

## **CS213 ASS.1**

### **Team:**

#### **Member one:**

Name: Mohamed Sayed Shebl

ID: 20220290

E-mail: modyshebl@gmail.com

Section no.: S20

#### **Member two:**

Name: Ali Emad Hodaib

ID: 20220210

E-mail: Alihodaib@gmail.com

Section no.:

### **Steps to collaborate on the project:**

1. We developed together the menu display function and the load and save functions.
2. We integrated all our functions together and tested all the functions.
3. We created a private repo on GitHub and maintained all needed files there.
4. We developed through the repository via work cycles of init, clone, change, add, commit, remote add origin, push.
5. We divided the responsibilities equally throughout the assignment

## Screen shot showing using Git work flow

The screenshot displays the GitHub interface for the repository 'Modyshebl/CS213-ASSIGNMENT-1'. The 'Commits' tab is selected, showing a list of commits on the 'main' branch. The commits are grouped by date, with the most recent ones from October 19, 2023, and older ones from October 18, 2023.

**Commits on Oct 19, 2023**

- Merge branch 'main' of <https://github.com/Modyshebl/CS213-ASSIGNMENT-1> (9bb9920)
- Modyshebl committed now
- this is the full integrated program and the bonus program (872926d)
- Modyshebl committed now
- Myfilters (99fd4d7)
- 3elwa18 committed 5 minutes ago
- Revert "These are my filters" (7f3afa7)
- Modyshebl committed 14 hours ago

**Commits on Oct 18, 2023**

- These are my filters (e0adb1b)
- Modyshebl committed 20 hours ago
- these are the essential files and libraries that we will work on (0eb5deb)
- Modyshebl committed 20 hours ago

The interface includes a search bar at the top, navigation links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The bottom of the screen shows a Windows taskbar with various application icons and system information.

## Typing Club Screenshots:

Member one Mohamed shebl:

The screenshot displays the TypingClub website interface for a user named Mohamed. The top navigation bar includes links for Home, Stats, Badges, and Typing Jungle. The user's profile shows 28% progress, 767 stars, and 379,118 points. The main content area features a grid of typing practice levels, each with a number, a keyboard icon, and a star rating. Levels 197 through 201 are completed, while levels 202 through 207 are locked. A 'Dynamic Practice' section is also visible. The bottom of the screen shows a Windows taskbar with various application icons and system information.

TypingClub Home Stats Badges Typing Jungle

28% progress | 767 stars | 379,118 points

197 198 199 200 201

int ship nth ear ore

202

Dynamic Practice

Basic Level 2 Basic Keys (Goal 30 WPM)

203 204 205 206 207

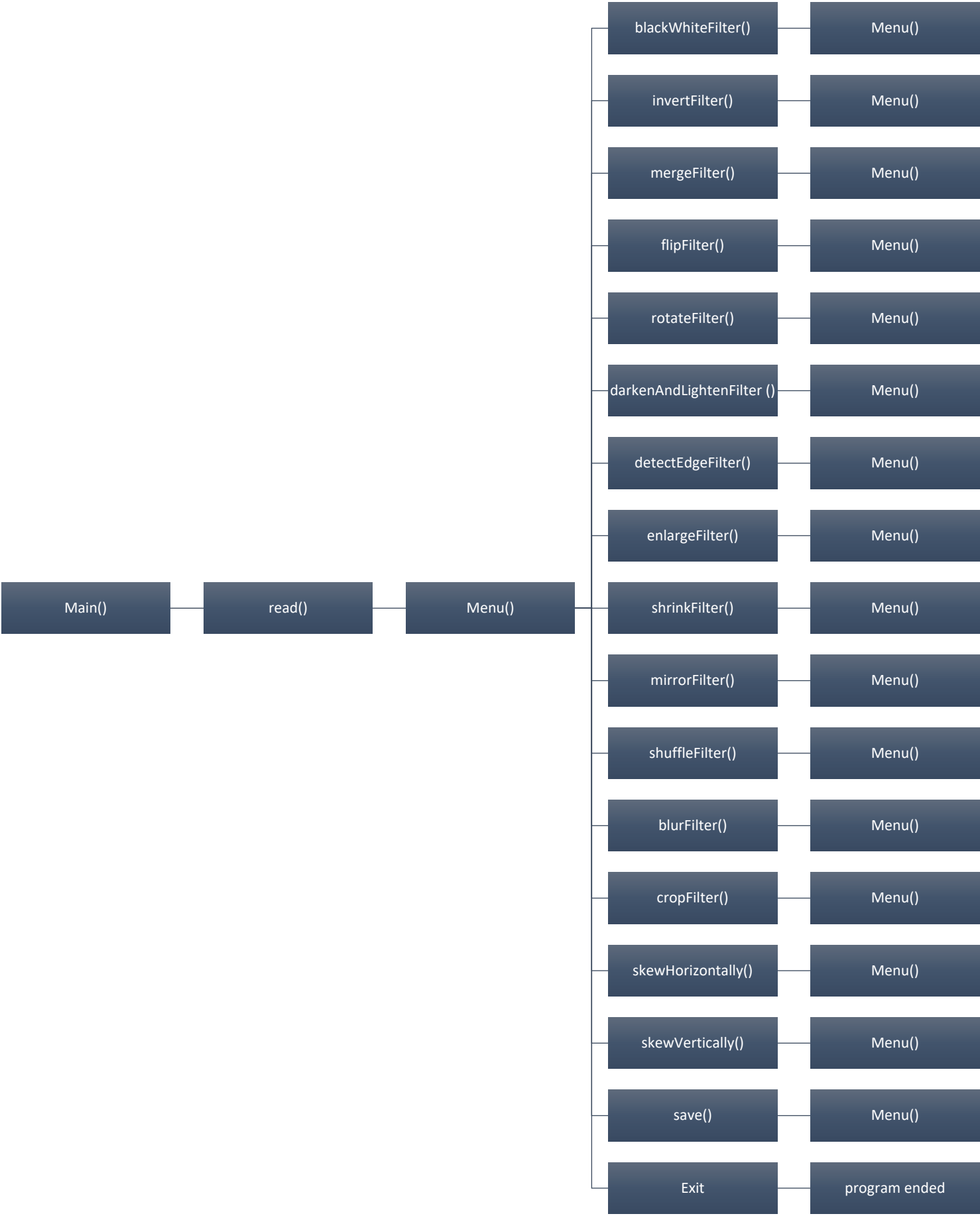
Cacao Tree Wooden Churches Vasco da Gama Solar System Play: Words

