

Project #1: ncurses를 이용한 TUI 화면 구성

C++ 프로그래밍
국민대학교 소프트웨어학부

Abstract

본 과제에서는 텍스트 사용자 인터페이스(Text User Interface, TUI) 라이브러리 중 하나인 ncurses 라이브러리를 이용하여 테트리스 게임의 화면을 구성하는 기본적인 기능을 구현한다.



1 과제 소개

본 과제에서는 텍스트 사용자 인터페이스(Text User Interface, TUI) 라이브러리 중 하나인 ncurses를 이용하여 테트리스 게임의 화면을 구성하는 기본적인 기능을 구현하도록 한다. 과제에서 작성할 프로그램은 Fig. 1과 같이 테트리스 게임 초기 화면을 출력하는 것이다.

2 과제 목표

본 과제의 목표는 중규모 이상의 프로그램을 객체지향언어인 C++로 개발함에 있어 필요한 개발요소를 학습하는 것이다.

- 요구사항을 만족하는 프로그램 개발: 복잡한 요구사항을 분석하여 이를 만족하는 프로그램을 개발할 수 있는 능력을 확보한다.
- 외부 라이브러리를 이용한 프로그램 개발: ncurses와 같은 외부 라이브러리를 이용하여 프로그램을 개발할 수 있는 능력을 확보한다.
- 여러 파일을 이용한 C++ 프로그램 개발: 하나의 소스파일로만 구성된 프로그램이 아니라, 클래스 별로 헤더파일과 소스파일을 구성하여 일정 규모 이상의 프로그램을 개발할 수 있는 능력을 확보한다.
- make를 이용한 C++ 프로그램 개발: 여러 소스파일로 구성된 프로그램을 컴파일/링크를 편리하게 하기 위한 도구로써 make를 종종 이용한다. 간단한 Makefile 파일을 만들고 make를 활용하여 여러 소스파일로 구성된 C++ 프로그램을 개발하는 능력을 확보한다.

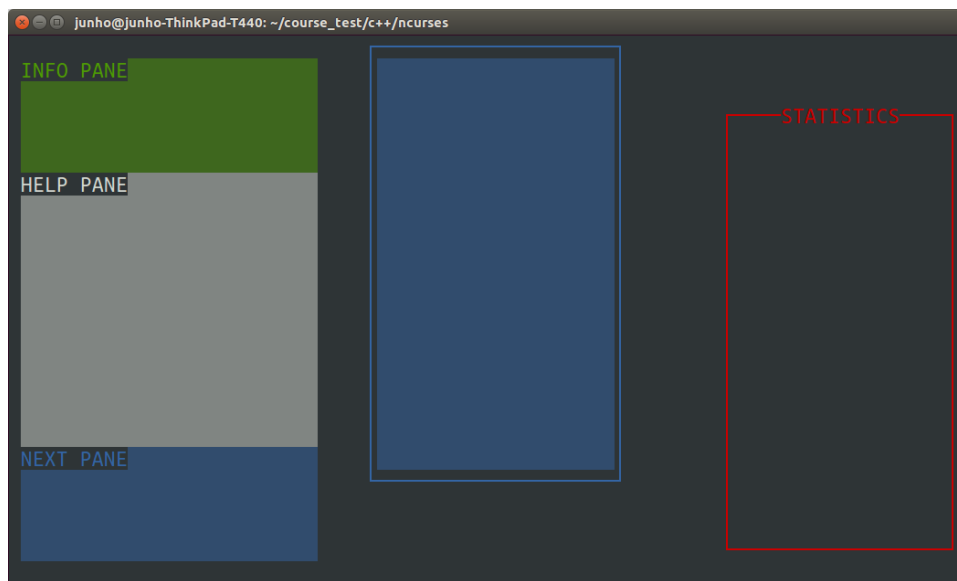


Fig. 1. ncurses를 활용하여 설계 과제 프로그램 초기 화면을 출력한 예

- 출제자: 김준호, 임은진

3 ncurses 설치

Ubuntu 16.04 LTS에서 ncurses를 설치하기 위해서는 다음의 명령을 터미널에서 수행하도록 한다.

```
sudo apt-get update
sudo apt-get install libncurses5-dev libncursesw5-dev
```

ncurses를 이용해서 작성한 소스코드 main.cpp가 있으면 다음과 같이 -lncurses 라이브러리 링크 옵션을 주고 컴파일하도록 한다.

```
g++ main.cpp -lncurses
```

4 과제 요구사항

다음의 세부 목표를 만족하는 프로그램을 디자인하도록 한다.

- TUI 기반 테트리스 화면 구성 (60점)
- 다중 파일을 이용한 C++ 프로그램 개발 (20점)
- make를 이용한 C++ 프로그램 개발 (20점)

4.1 TUI 기반 테트리스 화면 구성 (60점)

TUI 라이브러리 중 하나인 ncurses를 이용하여 테트리스 게임화면을 Fig. 1와 같이 구성한다. 테트리스 게임이 진행될 BoardPane의 크기만 규격을 준수하고, 그 이외의 사항은 자신의 개성에 맞게 자유롭게 화면을 구성하도록 한다.

