

Project #2: 테트리스 알고리즘 구현

C++ 프로그래밍
국민대학교 소프트웨어학부

Abstract

본 과제에서는 설계과제 #1에서 나아가 한 모양의 테트로미노 블록으로만 진행되는 테트리스 게임을 만든다. 본 과제의 목표는 오직 O-블록(■)만으로 진행되는 테트리스 게임을 구현하는 것이다.



1 과제 소개

본 과제에서는 설계과제 #1의 내용을 확장하여, 객체지향 설계(Object-oriented design)를 통해 하나의 모형으로 구성된 테트로미노 블록으로 진행되는 테트리스 게임 알고리즘을 완성한다. 과제에서 작성할 프로그램은 Fig. 1과 같이 오직 O-블록(■)만으로 진행되는 테트리스 게임이다.

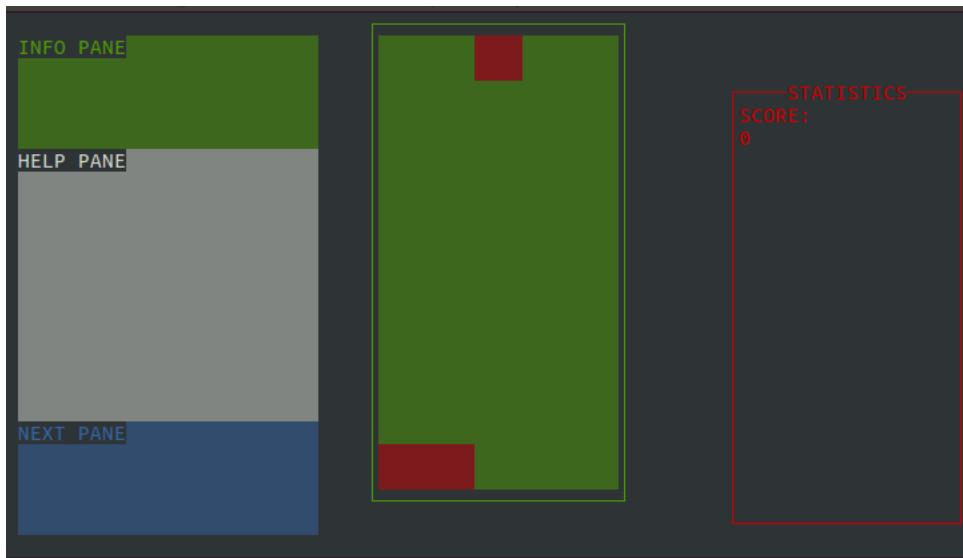


Fig. 1. O-블록(■)만으로 진행되는 테트리스 게임 예시

2 과제 목표

본 과제의 목표는 설계과제 #1 코드를 확장하여 O-블록(■)만으로 진행되는 테트리스 게임을 설계하고 구현하는 것이다.

- 기존 코드에 대한 재활용 능력 함양: 설계 과제 #1의 기능을 모두 지원한다.
- 요구 사항을 만족하는 알고리즘 개발 능력 함양: 알고리즘의 요구사항을 파악하여 이를 만족하는 프로그램 개발 능력을 기른다.

2.1 테트리스 게임 알고리즘

- 게임 시작 시 O-블록(■)은 Fig. 2와 같이 중앙에 배치된다.
- 사용자의 입력에 따라 블록의 위치를 조절할 수 있도록 한다.

- 사용자의 입력에 따른 동작은 다음과 같다.
- 블록은 좌, 우, 아래로 자유롭게 움직일 수 있지만 게임 화면을 벗어나거나 다른 블록의 영역을 침범할 수 없다.
- 블록이 더 이상 내려갈 수 없을 때에는 Fig. 3과 같이 그 자리에 고정되고 새로운 블록이 생성되어 상단 중앙에 배치된다.

