Image editing project(https://github.com/QuantLee/CppProject)

The purpose of this project is to achieve image editing in a few specific ways. The image we are dealing with is in a straight forward image format, called PPM imagery. This application allows user to edit the target image through the command line. The application also has comprehensive error handling for input stream, and ensure that potential invalid input from user will be properly handled and will not influence the whole editing process. The original image is a deer image below:

To help you run the application in your own computer. Here is a walk-though of the image editing process. After downloading the project folder. We use makefile under the UNIX system to compile all .cpp and .h files to output an executable file: proj3.exe. You can achieve this by navigating to the project folder in your terminal, your current working directory should be: imageEditing/.

And then type: **make** in the command line, you will see the compile and link process:

```
$ make
g++ -c project3.cpp -o project3.o
g++ -c ColorClass.cpp -o ColorClass.o
g++ -c ColorImageClass.cpp -o ColorImageClass.o
g++ -c RowColumnClass.cpp -o RowColumnClass.o
g++ project3.o ColorClass.o ColorImageClass.o RowColumnClass.o –o proj3.exe
```

Running the proj3.exe file and you could begin your editing process:

```
./proj3.exe
```

There is a sample editing process, the testing input is stored in in.txt file and you could use the following command to test:

```
./proj3.exe < in.txt
```

The interacting process is like the following:

```
Enter string for PPM image file name to load: deer.ppm


1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 1



1. Specify upper left and lower right corners of rectangle

2. Specify upper left corner and dimensions of rectangle

3. Specify extent from center of rectangle

Enter int for rectangle specification method: 3



Enter rectangle center row and column: 132 327

Enter int for half number of rows: 12

Enter int for half number of columns: 12


1. Red

2. Green

3. Blue

4. Black

5. White

Enter int for rectangle color: 1



1. No

2. Yes

Enter int for rectangle fill option: 1
```

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 4

Enter string for PPM file name to output: new1.ppm

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 1

1. Specify upper left and lower right corners of rectangle

2. Specify upper left corner and dimensions of rectangle

3. Specify extent from center of rectangle

Enter int for rectangle specification method: 1

Enter upper left corner row and column: 300 312

Enter lower right corner row and column: 364 337

1. Red

2. Green

3. Blue

4. Black

5. White

Enter int for rectangle color: 3

1. No

2. Yes

Enter int for rectangle fill option: 1

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 4

Enter string for PPM file name to output: new2.ppm

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 3

Enter string for PPM image file name to load: topHat.ppm

Enter upper left corner to insert image row and column: 35 280

1. Red

2. Green

3. Blue

4. Black

5. White

Enter int for transparecy color: 5

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 4

Enter string for PPM file name to output: new3.ppm

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 2

Enter string for file name containing pattern: ohdeerPattern.txt

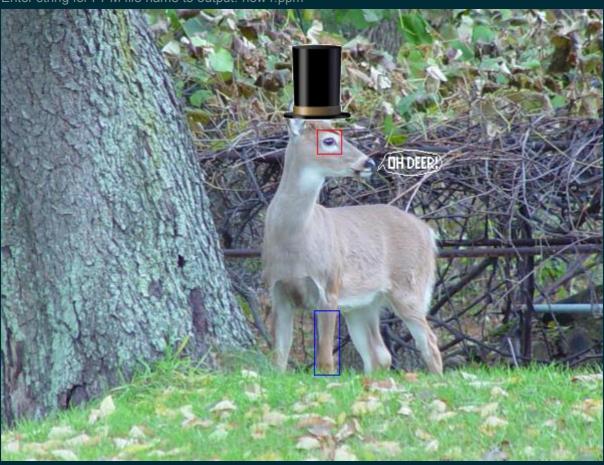Enter upper left corner to insert image row and column: 140 375

1. Red

2. Green

3. Blue

4. Black

5. White

Enter int for patternColor color: 5

1. Annotate image with rectangle

2. Annotate image with pattern from file

3. Insert another image

4. Write out current image

5. Exit the program

Enter int for main menu choice: 4

Enter string for PPM file name to output: new4.ppm

1. Annotate image with rectangle

The program achieves several algorithms to modify deer.ppm:

1. Allow rectangle to be draw on an image, rectangle outlines may be placed on an image to draw attention to a specific area, or filled rectangles may be placed on an image to block out a specific area.

2. An image may be annotated with a "pattern". A pattern, while rectangular overall, contains a description of a shape that is to be placed on an image. A pattern consists of a rectangle of only zeros and ones. When a pattern is placed over an image, the values in the pattern each fall over a specific pixel in the original image. A value of one in a pattern indicates that the pixel under it should be modified to be a certain color that is specified by the user. A zero in a pattern indicates that the pixel under it should NOT be affected by the pattern. Its original value is left intact, resulting in a sort of transparency.

3.The final image modification algorithm is the insertion of another (presumably smaller) PPM image at a specified location within the image being modified. This insertion simply reads another PPM image from a file and inserts the image contents where the user desires. PPM images are, by definition, rectangular. Since oftentimes, the image you want to insert is not rectangular, the application also supports a transparency color, such that any pixel in the image to be inserted which is the transparency color does not change the original image, but pixels that are not the transparency color will be used to replace the pixel value in the original image.

To see more details about this project, please refer to my github website.