

# Project #3: 상속과 다형성을 이용한 테트리스 게임 완성

C++ 프로그래밍  
국민대학교 소프트웨어학부

## Abstract

본 과제에서는 상속과 다형성을 이용한 객체지향 설계를 통해 총 7개의 테트로미노 블록이 무작위로 떨어지는 테트리스 게임을 완성한다.

## 1 과제 소개

본 과제에서는 설계과제 #1과 #2의 내용을 확장하여, 상속과 다형성을 이용한 객체지향 설계를 통해 총 7개의 테트로미노 블록이 무작위로 생성되는 테트리스 게임을 완성한다. 과제에서 작성할 프로그램은 Fig. 1와 같이 7개의 모든 블록을 사용하여 이루어지는 테트리스 게임을 구현하는 것이다.

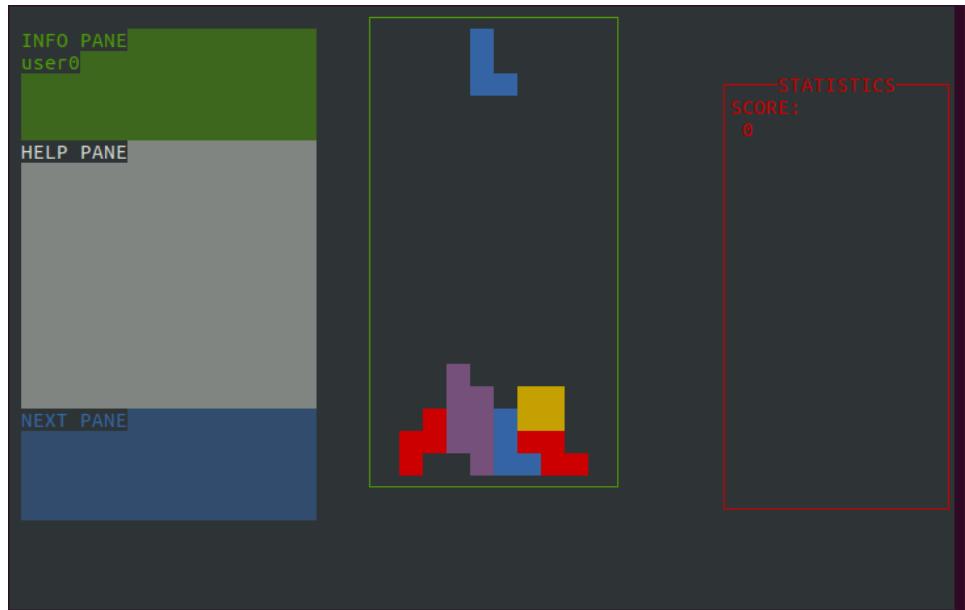


Fig. 1. 테트리스 프로그램 실행 화면 예시

## 2 과제 목표

본 과제의 목표는 상속과 다형성을 이용하여 자신이 작성한 설계과제 #2 코드를 확장하여 7가지 테트로미노 블록들로 구성된 테트리스 게임을 디자인 하는 것이다.

- 기존 코드에 대한 재활용 능력 함양: 설계 과제 #1과 #2의 기능을 모두 지원한다.
- 상속과 다형성을 이용한 프로그래밍 능력 개발: 이전 과제인 설계과제 #2에서는 하나의 테트로미노에 대해 동작하는 테트리스 코드를 작성함으로써 기본적인 알고리즘 개발에 주안점을 두었다. 본 설계과제 #3에서는 상속과 다형성을 통해 다양한 테트로미노에 대해 동작하는 테트리스 프로그램을 완성하도록 한다. 상속과 다형성의 성질을 잘 이해하여, 설계과제 #2에서 이미 작성한 코드는 알고리즘 코드로 재활용하고 새롭게 작성하는 설계과제 #3의 코드는 7개의 테트로미노에 대한 다형성이 반영되는 역할로 활용하도록 한다.
- 출제자: 김준호, 임은진

블록 번호 블록 이름 블록 색	기본 형태	1회 회전	2회 회전	3회 회전
(0) O-Block COLOR_YELLOW				
(1) L-Block COLOR_BLUE				
(2) J-Block COLOR_GREEN				
(3) S-Block COLOR_MAGENTA				
(4) Z-Block COLOR_RED				
(5) T-Block COLOR_CYAN				
(6) I-Block COLOR_WHITE				

