Photoformer the tiny image editor



By Guillermo Ramos and Alejandro López



Content

Diagrams

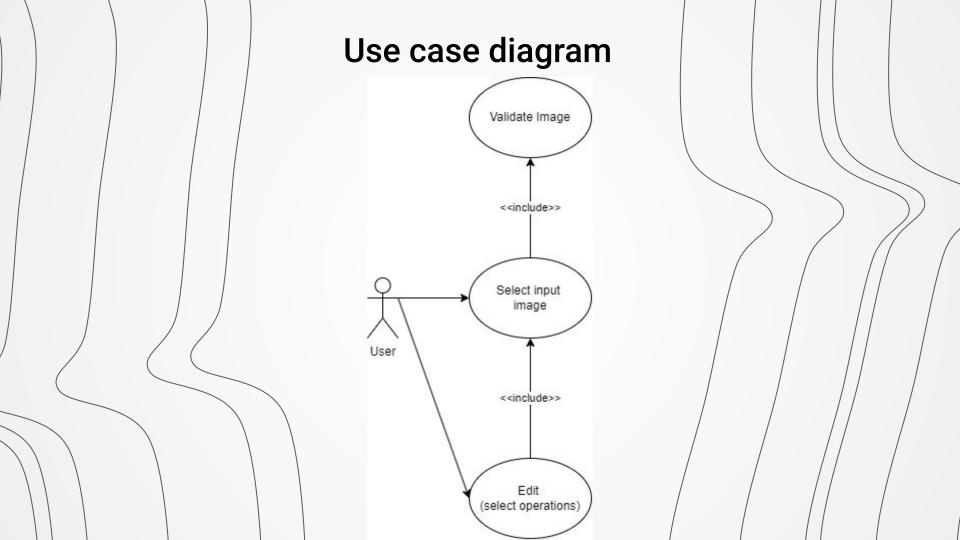
Reasoning

Adaptations

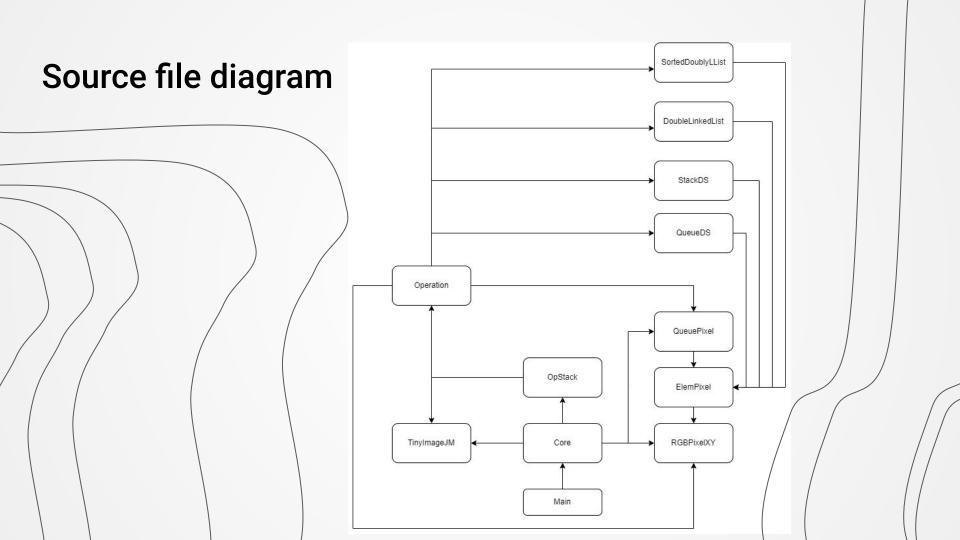
Explanations

Conclusions

Data flow diagram -Send confirmation--Send confirmations -Send confirmatiion--Send confirmation-User -Select Image-Verify Image Load Image Verify operation Select operation Load operation Run operations -Run operations-Produce output image



