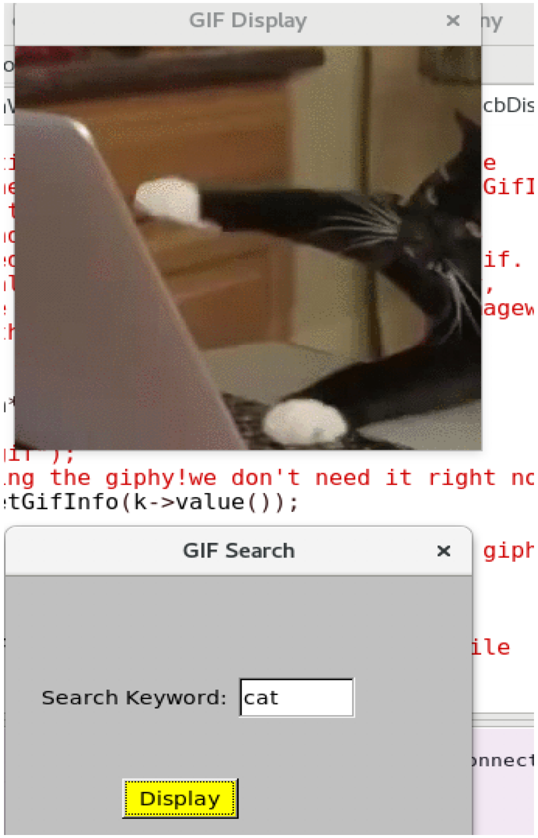
Specification



```
ng the giphy!we don't need it right nc
tGifInfo(k->value());
```

Search for a gif file related to a keyword the user types into a GUI textboxand display it in a separate window.

Analysis

Inputs A keyword entered into a textbox

Process Send query to mashape giphy API to retrieve information about a gif matching the user'skeyword. Parse the info to find a single gif to download using wget program.

Outputs A gif image related to the keyword

Design

main Call functions makeSearchWindow and makeDisplayWindow

getGifInfo Call searchGiphy to get information about gif from mashape

searchGiphy libcurl interface to communicate with internet

makeSearchWindow Create window to get input from user

makeDisplayWindow Create window to display the gif image

cbDisplay Callback function when user clicks button to call getGifInfo and put returned gif image inbox for display

Implementation

```
objs = main.o makeSearchWindow.o makeDisplayWindow.o \
cbDisplay.o getGifInfo.o searchGiphy.o

FLTKCXXFLAGS = `fltk-config --cxxflags --use-cairo --use-images`
FLTKLDFLAGS = `fltk-config --ldflags --use-cairo --use-images`

CXXFLAGS = -std=c++11 -I /home/debian/fltk-1.3.4-2 ${FLTKCXXFLAGS}
LDFLAGS = ${FLTKLDFLAGS} -lcurl

all: ${objs} lab.h
	g++ ${CXXFLAGS} -o lab ${objs} ${LDFLAGS}

clean:
	rm *.o
	rm lab

	#
```

```
#include "lab.h"

/*mainsets up the GUI interface and passes control to
FLTK subsystem.Before  returning  it  cleans  up  the
gif  by  removing  it  from  the   working directory.
*/

int main()
{
        makeSearchWindow()->show();
        makeDisplayWindow()->show();
        Fl::run();

        system("rm giphy* ");
        //We don't need it right now!

        return 0;
}
//
```

```
#include "lab.h"

Fl_Cairo_Window* cw;
Fl_Input* k;
Fl_Button *b;

Fl_Cairo_Window* makeSearchWindow()
{
        cw = new Fl_Cairo_Window(WIDTH,HEIGHT);
        cw->label("GIF Search");
        k = new Fl_Input(0.5*WIDTH, 0.25*HEIGHT, 0.25*WIDTH, 0.1*HEIGHT,
        "Search Keyword: ");
        b = new Fl_Button(0.25*WIDTH, 0.5*HEIGHT, 0.25*WIDTH, 0.1*HEIGHT,
        "Display");
        b->color(FL_YELLOW);
        b->callback((Fl_Callback * ) cbDisplay);
        return cw;


}
//
```

```
#include "lab.h"

Fl_Cairo_Window* dw;
Fl_Box * gb;

Fl_Cairo_Window* makeDisplayWindow()
{
        dw = new Fl_Cairo_Window(WIDTH,HEIGHT);
        dw->label("GIF Display");
        dw->position(cw->x()+cw->w()+MARGIN, cw->y());
        gb = new Fl_Box(0,0,WIDTH,HEIGHT);
        gb->image(new Fl_GIF_Image("giphy.gif"));
        return dw;

}
//
```

```
#include "lab.h"
extern Fl_Input* k;

/*
Write the Callback function for the button. It will use the
system call to delete the old giphy.gif file, thencall getGifInfo
(passing in the keyword the user typed into the input box.
Then use the wget commandin a system call to retrieve] the
gif from the url returned by getGifInfo, naming it giphy.gif.
Then usedynamic memory allocation to put the gif in memory,
and  call theFl_Boximage function to associatetheFl_GIF_Imagewith
the box.  Lastly, call the window redraw function.
*/

void cbDisplay(Fl_Button*, void*)
{
        system("rm giphy.gif");
        //This is for removing the giphy!we don't need it right now!!
        std::string url = getGifInfo(k->value());

        url = "wget " + url + " -O giphy.gif";//name the file giphy.gif

        system(url.c_str());

        gb->image(new Fl_GIF_Image("giphy.gif"));//find the file
        dw->redraw();
}
//
```

```
#include "lab.h"
#include <iostream>
#include <fstream>
#include <sstream>

/*
Call searchGiphy passing the in the keyword stored in s.
Parse the longstring returned to find the original gif
and extract the url needed to re-trieve it.  The string
returned by giphy is in json format.   We are usingstring
 functions to find unique words within it and then getting
  the sub-string.
*/

std::string getGifInfo (std::string s)
{
std::string gifInfo = searchGiphy(s);


std::string gifURL = gifInfo.substr(gifInfo.find("original"),200);


//declaring a string named gifURL then assign substr to gifInfo,
//the first word of gifInfo is original with the lenght of 200 characters
size_t p1 = gifURL.find("http");
size_t p2 = gifURL.find("\"",p1);
//size_t is an unsgined integer

gifURL = gifURL.substr(p1, p2 - p1);
return gifURL;

}
//
```

