Introduction

(1)background

I choose the second project—ASCII ART as my course work assignment. I write the code with help from lectures ,pasted link ,demonstrators and Google research.

(2)aim

The software allow user to select 1 image from 3 and convert it to ASCII ART image. (present the image with ASCII characters)User can input their desired colour for background and font(by rgb value). The final image can be saved into bmp file with a user input name.

(3)techniques

1.Counting loop

For converting image to ASCII, I need to divide the image to sub parts and calculate the brightness for each. Here I need a for loop to let the calculation going through sub parts by sub parts.

For my brightness calculation function,I need a concatenation to go through pixel by pixel for width and height and get rgb value to calculate the brightness.
I also use it in my ASCII table loading and calculation.

2.Simple branching(if functions)

I use this inside one of my switch case—mouth button down. I set a few numbers of event motion for each condition. While the user meets the condition(click in the area),the picture beside the button would be scaled to full screen mode.

3.global variables

I set a lot global variables try to reduce the word I need to type in the main function. Because I have a lot of counting loop, so I could just set a few global variables and assign them again and again in the for loop.Also I define the window width and height and sub width and height as global which allow me to change it once in the beginning but not changing anything in the main function.

4.multiple selection branching(switch)

I use this technique for the SDL event. It switch for the key down,key up,mouth button down and the sdl quit. And inside the case key down, I do another switch between the key s for save ,g for ASCII convert, Esc for quit and backspace for going back to the cover.

5.Function decomposition

Because the way to calculate each ASCII character brightness and sub images are the same. So I set an int function with parameter of surface,the start and end for the area to calculate. The function return brightness value of that area.
I called this brightness calculation function in the counting loop. Because of increment, it change the parameter while looping and reach my goal for calculate it sub by sub going through the whole window.

(4)algorithms

Step1:create window and build up cover

Step1.1:initialize SDL

Step1.2:Create a window
Step1.3:get window surface
Step1.4:load all the images materials and title text on several surface
Step1.5:Scaled all the loaded surface onto window surface

Step2:click button to scaled image
Step2.1:start a while not finished loop
Step2.2:use multiple branch to a mouse button down case
Step2.3:use single branching for the button area condition
Step2.4:free the surfaces for saving memory

Step3:press g to start convert to ASCII ART with asked colour
Step3.1:use multiple branch to a key button down case
Step3.2:another multiple branch under it to a key g pressed case
Step3.3:Calculation part for the ASCII
Step3.3.1:load in ascii table
Step3.3.2:calculate each ascii character brightness,load in ascii character table
Step3.3.3:calculate image brightness sub by sub using the increment,load in image brightness table
Step3.3.4:compare the brightness of character and sub image
Step3.3.5:use a counting loop ,for each sub bitmap,present the character which has the minimum difference
Step3.4:Present part for the ASCII

Step4:press s to save the work with asked name
Step4.1:create a empty char array
Step4.2:scan the user input name for output image
Step4.3:use the Saving function replace the name with the char array

Step5:press backspace going back to cover
Step5.1:multiple branch for a s pressed case
Step5.2:free the recent surface for saving memory
Step5.3:redo the whole Step1 to get a new cover

Step6:go back to step 2 until close window

(5)idea

At first,while I finished my interface part, I try to add code about calculation in it. But that make the code very long and it keep on going segmentation fault. So I divide my code into 2 parts write and test them separately .

1.interface part(the cover and the button and keyboard event)
2.calculation part(load ascii table,fill the image brightness table, compare them and present)

This make my work simpler because with short code I could find the mistake faster. I also do that for my calculation part. The calculation part is form by the ascii calculation part,the image calculation part,the compare part and the present part. I don't write them together but one by one and combine them at last.

Another thing I do to test and find mistake in my code is use the printf function in counting loops to print each elements that I load in the table to see whether the right number is load in the right place of the array.

