

FCAI - OOP Programming - 2023 - Assignment 1

Program Name : CS213-2023-20220204-20220177-20220509-A1-Part1.cpp

Last Modification Date : 9/10/2023

Author1 : Abdullah Mohamed Abdullah Mohamed

ID : 20220204

Section : S22 (B)

Author2 : Abdelrahman Fathy Mohamed Hammad

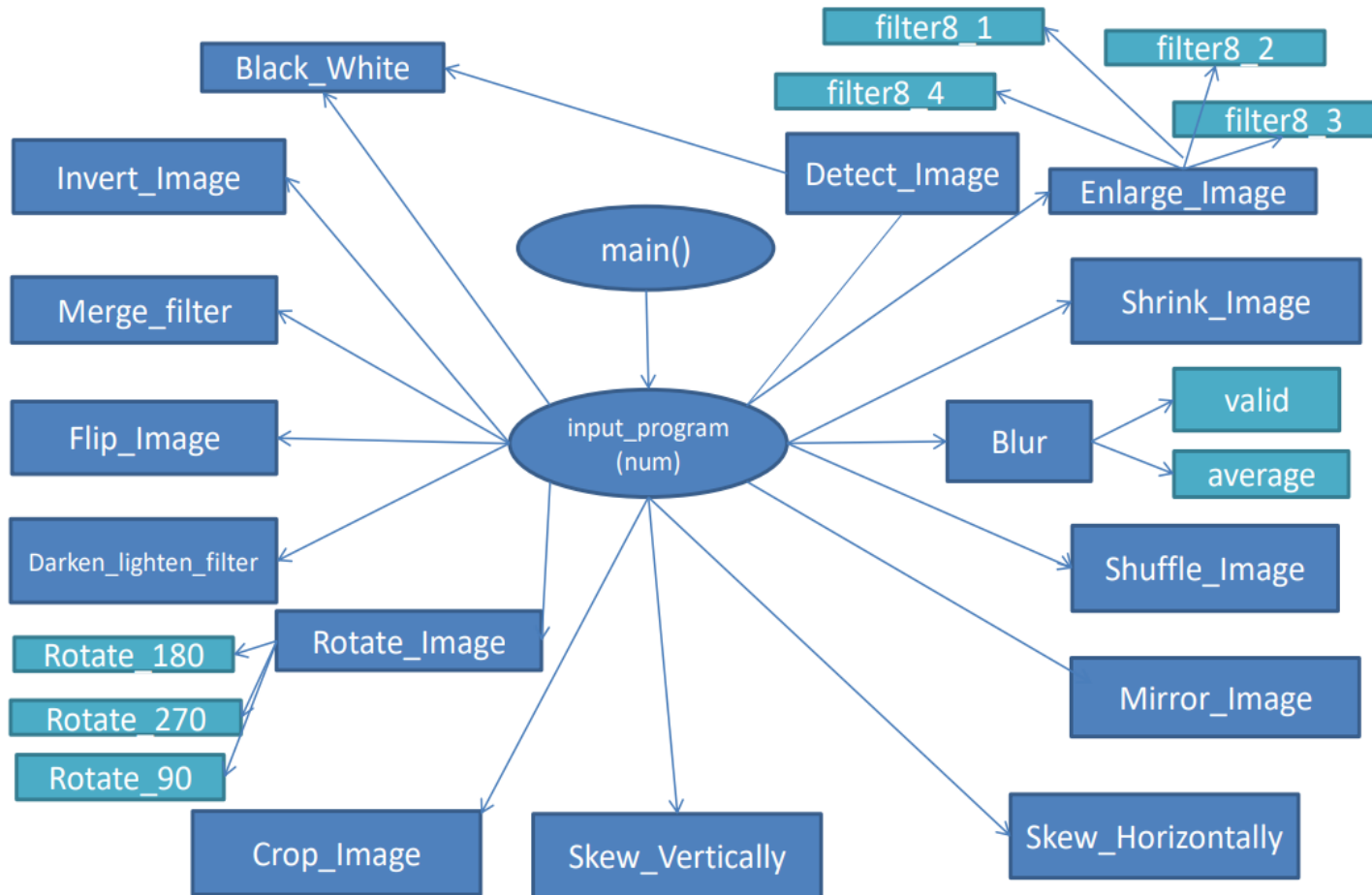
ID : 20220177

Section : S17 (B)

Author3 : Omar Khaled Elsayed Hussien

ID : 20220509

Functions diagram



The main function

The program starts with the `main()` function.

