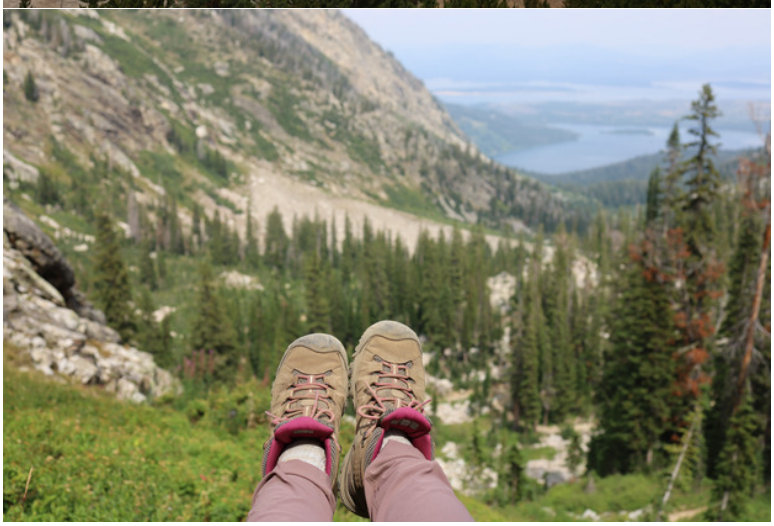
CS116 Ch10p06-GUI

1 Lab Description

In this lab, we need to design a photo effect program that allows users to select a photo and a photo effect. After submitting the selections, the next page will show the corresponding photos. There are a base class Effect and its three derived classes: Sunset, Grayscale and Invert. The three derived classes have inherited the function update_pixel from Effect and overridden the function process. The lab is meant to practice the polymorphism concepts in chapter 10. In order to allow the derived class to access the base class data member, I use "protected" insead of "private." To access the objects from different classes in a class hierarchy, I set up a vector of pointers to store the address of each object and create virtual functions to further access the overriden member function in derived classes. I reused the code of parsing query string from previous lab and make it a helper function in the photoEffect.cpp file. I also reimplemented the code from imagemod.cpp in the bigc3code to open and save the bmp files. In the first version, I did not use the "dirent" structure and directly outputed the image to html. The way I did in the previous version is static. However, in the real world situations, the dynamically creating html is much more prefered. To acheive that, I created another file to generate the html code and modified the code from chapter 11 worked example 1 To enhence the user experience of the program, I also added a back botton. Last but not the lease, I chose eight photos I took in Yellowstone National Park this summer in te program. Since the original size is too big to process, I resized them to the 10% of the origianl.

2 The Images in Photo Gallery



3 The Showcase of The Program

The Web Page of the Photo Program

Photo Effect Program

This program has selected three signature photo effects: Sunset, Grayscale, Invert

Photo Gallery


bison


elk


hikingboots


hollylake


mudgeyser


prism


sapphire


thermogeyser

Please select one photo in the gallery:

- ○ bison
- ◉ elk
- ○ hikingboots
- ○ hollylake
- ○ mudgeyser
- ○ prism
- ○ sapphire
- ○ thermogeyser

Please select your preferred photo effect:

○ Sunset ◉ Grayscale ○ Invert

Submit

The Next Page After Applying Effect

Original Photo

Photo With grayscale Effect

Back

I also made a video demostrating how the program works.

4 Source Code of Effect.h

```
1  #ifndef EFFECT_H
2  #define EFFECT_H
3
4  #include <fstream>
5  #include <vector>
6  using namespace std;
7
8  class Effect
9  {
10 public:
11     //constructor
12     Effect();
13     void update_pixel(string file_name);
14     virtual void process(int &red, int &green, int &blue);
15     //to access the protected member
16     string get_effect_name();
17
18 protected:
19     string effect_name;
20 };
21 /**
22    Gets an integer from a binary in_stream.
23    @param in_stream the in_stream
24    @param offset the offset at which to read the integer
25    @return the integer starting at the given offset
26 */
22 int get_int(fstream &stream, int offset);
23
24 #endif
```

