

C프로그래밍 실습

아스키 아트 생성 프로그램

진척 보고서 #1

제출일자:23/11/26

제출자명:김진강

제출자학번:233575

1. 프로젝트 목표

1) 배경 및 필요성

좁은 의미로서의 아스키 아트(ASCII Art) 란 아스키 코드 0x20~0x7e에 포함되는 문자, 기호를 사용한 그림을 말함. 요즘은 1바이트(8비트)의 문자 이외에, 2바이트(16비트)의 문자 같이 정확히 말하면 아스키 코드에 포함되지 않는 문자도 아스키아트라는 용어로 불리며, 이를 활용하여 더욱 다양한 아스키아트가 만들어지고 있음. 이러한 아스키코드는 사용자가 수작업으로 만드는데 어려움이 있음. 이러한 문제를 아스키아트를 자동으로 생성해주는 프로그램으로 해결할 수 있음.

2) 프로젝트 목표

사용자의 이미지를 사용자가 원하는 문자로 구성된 아스키아트로 만들어주는 프로그램.

3) 차별점

기존 프로그램은 정해진 문자로만 아스키아트를 만들어 줌. 이는 다양한 아스키아트의 수요에 맞추지 못할 수 있음. 우리는 아스키아트를 생성할 때 문자를 사용자가 직접 선택할 수 있게 하여 차별점이 있음.

2. 기능 계획

1) 사용자가 사용할 문자를 설정하는 기능

- 기본으로 설정된 문자 8개에서 사용자가 사용할 문자열을 변경할 수 있게 함.

(1) 사용할 문자는 8개로 배열에 명암이 오름차순으로 저장됨

2) 사용자의 이미지를 불러오는 기능

- 사용자에게 파일 이름을 입력 받아서 파일을 불러옴

- libjpeg 라이브러리를 활용하여 jpg이미지를 불러오고 처리한다.

(1) 파일 형식은 jpeg로 고정

3) 이미지를 아스키아트로 변환하는 기능

- 이미지를 흑백으로 변환하고 명암을 구한 후 그 명암에 맞는 글자를 출력함.

- libjpeg 라이브러리를 활용하여 jpg이미지를 불러오고 처리한다.

(1) 이미지의 RGB값을 흑백으로 변환함

(2) 명암에 따라 1)에서 저장한 문자를 선택하여 출력

3. 진척사항

1) 기능 구현

(1) 사용자의 이미지를 불러오는 기능

- "image.jpg"를 불러온다.(사용자가 입력한 파일 불러오는 기능은 후에 구현 예정)

- 1) fopen 함수를 사용하여 지정된 JPEG 파일을 이진 읽기 모드("rb")로 연다.(안 열리면 메시지 출력 후 종료)

2) jpg 구조체를 선언

3) 오류 처리기 설정. jpg파일에 문제가 있는 경우 에러를 출력하고 종료

4) jpeg_stdio_src 함수를 사용하여 JPEG 디코딩에 사용할 입력 파일을 설정

5) jpeg_read_header 함수를 호출하여 JPEG 파일의 헤더 정보를 읽음

6) jpeg_start_decompress 함수를 호출하여 이미지 디코딩을 시작

7) 이미지 데이터를 저장할 버퍼를 할당하고 (malloc 함수 사용), jpeg_read_scanlines 함수를 사용하여 행 단위로 이미지 데이터를 읽음.(이미지를 아래에서 위로 읽는다.)

- 적용된 배운 내용

파일 입출력(1), 반복문(7), 배열(7), 포인터(6),

- 코드 스크린샷

```

int main() {
    // JPEG 이미지 파일 경로
    const char* filename = "image.jpg";

    // JPEG 파일을 읽기 모드로 열기
    FILE* file = fopen(filename, "rb");
    if (!file) {
        printf("파일을 열 수 없습니다.");
        return 1;
    }

    // JPEG 구조체 선언
    struct jpeg_decompress_struct cinfo;
    struct jpeg_error_mgr jerr;

    // JPEG 오류 처리기 설정
    cinfo.err = jpeg_std_error(&jerr);

    // JPEG 구조체 초기화
    jpeg_create_decompress(&cinfo);

    // 입력 파일 지정
    jpeg_stdio_src(&cinfo, file);

    // 파일의 헤더 정보 읽기
    jpeg_read_header(&cinfo, TRUE);

    // 이미지 디코딩
    jpeg_start_decompress(&cinfo);

    // 이미지 데이터 저장을 위한 버퍼 생성
    unsigned char* image = (unsigned char*)malloc(cinfo.output_width * cinfo.output_height * cinfo.output_components);

    // 이미지 행 단위로 스캔하며 데이터 읽기
    while (cinfo.output_scanline < cinfo.output_height) {
        unsigned char* row_pointer = &image[(cinfo.output_height - cinfo.output_scanline - 1) * cinfo.output_width * cinfo.output_components];
        jpeg_read_scanlines(&cinfo, &row_pointer, 1);
    }
}

```

(2) 이미지를 아스키아트로 변환하는 기능

- 입력: 앞에서 값을 저장한 "image" 변수, 불러온 이미지의 가로 길이, 불러온 이미지의 세로 길이
- 2중 for 문에서 행은 뒤에서 앞으로 읽어온다.(libpeg는 이미지를 아래에서 위로 읽으므로) 픽셀을 하나씩 흑백으로 변환하고 명암을 구한 후 그 명암에 맞는 글자를 출력하고 행 for문이 한번 돌 때마다 줄바꿈함.
- 적용된 배운 내용

