2023 Spring OOP Assignment Report

과제 번호 : 3
학번 : 20220826
이름 : 김민서
Povis ID : kimminseo

명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

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

1. 프로그램 개요

- 본 프로그램은 ASCII Art generator를 구현한 것으로, 이미지와 설정 파일을 읽어들여 생성된 ASCII Art를 출력하고 파일에 저장한다.
- 프로그램의 디렉토리 구조는 src/에 cpp 소스 파일, include/에 헤더 파일로 구성되어 있다.

  소스 파일의 main.cpp는 프로그램의 진입점을 정의한다. artist/ 디렉토리에는 artist 클래스 및 파생 클래스를 정의하는 파일이 포함된다. drawer/ 디렉토리에는 drawer 및 파생 클래스를 정의하는 파일이 포함된다. parser/ 디렉토리에는 parser 클래스를 구현한 파일이 포함된다.
- 실행 파일 생성을 위한 Makefile을 제공한다. make 명령을 수행하면 실행 파일(이름: assn3)이 생성된다.

2. 프로그램의 구조 및 알고리즘

- 본 프로그램의 작성을 위해 구현된 class는 다음과 같다.
  - StudentDatabase
  - Artist (virtual)
    - classic
    - iclassic
    - sobelx
    - sobely
    - gradient
    - custom_artist

- drawer
  - upsample
  - downsample
  - scale
- parser

프로그램 실행 시 argument로 사진 파일, 설정 파일, 출력 파일의 경로를 입력받는다. 이후 parser 객체 생성 및 사진 파일, 설정 파일을 parse하여 이미지의 폭/높이 및 픽셀 배열, ASCII Art 생성에 사용할 artist 및 drawer의 정보를 얻는다.

Parsing 결과를 바탕으로 artist 및 drawer의 인스턴스를 생성한다. 이때, 파생 클래스의 인스턴스를 base 클래스 타입의 포인터로 참조한다. 이렇게 함으로서 부모 클래스는 공통된 인터페이스를 제공하고, 세부 구현은 파생 클래스에서 다르게 작성하여 다형성을 이룬다.

drawer 객체는 artist 객체의 포인터를 멤버로 갖는다. draw() 메서드는 ASCII Art를 그린 결과를 반환하는데, 이때 특정 픽셀에 대한 ASCII 변환 값을 얻어오기 위하여 멤버의 artist 객체가 제공하는 mapper() 메서드를 사용한다.

artist 객체의 mapper() 메서드는 특정 픽셀에 대응되는 ASCII 값을 반환한다. 파생 클래스마다 세부 구현이 다르다.

drawer 객체의 소멸 시, 멤버인 artist 객체의 포인터를 통해 원본 artist 객체까지 소멸시킨다. 이를 구현하기 위하여 drawer 클래스에 destructor를 정의하였다. 현재의 파생 클래스 구현상 destructor를 오버라이딩 할 일이 없으므로 dynamic binding이 아니어도 문제가 없으나, 만일 새로운 파생 클래스를 작성할 때 destructor를 재정의해야 하는 경우를 위하여 destructor를 virtual method로 정의하였다.

artist 객체는 현재 구현된 파생 클래스에 대해서 소멸 시 실행되어야 할 작업이 존재하지 않지만, 그럼에도 만일 추후 파생 클래스를 추가할 때 구현상 destructor를 정의해야 하는 경우를 위하여 virtual destructor를 정의해 두었다.

3. 토론

- 본 과제를 수행하며 virtual function을 통해 파생 클래스의 서로 다른 구현을 부모 클래스의 공통된 인터페이스를 통하여 접근함으로써 다형성의 개념을 익힐 수 있었다.

