

# 최종 프로젝트 보고서

팀명:교수님학점잘조

20202106 OSHIMA ASUKA

## 1. 어플리케이션의 목표

Libpng 라이브러리를 활용해 이미지를 다양하게 처리하고자 했다.

## 2. 어플리케이션의 주기능 및 동작 튜토리얼

다양한 그래프나 모양을 그리는 코드를 작성하여 png 파일로 출력하는 어플리케이션이다.  
또한 입력 받은 png 파일을 뒤집은 어플리케이션을 만들었다.

## 3. 구현 세부사항

공지에서 a.그래프, b.모양, c. 텍스트를 출력하는 기능과 d. 이미지를 뒤집기 의 4 개의 기능을 만들었다. 이번 프로젝트에서는 배포된 lines.c: 랜덤한 줄을 출력하는 파일을 참고해서 구현했다. a,b,c 는 같은 파일의 작성을 했으며,  $256 \times 256$  의 화소를 가진 이미지에 그림을 그린다. 배경은 검은색으로 초기화되어 있다.

### a.그래프 출력하기

그림 1 은  $y=x*x/256$  의 그래프를 출력하는 코드다. 행렬의 Colum 를 x 변수로 row 를 y 변수로 생각한다. for 문을 이용하여 x 의 값을 결정해 고정한 다음에 y 를 움직여서 선을 그린다. 그림 2 는 출력 결과다. 그림 2 부터 2 차 함수의 그래프가 출력된 것을 확인할 수 있다. 또한 그림 1 의 77 째 줄을 다른 함수로 바꿔 쓰면 다른 함수를 그릴 수 있다.

```
70
71 void draw_function(Pixel **row_pointers)
72 {
73     for (int col = 0; col < WIDTH; ++col)
74     {
75         for (int row = 0; row < HEIGHT; ++row)
76         {
77             int y = col * col / 256;
78             /* draw a graph */
79             Pixel *pixel_ptr = &row_pointers[row][col];
80             if (HEIGHT - row <= y)
81             {
82                 pixel_ptr->r = 0; // red
83                 pixel_ptr->g = 255; // green
84                 pixel_ptr->b = 0; // blue
85                 pixel_ptr->a = 255; // alpha (opacity)
86             }
87         }
88     }
89 }
90
```

그림 1 그래프 코드

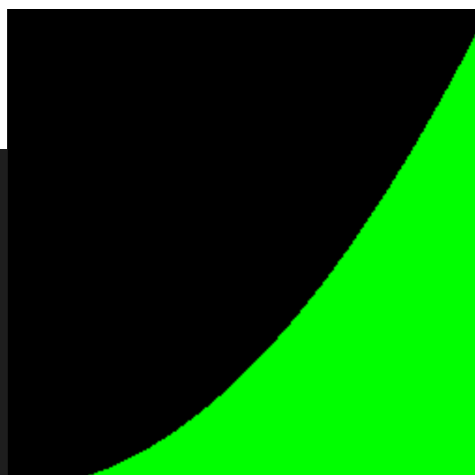


그림 2 그래프 출력결과

### b.모양 출력하기

모양을 출력하는 코드를 작성했고 모양은 사각형, 원, 삼각형의 3 개를 출력하는 코드를 작성해 보았다. 그림 3 은 사각형을 그리는 코드다. 먼저 삼각형의 좌측 상단 좌표를 정하고, 길이와 높이를 정한다. 그리고 사각형 안에 점을 찍어간다. 만약 삼각형이  $256 \times 256$  의 사이즈밖으로 나올 때는 에러문이 나온다. 그림 4 는 입력 값과 그 출력결과다. 좌측 상단 좌표 (50,125)를 기준으로 사각형을 그렸다.

```

91 void draw_rectangle(Pixel **row_pointers)
92 {
93     int length, height;
94     int init_x, init_y;
95
96     printf("Determine left upper coordinate of rectangle (x, y) : ");
97     scanf("%d %d", &init_x, &init_y);
98     if (init_x > WIDTH || init_y >= HEIGHT)
99     {
100         printf("Left upper coordinate of rectangle is out of image!\n");
101         return;
102     }
103     printf("Length and height of rectangle : ");
104     scanf("%d %d", &length, &height);
105     if (length + init_x >= WIDTH || height + init_y >= HEIGHT)
106     {
107         printf("Some part of rectangle is out of image!\n");
108         return;
109     }
110     for (int col = 0; col < WIDTH; ++col)
111     {
112         for (int row = 0; row < HEIGHT; ++row)
113         {
114             /* draw a rectangle */
115             Pixel *pixel_ptr = &row_pointers[row][col];
116             if (init_y < row && row < init_y + height && init_x < col && col < init_x + length)
117             {
118                 pixel_ptr->r = 0; // red
119                 pixel_ptr->g = 255; // green
120                 pixel_ptr->b = 0; // blue
121                 pixel_ptr->a = 255; // alpha (opacity)
122             }
123         }
124     }
125 }
126