Fig. 2. 테트리스 게임을 구성할 7가지 테트로미노 블록들

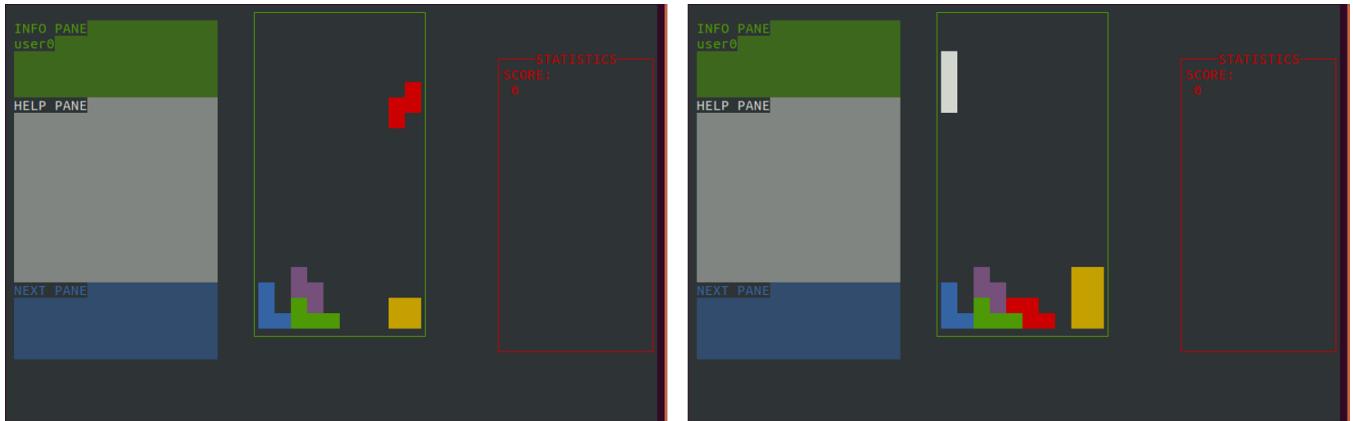


Fig. 3. 떨어지는 블록의 옆이 벽이거나 다른 블록인 경우

## 2.1 7가지 테트로미노 블록을 모두 지원하는 테트리스 게임 완성

- 게임블록은 Fig. 2과 같이 색과 모양이 다른 7가지 블록(테트로미노)으로 구성한다.
  - 7가지의 모든 블록은 Fig. 2와 같이 각각의 번호, 이름, 색을 갖는다.
  - 블록 번호(0 ~ 6)는 블록을 생성할 때 사용될 블록모양의 번호이다.
    - \* 블록을 생성할 때 나오는 블록 모양은 난수 생성(random number generator)을 통해 무작위로 생성되도록 한다
  - 블록 색은 Fig. 2에서 제시한 것과 같이 설정한다.
  - 블록은 회전 키를 입력 한다면 Fig. 2와 같이 회전한다.
    - \* 한 번 회전할 때마다 Fig. 2와 같이 반시계 방향으로 회전하고 네 번 회전하면 다시 기본 형태가 된다.

- \* 만일 Fig. 3과 같이 떨어지는 블록의 옆이 벽이거나 다른 블록인 경우, 회전을 할 때 벽이나 다른 블록을 넘거나 침범하게 되므로 회전할 수 없다.
- 설계 과제 #2와 마찬가지로 사용자 입력에 따라 블록의 위치를 조절할 수 있도록 한다.
  - 사용자 입력에 따른 동작은 다음과 같다(설계 과제 #3에 추가된 것은 굵은 글씨로 표시).
  - \* 블록을 왼쪽으로 한 칸 이동
  - \* 블록을 오른쪽으로 한 칸 이동
  - \* 블록을 아래쪽으로 한 칸 이동
  - \* **블록을 반시계 방향으로 회전**
  - \* 블록을 아래쪽으로 떨어뜨리기
  - \* 게임 종료
  - 블록은 좌, 우, 아래로 자유롭게 움직이거나 반시계 방향으로 회전할 수 있지만, 게임 화면을 벗어나거나 다른 블록의 영역을 침범할 수 없다.
  - 블록이 더 이상 내려갈 수 없을 때에는 그 자리에 고정되고 7가지 모양 중 무작위로 뽑힌 새로운 블록이 생성되어 상단 중앙에 배치 된다.
  - 블록이 쌓여 가로 줄이 빙름 없이 채워지게 되면, 채워진 줄은 없어지고 그 위에 남은 블록은 아래 이동한다.
  - 없어진 줄 수만큼 점수가 올라가게 된다.
  - 블록이 쌓여 게임 화면의 가장 윗줄에 닿으면 게임은 강제 종료 된다.