Waiting for securepubads.g.doubleclick.net...

Type here to search

26°C مشمس 12:38 PM 10/16/2023

**Function Decomposition Diagram**



## Algorithms of the filters

In every filter we will use nested loop to loop over all its pixels.

### Black and White filter:

Any pixel its number greater than 127 is nearer to white, so it will be equal to 255, otherwise it is nearer to black, so it will be zero.

### Invert Image filter:

Every pixel value changes to its negative value by subtracting it from 255.

### Merge Images filter:

First, load the second picture to merge it with the original.

Second, to merge the two pictures we get the average of each corresponding pixel in each picture and store it in the new picture.

### Flip Image filter:

In horizontal flip, horizontal flip means row value will be constant while column value will be (255 - column value).

In vertical flip, vertical flip means column value will be constant and row value will be (255 - row value).

### Rotate Image filter:

If we want to rotate it by **90 degrees**: we change rows from top to bottom to be columns from right to left by making:

value of row = 255 - (value of column)

value of column = value of row

If we want to rotate it by **180 degrees**, we will make:

value of row = 255 - (value of row)

value of column = 255 - (value of column)

If we want to rotate it by **270 degrees**, we will make:

Value of row = value of column

value of column = 255 - (value of row)

### Darken and Lighten Image filter:

**For lighten:** Increase each pixel's value by adding to it half its value. Therefore increasing its brightness.

**For darken:** Decrease each pixel's value by subtracting from it half its value. Therefore decreasing its brightness.

### **Detect Image Edges filter:**

First turn the picture black and white so it is easier to deal with.

Then check if the pixel is not the same value as the next horizontal or vertical pixel. If it is, then make it white. If it isn't, make it black

### **Enlarge Image filter:**

Extract the quarter that we will need to enlarge into another array.

Then overwrite the original picture by filling it out with the enlarged version of the quarter.

The enlarged version is created by filling out each 4 places [shaping a square] in the original array by 1 pixel of the quarter

### **Shrink Image filter:**

Copy the original picture then overwrite the original array with the shrunk version of the picture.

The shrunk version is created by cutting out a specific number of pixels in between the pixels that are written into the array

The specific number of pixels is dependent on the user's choice of the shrinking factor of the dimensions

If its half then two pixels get cut out each in between pixels, if its one third then 3 pixels get cut out, if its quarter then 4 pixels get cut out.

### **Mirror Image filter:**

First copy the original picture then according to what side you want to mirror we flip the copied version then overwrite the counter side of the side you want to mirror with the flipped version of itself in the original picture.

### **Shuffle Image filter:**

We first take the order that the user inputs (let's say 4231) then based on that order we extract the 4<sup>th</sup> quarter then overwrite the first 128x128 pixels in the original array using a for loop with iterators for the quarter and different ones for the original array with it then we continue this process in chronological order until we fill the 4 quarters.

### **Blur Image filter:**

We use the box blur method where we add each pixel to the surrounding 8 pixels then multiplying their sum by  $1/9$  then overwriting the value with the result.

### **Crop Image filter:**

Copy the picture then turn the original array all black then using for loops that which have constraints according to the length and the width inputted by the user with also different iterators for each array (the copied one and the original one) write the cropped segment into the black array(the original array)