

## CS 102 Lab 4: Release the Gifs

# Specification

---

This program will allow the user to type in a search word and then search the Giphy website to find a gif related to the word they searched. Once it finds the gif it will display it.

The API for giphy can be found for free on the mashape marketplace

# Analysis

---

We will use the `market.mashape.com/exploremashape` market to get the API for the Giphy website from `market.mashape.com/giphy/giphy`. We will use the CURL API, from `curl.haxx.se/libcurl` to get the gif file. The gif will be animated using FLTK and Cairo from `www.fltk.org/doc-1.3/group__group__cairo.html`

Each url is a hyperlink which leads to a webpage where you can learn more about each part of the program

# Design

---

- labh
  - Contains all shared headers and variables
- makeSearchWindow
  - Makes a window with a search box that people can use to find Gif
- getGifInfo
  - Get data, in json format, from Giphy API from mashape specific to the users keywords
- extractGifInfo
  - Given a string in json format with all the info about the gif and return the information
- cbDisplay
  - Uses given keyword, finds appropriate gif, and displays the gif
- main
  - Use console to test getting and displaying a gif then using a GUI to let user choose their own file

Each of the implementation slides, and therefore each .cpp file, has one function in it. The function shares the name of the file aswell.

# Implementation lab.h

---

```
#include <config.h>
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <FL/Fl_Cairo_Window.H>
#include <FL/Fl_Input.H>
#include <FL/Fl_Button.H>
#include <FL/Fl_Box.H>
#include <FL/Fl_GIF_Image.H>
const int width = 300;
const int height = 300;
Fl_Cairo_Window * makeSearchWindow();
Fl_Cairo_Window * makeDisplayWindow();
void cbDisplay(Fl_Button*,void*);
extern Fl_Cairo_Window * dw;
extern Fl_Box * g;
extern Fl_Input * r;
extern Fl_Cairo_Window * cg;
std::string extractGifInfo(std::string g);
std::string getGifInfo(std::string s);
```

- Contains all variables and headers files used by other functions

# Implementation makeSearchWindow.cpp

---

```
#include "lab.h"
Fl_Cairo_Window * cw;
Fl_Input * r;
Fl_Button * b;

Fl_Cairo_Window * makeSearchWindow(){
    cw = new Fl_Cairo_Window(width,height);
    cw->label("Search For A Gif");
    cw->color(fl_rgb_color(74,189,172));
    r = new Fl_Input(.5*width,.25*height,.25*width,.1*height);
    r->label("Keyword: ");
    b = new Fl_Button(.5*width,.5*height,.25*width,.1*height);
    b->label("Display");
    b->color(fl_rgb_color(247,183,51));
    b->callback((Fl_Callback*) cbDisplay);
    return cw;}
```

- Make a Window to obtain keyword
  - Makes a Window called "Search For A Gif"
  - Makes type box for keyword
  - Makes button to send keyword to other functions

# Implementation getGifInfo.cpp

---

```
#include "lab.h"
#include <curl/curl.h>
const std::string url = "https://giphy.p.mashape.com";
const std::string key =
"X-Mashape-Key: N2gdkwRwq8mshigJ0yEc4T1ibhjqp160c4SjsnPVK3HCrfVhMo";
const std::string js = "Accept: application/json";
//handleData (sequence of bytes, , how many elements will it return,
size_t handleData(void* c, size_t s, size_t n, void* g)
{
    *static_cast<std::string*>(g) += static_cast<char*>(c);return s * n;
}
std::string getGifInfo(std::string s)
{std::string j;    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 +
    "/v1/gifs/search?api_key=dc6zaT0xFJmzC&q=" + s;
    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,&j); curl_easy_perform(hnd);
    j.erase(j.find_last_of("}")+1);return j;
}
```

- goal: get the json info from Giphy API
  - expect string parameter with keyword
  - use the CURL library to retrieve the data
  - return the json data

# Implementation extractGifInfo.cpp

```
#include <iostream>
std::string extractGifInfo(std::string g)
{
    std::cout << g << std::endl;
    std::string t = g.substr(g.find("original"),100);
    size_t p1 = t.find("http");
    std::string r = t.substr(p1);
    size_t p2 = r.find("\");
    std::string url = t.substr(p1,p2);
    return url;
}
```

- goal: extract from the json info about the gif
  - expect string parameter with data about gif in json format
  - use parse function from json api from github
  - return the url
    - \* later we need to extract the rest of the information

```
- "original": {
    "url": "http://media4.giphy.com/media/ivelTBasMg67S/giphy.gif",
    "width": "500",
    "height": "350",
    "size": "1004245",
    "frames": "31",
    "mp4": "http://media.giphy.com/media/ivelTBasMg67S/giphy.mp4"
},
```



# Implementation makeDisplayWindowDoc.cpp

---

```
#include "lab.h"
FL_Cairo_Window * dw;
FL_Box * g;

FL_Cairo_Window * makeDisplayWindow(){
    dw = new FL_Cairo_Window(500,500);
    dw->label("Display of Gif");
    dw->color(fl_rgb_color(74,189,172));
    g = new FL_Box(FL_FLAT_BOX, 0,0,500,500,"");
    return dw;}
```

- goal: Make a window that displays the gif
  - Make a window call "Display of Gif"
  - Makes a box for the gif to be put in

# Implementation cbDisplay.cpp

---

```
#include "lab.h"
void cbDisplay(Fl_Button*,void*)
{
    //std::cout << r->value() << std::endl;
    std::string s = r->value();
    std::string j = getGifInfo(s);
    std::string url = extractGifInfo(j);
    std::string cmd = "curl " + url + " > 200.gif";
    system(cmd.c_str());//c_str() extracts a C-String from C++ String
    g->image(new Fl_GIF_Image("200.gif"));
    dw -> redraw();
}
```

- goal: Get the Gif and display it in window
  - Get keyword from search and save in s
  - Put s in getGifInfo and save result in j
  - Put j in extractGifInfo and save result in url
  - Add "curl" and "j 200.gif" and saves it to cmd
  - Turn cmd from c++ string to c string
  - Inputs image 200.gif into box g
  - Redraws Display Gif window

# Implementation main.cpp

---

```
#include "lab.h"
//void killAnimate();

int main()
{
    makeSearchWindow() -> show();
    makeDisplayWindow() -> show();
    Fl::run();
    //system("animate 200.gif &"); // & means run in background
    //killAnimate();
}
```

- Retrieve the gif file
- Display it

# Test

- The user can search whatever they want
  - If the user searches for "cats" they will get a cat related gif
  - if the user searches for "gravity+falls" you will get another related gif

The user cannot type in spaces, but can use "+" as an alternative