-
- 출제자: 김준호, 임은진

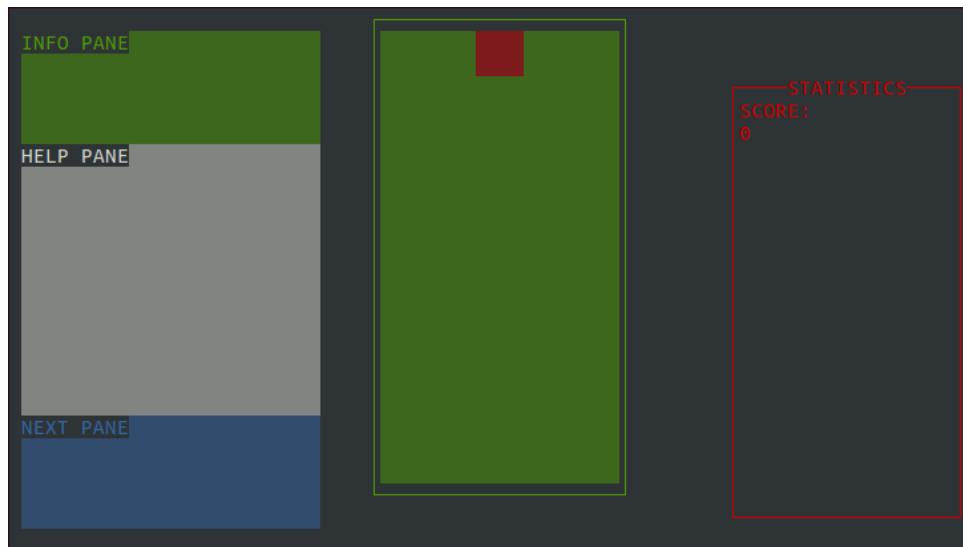


Fig. 2. 프로그램 실행시 초기화면 예시

- d) Fig. 4과 같이 블록이 쌓여 가로 줄이 빈틈 없이 채워지게 되면, 채워진 줄은 없어지고 그 위에 남은 블록은 아래 이동한다.
- e) Fig. 4과 같이 없어진 줄 수만큼 점수가 올라가게 된다.
- f) 도형이 쌓여 게임 화면의 가장 윗줄에 닿으면 게임은 강제종료 된다.



Fig. 3. 하단으로 이동한 블록의 고정 및 새로운 블록의 생성



Fig. 4. 채워진 줄이 사라지고 점수가 상승된 화면 예시

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, 실시간으로 키보드 입력을 받아 게임이 진행되도록 한다.

```
./tetris
```

키보드 입력은 다음과 같이 처리하도록 한다.

- ← (왼쪽 방향키) - 블록을 왼쪽으로 한 칸 이동
- → (오른쪽 방향키) - 블록을 오른쪽으로 한 칸 이동
- ↓ (아래쪽 방향키) - 블록을 아래로 한 칸 이동
- 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 ASCII 텍스트 파일이며, 다음과 같이 테트로미노의 움직임을 나타내는 문자들로 구성되어 있다.

- l - 왼쪽으로 한 칸 이동
- r - 오른쪽으로 한 칸 이동
- g - 아래로 한 칸 이동
- d - 최 하단으로 이동
- q - 게임 종료

예를 들어, 입력 파일의 내용이 아래와 같다면 그 결과는 Fig. 5과 같다. (* 게임 진행과정을 보여줄 필요는 없으며, 게임 최종결과만 확인할 수 있으면 됨)

```
l l l l g d l l d d r r d r r r r r q
```