The variable `num` is declared as a `char`, and the `input_program()` function is called to obtain user input and assign it to `num`.

The program enters a while loop that continues as long as `num` has a non-zero value.

Inside the while loop, a switch statement is used to perform different image processing operations based on the value of num.

Each case in the switch statement corresponds to a different operation. For example, case '1' represents the operation to convert the image to black and white, case '2' represents the operation to invert the image, and so on.

After each operation is completed, the input_program() function is called again to get the next user input new value.

If the value of num is '0', the while loop is exited, and the end program.

function input_program

This is a C++ function that displays an interface for a program that applies various filters to an image. The function prompts the user to select a filter to apply or exit the program. The available filters are:

1. Black & White Filter
2. Invert Filter
3. Merge Filter
4. Flip Image
5. Darken and Lighten Image
6. Rotate Image
7. Detect Image Edges
8. Enlarge Image
9. Shrink Image
 - a. Mirror 1/2 Image
 - b. Shuffle Image
 - c. Blur Image
 - d. Crop Image
 - e. Skew Image Right

f. Skew Image Up

s. Save the image to a file

The user can select a filter by entering the corresponding letter or number.

Function1 : Black and white image

The code works by iterating over the image array and comparing each pixel value to 128. If the pixel value is greater than 128, the code sets the corresponding pixel in the new_image array to 255 (white). Otherwise, the code sets the corresponding pixel in the new_image array to 0 (black).

Function2 : Invert image

The code works by iterating over the image array and subtracting each pixel value from 255. This results in an image where the dark pixels become light and the light pixels become dark.

Function3: Merge Images

It takes two 2D arrays of unsigned characters as input parameters, image and new_image. The function reads an image file in grayscale format, specified by the user, and stores it in a 2D array named image_marge. The function then iterates over each pixel of the input images and calculates the average gray level of the corresponding pixels in both images. The new image is created with every pixel equal to this average gray level. Finally, the new image is stored in the new_image array.

Function4: Flip Image

The code you provided flips an image horizontally or vertically, depending on the user's choice. The code works by first creating a new image array to store the flipped image. Then, it iterates over the pixels of the original image and copies them to the new image array, flipped according to the user's choice. Finally, the code copies the flipped image back to the original image array.

Function5: Rotate Image

The function loops through every pixel in the image and rotates it by 90 or 180 or 270 degrees clockwise. The result is then stored in the corresponding pixel in the new image. Finally, the original image is updated with the new image.

Function 6: Darken and Lighten Image

The function prompts the user to select either a darken or lighten operation. If the user selects darken, the function iterates over each pixel of the input image and assigns the corresponding pixel in the new image to half of the original pixel value. If the user selects lighten, the function iterates over each pixel of the input image and assigns the corresponding pixel in the new image to a value that is 50 greater than the original pixel value, with a maximum value of 255. Finally, the new image is stored in the new_image array.

Function 7: Detect Image Edges

The function first applies a filter to convert the input image to black and white. The function then iterates over each pixel of the input image and checks if it is a boundary pixel. A pixel is considered a boundary pixel if it is black (0) and has at least one neighboring white (255) pixel. If a pixel is a boundary pixel, the corresponding pixel in the new image is set to white (255). Finally, the new image is stored in the new_image array.

Function 8: Enlarge Image

enlarge of The filter8_1, filter8_2, filter8_3, and filter8_4 functions are responsible for extracting one of the four quarters of the image array and storing it in the new_image array. Each function uses nested loops to iterate over each pixel of the image array. The selected quarter is determined by the given formula: $\text{image}[(i / 2) + 128][(j / 2) + 128]$, $\text{image}[(i / 2) + 128][j / 2]$, $\text{image}[i / 2][(j / 2) + 128]$, or $\text{image}[i / 2][j / 2]$, depending on the function. The extracted quarter is then stored in the corresponding position in the new_image array.