반복문, 함수, 포인터

```

// 이미지 출력
changeToAscii(image, cinfo.output_width, cinfo.output_height);

```

```
// 아스키 문자 세트
char ascii_chars[] = "@%#*+=-:. ";

// 이미지를 아스키 아트로 변환하는 함수
void changeToAscii(unsigned char* image, int width, int height) {
    int gray;
    int index;

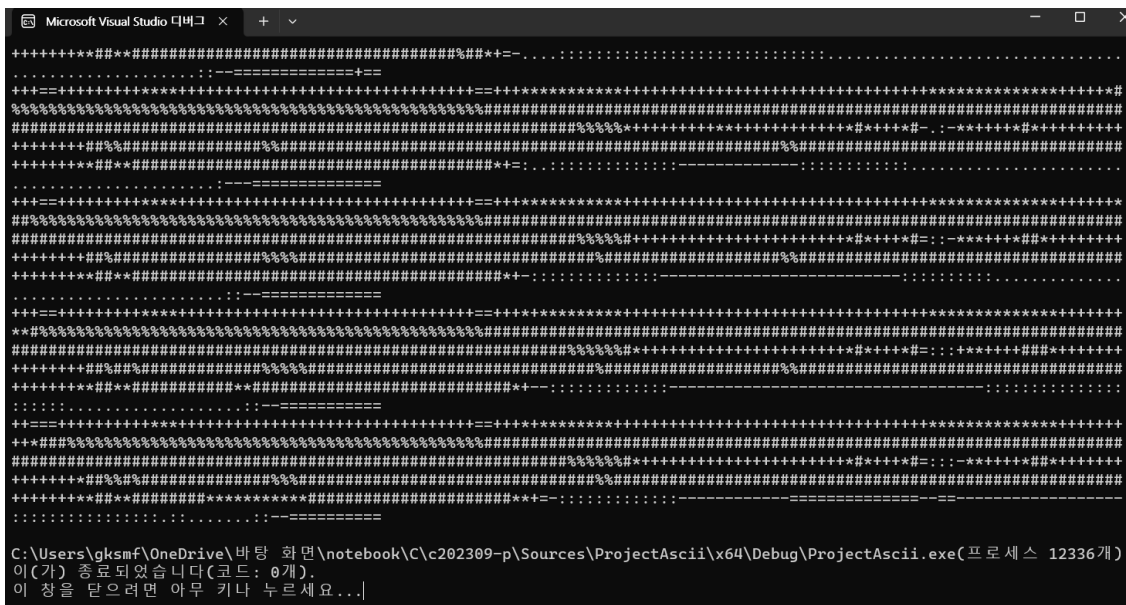
    for (int i = height; i > 0; i--) {
        for (int j = 0; j < width; j++) {
            index = (i * width + j) * 3;
            // RGB 값을 흑백으로 변환
            gray = (int)(0.2989 * image[index] + 0.5870 * image[index + 1] + 0.1140 * image[index + 2]);

            // 밝기에 따라 아스키 문자 선택하여 출력
            int char_index = (gray * (sizeof(ascii_chars) - 1)) / 255;
            printf("%c", ascii_chars[char_index]);
        }
        printf("\n");
    }
}
```

2) 테스트 결과

(1) 사용자의 이미지를 불러오는 기능과 이미지를 아스키아트로 변환하는 기능

- 1) "image.jpg"를 불러온다.(사용자가 입력한 파일 불러오는 기능은 후에 구현 예정)
- 2) 2중 for 문에서 행은 뒤에서 앞으로 읽어온다.(libpeg 라이브러리는 이미지를 아래에서 위로 읽으므로) 픽셀을 하나씩 흑백으로 변환하고 명암을 구한 후 그 명암에 맞는 글자를 출력하고 행 for문이 한번 돌때마다 줄 바꿈.
- 테스트 결과 스크린샷(이미지 크기는 640x480로 픽셀 하나에 한 문자를 할당해서 한 화면에 담기지 않음)



메모장에 따로 복사한 사진



4. 계획 대비 변경 사항

1) 픽셀 수 변경

- 이전: 한 픽셀에 하나씩 문자를 할당
- 이후: 여러 픽셀의 명암을 평균내서 하나의 문자를 할당
- 사유: 한눈에 보기 힘들고 글자수가 너무 많으면 활용도가 떨어짐

5. 프로젝트 일정

| 업무 | | 11/3 | 11/10 | 11/17 | 11/23 | 12/10 | 12 |
|--------------------------------------------|----------------------------------------------------------------------|------|-------|-------|--------|----------------------------|----|
| 제안서 작성 | | 완료 | | | | | |
| 1) 사용 자가 사 용할 문 자를 설 정하는 기능 | 세부기능1 | | | | -----> | | |
| 2) 사용 자의 이 미지를 불러오는 기능 | 세부기능1 | | | 완료 | | | |
| 2) 이미 지를 아 스키아트 로 변환 하는 기 능 | (1) 이미지의 RGB값을 흑 백으로 변환함 (2) 명암에 따라 1)에서 저장한 문자를 선택하여 출력 | | | | | 세부기 능 3) 완료하 면 완료 | |
| | 3) 여러 픽셀의 명암을 평균내서 하나의 문자를 할당 | | | | | -----> | |