5 Source Code of Effect.cpp

```cpp
#include "Effect.h"
#include <iostream>

using namespace std;

Effect::Effect()
{
    effect_name = "no_effect";
}
void Effect::update_pixel(string selected_file)
{
    fstream in_stream;
    string file_name = "original_photo/" + selected_file + ".bmp";
    // Open as a binary file
    in_stream.open(file_name, ios::in | ios::binary);

    fstream out_stream;
    out_stream.open("output_photo/" + selected_file + "_" + effect_name + ".bmp", ios::out
| ios::binary);

    int file_size = get_int(in_stream, 2); // Get the image dimensions
    int start = get_int(in_stream, 10);
    int width = get_int(in_stream, 18);
    int height = get_int(in_stream, 22);

    // Scan lines must occupy multiples of four bytes
    int scanline_size = width * 3;
    int padding = 0;
    if (scanline_size % 4 != 0)
    {
        padding = 4 - scanline_size % 4;
    }

    if (file_size != start + (scanline_size + padding) * height)
    {
        cout << "Not␣a␣24-bit␣true␣color␣image␣file." << endl;
        return;
    }

    in_stream.seekg(0);
    out_stream.seekp(0);
    for (int i = 0; i < start; i++)
    {
        out_stream.put(in_stream.get());
    }

    int pos = start;
    for (int i = 0; i < height; i++) // For each scan line
    {
        for (int j = 0; j < width; j++) // For each pixel
        {
            in_stream.seekg(pos); // Go to the next pixel
```

```
            int blue = in_stream.get();
            int green = in_stream.get();
            int red = in_stream.get(); // Read the pixel

            // Process the pixel
            process(red, green, blue);

            out_stream.seekp(pos); // Go back to the start of the pixel

            out_stream.put(blue); // Write the pixel
            out_stream.put(green);
            out_stream.put(red);
            pos = pos + 3;
        }

        in_stream.seekg(padding, ios::cur); // Skip the padding
        pos = pos + padding;
    }

    in_stream.close();
    out_stream.close();
}

void Effect::process(int &red, int &green, int &blue)
{
}
string Effect::get_effect_name()
{
    return effect_name;
}

int get_int(fstream &in_stream, int offset)
{
    in_stream.seekg(offset);
    int result = 0;
    int base = 1;
    for (int i = 0; i < 4; i++)
    {
        result = result + in_stream.get() * base;
        base = base * 256;
    }
    return result;
}
```

6 Source Code of Sunset.h

```
#ifndef SUNSET_H
#define SUNSET_H
#include "Effect.h"

using namespace std;

class Sunset : public Effect
{
public:
    //constructor
    Sunset();
    void process(int &red, int& green, int& blue);
private:

};


#endif
```

7 Source Code of Sunset.cpp

```
#include "Sunset.h"
using namespace std;

Sunset::Sunset(){
    effect_name = "sunset";
}
void Sunset::process(int &red, int& green, int& blue)
{
    red = red;
    green = 0.7*green;
    blue = 0.7*blue;
}
```

8 Source Code of Grayscale.h

```h
#ifndef GRAYSCALE_H
#define GRAYSCALE_H
#include "Effect.h"

using namespace std;

class Grayscale : public Effect
{
public:
    //constructor
    Grayscale();
    void process(int &red, int& green, int& blue);

private:

};


#endif
```

9 Source Code of Grayscale.cpp

```cpp
#include "Grayscale.h"

Grayscale::Grayscale(){
    effect_name = "grayscale";
}
void Grayscale::process(int &red, int& green, int& blue)
{
    int grayness_level = 0.3*red + 0.59*green + 0.11* blue;
    red = grayness_level;
    green = grayness_level;
    blue = grayness_level;
}
```

10 Source Code of Invert.h

```
#ifndef INVERT_H
#define INVERT_H
#include "Effect.h"

using namespace std;

class Invert : public Effect
{
public:
    //constructor
    Invert();
    void process(int &red, int& green, int& blue);
private:

};


#endif
```

11 Source Code of Invert.cpp

```
#include "Invert.h"

Invert::Invert()
{
    effect_name = "invert";
}
void Invert::process(int &red, int &green, int &blue)
{
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
}
```

12 Source Code of direntry.h

```
#ifndef DIRENTRY_H
#define DIRENTRY_H

#include <string>
#include <vector>

using namespace std;

class DirectoryEntry
{
public:
   /**
      Constructs a directory entry representing a directory.
   */
   DirectoryEntry(string directory_name);
   /**
      Yields the children of this directory.
   */
   vector<DirectoryEntry> children() const;
   /**
      Checks whether this entry represents a directory.
   */
   bool is_directory() const;
   /**
      Yields the name of this entry.
   */
   string name() const;
   /**
      Yields the extension of this entry, or "" if this entry
      has no extension.
   */
   string extension() const;
private:
   DirectoryEntry();
   DirectoryEntry child(string name, int childtype) const;

   string entryname;
   int type;
};

#endif
```