Function 9: Shrink Image

This code allows the user to shrink the image dimensions to 1/2, 1/3 or 1/4 of the original dimensions. The user is prompted to choose one of these options. The code then creates a new image with the chosen dimensions and fills it with white pixels. Finally, the code calculates the average pixel value for each block of pixels in the original image and assigns that value to the corresponding pixel in the new image.

Function a: Mirror Image

The function prompts the user to select which side of the image to mirror: left, right, upper, or down. If the user selects left or right, the function iterates over each pixel of the input image and assigns the corresponding pixel in the new image to the pixel at the opposite end of the row. If the user selects upper or down, the function iterates over each pixel of the input image and assigns the corresponding pixel in the new image to the pixel at the opposite end of the column. Finally, the new image is stored in the `new_image` array.

Function b: Shuffle Image

The code first prompts the user to enter the new order of the quarters in the shuffled image. It then iterates over the `new_image` array, and for each pixel, it copies the pixel value from the image array at the corresponding location in the new order.

For example, if the user enters the order 1 2 3 4, then the code will copy the pixel value from `image[i/2][j/2]` to `new_image[i/2][j/2]`, the pixel value from `image[(i/2)][(j/2)+128]` to `new_image[i/2][(j/2)+128]`, and so on.

Once the `new_image` array has been populated, the code copies the contents of the `new_image` array back to the `image` array. This is necessary because the `image` array is the input and output argument to the function.

Function c: Blur Image

The function loops through every pixel in the image and calculates the average value of the pixel and its surrounding pixels. The result is then stored in the corresponding pixel in the new image. Finally, the original image is updated with the new image.

Function d: Crop Image

The function prompts the user to enter the x and y position of the top-left corner of the square to keep, as well as the length and width of the square. The function then creates a new image by copying the pixels within the specified square from the input image and storing them in the new image. Finally, the new image is stored in the image array.