Class diagram StackDS RGBPixelXY + top: ElemPixel + «destroy» StackDS ()> + coordY: unsigned int = 0 + push (ppix : RGBPixelXY + colorR: unsigned char = 0 + isEmpty (): bool colorG: unsigned char = 0 + pop (): ElemPixel + colorB: unsigned char = 0 TinylmageJM + «create» RGBPixelXY (paramX : unsigned int, paramY : unsigned int) + «create» RGBPixelXY (paramX : unsigned int, paramY : internal height: int unsigned int, paramR: unsigned char, paramG: unsigned internal_image: unsigned char char, paramB : unsigned char) internal_palette: unsigned char + «destroy» RGBPixelXY () internal_name: std::string + getX (): unsigned int internal_error: std::string + getY (): unsigned int imageValid: bool + getR (): unsigned char filePointer: FILE getG (): unsigned char ElemPixel getB (): unsigned char openFile (p_flags : std::string): bool setComponents (paramR : unsigned char, paramG : pixel: RGBPixelXY + «create» TinyImageJM (p_name : std::string) next: ElemPixel + loadImageFromDisk (): bool + normalizeToRedGreenOrBlue () isValid (): bool + isBlack (): bool «create» ElemPixel (ppixel: RGBPixelXY) getImagePointertoInternal (): unsigned char QueuePixel + isNotBlack (): bool getWidth (): int *create* ElemPixel (ppixel: RGBPixelXY, pnext: Elem + getDistACentro (width : unsigned int, height : unsigned getHeight (): int int): unsigned int + rear: ElemPixel + «destroy» ElemPixel () + getInternalError (): std::string + getFromRaw (image : unsigned char, width : unsigned int + «destroy» QueuePixel () getPix (): RGBPixelXY releaselmageMemory (): bool > , height : unsigned int, p_x : unsigned int, p_y : unsigned int + isEmpty (): bool getNext (): ElemPixel setNewImagePointerWithOldRemoval (p. image + dequeue (): ElemPixel getPrevious (): ElemPixel + setIntoRaw (image : unsigned char, width : unsigned int, insigned char, p_width : int, p_height : int) + enqueue (ppix : RGBPixelXY) setNext (pnext : ElemPixel) saveImageToDisk (p_name : std::string): int height: unsigned int, p_x: unsigned int, p_y: unsigned int): setPrevious (pprevious : ElemPixel) + copyQueue (): QueuePixel «destroy» TinvImageJM () SortedDoublyLList header: ElemPixel + last: ElemPixel isEmpty (): bool insertSorted (ppix : RGBPixelXY) retrieveFirst (): ElemPixel DoublyLinkedList QueueDS Core Operation front: ElemPixel header: ElemPixel next: Operation rear: ElemPixel last: ElemPixel operation: std::string + boot () + «destroy» QueueDS () «destroy» DoublyLinkedList () + «create» Operation (operation : std::string) enqueue (ppix : RGBPixelX) insertLast (ppix: RGBPixelXY) + «create» Operation (operation : std=string, p_next : Operation) isEmpty (): bool retrieveLast (): ElemPixel Main dequeue (): ElemPixe isEmpty (): bool «destroy» Operation () OpStack getNext (): Operation main (): int + top: Operation getNegative (queuePix : QueuePixel, mylmg : Tinylmage JM, imageName : std::string) + «destroy» OpStack () + getFlipped (queuePix : QueuePixel, mylmg : TinyImage + pop (): Operation + push (operation : std::string) JM, imageName : std::string) isEmpty (): bool getDarken (queuePix : QueuePixel, mylmg : Tinylmage IM. imageName : std::string) - getFlopped (gueuePix : QueuePixel, mylmg : Tinylmage JM, imageName : std::string) getSorted (queuePix : QueuePixe imageName: std::string) getOp (): std::string



ADTs REASONING

getNegative - Queue
getDarken - Queue
getFlipped - Stack
getFlopped - Doubly Linked List

getSorted - Sorted Doubly Linked List

ADTs ADAPTATIONS: QUEUEPIXEL

```
SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue
dequeue: queue -> item

front: queue -> item
makenull: queue -> queue
empty: queue -> boolean
```

ENDSPEC

```
SPEC QUEUE[ITEM]

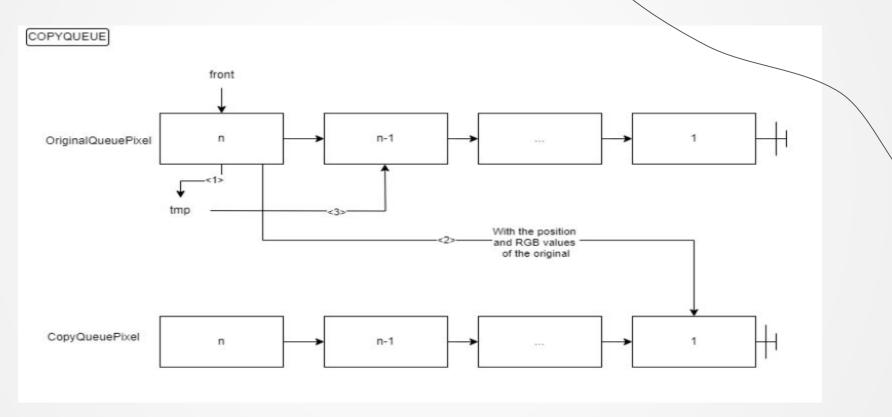
GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue
dequeue: queue -> item
empty: queue -> boolean
copyQueue: queue -> queue
```

ENDSPEC

ADTs ADAPTATIONS: QUEUEPIXEL



ADTs ADAPTATIONS: QUEUEDS

```
SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue
dequeue: queue -> item
front: queue -> item
makenull: queue -> queue
empty: queue -> boolean
```

```
SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue

dequeue: queue -> item

empty: queue -> boolean
```