13 Source Code of htmlGenerator.cpp

```cpp
#include <iostream>
#include <fstream>
#include "direntry.h"

using namespace std;

void find(DirectoryEntry directory, string extension, vector<DirectoryEntry> &results)
{

    vector<DirectoryEntry> entries = directory.children();
    for (int i = 0; i < entries.size(); i++)
    {
        DirectoryEntry entry = entries[i];
        if (entry.extension() == extension)
        {
            results.push_back(entry);
        }
    }
}

int main()
{
    ofstream output_file("photoEffect1.html", ofstream::out);
    string current_dir_name = "original_photo";

    output_file << "<html><head><title>Photo␣Effect␣Program</title></head>";
    output_file << "<body>␣\
␣␣␣␣<h1>Photo␣Effect␣Program</h1>␣\
␣␣␣␣<h3>This␣program␣has␣selected␣three␣signature␣photo␣effects:␣Sunset,␣Grayscale,␣Invert␣<br></h3>
␣␣␣␣<h1>Photo␣Gallery</h1>";

    DirectoryEntry current_dir(current_dir_name);

    vector<DirectoryEntry> bmp_entries;
    find(current_dir, "bmp", bmp_entries);
    vector<string> file_names;

    for (int i = 0; i < bmp_entries.size(); i++)
    {
        string entry_name = bmp_entries[i].name();
        string file_name = entry_name.substr(
            current_dir_name.length() + 1,
            entry_name.length() - current_dir_name.length() - 5);
        file_names.push_back(file_name);

        output_file << "<figure>\n";
        output_file << "<img␣width=\"20%\"␣src=\""
                    << entry_name << "\"␣alt=\"Photo"
                    << i << "\"␣/>";
        output_file << "<figcaption>" << file_name << "</figcaption>"
                    << "</figure>";
    }
```

```
50
51      output_file << "<form␣action=\"cgi-bin/photoEffect.cgi\">"
52                  << "<h2>Please␣select␣one␣photo␣in␣the␣gallery:</h2>";
53
54      for (int i = 0; i < file_names.size(); i++)
55      {
56          output_file << "<div>";
57          output_file
58              << "<input␣type=\"radio\"␣name=\"picture\"␣"
59              << "id=\"" << file_names[i] << "\"␣"
60              << "value=\"" << file_names[i] << "\"␣>"
61              << "\n"
62              << "<label␣for=\"" << file_names[i] << "\">"
63              << file_names[i] << "</label>";
64
65          output_file << "</div>";
66      }
67      output_file << "<p>Please␣select␣your␣preferred␣photo␣effect:</p>";
68      output_file << "<div>";
69      output_file << "<input␣type=\"radio\"␣id=\"Sunset\"␣name=\"effect\"␣value=\"0\">␣\n␣\
70  ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣<label␣for=\"Sunset\">Sunset</label>␣\n"
70                  << "<input␣type=\"radio\"␣id=\"Grayscale\"␣name=\"effect\"␣value=\"1\">␣\n␣\
71  ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣<label␣for=\"Grayscale\">Grayscale</label>"
71                  << "<input␣type=\"radio\"␣id=\"Invert\"␣name=\"effect\"␣value=\"2\">␣\n␣\
72  ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣<label␣for=\"Invert\">Invert</label>␣\
73  ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣</div>";
72      output_file << "<div>␣\n␣\
73  ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣<button␣type=\"submit\">Submit</button>␣\n␣\
74  ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣</div>";
73
74      output_file << "</form></body></html>";
75      output_file.close();
76      return 0;
77  }
```