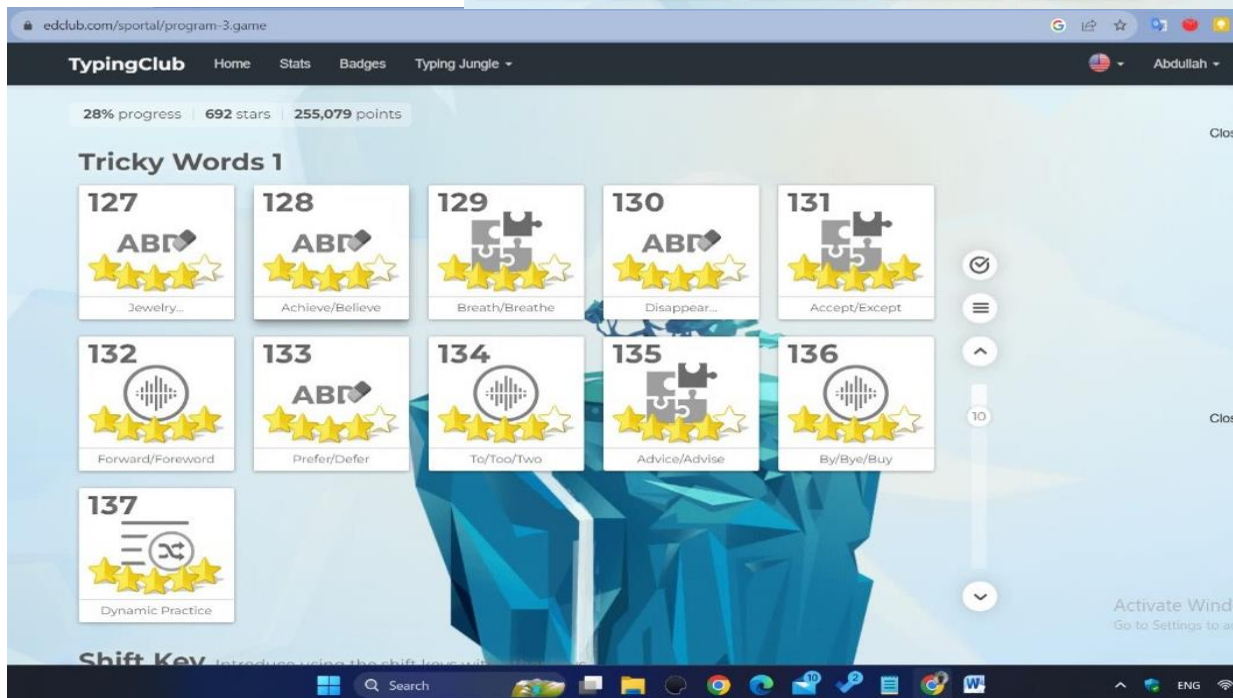
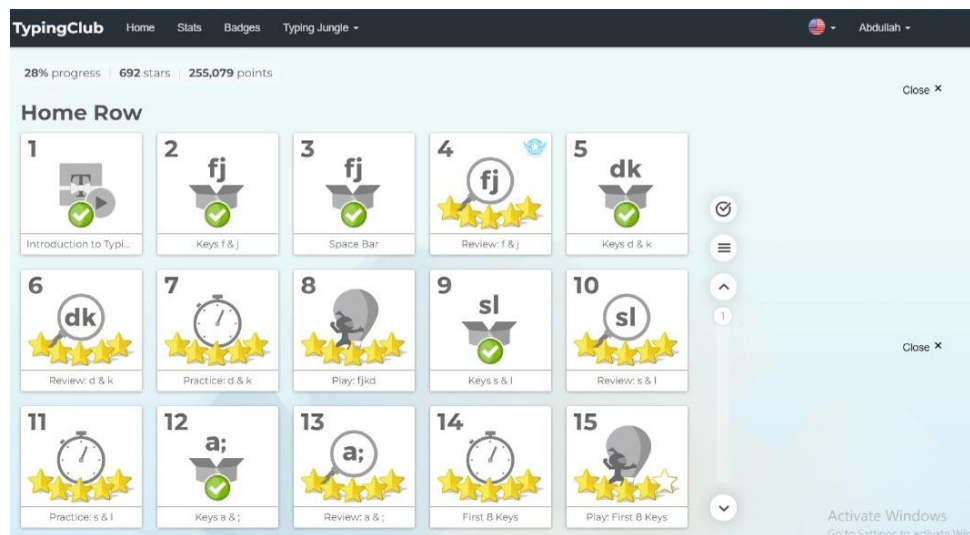
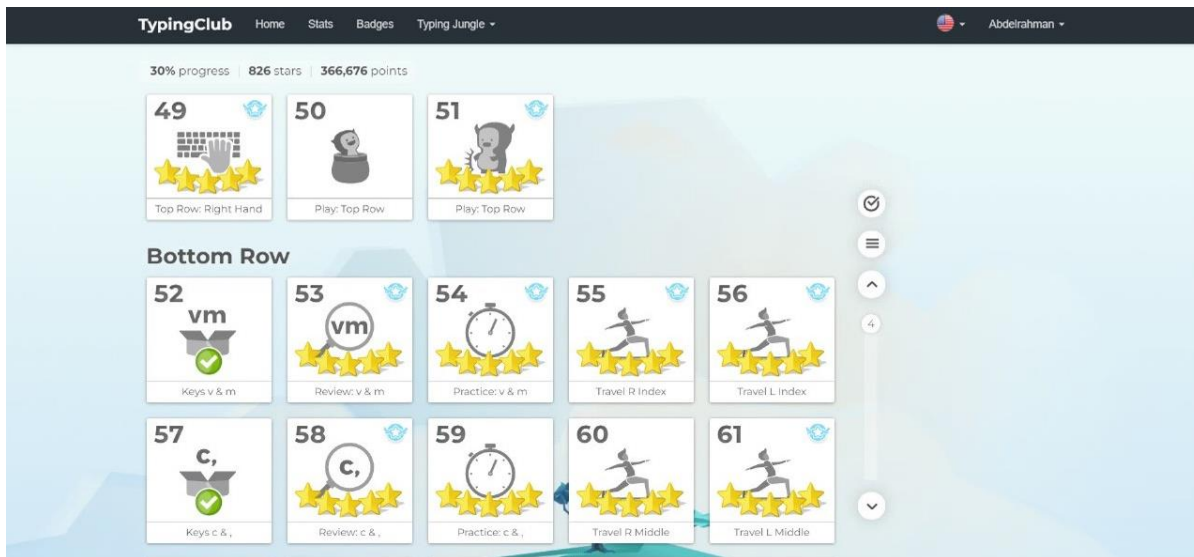
Function e, f: Skew Horizontally / Vertically

The code contains two functions named `Skew_Horizontally` and `Skew_Vertically`.

