**2023 Spring OOP Assignment Report**

과제 번호 : 3

학번 : 20220778

이름 : 표승현

Povis ID : hyeony312

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

프로그램을 하다 보면 결정해야 할 세부 사항이 많은데, 이러한 세부 사항을 처리한 방법과 이유를 보고서에 쓰십시오.

독창적인 아이디어와 추가 기능은 보너스 점수를 받을 수 있으므로, 보고서에 명확히 기재하십시오.

문제가 여러 개인 경우, 각 문제별로 정리해서 작성합니다.

아래 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

1. **프로그램 개요**

제시된 클래스들을 이용하여 이미지를 처리하는 프로그램이다. parser 클래스는 이미지 파일과 구성 파일을 읽어들이고, 결과를 파일로 출력하는 역할을 한다. artist 클래스는 이미지를 처리하는데 필요한 기본적인 기능을 제공하며, 이 클래스를 상속받아 각종 이미지 처리 알고리즘을 구현한 클래스들이 존재한다. drawer 클래스는 artist 클래스를 이용하여 이미지를 그리는 역할을 수행한다. 이 클래스를 상속받아 다운샘플링, 업샘플링, 이미지 크기 조절 등 다양한 이미지 처리 알고리즘을 구현한 클래스들이 존재한다. ASCII 문자를 이용하여 너비, 높이, 그리고 각각의 픽셀값을 차례로 저장한 그림 파일을 표현한다. 프로그램을 실행할 때 input, config, output 파일의 이름이나 경로를 입력하면 자동으로 ASCII art를 구현한다.

프로그램은 main.cpp와 parser파일(parser.h&parser.cpp), artist파일(artist.h&artist.cpp), drawer파일(drawer.h& drawer.cpp)으로 구성된다. 각각의 파일은 parser class, artist class, drawer class의 정의 및 관련 methods를 포함한다. 각 class에 대한 자세한 설명은 2. 프로그램의 구조 및 알고리즘에서 다룬다.

1. **프로그램의 구조 및 알고리즘**

parser class

너비, 높이 그리고 각 픽셀의 값을 담고 있는 텍스트 파일을 프로그램에서 활용할 수 있도록 적절한 형태로 변환하는 기능을 수행한다. 더불어 변환된 ASCII 문자 그림을 출력하는 역할도 수행한다.

artist class

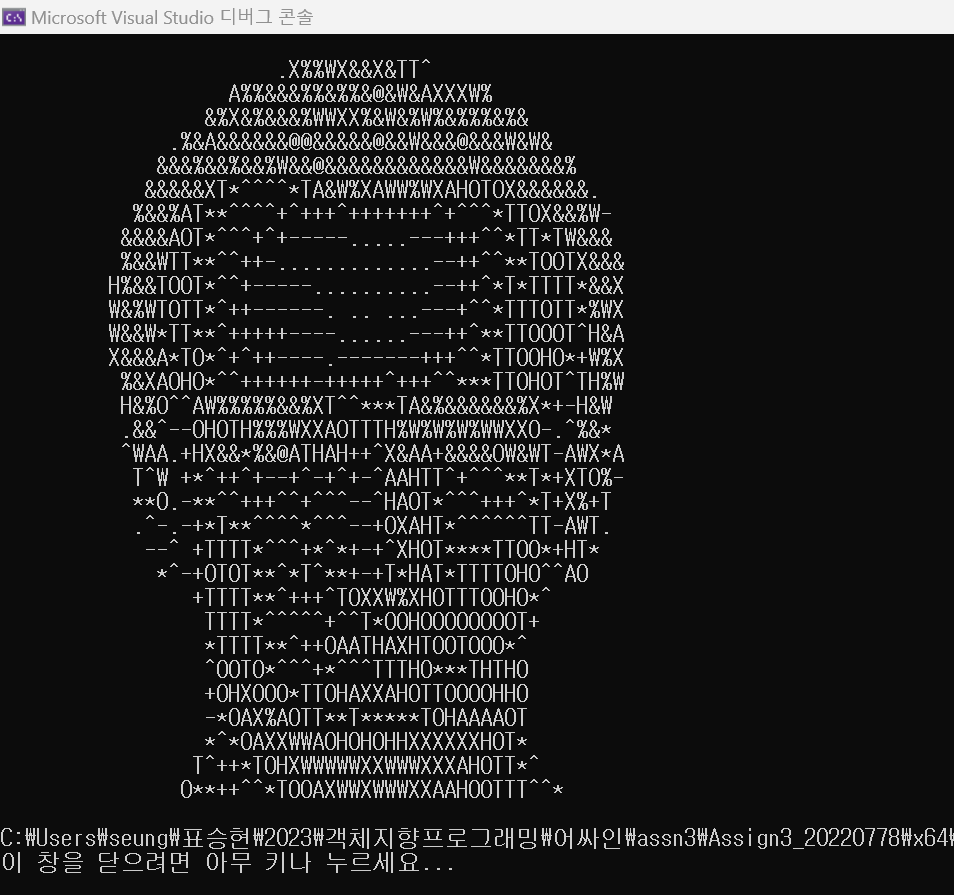
그림을 표현할 style의 종류를 구현하는 class이다. mapper 함수를 이용해 각 픽셀값을 참고하여 값에 따라 적절한 문자로 변환하여 저장한다. 이를테면 classic class의 경우 각 픽셀의 좌표를 인수로 받고 해당 값이 높을수록 복잡한 문자로 변환하여 시각적 density를 구현한다. Iclassix class의 경우에는 classic의 반대로 픽셀값이 높을수록 dense가 낮은 문자로 변환한다.



