Cairo University

Faculty of Computers and Artificial Intelligence

CS112

Structured Programming

Assignment #3 Report

2022

Supervise by: Dr. Mohammed El-Ramly

m.elramly@fci-cu.edu.eg

IDs:

Seif Elden Mahmoud Helmy --> 20210169

Mina Albert Saeed --> 202120417

Ziad Ahmed Mohamed --> 20210143

E-mails:

Mina ❼ mina.albert33@gmail.com

Ziad ❼ ziadeliwa93@gmail.com

Seif ❼ seifhelmy777@gmail.com

Algorithms:

Ziad's Algorithm:

for filter black and white i put a variable avg this variable improve the quality of the filter , first for loop is for rows in the matrix , the second is for columns the avg will increase by the matrix , the next assign for the avg is to make avg more efficient if statements is to check the limit for black and white since black and white is from 0 to 255 .

for filter Mirror .
first in the Mirror filter is the up Mirror I made a for loop that check the rows from the size and ends to 0 because i want to do Up filter.
the second for loop is to check from the 0 to size for the image , then the image will increase by the size - rows and size of columns .
The left Mirror filter i made the same for loops from the Up filter the changes here is the image will be size of rows and size of (size-columns) .
The right Mirror filter first for loop is to check the rows from 0 to size second is to check size in columns from size to 0 and will decrement,
The image will be Mirrored right because the image will be image for size in rows and in columns will be (size - columns).
The down Mirror filter same for loops for Right Mirror filter the difference is the image will be for rows will be (size - rows) and for columns will be the columns only. the remainder from the code is to know from the user what does he want .
if he wants Mirror up will press 1 if he wants Mirror down will press 2 if he wants Mirror right will press 3 if he wants Mirror left will press 4

if he inputs any other numbers the Mirror filter will not be executed.

For flip image filter.

First is the Vertical flip first for loop is to check the rows from 0 to The Size,

the second is to check the same on columns starts from 0 to The Size also next the image will be on the rows will be the (size – rows -1) as rows and as columns will be all columns.

second is the Horizontal flip same for loops from the Vertical first for loop is to check the rows from 0 to The Size,

the second is to check the same on columns starts from 0 to The Size also next the image will be on the rows will be all rows, but the columns will be (size – columns -1).

I made a function doFlip, to make the user to choose from the vertical flip or the horizontal flip is he wants the horizontal flip the choice will be 1, if he wants the horizontal flip the choice will be 2, any number expect the filter will not be executed.

For detect image filter.

I use the Sobel detection law first I made a for loop for rows starts from 1 to Size - 1 to avoid negative subtraction number next for loop starts also from 1 to Size - 1 and first array will be {(rows-1) and (columns -1) multiplies by 1 } + {(rows) and (columns -1) multiplies by 2 } + {(rows + 1) and (columns -1) multiplies by 1 } + {(rows + 1) and (columns -1) multiplies by -1 } + {(rows) and (columns + 1) multiplies by -2 } + {(rows + 1) and (columns + 1) multiplies by -1 } .

Same for loops I uses then the second array {(rows-1) and (columns -1) multiplies by -1 } + {(rows) and (columns -1) multiplies by -2 } + {(rows + 1) and (columns -1) multiplies by 1 } +{(rows + 1) and (columns -1) multiplies by 1} + {(rows) and (columns + 1) multiplies by 2 } + {(rows + 1) and (columns + 1) multiplies by 1} .

The image will be (√(first array ^2 + second array ^2)).