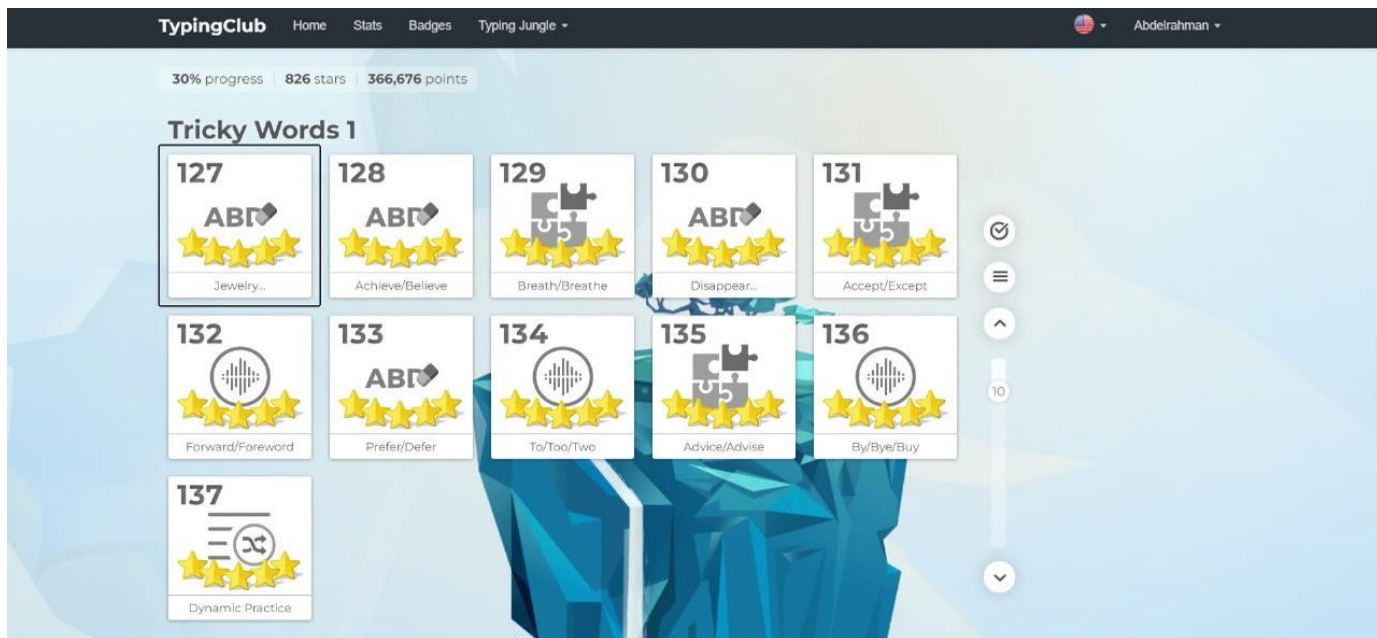
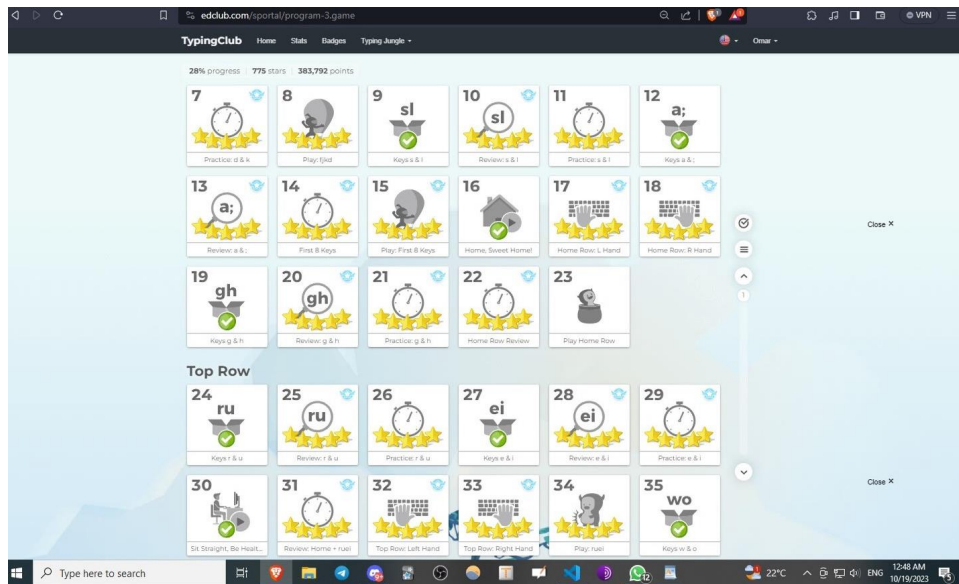
The `Skew_Horizontally` function takes two 2D arrays of unsigned characters as input parameters, `image` and `new_image`. The function prompts the user to enter a degree value to skew the image horizontally. The function then creates a new image by skewing the input image horizontally by the specified degree value. The new image is stored in the `new_image` array, and then copied to the `image` array.