위 그림은 이번 assigment에서 주어진 예시 파일을 iclassic style로 변환한 것이다.

drawer class

artist class를 이용하여 각각의 픽셀값을 문자로 변환한 것을 draw 함수를 이용하여 화면 상에 출력하는 class이다. 이때 draw의 방법에는 세 가지 옵션이 있는데 downsample과 upsample 그리고 scale이다. Downsample은 변환된 ASCII art의 크기를 두 배 줄여서 출력하고, upsample은 반대로 두 배 늘려서 출력한다. Scale의 경우 단순히 두 배 줄이고 늘리는 데에서 그치는 것이 아니라 너비와 높이의 변환값을 추가 인자로 받아 각각을 n배 늘리거나 줄이는 옵션이다.



위 사진은 “classic|scale|1|-2” config를 이용하여 출력한 결과이다.

1. **토론 및 개선**

문제4> drawer 생성자가 artist가 아닌 artist\*를 인자로 받는 이유를 설명하라

Artist를 인자로 받는 생성자는 artist에 대한 메모리 복사가 일어나기 때문에 불필요한 메모리 및 시간을 소요한다. 그러나 artist\*를 인자로 받으면 주소값만을 저장하면 되기 때문에 불필요한 메모리 복사를 하지 않아도 된다.

문제5> artist 클래스를 상속받아서 자신만의 style을 정의하고 그림을 출력하라

Int main() { …

else if (style\_target == "my\_style") {

style = new my\_style(width, height, vals);

}

… }

char my\_style::mapper(int y, int x)