RGB ALGORITHIMS

Seif's Algorithm:

1- merge filter

- adding 2d array to move in every pixel in the image

- to merge we need to get the average pixels in the images

- so new image = (image1 pixels + image2 pixels)/2

2- lighten and darken filter

- let the user to choose what he wants (darken or lighten)

- if the user choose 1 he will dark the image

- adding 2d array to move in every pixel in the image and dividing by 2 to dark the image

- if the user choose 2 he will light the image

- adding 2d array to move in every pixel in the image and get the image to be equals 200

- to let every pixel = 200 (white gray)

- and then merge the (white gray) image with the original image

3- shrink

  - let the user choose what he wants (1/2-1/3-1/4)

  - getting the avg for 2 * 2 matrix for indexes (00, 01, 10, 11)

  - divide every pixel by the number which he choose (1/2-1/3-1/4) -----> image[i/2][j/2]

4- blur

  - getting the avg for a matrix 5 * 5 to get the middle element to blur the image

  - append the avg on the new image to blur it

Mina's Algorithm:

1-Invert Filter:

Make a function that loops through the image 2d- array and do a bitwise NOT (~) operation on every element to get its opposite

2- Rotate Filter:

Make a function that do the following:

First, we will transpose our image 2-d array (matrix) by swapping the rows by columns using this for loop:

```
for(int i = 0; i < SIZE; i++)
for(int j = i; j < SIZE; j++)
swap(image[i][j], image[j][i])
```

Second, we will swap the columns using two pointers approach using this for loop:

```
For (int i = 0; i<size of the array; i++)
        For(int j = 0; j < (size of the array/2); j++)
                Swap(image[i][j], image[i][size of the array - i  -j]
```

This filter rotate image by 90 degrees if we want to rotate the image by 180: we will call this function twice. If we want to rotate the image by 270: we will call this function three times.

3- Enlarge Filter:

Using two pointers approach, based on the user's input.

We will set the start and the end of our pointers to scan the selected quarter.

Two pointers for the old image (I, j)

Two pointer for the new image (new_i, new_j) this will increment by two every iteration

Using two nested for loops looping through the old image and load its pixels in the new image four times in a shape of square to cover the whole image.

4- Shuffle Filter:

First, we check if the user inputs a right input or not

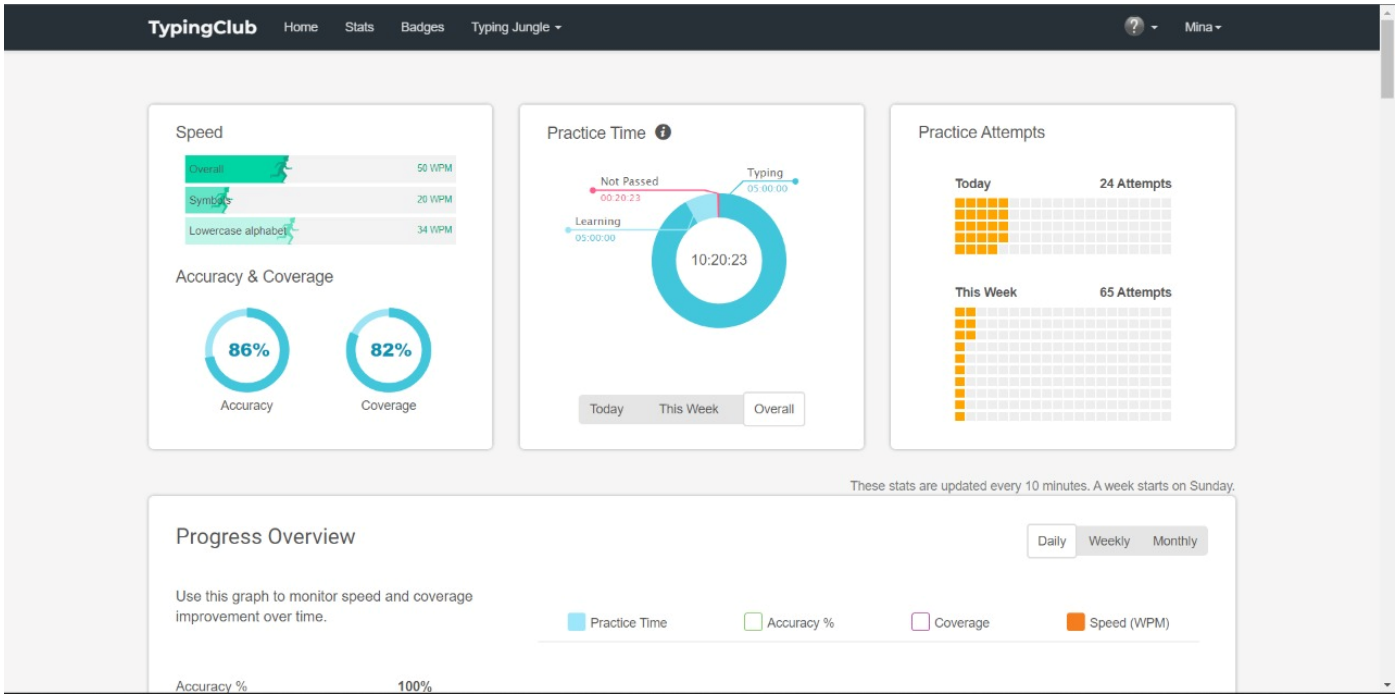
Second, we initialize the start and the end of the pointers we are going to use to scan the quarters by checking every char in the input