The `Skew_Vertically` function also takes two 2D arrays of unsigned characters as input parameters, `image` and `new_image`. The function prompts the user to enter a degree value to skew the image vertically. The function then creates a new image by skewing the input image vertically by the specified degree value. The new image is stored in the `new_image` array, and then copied to the `image` array.

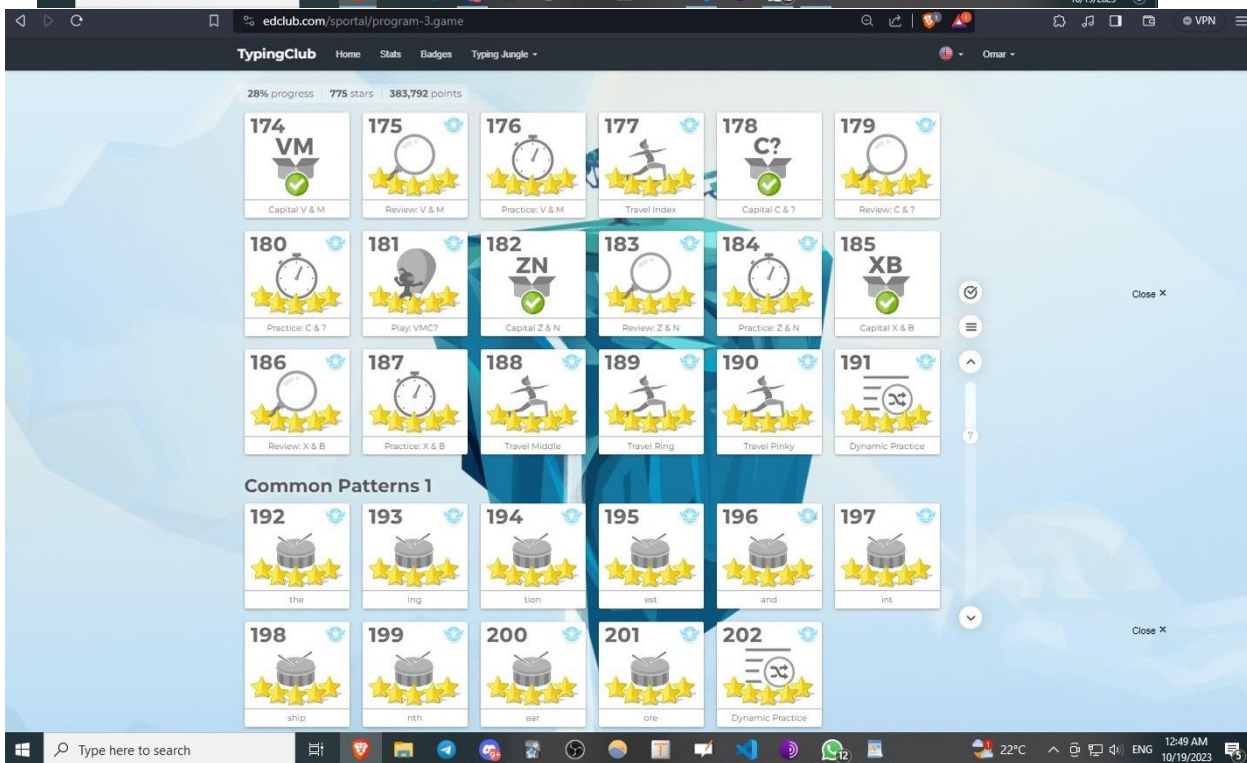
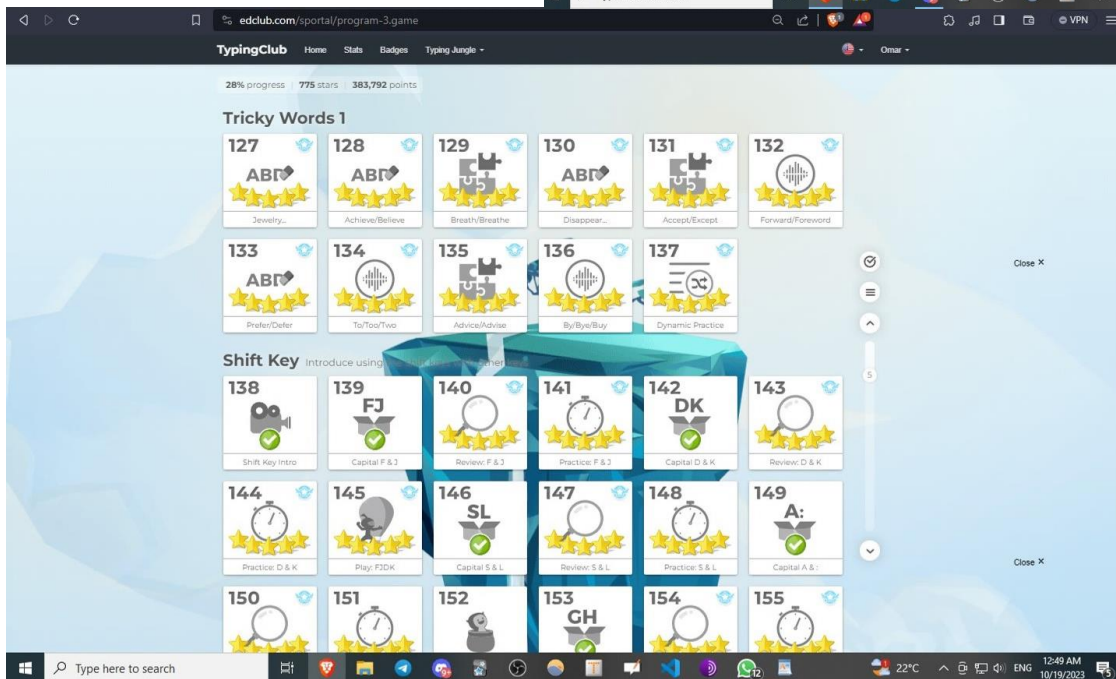
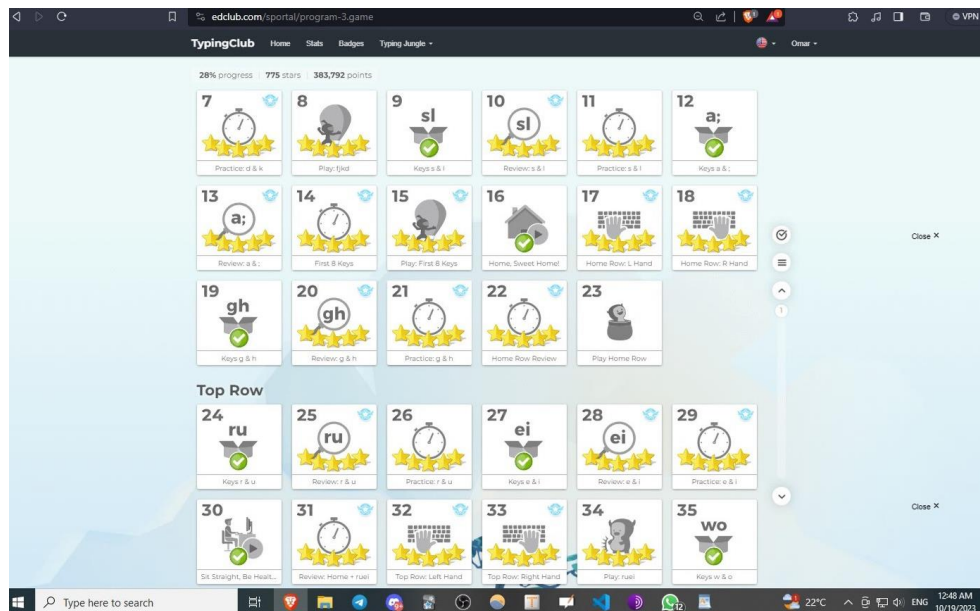
Abdullah mohamed
20220204