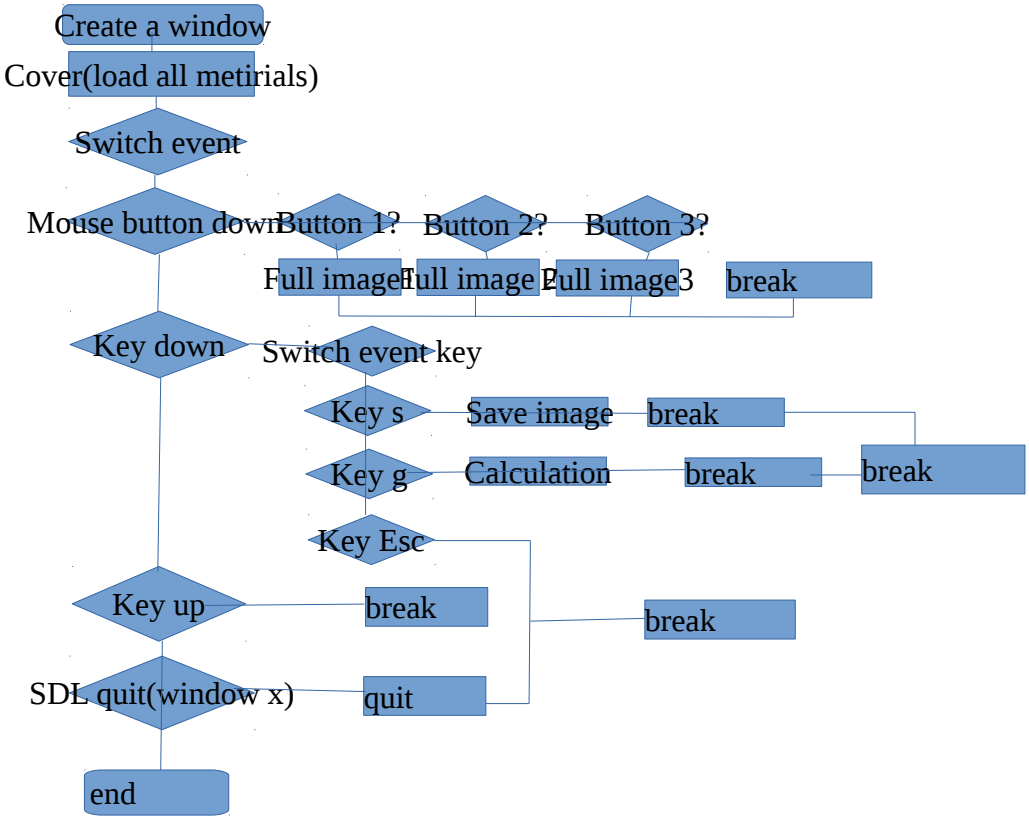
(I make a mistake by just pasting my code from another file and ignore the readability. The demonstrator help me with the interface part readability and I fixed the calculation

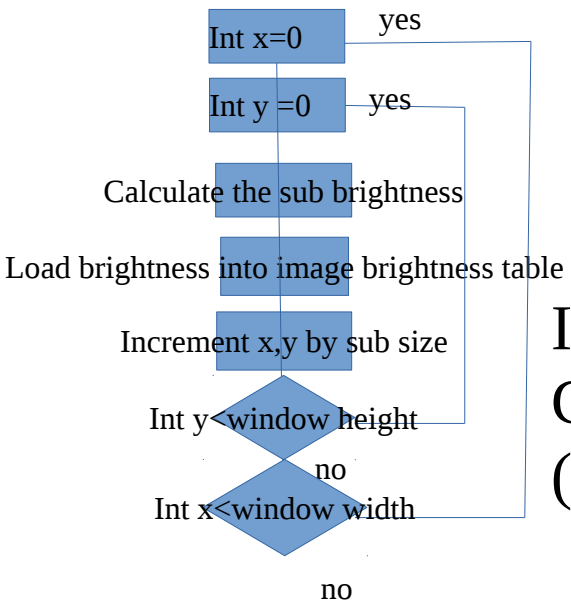
part readability myself.)

