6 Paint

The main goal of this project is to make simple image editing/batch convert application. You cannot use any image library for working with images, all image editing, loading and saving functions have to be implemented by you.
You have to support 2 image formats, it is recommended to use lossless formats or their lossless variants as they are easier to work with e.g.: BMP, TGA, TIFF, PNG.
There will be 3 applications. First application will be used from CLI, you will write commands directly to the command line. Second application will take commands from file, it can be used for editing/converting multiple images at once. Third application will read commands from file and generate image based on those commands.

6.1 Command (functions) to implement

- required parameter will be passed right after the command separated by space
- optional parameter will be passed after the required parameters inside of curly brackets {} , values inside {} can have different order, only some of the optional parameters can be used
- option %|PX means whether to use relative (percentage) OR absolute (pixel) values
- you can use some library like OpenCV to visualize the changes on screen (optional, do this only if you want)

- **LOAD ./folder/*.bmp**- load images in folder or one image
- **SAVE ./folder/*.bmp**- where the loaded image(s) will be saved and in which format
- **COLOR r g b**- set global color for your next commands
- **Line** - draw line between two points
  - LINE %|PX x1 y1 x2 y2 {width: number, color: {r: number, g: number, b: number}}
  - width - sets width of line, default=1
  - color - sets color of line, this color will be used instead of global settings
- **Circle** - defined by its center point and radius
  - CIRCLE %|PX x1 y1 radius {fill: bool, fill-color: {r: number, g: number, b: number}, border-color: {r: number, g: number, b: number}, border-width: number}
  - fill - whether the inside of the circle will be filled or not
  - fill-color - sets fill color, this color will be used instead of global settings
  - border-color - sets border color, this color will be used instead of global settings
  - border-width - sets width of border line, default=0
- **Bucket** - sets color to the area with the same color (Flood fill)
  - BUCKET %|PX x1 y1 {color: {r: number, g: number, b: number}
  - color - sets color of the fill, this color will be used instead of global settings
- **Resize** - resize the entire image

  - RESIZE %|PX width height - with % use relative size

- **Rotation** - rotate image by +-90 degrees
  - ROTATE CLOCK - rotate image clockwise
  - ROTATE COUNTERCLOCK - rotate image counter clockwise
- **Invert colors**
  - INVERTCOLORS
- **Convert to grayscale**
  - GRAYSCALE
- **Crop** - define crop by top left corner and bottom rigth corner
  - CROP %|PX x1 y1 x2 y2 - with % use relative coordinates
- **UNDO** - undo previous command (can be used multiple times)
- **REDO** - redo previously undone commands (can be used multiple times)

6.2 Batch edit/convert

This application will have 1 arguments, input file with commands.

File input example *exampleFileBatchConvert.txt*:
LOAD ./images/
GRAYSCALE
RESIZE 71
COLOR 0 255 0
LINE % 0 0 100 100
COLOR 255 0 0
LINE % 0 100 100 0
SAVE ./images-converted/*.bmp

Commands in this file will load all supported images from folder *./images/*, it will convert them to grayscale, it will draw line from top left corner to right bottom corner of the image in green color, then it will draw line from bottom left corner to right top corner of the image in red color, then it will be resized to 71% size of original image. Finally they will be all saved as BMP (one of the supported formats) into *./images-converted* folder.

6.3 Generate image based on commands

This application will have 3 arguments, input file with commands, output resolution, output filename. It will generate image based on commands in file.

File input example *drawCzFlagCommands.txt*:
LINE % 0 100 50 50 {color: {r: 255, g: 0, b:, 0}}
LINE % 0 0 50 50 {color: {r: 255, g: 255, b:, 255}}
LINE % 50 50 100 50 {color: {r: 255, g: 255, b:, 255}}
BUCKET % 95 95 {color: {r: 255, g: 0, b:, 0}}

BUCKET % 95 5 {color: {r: 255, g: 255, b:, 255}}
BUCKET % 5 50 {color: {r: 0, g: 0, b:, 255}}

Usage example: *commandsToImage.exe drawCzFlagCommands.txt 3000x2000 czFlag.bmp*
This should draw flag of Czech republic to file czFlag.bmp with resolution 3000x2000.