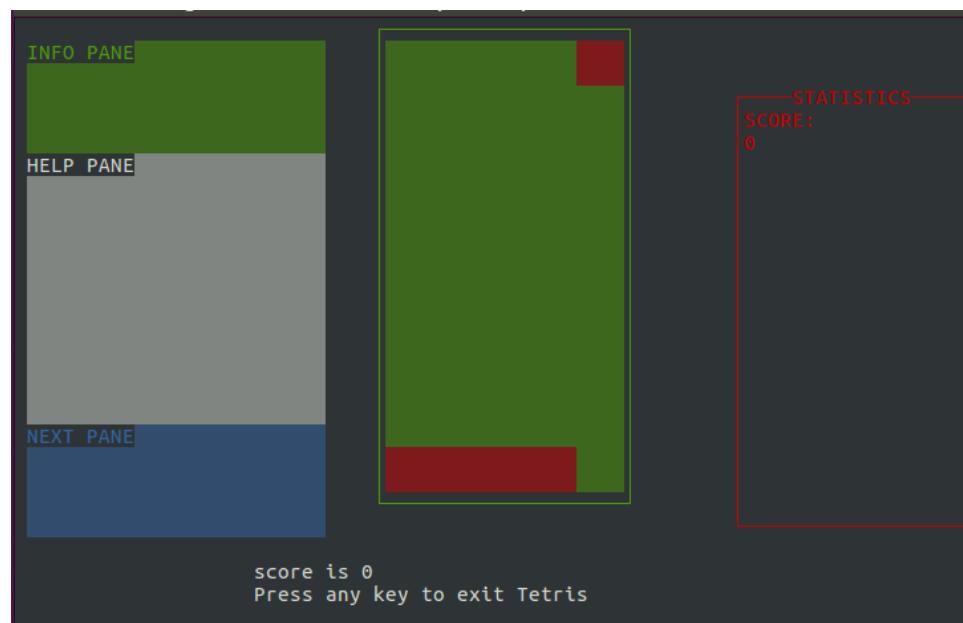


Fig. 5. 입력 파일에 의한 테트리스 게임 최종결과

3.3 추가 기능 구현

다음의 기능이 추가 구현될 경우 보너스 점수를 부여한다.

- 사용자의 키보드로 진행되는 게임 모드에서 일정시간(예: 1초)이 지나면 테트로미노 블록이 자동으로 아래칸으로 떨어지는 기능 구현 (보너스 점수: 20점)

4 예제 코드

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

```
#include <iostream>
#include <fstream>

// 게임에 사용될 블록 클래스를 정의한다.
class Block
{
};

int main(int argc, char* argv[])
{
    std::ifstream inStream;
    bool isGameOver = false;
    char cur_key;
    // i) arg를 확인하여, 입력파일이 있으면 파일을 연다. (없으면 그냥 통과)

    // ii) 테트리스 게임의 초기 화면을 구성한다.
    init();

    // iii) 종료가 될 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.
    }

    // iv) 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
- 색 구성이 Fig. 1와 동일할 필요는 없다.

5 과제 제출방법 (매우 중요!!!)

- 프로그램의 각 기능별로 함수들을 분리해서 구현한다.
- 프로그램 구현 시 반드시 하나 이상의 class를 사용하고, 두 개 이상의 소스파일로 구성한다.
- 5월 19일(금) 23시 59분까지 가상대학에 업로드하도록 한다.
- 과제 코드는 **Ubuntu 16.04 LTS 환경에서 make 명령으로 컴파일 가능하도록** 작성한다.
- 과제 코드는 다음의 파일들은 하나의 압축파일로 묶어 **tar.gz** 파일 형식이나 표준 **zip** 파일 형식으로만 제출하도록 한다.
이때, 압축파일의 이름은 반드시 'OOOOOOOO_PROJ_02.tar.gz (OOOOOOOO은 자신의 학번)'과 같이 자신의 학번이 드러나도록 제출한다.
 - 1) 모든 소스파일 (*.h, *.cpp)
 - 2) Makefile
- 과제에 관한 질문은 오피스아워를 활용하도록 한다. 교육조교(teaching assistant, TA)에게 메일로 약속시간을 정한 후, 교육조교가 있는 연구실로 방문하여 물어보는 것도 매우 권장하는 방법이다.