2Implementation

(1)structure

A simple flow chart for my event loop

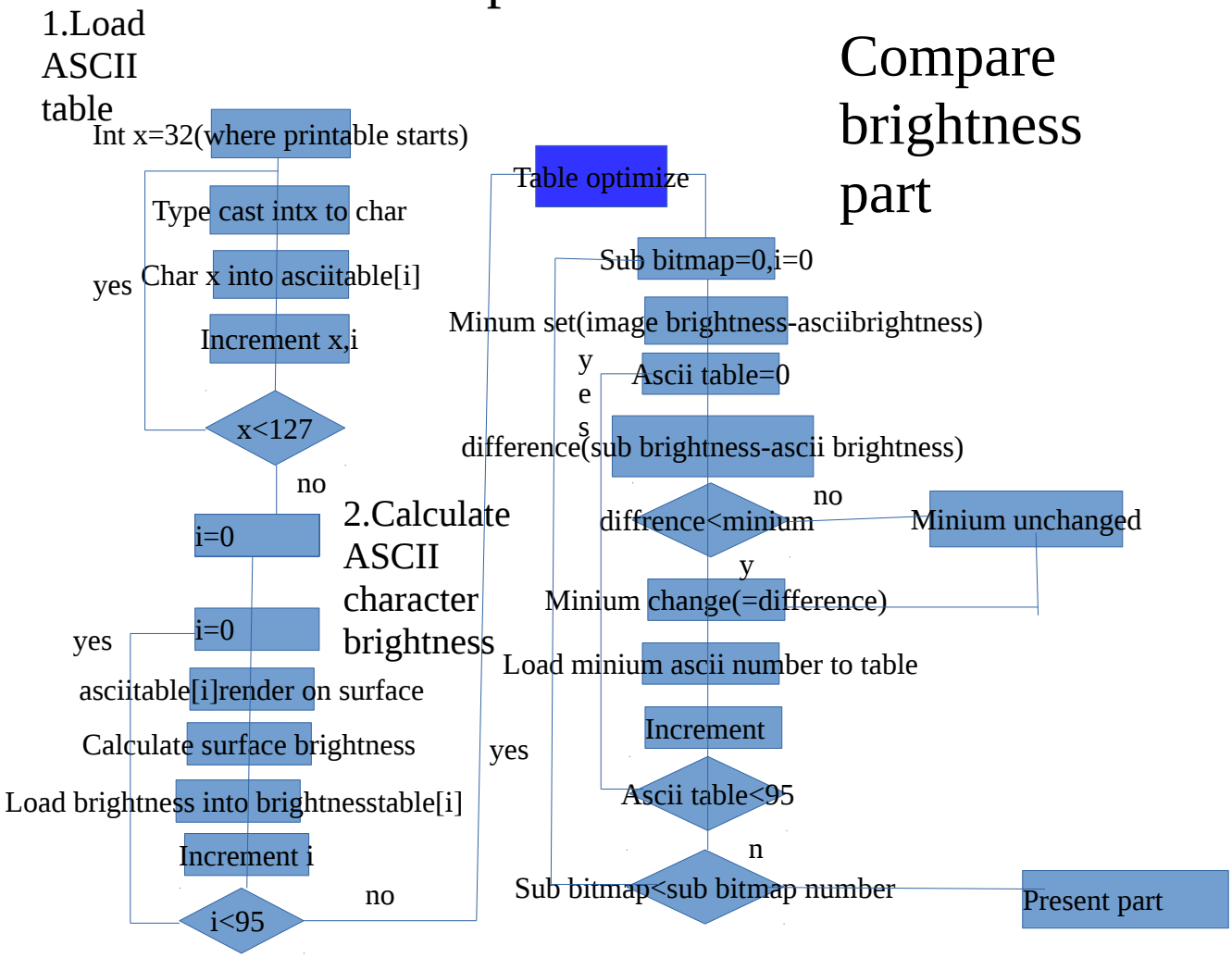




Simple flow chart for my calculation part

Image Calculation part (sub by sub)

ASCII Calculation part



(2)flow of control

My work is a water fall model,it doesn't allow the retrospective changes.

(3)most important functions

(1)the brightness calculation function

This is the centre function for the whole source code. The main goal for the software is fill sub images with same brightness character.So without this function,all the code would not work.

For this part of code,I used the source code from gigi lab and put it in a counting loop and make this function return a interger value.

I used to use the brightness=r+b+g/3 function,but I find the out put is more accurate with the brightness=0.212671f*r+0.715160f*g+0.072169f*b.

I used to struggle how to add all per pixel brightness together by counting loop ,but I find out that using a increment in the loop can solve the problem with just one line.

(2)character brightness table optimization.

Before writing the brightness optimization,I struggling with the font size of each single character because I think many blank would influence the final result.But after looking at the pasted link, I find out that the character brightness is not the final brightness use to map with sub images.It also need a table optimization to strecth it to a proper range from 0-255.When I compile without optimization,I couldnt print out a proper image.