```

그림 3 사각형 코드

```

Determine left upper coordinate of rectangle (x, y) : 50 125
Length and height of rectangle : 70 110

```

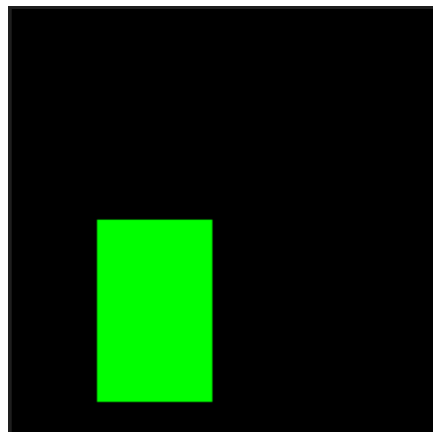


그림 4 사각형: 입력과 출력결과

다음은 원을 그리는 함수에 대해 설명한다. 그림 5 는 원을 그리는 코드다. 원의 중심의 좌표와 반지름의 입력을 받아 원의 중심과 임의의 점(x,y)가 원안에 존재하는 경우에는 점을 색깔로 찍도록 코드를 작성했다. 중심의 좌표(x1,y1)=(70,125),반지름은 70 라 했을 때 출력결과는 그림 6 와 같다.

```

128 void draw_circle(Pixel **row_pointers)
129 {
130     int dx, dy;
131     int x1, y1, radius;
132
133     printf("center of the circle(x1 y1): ");
134     scanf("%d %d", &x1, &y1);
135     if (x1 > WIDTH || y1 > HEIGHT)
136     {
137         printf("Center coordinate of the circle is out of image!\n");
138         return;
139     }
140     printf("radius: ");
141     scanf("%d", &radius);
142     if (x1 + radius > WIDTH || x1 - radius < 0 || y1 + radius > HEIGHT || y1 - radius < 0)
143     {
144         printf("Some part of the circle is out of image!\n");
145         return;
146     }
147
148     for (int y = 0; y < HEIGHT; y++)
149     {
150         for (int x = 0; x < WIDTH; x++)
151         {
152             dx = (int)x - (int)x1;
153             dy = (int)y - (int)y1;
154             Pixel *pixel_ptr = &row_pointers[y][x];
155             /* draw a circle */
156             if ((dx * dx) + (dy * dy) <= radius * radius)
157             {
158                 pixel_ptr->r = 0; // red
159                 pixel_ptr->g = 255; // green
160                 pixel_ptr->b = 0; // blue
161                 pixel_ptr->a = 255; // alpha (opacity)
162             }
163         }
164     }
165 }