### 3 과제 요구사항

다음의 요구사항을 만족하는 프로그램을 디자인하도록 한다.

- 키보드 기반 게임 진행: 사용자의 키보드 입력을 통해 진행되는 테트리스 게임을 구현한다. (60점)
- 리플레이 파일 기반 게임 진행: 텍스트 파일로부터 입력을 받아 진행되는 테트리스 게임을 구현한다. (30점)
- 다중 소스코드 기반 개발: 설계과제 #1에서와 같이 다중 소스코드로 개발하며 make를 통해 Makefile로 프로그램이 빌드되도록 소스코드를 구성한다. (10점)

#### 3.1 키보드로 부터 입력

다음과 같이 입력 파일 없이 프로그램을 구동하는 경우, 게임 시작에 앞서 사용자의 이름을 입력 받을 수 있도록 한다. 사용자 이름을 입력하고 Enter(엔터)키를 누르면 Fig. 4과 같이 사용자 이름으로 구성된 테트리스 게임을 시작할 수 있게 설정한다. 사용자 이름은 설계 과제 #1에서 구현했던 사용자 이름 윈도우에 출력될 수 있도록 한다.

---

```
./tetris
테트리스 게임을 시작합니다.
사용자의 이름을 입력하세요.
이름: user
```

---

- 1) 입력 파일이 없는 경우에는 실시간으로 키보드의 입력을 받아 게임을 진행한다.
- 2) 키보드 입력에서는 현재 시각에 따라 무작위로 생성되도록 하는 time(NULL)을 이용하여 srand() 함수의 seed값을 설정 한다.
  - ← (왼쪽 방향키) - 블록을 왼쪽으로 한 칸 이동
  - → (오른쪽 방향키) - 블록을 오른쪽으로 한 칸 이동
  - ↓ (아래쪽 방향키) - 블록을 아래로 한 칸 이동
  - ↑ (위쪽 방향키) - 블록을 반시계 방향으로 회전
  - SpaceBar - 블록을 최하단으로 이동
  - q (영문자 q키) - 게임 종료

#### 3.2 텍스트 파일로부터 입력

다음과 같이 입력 파일과 받아 프로그램을 구동하는 경우, 입력 파일을 리플레이 파일로 간주하여 최종게임 결과를 확인할 수 있다.

---

```
./tetris input.txt
```

---

입력 파일은 다음의 형식을 갖춘 ASCII 텍스트 파일이다.

---

```
[사용자 이름] [seed 값] [블록 위치 조절 ...]
```

---

- 사용자 이름: 한 단어로 표시된 사용자 이름
- seed 값: srand() 함수의 입력인 seed 값으로 정수형 숫자이다. 이 값에 따라 이후 rand() 함수로 생성되는 난수가 달라짐
- l - 왼쪽으로 한 칸 이동