We use I ,j to scan the quarter from the old image

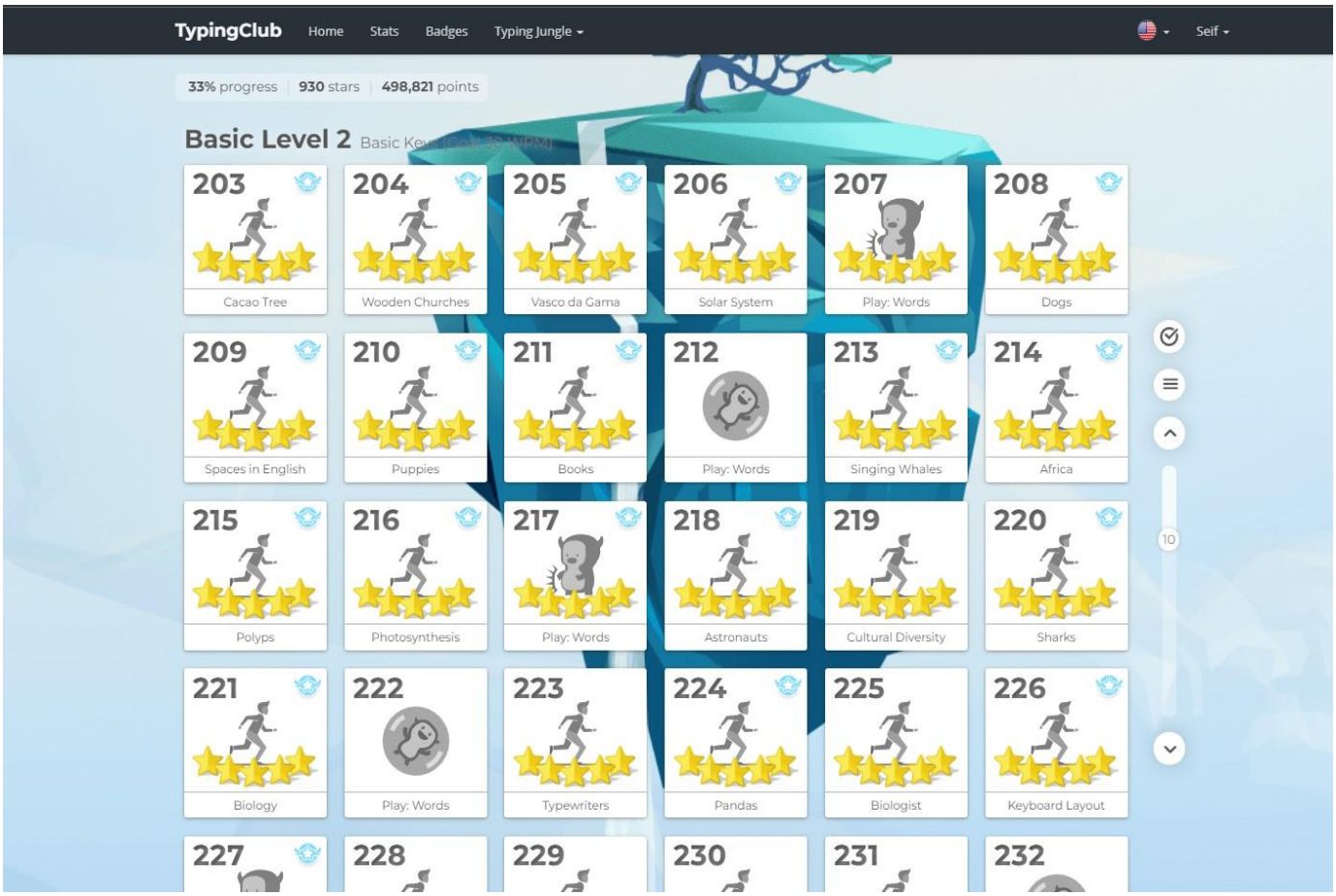
We use new_i, newj to traverse the quarter in the new image.