```

그림 5 원그리기 코드

```

center of the circle(x1 y1): 70 125
radius: 70

```

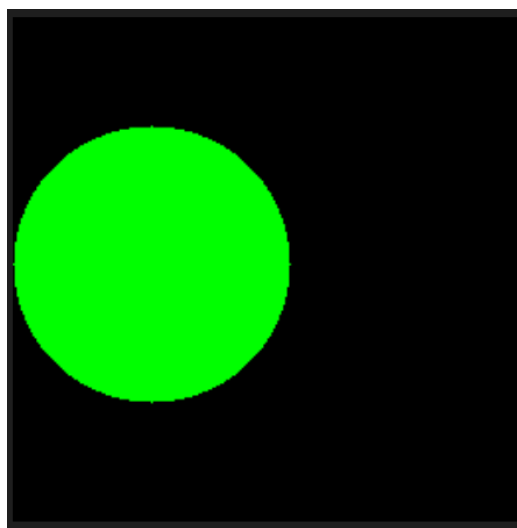


그림 6 원: 입력과 출력 결과

마지막으로 삼각형에 대해다. 그림 7 은 삼각형을 그리는 코드다. 사각형을 출력하는 코드와 동일하게 좌측 상단 좌표를 정하고, 길과 높이를 정하고 직각삼각형을 그린다. 지금까지 살펴보았던 사각형과 원과 동일하게 삼각형안에 존재하는 점을 찍었다. 그림 8 은 입력과 출력 결과이고 결과가 직각삼각형이 되어 있는 것을 확인할 수 있다.

```

166
167 void draw_triangle(Pixel **row_pointers)
168 {
169     float length, height;
170     float init_x, init_y;
171
172     printf("Determine left upper coordinate of triangle (x, y) : ");
173     scanf("%f %f", &init_x, &init_y);
174     if (init_x > WIDTH || init_y > HEIGHT)
175     {
176         printf("Left upper coordinate of triangle is out of image!\n");
177         return;
178     }
179     printf("Length and height of triangle : ");
180     scanf("%f %f", &length, &height);
181     if (length + init_x > WIDTH || height + init_y > HEIGHT)
182     {
183         printf("Some part of triangle is out of image!\n");
184         return;
185     }
186
187     for (int y = 0; y < HEIGHT; y++)
188     {
189         for (int x = 0; x < WIDTH; x++)
190         {
191             float rate = length / height;
192             Pixel *pixel_ptr = &row_pointers[y][x];
193             /* draw a triangle */
194             if (init_y <= y && y < init_y + height && init_x <= x && x < init_x + (rate * (y - init_y)))
195             {
196                 pixel_ptr->r = 0; // red
197                 pixel_ptr->g = 255; // green
198                 pixel_ptr->b = 0; // blue
199                 pixel_ptr->a = 255; // alpha (opacity)
200             }
201         }
202     }
203 }

```

그림 7 삼각형 코드

```

Determine left upper coordinate of triangle (x, y) : 125 20
Length and height of triangle : 100 120

```

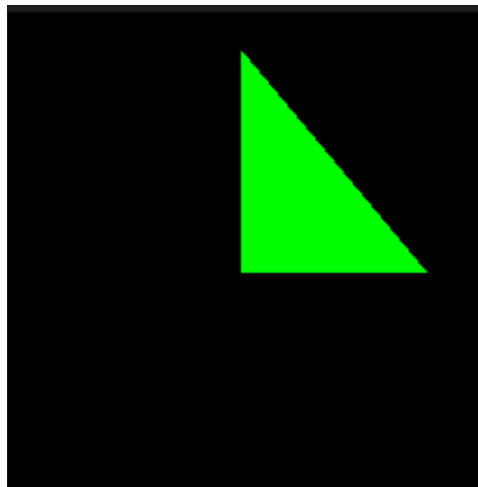


그림 8 삼각형: 입력과 출력결과

### c. 텍스트를 출력하기

b. 모양 출력하기에서는 함수조건을 충족하는 점을 색깔로 찍어서 모양을 출력했다. 텍스트 출력하기에서도 조건을 충족하는 점을 찍으면 된다. 따라서 먼저 각 텍스트 입력에 대한 조건을 만들기 위해 입력한 텍스트와 맞는 텍스트를 표현하는 헤더파일(Fon.h)을 만들었다.(그림 9) 이 파일은 font\_array 배열 0~25 에 소문자 알파벳을 표현하는 Font 배열을 저장했다. font\_array[0]을 보면 요소 1 로 'a'라는 소문자가 표현되어 있는 것을 알 수 있다. 6 째 열이 0 이 되어 있는 것은 텍스트를 복수 출력할 때 텍스트와 텍스트가 붙어서 잘

