

Developer’s manual

Table of Contents

Developer’s manual.....1

 Modules.....2

Compilation and execution

Command	Description
Make	Compile all the necessary file for the creation of the executable file (named Main)
./Main	Execute the executable file

Modules

The program is divided into 19 modules :

- **Main** : it contains the main function that will execute the program
- **Quadtree** : This module manage all the operation to the Quadtree structure such as the creation of a Quadtree or the initialization of a Quadtree. This structure is used only to represents an image.
- **Hashtable** : It is a module that is necessary for this program. The purpose of this module is to maintain a high performance during the creation of the Quadtree. Instead of reading the whole Quadtree to find the leave with the highest error value, the hashtable can look directly into the leaves of the Quadtree.
- **Pixel** : The Pixel module manage all the operation to the Pixel structure. This structure contains information about a pixel such as the position in the window and its width and height.
- **Color** : The Color module manage all the operation on an rgba array.
- **BitFile** : This module is necessary for the Compression and Decompression module. This module manage the IO of a file.
- **Compression** : This module manage the save of a Quadtree into a file.
- **Decompression** : This module manage the load of a Quadtree from a file.
- **Minimisation** : This module manage the minimization of a Quadtree
- **Calcul** : This module manage all the calcul linked to the creation of a Quadtree such as error value
- **Input_img** : This module manage the creation of a Quadtree representing an image. It groups the main modules for the generation of the Quadtree : Quadtree, Color, Pixel, Hashtable, Calcul.
- **Menu** : It manage the creation of the menu interface into an MLV window
- **Button** : It manage the creation of the clickable button on an MLV window
- **Extension** : It manage the verification of the extension of a file name

- **Actions** : It manage all the actions that can be done after clicking into a button
- **Image** : it manage the operations that can be done on an Image structure
- **Graphic** : It manage the display of an image in the MLV window
- **Test** : This module execute test on the main functions of the program (if it is possible to test).
- **Display_qt** : This module is used only for the test purposes. It can simplify the display of a Quadtree and make it more readable for the user.

IMPORTANT : the dot command is required to create the graph of the Quadtree and execute the test function.

Hashtable

The purpose of this module is to create a Quadtree efficiently.

Important point:

- Every time leaves are created from a tree, they are added into the hash table (every element of the list are sorted because we use the sorted add).
- The program find the node with the highest error value by searching the first element of each list in the hash table (because each list are sorted by the error value)
- Every time a leave becomes a node, this node is removed from the hash table

Compression & Decompression

To save or load a file, we use the bit to bit operation. Those operations are managed by the BitFile module (function to read/write a bit, open a file ...).

Those module uses function that browse a tree recursively.

Test

Test is module that will test the most important functions in this project (if possible).

Test is not fully updated, it only tests the compression and decompression function. It also tests the BitFile module.

In the Test module, Node, Pixel and color are randomly generated to compare a lot more varieties of variables.

Minimization

The minimization method used in this project is :

- When all the sons of a node have the same color, all those sons are removed and replaced by this node with the same color as its sons.

Options

No option have been implemented