Results

(1)images

While picking images I hasn't choose those with very high resolution.Because my code go through each pixel in a sub image, so if the original loaded image resolution is very high,the brightness calculation counting loop excute for too much time,the software would crash.

Different sub image size and font size influence the final present picture.I define the sub iamge width and height as global variables and I try to compile the code with several different number for them.The out put for smaller sub settings are more clearly but takes more time to calculate.So finally I choose 8x8sub and font 8 balancing between out put present quality and waiting time.

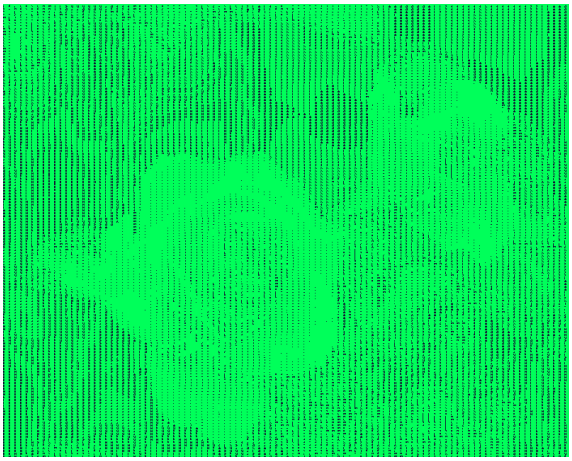
Original imaes

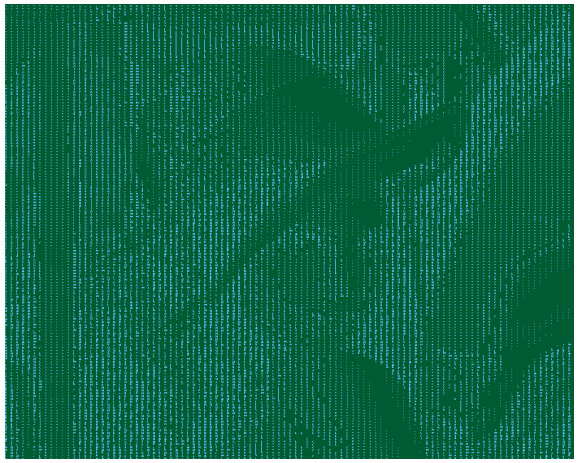
*ref2

*ref1

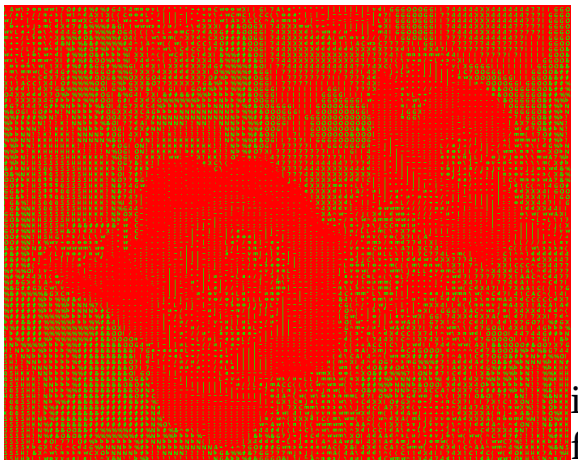
*ref3

sub image for 8x4 and font size of 4

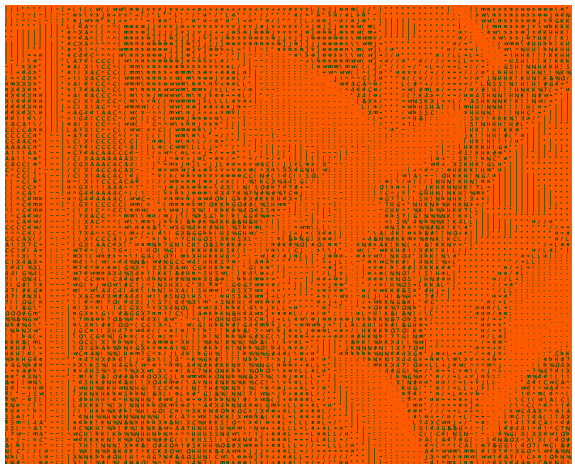


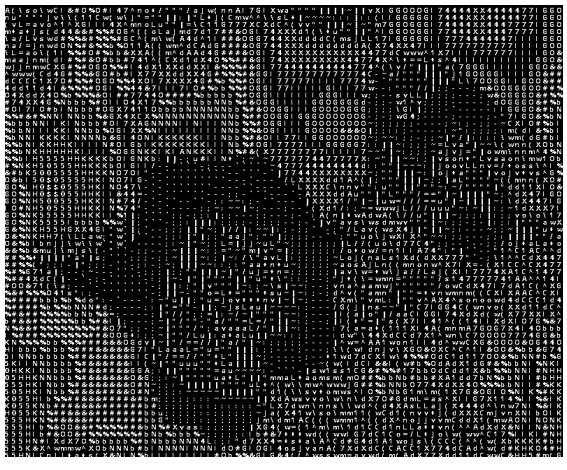


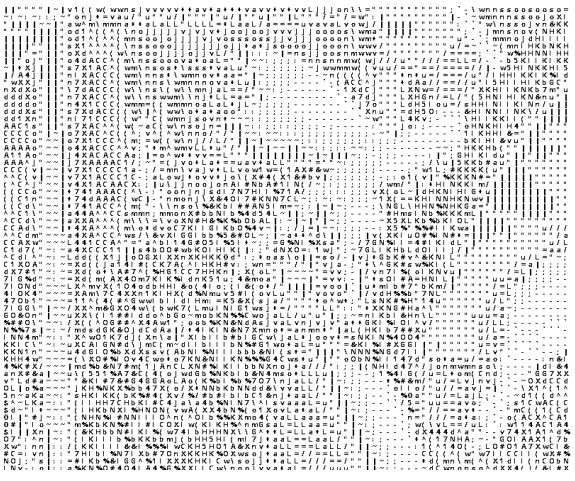
sub image for 8x8 and font size of 8



sub image for 10x10 and font size of 10





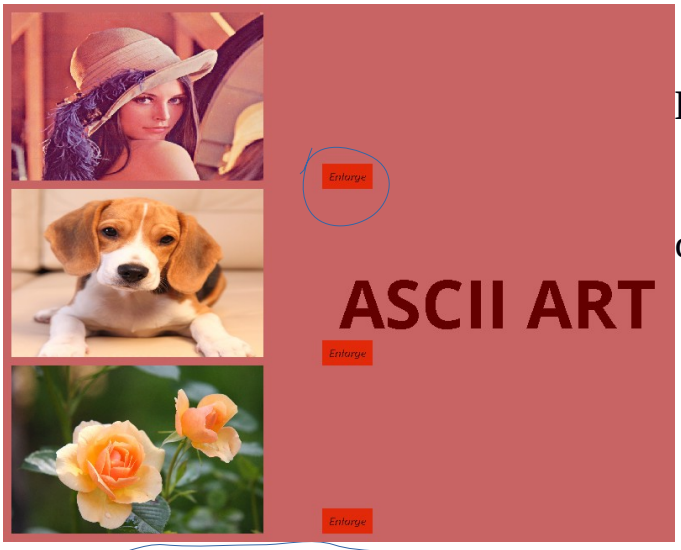


(2)explain the work

the work start with type in the make for make file.
Then type ./test3 to open the window.