안보이게 되는 것을 방지하기 위해서다. 이 헤더 파일을 이용해서 font\_array 가 1 일 때 점을 찍으면 원하는 글씨를 출력할 수 있다. 따라서 입력과 Font.h 의 배열을 연관시켜야 한다. 그림 10, 51 째 줄의 draw\_string 함수에서는 입력 텍스트를 받아 그 입력 텍스트 부터 font\_array 배열의 대응하는 index(n)를 구한다. 37 째 줄의 draw\_char 함수와 19 째 draw\_pixel 함수에서는 font\_array 배열의 요소가 1 일 때만 점을 찍도록 코드를 구현했다. 'sogang'라 입력했을 때 출력 결과는 그림 11 이다.

```
1  #include <stdio.h>
2
3  #define FONT_WIDTH 6
4  #define FONT_HEIGHT 5
5
6  typedef int Font[5][6];
7
8  Font font_array[26] =
9  {
10     // a
11     {{1, 1, 1, 1, 0, 0},
12      {0, 0, 0, 0, 1, 0},
13      {0, 1, 1, 1, 1, 0},
14      {1, 0, 0, 0, 1, 0},
15      {1, 1, 1, 1, 1, 0}},
16     // b
17     {{1, 0, 0, 0, 0, 0},
18      {1, 0, 0, 0, 0, 0},
19      {1, 1, 1, 1, 1, 0},
20      {1, 0, 0, 0, 1, 0},
21      {1, 1, 1, 1, 0, 0}},
22     // c
23     {{0, 1, 1, 1, 1, 0},
24      {1, 0, 0, 0, 0, 0},
25      {1, 0, 0, 0, 0, 0},
26      {1, 0, 0, 0, 0, 0},
27      {0, 1, 1, 1, 1, 0}},
```

그림 9 Font.h

```

18
19 void draw_pixel(Pixel **row_pointers, int x, int y, int width, int height)
20 {
21     for (int col = 0; col < width; ++col)
22     {
23         for (int row = 0; row < height; ++row)
24         {
25             Pixel *pixel_ptr = &row_pointers[row][col];
26             if (row == y && col == x)
27             {
28                 pixel_ptr->r = 0; // red
29                 pixel_ptr->g = 255; // green
30                 pixel_ptr->b = 0; // blue
31                 pixel_ptr->a = 255; // alpha (opacity)
32             }
33         }
34     }
35 }
36
37 void draw_char(Pixel **image, int cursor, int font_arr, char *str)
38 {
39     /* draw font */
40     int image_length = strlen(str) * FONT_WIDTH;
41     for (int x1 = 0; x1 < FONT_WIDTH; ++x1)
42     {
43         for (int y1 = 0; y1 < FONT_HEIGHT; ++y1)
44         {
45             if (font_array[font_arr][y1][x1] == 1)
46                 draw_pixel(image, cursor + x1, y1, image_length, FONT_HEIGHT);
47         }
48     }
49 }
50
51 void draw_string(Pixel **row_pointers)
52 {
53     /* scan string */
54     char str[10];
55     printf("Input string : ");
56     scanf("%s", str);
57
58     /* draw characters */
59     for (int i = 0; i < strlen(str); ++i)
60     {
61         int n = (int)str[i] - 'a';
62         draw_char(row_pointers, i * FONT_WIDTH, n, str);
63     }
64 }
65

```

그림 10 텍스트: 코드



그림 11 출력결과

#### d. 이미지를 뒤집기

마지막으로 이미지를 뒤집은 코드에 대해 설명한다. 그림 12은 사진을 좌우로 뒤집은 코드이고 화소의 요소를 좌우로 바꾸고 있다. 그림 13은 입력했던 사진과 그 사진을 좌우로 뒤집은 결과 사진이다. 이와 같은 방법을 이용해서 위아래로 뒤집은 코드도 작성했다. (그림 14) 그림 11의 사진을 입력사진으로 한 결과 사진은 그림 15이다. 여기서 사진을 읽고 출력하는 코드는 Final Project Starter Kit 사이트를 참고하였다.