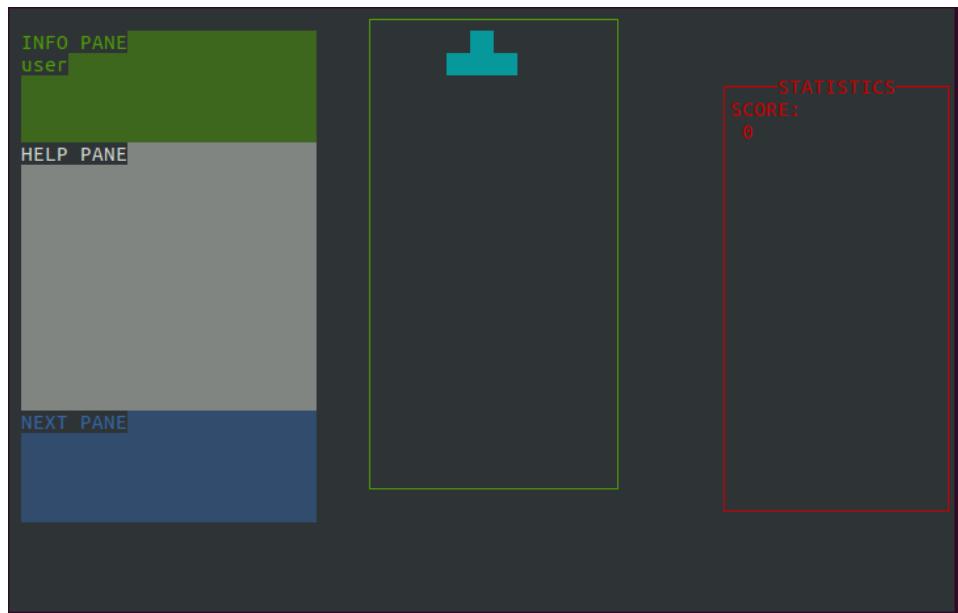


Fig. 4. 입력 받은 사용자 이름이 표시된 게임 화면

- r - 오른쪽으로 한 칸 이동
- g - 아래로 한 칸 이동
- t - 블록을 반시계 방향으로 회전
- d - 쪼 하단으로 이동
- q - 게임 종료

예를 들어, 입력 파일의 내용이 아래와 같다면 사용자 이름은 user이며, srand() 함수의 seed값으로 0이 설정된 상태로 게임이 진행되며, 그 결과는 Fig. 5과 같다 (\* 게임 진행과정을 보여줄 필요는 없으며, 게임 최종결과만 확인할 수 있으면 됨)

---

```
user 0 g t l l l l d l d r d r r r r d q
```

---

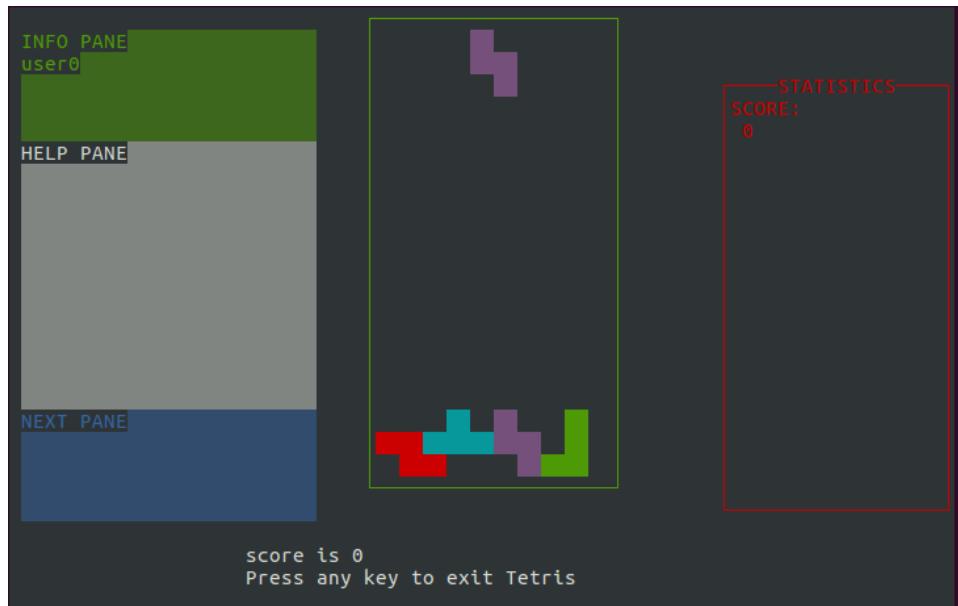


Fig. 5. 입력 텍스트 파일에 의한 게임 최종결과

### 3.3 추가 기능 구현

다음의 기능이 추가 구현될 경우 보너스 점수를 부여한다.