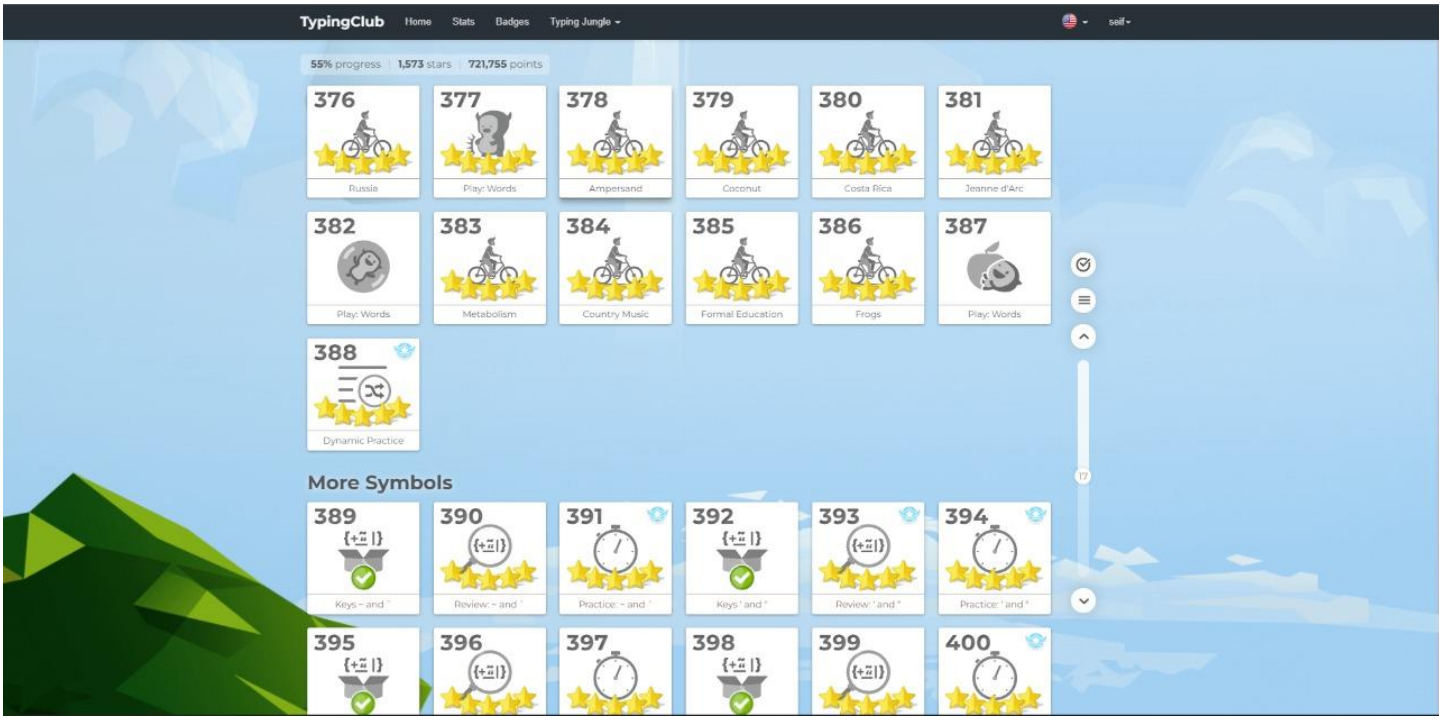
Typing screenshots:

Mina Albert's screenshots:

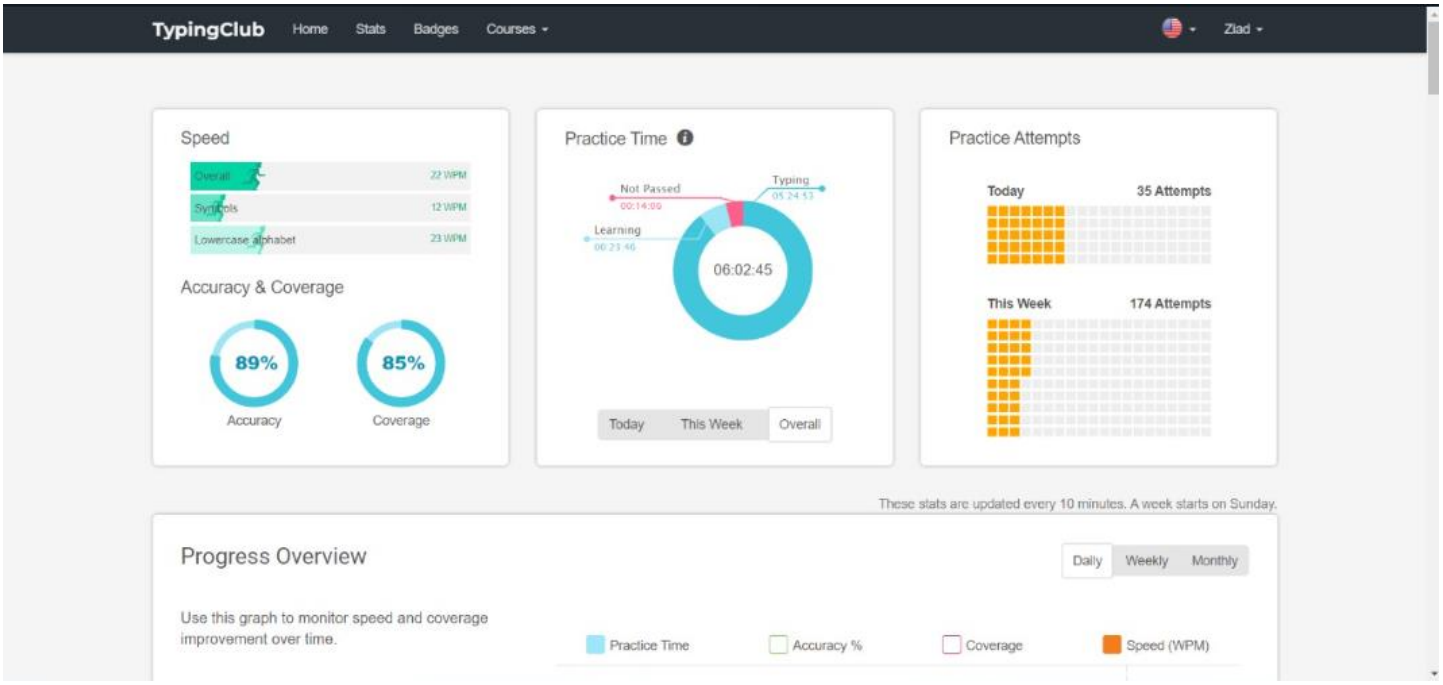


Seif’s screenshots:





Ziad’s screenshots:



The Diagram of the Functions: