Summary

Introduction

◆ Objective of the project

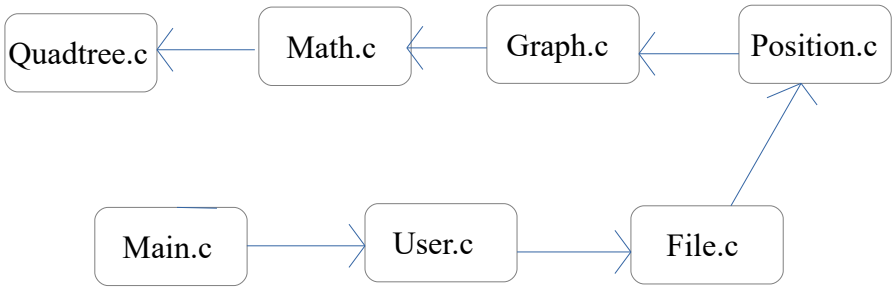
This project has as objective to make an application for treatment on image by using quadtree implementation in C language. This application will be able to decode, encode and compress every image.

We will only use the **MLV library** to make this application.

Development

Dev part

◆ Modulation of the program



1. Quadtree.c →contains functions for quadtree treatment.
2. Math.c → contains functions which calculate something for exemple to calculate average color of zone, the error of the zone…
3. Graph.c → contains every functions to draw on the window.
4. Position.c →contains functions to manage the position of zone on the image.
5. File.c →contains functions to manage files (save tree, read tree…).
6. User.c → for what the user can do.
7. Main.c → Main program.

◆ Structures

To make our application we needed to define the following structures :

Pixel union structure which contains 2 fileds

bit for B&W image and color for color image (rgba).

```
union Pixel {
int bit;
int color[4];
```

```
};
```

Position structure which is the position of a zone.
x, y → the position at the North West.
x1, y1 → the position at the south East.

```
typedef struct position {
int x, y;
int x1, y1;
} Position;
```

Finaly the structure for the quadtree which contains the 4 nodes northWest, northEast, southWest , southEast and a filed error, pos, pixel for every node.

```
typedef struct quadtree {
long error;
Position pos;
union Pixel pixel;
struct quadtree *northWest;
struct quadtree *northEast;
struct quadtree *southWest;
struct quadtree *southEast;
} Quad, *QuadTree;
```

◆ Image

The images are coding by a **'int'** for black and white image (1 for black and 0 for white).

Colors images are coding by an **array of 4 int** for rgba.

To know if we put 1 or 0 for black and white image we calculte the average color of the current zone, then we watch if the value is more closest than 255 we attribute the value 0 else 1 for black.

◆ Save & Read Files

Save black & white : Nodes are coding by 0 and leafs are coding by 1 following by their color 1 for black and 0 for white.

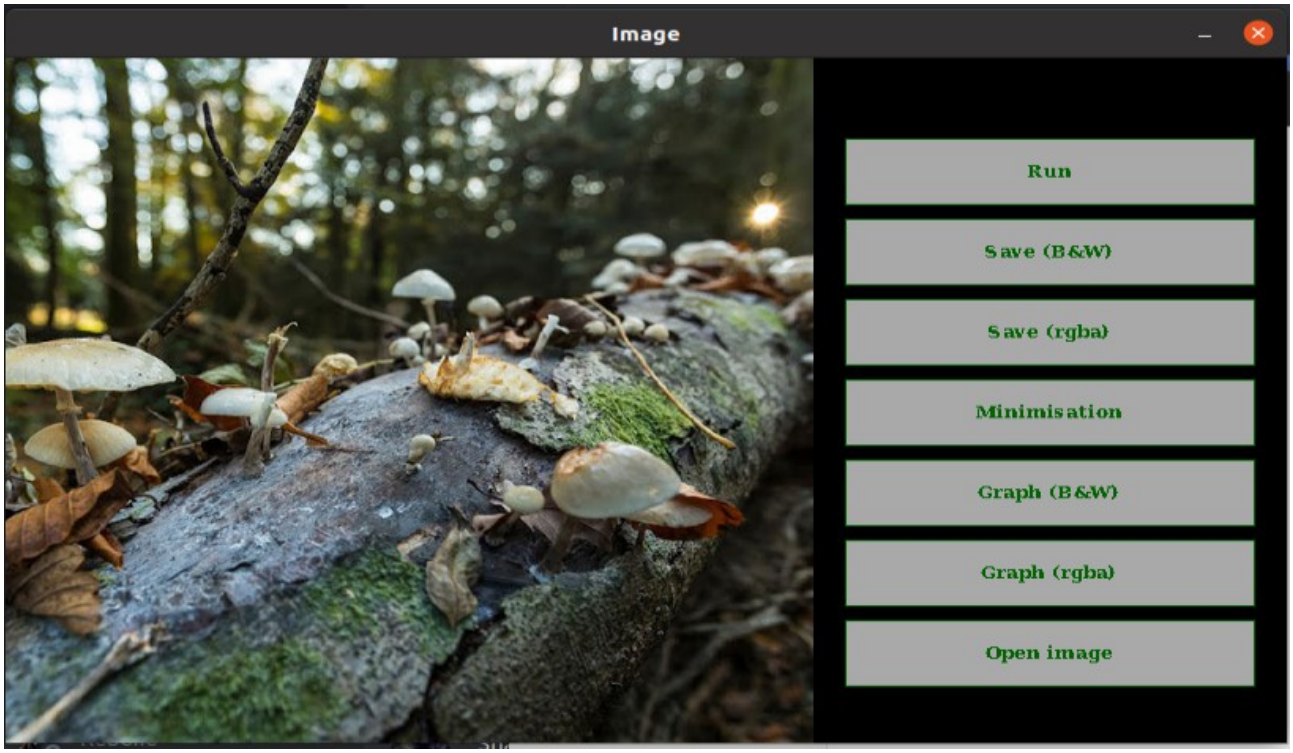
Save color : Nodes are also coding by 0 + space and leafs by 1 + space following by his rgba color

split by space.

Read files : read the file .qtc or .qtn and the file pos.txt.

User part

- ***Use of program***



Run → One sentence will appear in the terminal to ask you the stop value of the program it corresponds to the error that the image must have.

```
How much do you want as error for the image :
```

Save (B&W) → if your image is B&W it will save in the file img.qtn.

Save (rgba) → if your image is rgba it will save in the file img.qtc.

Minimisation → launches the program but minimizing the knots of the tree corresponding to the image.

However, it's not operational at the moment.

Graph (B&W) → Displays the graph of the tree corresponding to the image (B&W) that you have minimized.

However, it's not operational at the moment.

Graph (rgba) → Displays the graph of the tree corresponding to the image (rgba) that you have minimized.

However, it's not operational at the moment.

Open image → The application will ask you to write the name of the image that you want to open.

Then you will need to put 1 if your pick image is in color or 0 if it’s in B&W.

Conclusion

- **Bug & Partially processed**

We had some difficulties on the realization and the understanding of the minimization part of the project which is not functional in our program however we tried to do the maximum of what was asked of us, most of the requested functionality is present.

Annex

- **Compilation**

To compile you need to make the following commands :

Firstly → unzip the folder by a right click or in the terminal.

Then → “make” on the terminal.

Finaly → “./Project” one the terminal.

The application will appear.