- 텍스트 파일을 통해 리플레이 형태로 진행되는 게임 모드에서 게임의 최종 결과가 아니라 게임이 진행되는 과정이 모두 보이도록 하는 기능 구현(보너스 점수: 30점)

## 4 예제 코드

과제의 목표를 달성하도록 하는 프로그램의 개략적인 코드 전개는 다음과 같다. (주의사항: 아래의 코드는 프로그램의 개략적인 뼈대만을 알려줄 뿐이며, 학생 개개인은 아래의 코드를 참고하여 과제의 목표 및 요구사항을 만족하도록 자신만의 코드를 구현해야 함)

---

```
#include <iostream>
#include <ifstream>
#include <ctime>

class Block
{
protected:
    int state;
public:
    virtual void draw();
    virtual void rotate();
};

class O_Block : public Block
{
public:
    virtual void draw();
    virtual void rotate();
};

// ...

class I_Block : public Block
{
public:
    virtual void draw();
    virtual void rotate();
};

int main(int argc, char* argv[])
{
    std::ifstream inStream;
    bool isGameOver = false;
    char cur_key, username[8];
    int seed;

    // i) argc를 보고, 입력파일이 있으면 파일을 연다. (없으면 그냥 통과)

    // ii) 입력파일이 있으면 입력파일로부터 '사용자 이름'을 입력 받는다.
    //      입력파일이 없으면 키보드로부터 '사용자 이름'을 입력 받는다.

    // iii) 입력파일이 있으면 입력파일로부터 srand() 함수의 seed값을 입력 받는다.
    //       입력파일이 없으면 time(NULL)을 통해 srand() 함수의 seed값을 무작위로 설정한다.

    if(argc > 1) inStream >> seed;
    else seed = time(NULL);
    srand(seed);

    // iv) 테트리스 게임의 초기 화면을 구성한다.
```

```

init();

// v) 종료될 때까지 게임을 진행한다.
while (!isGameOver)
{
    // 입력파일이 있으면 파일내용에 따라 블록의 위치를 조정한다.
    // 입력파일이 없으면 키보드 입력에 따라 블록의 위치를 조정한다.
    if (argc > 1) inStream >> cur_key;
    else cur_key = input_key();

    isGameOver = update(cur_key); // 입력 받은 키 처리를 담당한다.
    display(); // 키보드 처리 후의 화면을 출력한다.
}

// vi) 테트리스 게임에 쓰인 각종 리소스를 반환한다.
finalize();
return 0;
}

```

---

본 과제에서는 위와 같이 자신이 작성한 다중 파일들로 구성된 프로그램이 make를 통해 정상적으로 컴파일되도록 Makefile을 작성하는 것을 목표로 한다. Ubuntu 16.04 LTS의 터미널 환경에서 make로 정상적으로 컴파일되도록 하는 것이 목표이다.

## 참고할 정보

- 과제와 함께 첨부된 ncurses library 튜토리얼 자료를 참고한다.
- ncurses에 대해 더 자세히 알고 싶은 경우, 아래 사이트를 참고한다.
  - <http://pubs.opengroup.org/onlinepubs/7908799/xcurses/curses.h.html>
  - (한국어) NCURSES-Programming-HOWTO
- make에 대해 더 자세히 알고 싶은 경우, 아래 사이트를 참고한다.
  - 위키백과 make
  - GNU Make 강좌 - 임대영
- 난수 생성(random number generator)에 관련된 함수는 다음의 사이트들을 참고한다.
  - srand() 함수: 난수 생성의 seed 값을 설정하는 함수
  - rand() 함수: 난수 값을 생성해 반환하는 함수

## 5 과제 제출방법 (매우 중요!!!)

- 프로그램의 각 기능별로 함수들을 분리해서 구현한다.
- 프로그램 구현 시 반드시 하나 이상의 class를 사용하고, 두 개 이상의 소스파일로 구성한다.
- 6월 2일(금) 23시 59분까지 가상대학에 업로드하도록 한다.
- 과제 코드는 Ubuntu 16.04 LTS 환경에서 make 명령으로 컴파일 가능하도록 작성한다.
- 과제 코드와 보고서 파일들은 하나의 압축파일로 묶어 tar.gz 파일 형식이나 표준 zip 파일 형식으로만 제출하도록 한다. 이때, 압축파일의 이름은 반드시 'OOOOOOOO\_PROJ\_03.tar.gz (OOOOOOOO는 자신의 학번)'과 같이 자신의 학번이 드러나도록 제출한다.
  - 1) 모든 소스파일 (\*.h, \*.cpp)
  - 2) Makefile
  - 3) 보고서 파일 (\*.doc)
- 과제에 관한 질문은 오피스아워를 활용하도록 한다. 교육조교(teaching assistant, TA)에게 메일로 약속시간을 정한 후, 교육조교가 있는 연구실로 방문하여 물어보는 것도 매우 권장하는 방법이다.