ENDSPEC

ENDSPEC

ADTs ADAPTATIONS: OPSTACK

```
SPEC STACK[ITEM]

GENRES stack, item

OPERATIONS:

push: stack item -> stack

pop: stack -> item

top: stack -> item

makenull: stack -> stack

empty: stack -> bool

ENDSPEC
```

```
SPEC STACK[ITEM]

GENRES stack, item

OPERATIONS:

push: stack item -> stack

pop: stack -> item

empty: stack -> bool

ENDSPEC
```

ADTs ADAPTATIONS: STACKDS

```
SPEC STACK[ITEM]

GENRES stack, item

OPERATIONS:

push: stack item -> stack

pop: stack -> item

top: stack -> item

makenull: stack -> stack

empty: stack -> bool

ENDSPEC
```

```
SPEC STACK[ITEM]

GENRES stack, item

OPERATIONS:

push: stack item -> stack

pop: stack -> item

empty: stack -> bool

ENDSPEC
```

ADTs ADAPTATIONS: DOUBLYLINKEDLIST

```
SPEC LIST[ITEM]
       GENRES list, item, position
       OPERATIONS:
               insert: item position list -> list
               delete: position list -> list
               locate: item list -> position
               retrieve: position list -> item
               next: position list -> item
               previous: position list -> item
               makenull: list -> list
               empty: list -> bool
ENDSPEC
```

```
SPEC LIST[ITEM]

GENRES list, item

OPERATIONS:

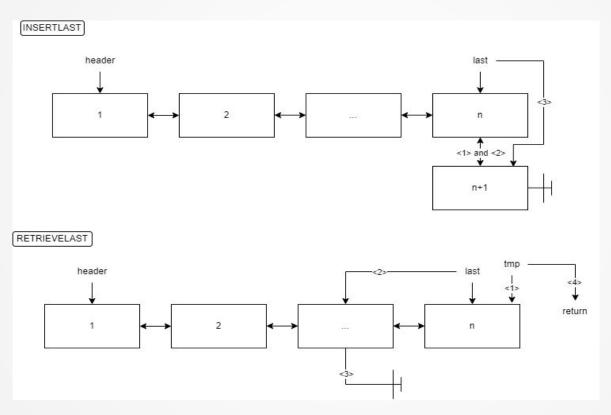
insertLast: item list -> list

retrieveLast: list -> item

empty: list -> bool

ENDSPEC
```

ADTs ADAPTATIONS: DOUBLYLINKEDLIST

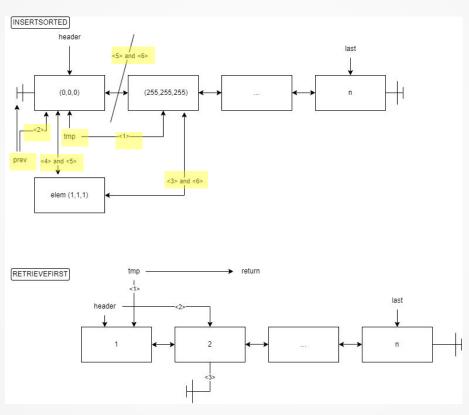


ADTs ADAPTATIONS: SORTED DOUBLYLINKEDLIST

ENDSPEC

```
SPEC LIST[ITEM]
       GENRES list, item, position
       OPERATIONS:
                                                             SPEC LIST[ITEM]
               insert: item position list -> list
                                                                    GENRES list, item
               delete: position list -> list
                                                                    OPERATIONS:
               locate: item list -> position
                                                                            insertSorted: item list -> list
               retrieve: position list -> item
                                                                            retrieveFirst: list -> item
               next: position list -> item
                                                                            empty: list -> bool
               previous: position list -> item
                                                             ENDSPEC
               makenull: list -> list
               empty: list -> bool
```

ADTs ADAPTATIONS: SORTED DOUBLYLINKEDLIST



CLASSES EXPLANATION

Core:

void boot()

ElemPixel

Operation:

- void getNegative()
- void getFlipped()
- void getDarken()
- void getFlopped()
- void getSorted()

OpStack

QueueDS

QueuePixel:

QueuePixel* copyQueue()

StackDS

DoublyLinkedList

SortedDoublyLList:

void insertSorted()

RUNNING TIME EXPLANATION

LOADING PHASE

Pixel queue creation -> O(1)

Validation of the image format -> O(1)

Pixel enqueuing -> O(N)

USER INTERACTION PHASE

Insert operations in OpStack -> O(1)

Where N is the total number of pixels in the image

RUNNING TIME EXPLANATION

DATA PROCESSING PHASE

getNegative() ->
$$O(N + N + N)$$
 -> $O(3N)$ -> $O(N)$
getFlipped() -> $O(N + N + N + N)$ -> $O(4N)$ -> $O(N)$
getFlopped() -> $O(N + N + N + N)$ -> $O(4N)$ -> $O(N)$
getDarken() -> $O(N + N + N + N)$ -> $O(4N)$ -> $O(N)$

