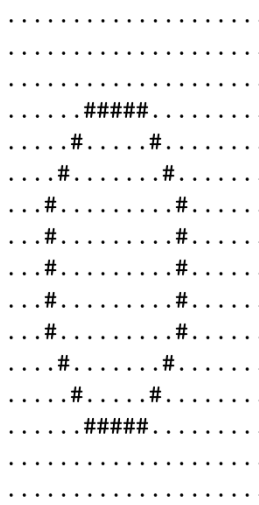
Carry out the following orders

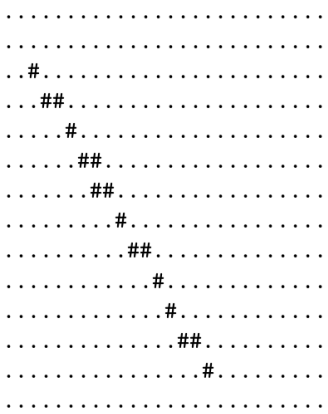
—**clear**: clear the screen
—**exit**: exit the program
—**point x y**: add a point to the array of objects
—**line x1 y1 x2 y2**: draws a segment connecting the two points (x1, y1) and (x2, y2)
—**circle x y radius**: draws a circle with center (x, y) and radius radius
—**square x y length**: draws a square whose upper left corner is (x, y) and side length.
—**rectangle x y width height**: draws a rectangle whose upper left corner is (x, y), width width and length height.
—**polygon x1 y1 x2 y2 x3 y3** ... ...
—**plot**: displays all the objects of the same table on the screen.
—**list**: displays the list of objects that make up the drawing with their identifiers (id).
—**delete id**: delete the object whose id is equal to id.
—**erase**: delete all objects from the same array.
—**help**: displays the list of commands as well as a mini manual allowing the user to use the commands correctly

# Objectives of the project

## Drawing on screen

- The design itself will be done through ASCII characters
- To generate the geometric shape, we will rely on algorithms already known in computer graphics such as :
  - Segment drawing algorithm to draw a line
  - Bresenham's segment drawing algorithm for drawing a line
  - Andres circle drawing algorithm to draw a circle





# Objectives of the project

## Pedagogical interest

- Reinforce your learning on structures
- Reinforce your learning on the arrays
- Reinforce your learning on pointers
- Reinforce your learning on the modularity of a program
- Reinforce your learning on dynamic allocation
- Reinforce your learning about generics
- Acquire knowledge on advanced concepts

04

# The basic structures

04

# The basic structures

Several structures are to be developed:

- The Point structure
- The Line structure
- The Circle structure
- The square structure
- The Rectangle structure
- The Polygon structure

Associate to each geometric shape, three functions:

- Create the geometrical shape in memory by means of dynamic allocation
- Delete the geometric shape from the memory
- Display the geometric shape

# The basic structures

Use of the void* pointer for generics :

- Creation of the Shape structure allowing to associate a unique identifier to a geometrical shape
- Method of generating a unique ID
- Creation of generic functions associated with geometric shapes

To go further :

- Implementation of the drawing areas
- Layer implementation
- Use of linked lists
- .....