<sup>1</sup> [A simple libpng example program \(zarb.org\)](http://zarb.org)

```

125
126 /* 사진 좌우 뒤집기 */
127 for (int y = 0; y < height; ++y)
128 {
129     for (int x = 0; x < width / 2; ++x)
130     {
131         Pixel tmp = row_pointers[y][width - 1 - x];
132         row_pointers[y][width - 1 - x] = row_pointers[y][x];
133         row_pointers[y][x] = tmp;
134     }
135 }

```

그림 12 이미지 좌우 뒤집기 코드

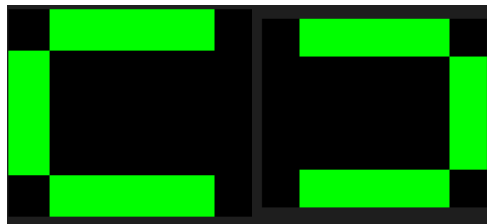


그림 13 입력 이미지/ 출력 이미지

```

125
126 /* 사진 위아래 뒤집기 */
127 for (int y = 0; y < HEIGHT / 2; ++y)
128 {
129     Pixel *tmp = row_pointers[HEIGHT - y - 1];
130     row_pointers[HEIGHT - y - 1] = row_pointers[y];
131     row_pointers[y] = tmp;
132 }
133

```

그림 14 이미지 위아래 뒤집기



그림 15

#### 4. 작동을 위한 설치 튜토리얼<sup>2</sup>

1. 먼저 [WindowsPowerShell] 를 관리자 권한으로 실행한다. (그림 16)

<sup>2</sup> Windows10 에 리눅스 설치하기(WSL2)

<https://blog.naver.com/goldrushing/222175975517>

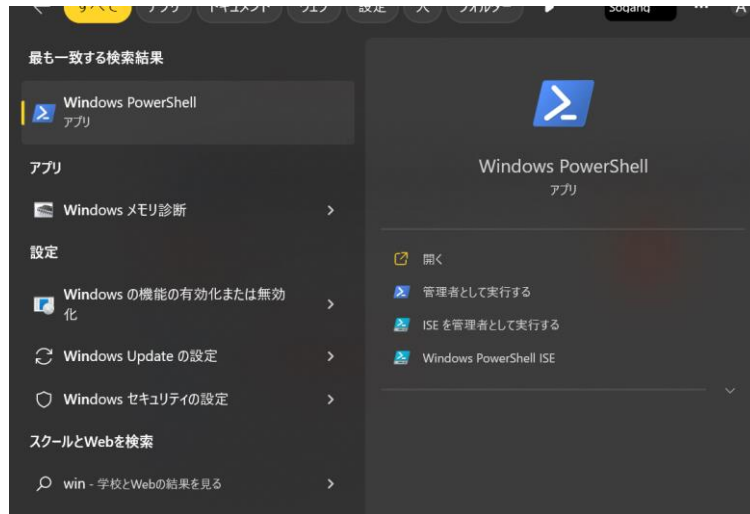


그림 16

2.다음은 아래와 같은 명령어를 실행시킨다.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

3. 다음 명령을 실행해서 가상머신 플랫폼 기능을 활성화 시킨고 노트북을 재시작한다.

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

4. wsl 는 사용 가능이 됐지만 최신 버전 다운로드해야 한다. 업데이트 패키지를 다운로드하여 실행해 주어야 한다.

( [https://learn.microsoft.com/ko-kr/windows/wsl/install?WT.mc\\_id=DT-MVP-4039897#step-4---download-the-linux-kernel-update-package](https://learn.microsoft.com/ko-kr/windows/wsl/install?WT.mc_id=DT-MVP-4039897#step-4---download-the-linux-kernel-update-package))

5. package 를 설치한다.

6.wsl 를 관리자 모드를 실행시키고 아래의 명령어를 입력한다.

```
wsl --set-default-version 2
```

7. Ubuntu 를 다운로드 한다.

8. 배포된 코드를 Ubuntu 를 이용해서 실행시킨다.

## 5. 결론 및 논의



이번 프로젝트에서는 교수님이 공지에 올려주셨던 예시를 만들었다. 하지만 많은 학우분들은 스스로 하고 싶은 주제를 찾고 프로젝트를 만들고 있었다. 앞으로 프로그램을 만들기 위해서는 주어진 과제를 하는 뿐만 아니라 스스로 주제를 찾는 필요가 있다고 생각했다. 또한 테스트 출력하기에서는 지금까지 배웠던 함수들만 이용했는데 처음에 텍스트를 배열로 표현하는 아이디어가 생각나지 않아서 완성할 때까지 시간이 오래 걸렸다. 이러한 아이디어를 키우기 위해서 많이 코딩을 해야겠다고 생각했다.