{

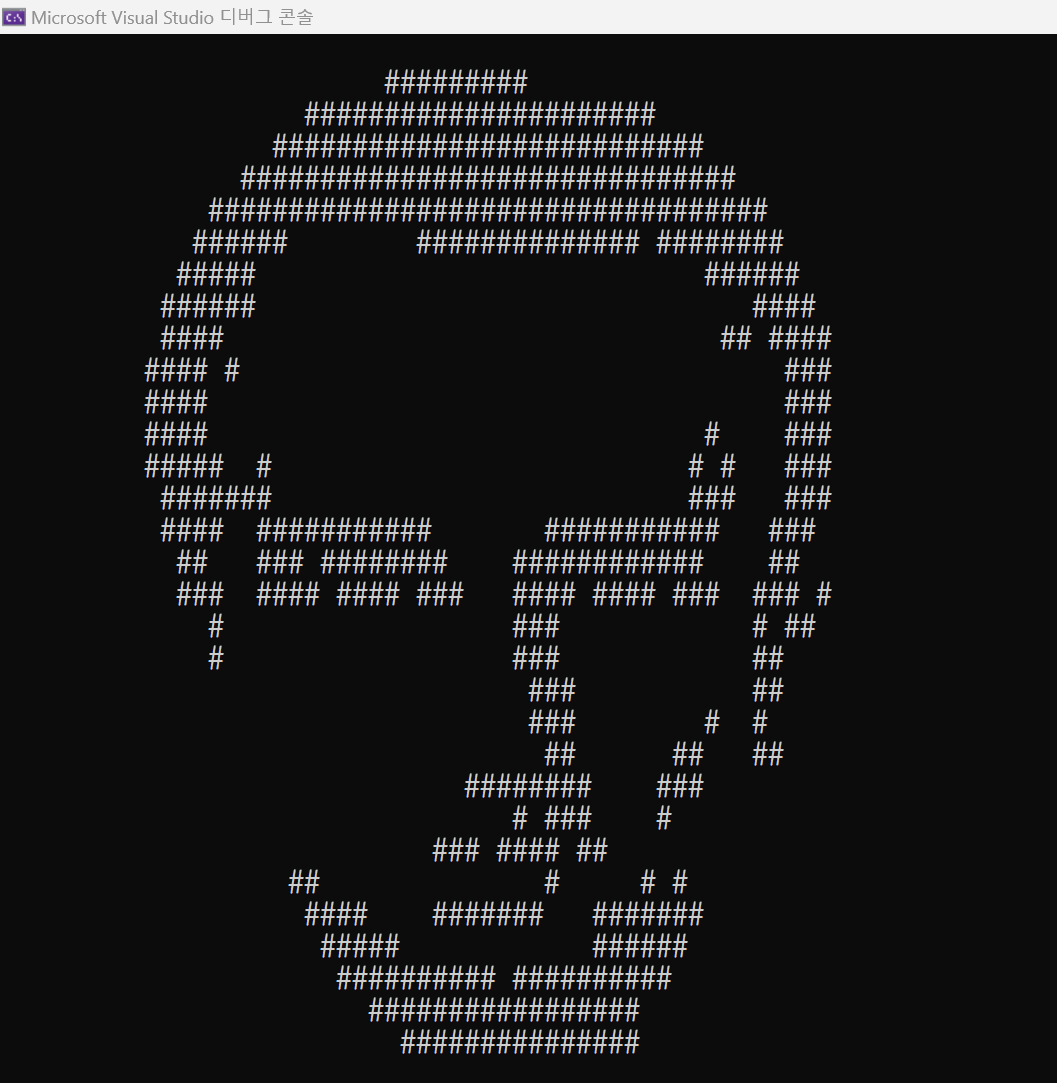
int num = artist::find\_pixel(y, x);

if (num < 127) return '#';

return ' ';

}

클래스를 정의하여 픽셀값이 127 미만인 경우 #을 출력하도록 하여 명암 표시를 할 수 있게 해보았다. Config 내용은 “my\_style|scale|1|-2”이며 실행한 결과는 아래와 같다.



문제6> 제시된 클래스들을 객체 지향 프로그래밍 관점에서 개선할 수 있다면 어떤 점을 더 개선할 수 있을지 토의하라

다양한 기능을 추가해본다. artist 클래스에서는 기본적인 그림 그리기 기능만 제공하고 있다. 이를 보완하기 위해 다양한 기능을 추가할 수 있다. 예를 들어, 이미지를 회전하는 기능, 이미지를 자르는 기능, 이미지를 필터링하는 기능 등을 추가할 수 있다. 이러한 기능을 추가함으로써, 클래스의 활용도가 높아지고 사용자가 다양한 이미지 처리 기능을 수행할 수 있다.

1. **참고 문헌**