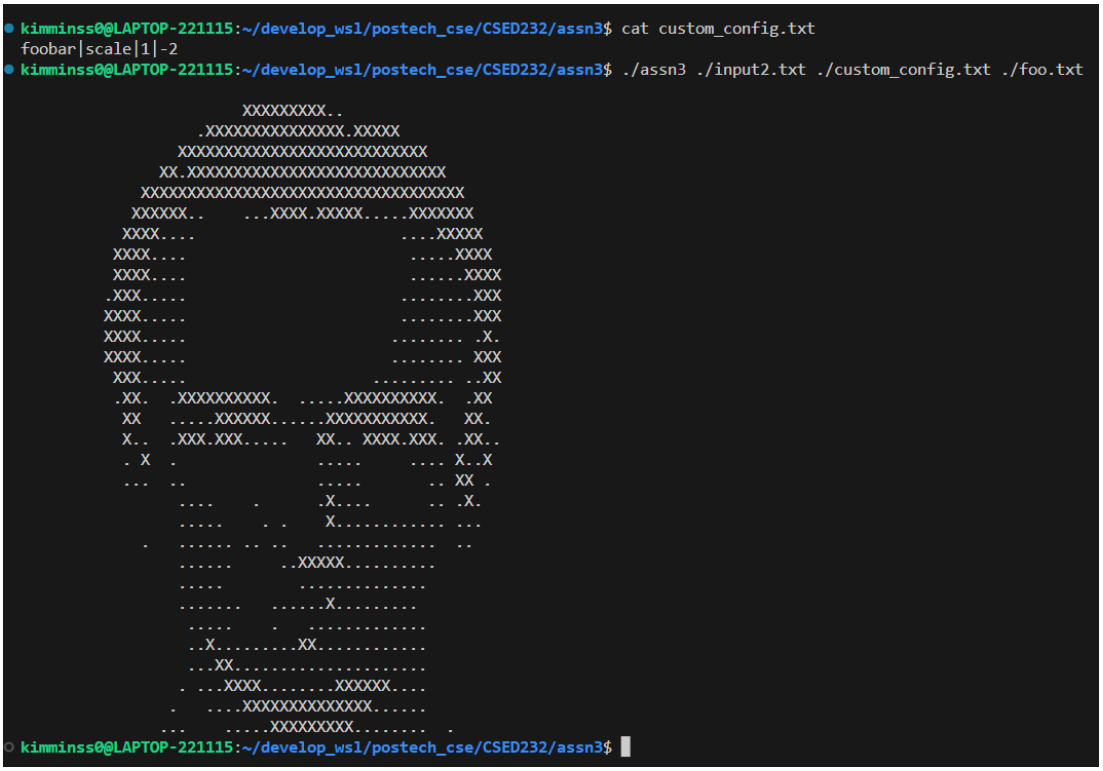
4. 문제 4, 5, 6번 답안

- [문제 4] drawer 생성자가 artist가 아닌 artist*를 인자로 받는 이유를 설명하라.

  drawer 객체에서 artist 파생 클래스의 인스턴스를 base class인 artist의 포인터로 참조함으로서 자식 클래스에서 오버라이드된 메서드를 호출할 수 있다. 만일 생성자에서 artist class 타입으로 artist 파생 클래스의 인스턴스를 인수로 받는다면 artist의 (default) copy constructor가 호출되며, 생성된 객체는 파생 클래스의 타입이 아닌 artist의 타입이므로 메서드 호출 시 static binding이 일어나 다형성이 깨진다. Pointer 또는 reference로 받아야 dynamic binding에 의해 다형성을 이룰 수 있다.

- [문제 5] artist 클래스를 상속받아서 자신만의 style을 정의하고 그림을 출력하라.

  classic을 변형하여 3단계의 명암을 갖도록 하였다.



- [문제 6] 개선 사항

  main 함수에서 artist 타입의 인스턴스가 동적 할당되나 그 해제는 drawer 객체의 소멸자에서 발생하고 있다. 할당이 이루어진 main 함수에서는 생성된 artist 인스턴스의 생명 주기를 예측하기 쉽지 않다. 또한, drawer 객체가 해당 artist 객체를 완전히 소유하는 관계가 아니기 때문에 drawer 객체에서 artist 객체를 소멸하는 것이 타당하지 않다고 보인다. 따라서, artist 객체의 소멸 또한 main에서 이루어지는 것이 타당하다고 생각한다.

5. 참고 문헌

-