Abdelrahman Fathy
20220177



Omar Khalid
20220509



github.com/Abdallahyones/Filter_Image_RGB/commits/main

Abdallahyones / Filter_Image_RGB

Code Issues Pull requests Actions Projects Security Insights

Commits

main

Commits on Oct 19, 2023

- Update Rgbfilter (Blur and Skew Filters C, F).cpp Verified 9877711 <> Tip
OmarKhaleel2005 committed 1 minute ago
- Update Rgbfilter (Shrink Filter 9).cpp Verified 435c981 <> Tip
OmarKhaleel2005 committed 16 minutes ago
- Update Rgbfilter (Darken and Lighten filter 6).cpp Verified 288746f <> Tip
OmarKhaleel2005 committed 23 minutes ago
- Update Rgbfilter (Merge Filter 3).cpp Verified 365a843 <> Tip
OmarKhaleel2005 committed 24 minutes ago

Commits on Oct 18, 2023

- Update Rgbfilter.cpp for filter 2,5,8,b,e Verified e1f7e33 <> Tip
abdelrahman27104 committed 9 hours ago
- Update Rgbfilter.cpp for filter 2,5,6,b,e Verified 483f804 <> Tip
abdelrahman27104 committed yesterday

Commits on Oct 15, 2023

- Update Rgbfilter.cpp for filter 2,5,8,b Verified 38d7e62 <> Tip
abdelrahman27104 committed 3 days ago

Commits on Oct 14, 2023

- Interface of program and all filter 1,4,7,a Verified 2c0ae33 <> Tip
Abdallahyones committed 4 days ago
- Create README.md Verified b31280f <> Tip
Abdallahyones committed 4 days ago

Newer Older

Type here to search

23°C 12:54 AM 10/19/2023

github.com/Abdallahyones/Filter_Image/commits/main

Abdallahyones / Filter_Image

Code Issues Pull requests Actions Projects Security Insights

Commits

main

Commits on Oct 19, 2023

- Update Filters_gray (Filter F).cpp Verified 5db3a80 <> Tip
OmarKhaleel2005 committed 50 minutes ago

Commits on Oct 18, 2023

- Update Filters_gray.cpp for filter 2,5,8,b,e Verified 477abd6 <> Tip
abdelrahman27104 committed 9 hours ago
- Update Filters_gray.cpp for filter 2,5,8,b,e Verified 7d5a2ae <> Tip
abdelrahman27104 committed yesterday
- Update Filters_gray.cpp for filter 2,5,8,b,e Verified a866146 <> Tip
abdelrahman27104 committed yesterday
- Update Filters_gray.h for filter 2,5,8,b,e Verified da9e88f <> Tip
abdelrahman27104 committed yesterday

Commits on Oct 14, 2023

- Update Filters_gray.h (filters 3, 6, 9, c) Verified 51aeac3 <> Tip
OmarKhaleel2005 committed 4 days ago
- Update Filters_gray.cpp for filters 2, 5, 8, b, e Verified 3c22d44 <> Tip
abdelrahman27104 committed 4 days ago
- Update Filters_gray.h for filters 2, 5, 8, b, e Verified 5ac61c9 <> Tip
abdelrahman27104 committed 4 days ago
- Interface of program and all filter 1,4,7,a,d Verified d96a40f <> Tip
Abdallahyones committed 5 days ago
- Initial commit Verified 9429b25 <> Tip
Abdallahyones committed 5 days ago

Type here to search

23°C 1:01 AM 10/19/2023