14 Source Code of photoEffect.html

```
<html>

<head>
    <title>Photo Effect Program</title>
</head>

<body>
    <h1>Photo Effect Program</h1>
    <h3>This program has selected three signature photo effects: Sunset, Grayscale, In-
vert <br></h3>
    <h1>Photo Gallery</h1>
    <figure>
        <img width="20%" src="original_photo/bison.bmp" alt="Photo0" />
        <figcaption>bison</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/elk.bmp" alt="Photo1" />
        <figcaption>elk</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/hikingboots.bmp" alt="Photo2" />
        <figcaption>hikingboots</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/hollylake.bmp" alt="Photo3" />
        <figcaption>hollylake</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/mudgeyser.bmp" alt="Photo4" />
        <figcaption>mudgeyser</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/prism.bmp" alt="Photo5" />
        <figcaption>prism</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/sapphire.bmp" alt="Photo6" />
        <figcaption>sapphire</figcaption>
    </figure>
    <figure>
        <img width="20%" src="original_photo/thermogeyser.bmp" alt="Photo7" />
        <figcaption>thermogeyser</figcaption>
    </figure>
    <form action="cgi-bin/photoEffect.cgi">
        <h2>Please select one photo in the gallery:</h2>
        <div><input type="radio" name="picture" id="bison" value="bison">
            <label for="bison">bison</label>
        </div>
        <div><input type="radio" name="picture" id="elk" value="elk">
            <label for="elk">elk</label>
        </div>
        <div><input type="radio" name="picture" id="hikingboots" value="hikingboots">
```

```
52              <label for="hikingboots">hikingboots</label>
53          </div>
54          <div><input type="radio" name="picture" id="hollylake" value="hollylake">
55              <label for="hollylake">hollylake</label>
56          </div>
57          <div><input type="radio" name="picture" id="mudgeyser" value="mudgeyser">
58              <label for="mudgeyser">mudgeyser</label>
59          </div>
60          <div><input type="radio" name="picture" id="prism" value="prism">
61              <label for="prism">prism</label>
62          </div>
63          <div><input type="radio" name="picture" id="sapphire" value="sapphire">
64              <label for="sapphire">sapphire</label>
65          </div>
66          <div><input type="radio" name="picture" id="thermogeyser" value="thermogeyser">
67              <label for="thermogeyser">thermogeyser</label>
68          </div>
69          <p>Please select your preferred photo effect:</p>
70          <div><input type="radio" id="Sunset" name="effect" value="0">
71              <label for="Sunset">Sunset</label>
72              <input type="radio" id="Grayscale" name="effect" value="1">
73              <label for="Grayscale">Grayscale</label><input type="radio" id="Invert" name="effect"
value="2">
74              <label for="Invert">Invert</label>
75          </div>
76          <div>
77              <button type="submit">Submit</button>
78          </div>
79      </form>
80  </body>
81
82  </html>
```

15 Source Code of photoEffect.cpp

```
#include <fstream>
#include <string>
#include <iostream>
#include "Effect.h"
#include "Sunset.h"
#include "Grayscale.h"
#include "Invert.h"
#include "dirent.h"

using namespace std;

//To parse the query string from html
void parse_query_string(string s, vector<string> &parsed_answers)
{
    //parse the sentence from html
    //i.e. Q1=hi&Q2=Denmark
    string delimiter1 = "&";
    string delimiter2 = "=";

    int position1 = 0;
    int position2 = 0;
    string token;
    string yourAnswer;
    //parse the sentence into token: i.e q1=hi
    int counter = 0;
    while ((position1 = s.find(delimiter1)) != string::npos)
    {
        token = s.substr(0, position1);
        //parse the token into answer: i.e hi
        position2 = s.find(delimiter2);

        yourAnswer = token.substr(position2 + 1, position1 - (position2 + 1));
        parsed_answers.push_back(yourAnswer);

        s.erase(0, position1 + delimiter1.length());
    }

    position2 = s.find(delimiter2);
    yourAnswer = s.substr(position2 + 1, position1 - (position2 + 1));
    parsed_answers.push_back(yourAnswer);
}

int main()
{
    cout << "Content-type:␣text/html\n\r\n\r";

    //to get answers from user input in the cgi progrm
    string s = getenv("QUERY_STRING");
    vector<string> answers;
    parse_query_string(s, answers);
    //to store address of the Effect class

```

```
53      vector<Effect *> effects;
54      effects.push_back(new Sunset());
55      effects.push_back(new Grayscale());
56      effects.push_back(new Invert());
57      //answer[i] is string type and we need to convert the string to int
58      //so we can use it as the index of effects
59      effects[stoi(answers[1])]->update_pixel(answers[0]);
60      string effectName = effects[stoi(answers[1])] -> get_effect_name();
61      string fileWithEffect = answers[0] + "_" + effectName + ".bmp";
62
63      cout << "<hl>Original␣Photo␣<br></hl>";
64      cout << "<img␣width=\"50%\"␣src=\"../original_photo/"
65              + answers[0] + ".bmp\"␣alt=\"photo␣1\"␣/><br>";
66      cout << "<hl>Photo␣With␣" << effectName << "␣Effect␣<br></hl>";
67      cout << "<img␣width=\"50%\"␣src=\"../output_photo/"
68              + fileWithEffect + "\"␣alt=\"photo␣1\"␣/>";
69
70      // A button to go back to the previous page
71      cout << "<div><button␣onclick=\"history.back()\">Back<button></div>";
72
73      return 0;
74  }
```