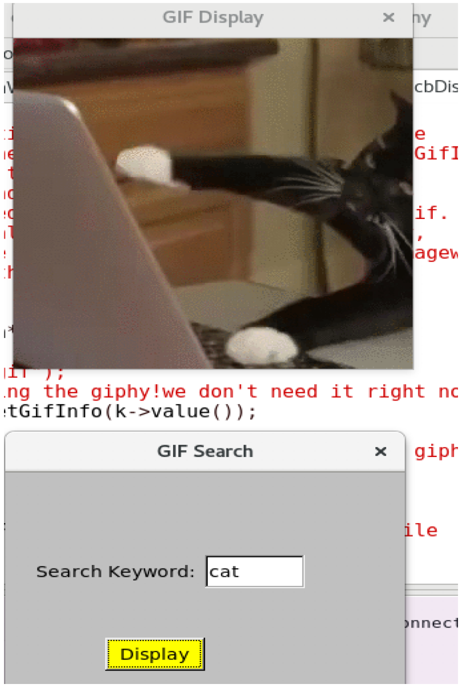
```
#include <iostream>
#include "mashape.h"
#include <curl/curl.h>

/*curl is a C API to interface with internet sites.  We have
to set up thequery string and collect the page(s) returned into
a string.  This is becauselarge pages will be sent in packets
rather than all at once.  handleData isa way to collect and
recombine these packets.  Read this link for information about
the 'callback' function */
/*
https://curl.haxx.se/libcurl/c/CURLOPT_WRITEFUNCTION.html
The include file has the specific `keys` needed for mashape.
*/

//libcurl
size_t handleData(void* c, size_t s, size_t n, void* j)
{
        *static_cast<std:: string*>(j) += static_cast<char*>(c);
        return s * n;
}
std::string searchGiphy(std::string k)
{
        std::string s = "";
        struct curl_slist* slist1 = NULL;
        slist1 = curl_slist_append(slist1,key.c_str());
        slist1 = curl_slist_append(slist1,js.c_str());
        std::string q = url + api + "&q=" + k;
        CURL* hnd = curl_easy_init();
        curl_easy_setopt(hnd,CURLOPT_URL,q.c_str());
        curl_easy_setopt(hnd,CURLOPT_HTTPHEADER,slist1);
        curl_easy_setopt(hnd,CURLOPT_WRITEFUNCTION,handleData);
        curl_easy_setopt(hnd,CURLOPT_WRITEDATA, &s);
        curl_easy_perform(hnd);
        curl_easy_cleanup(hnd);
        return s;
}
//
```

Test

Testcase 1

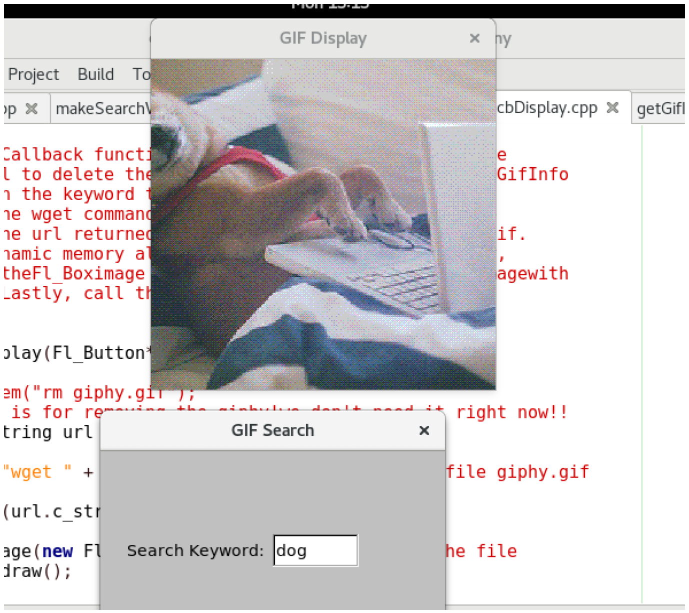


```
ng the giphy!we don't need it right nc
tGifInfo(k->value());
```

Search Keyword: cat

When the user typed cat as the key word, the program will show the user the cute cat.

Testcase 2



```
Callback functi                               e
l to delete the                               GifInfo
n the keyword t
ne wget command
ne url returned                               if.
namic memory al                               ,
theFl_Boximage                                agewith
Lastly, call th

play(Fl_Button*

em("rm giphy.gif");
 is for re                               right now!!
tring url

"wget " +                                file giphy.gif

(url.c_str

age(new Fl                               he file
draw();
```

Search Keyword: dog

When the user typed dog as the key word, the program will show the user the lovely dog.

Conclusion of my testcases

As we can see in my testcases, all of my codes run well.

I successfully acheived the goal of this lab assignment!