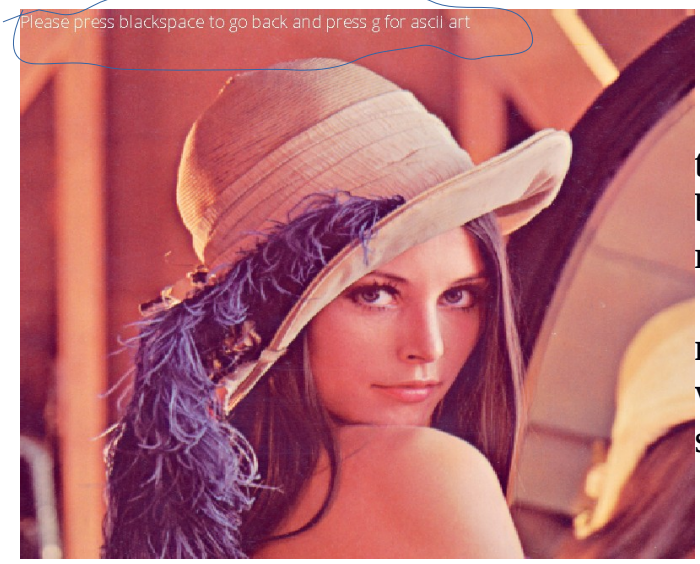
```
File  Edit  View  Search  Terminal  Help
s5068691@w11541:~/pcoureswork$make
clang test3.c -L/usr/local/lib/ -lSDL2 -lSDL2_ttf -lSDL2_image -lm -o test3
s5068691@w11541:~/pcoureswork$./test3
Clicked on button 1
Clicked on backspace
Clicked on button 2
Clicked on backspace
Clicked on button 1
n,Please Enter name for picture:
hhhhhh
The picture is saved!
Clicked on backspace
Clicked on button 2
g is pressed,please wait for the calculation
n,image brightness calculation is finished,please wait
ascii brightness is ready please keep waiting
compart is finished,it is almost done!
pleae input the rgb colour u want for the background and font:200,10,90,0,0,100,
present finished!
press s for saving your work!
n,Please Enter name for picture:
The picture is saved!
Please Enter name for picture:
pinkd
```

p://www.dreamincode.net/forums/topic/175010-creating-buttonsmenus-in-sdl/*/



While the window open,it should be like this

It contain three image wait for choosing by 3 buttons and a title



While the user click on the button, the terminal would type “click on button x” to make sure the event is received.

The image would be scaled and a note would present on the left top writing the instruction for the next step.

While user press the button g,the terminal would type”g is pressed ,please wait for calculation.”

While the calculation part is finished,the terminal would ask user to input 6 intergers ,the first 3 is link to the rgb value for the background color and the last 3 is link to the rgb calue for the font.

When the present part is finished the terminal would say”Present finished! Please press s to save your work!”and ask for a name for the saving image. After this, if user press backspace.It would go back to cover and let the user choose again from beginning .

Improvement & Issues

(1)Speed

The speed for my calculation gets slower and even the software crash at the third (depends on the size of sub bitmap)time when user try to redo the calculation.Which mean the user can only produce 2 ASCII image each time open the program.The problem should be running out of memory to run the program.I try to use “meset” function and counting loop to clean up the tables for image sub brightness but it doesn't work. I think I should try to free the dynamic allocation after each time after saving the picture.

(2)quick sort

Because I used printf funtion to check the function each time, so I set the max and min of the character brightness table myself after looking at the terminal printed number.But I think the code should be better working itself without my own influence.So I think it would be better to write a sort function to arrange the table from small to large number.

(3)more decomposition

Although I have do some functional decomposition(the brightness calculation function),I still think the function under the pressed g case is too long. It would be better to turn them into a more short and composite way by turning each part to a small functions.

(4)average color function

I only let the user to choose the background color and font color. I think I should do more on makking a choice of converting it to ASCII with average colour.I have the rgb for each

sub groub,I could use another counting loop to calculate the average r average g and average b and load them into another table and print each different sub group with font number of this r,g and b.

(5)printf

There is a little problem with my print please input image name line.It would sometime junp through the scanf line dirctly to the image saved!line.Although I add an if function to avoid this happen but it still some time jump out.

Conclusion

I've learned and used a lot programming knowledge while doing this project.At past they are just character in lecture to me but after writing coding myself I start to understand and memorize them in my heart.By learning using SDL I discover a lot coding websites(lazy foo,gigi lab).I also learn the way to do reasearch and self study using Google.The feeling from knowing nothing to actualy solve a task is so cool!I still got a lot more to learn and I start to find coding interesting. Although I have some times upsetting and struggling with the algrithms but looking backwards now,the whole experience is fantastic.

Reference

ref1: Dwight.H.Dwight Hooker.(1972).Lenna.[image] Available at:https://en.wikipedia.org/wiki/Lenna

ref2:Reinbacher.R.Reinbacher.(2014).Canon EOS 40d.[image]Available at:https://pixabay.com/en/rose-flower-blossom-bloom-616013/

ref3:WhatdogRobot.W.WhatdogRobot.(unkown).Beagle.[image]Available at:https://www.what-dog.net/#results