- TUI 모드 사용 시작/종료
 - 프로그램 시작 시에 TUI mode를 시작하고 프로그램 종료 시에 TUI mode를 종료하도록 한다.
- 컬러쌍(color pair) 설정 기능
 - Fig. 1와 같이 각 윈도우의 화면 구성에 사용할 컬러 쌍들을 설정한다.
- 윈도우들의 생성 기능
 - 사용자 정보 윈도우인 InfoPane 생성
 - * 사용자의 이름을 표시할 윈도우이다.
 - * 본 과제에서는 해당 pane의 제목만 표시되도록 하고, 내용은 채워넣지 않는다.
 - 도움말 정보 윈도우인 HelpPane 생성
 - * 테트리스 게임 활용키 등과 같은 도움말을 표시할 윈도우이다.
 - * 본 과제에서는 해당 pane의 제목만 표시되도록 하고, 내용은 채워넣지 않는다.
 - 다음 테트로미노 표시 윈도우인 NextPane 생성
 - * 다음에 떨어질 테트로미노를 알려주는 윈도우이다.
 - * 본 과제에서는 해당 pane의 제목만 표시되도록 하고, 내용은 채워넣지 않는다.
 - 게임 윈도우인 BoardPane 생성
 - * BoardPane은 게임 진행시에 테트로미노(tetromino, 일명 테트리스 블록)들이 표시될 윈도우이다.
 - * BoardPane은 가로 $22(= 1 + 2 \times 10 + 1)$, 세로 $22(= 1 + 20 + 1)$ 로 설정한다.
 - Board pane의 외곽선을 표현하기 위해, 가로에서 가장 왼쪽 한줄과 오른쪽 한줄, 세로의 가장 위쪽 한줄과 아랫쪽 한줄을 윤곽선으로 표시한다.
 - 테트로미노가 움직일 수 있는 공간의 크기는 가로 10, 세로 20의 공간이다.
 - 그러나 터미널의 한 문자가 찍히는 공간이 가로:세로 비율이 1:2이기 때문에 연속 두 칸 checker board (ACS_CKBOARD)를 두번 찍어 정사각형 모양을 만들고 이를 테트로미노의 한 칸으로 가정하도록 한다.
 - 따라서, TUI의 가로 × 세로 공간 크기를 20×20 으로 설정하는 방식으로 테트로미노가 움직일 수 있는 공간 크기 10 × 20을 표현할 수 있도록 한다.
 - 점수 윈도우인 StatPane 생성
 - * 사용자의 현재 점수를 표시 할 윈도우이다.
 - * 본 과제에서는 해당 pane의 제목만 표시되도록 하고, 내용은 채워넣지 않는다.
- 윈도우 바깥 경계선 설정 기능
 - ncurses의 box() 함수를 사용하여 윈도우 바깥 경계선을 표시한다.

4.2 다중 파일을 이용한 C++ 프로그램 개발 (20점)

하나의 소스코드(예: main.cpp)로 프로그램을 개발하게 되면 협업을 통한 프로그램 개발이 어려워지고 관리가 복잡해 진다는 단점이 있다. 간단한 프로그램을 개발할 경우라도 의미있는 여러 개의 파일로 쪼개서 개발하는 경우가 일반적이다.

C++ 언어를 이용한 개발에서는 하나의 클래스를 헤더파일 하나와 소스파일 하나로 표현하는 것이 일반적이다. 예를 들어, Pane 클래스가 있다면 Pane 클래스의 선언은 Pane.h라는 헤더파일에 클래스의 구현은 Pane.cpp 소스파일에 표현되도록 한다.

본 과제에서는 등장하는 모든 클래스에 대해 각 클래스가 자신만의 헤더파일과 소스파일을 가지도록, 다음과 같이 다중 파일을 이용하여 C++ 프로그램을 개발하도록 한다.

```
/tetris/
```

다중 파일로 개발할 경우, 헤더파일이 중복 포함될 수 있기 때문에 헤더파일이 중복 포함되지 않도록 모든 헤더파일에 다음과 같이 `#ifndef #define #endif` 문을 적절하게 활용하도록 한다. 자세한 사항은 교재 323-324 페이지를 참고한다.

```
// Pane.h
#ifndef __PANE_H__
#define __PANE_H__

...

#endif //__PANE_H__
```

4.3 make를 이용한 C++ 프로그램 개발 (20점)

프로그램 개발 규모가 어느정도 커지면 앞서 설명한 바와 같이 다중 파일로 개발하는 경우가 많아지게 된다. 이런 경우 매번 다음과 같이 모든 소스파일들을 g++ 컴파일러에게 알려줘서 프로그램을 컴파일해야 하는데, 이는 여간 귀찮은 일이 아니다.

```
g++ -o tetris tetris.cpp Pane.cpp InfoPane.cpp HelpPane.cpp ... StatPane.cpp -lncurses
```

이러한 작업을 귀찮은 작업을 간단하게 해 주는 도구로 널리 쓰이는 것이 make이다. make는 현재 디렉토리에 Makefile의 내용을 기반으로 작업을 수행하는 도구인데 일반적으로 C++ 프로그램 개발에서는 다음과 같이 활용하도록 한다.

- 프로그램 소스코드가 있는 디렉토리에 Makefile이라는 텍스트 파일을 만든다.
- Makefile의 내용을 g++ 컴파일 내용으로 채운다.
- 매번 새롭게 컴파일이 필요한 경우 터미널에서 간단하게 make를 수행하면 Makefile의 내용에 따라 컴파일이 된다.

make를 활용하기 위해서는 Makefile을 잘 작성해야 하는데 다음은 C++ 프로그램 개발에서 널리 쓰이는 일반적인 형태의 Makefile이다.

```
# Makefile example
CC = g++
TARGET = tetris
SOURCES = tetris.cpp \
           Pane.cpp \
           BoardPane.cpp \
           InfoPane.cpp \
           HelpPane.cpp \
           StatPane.cpp \
           NextPane.cpp
LDLAGS = -lncurses

all:
    $(CC) -o $(TARGET) $(SOURCES) $(LDLAGS)

clean:
    rm -rf *.o $(TARGET)
```

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

5 과제 제출방법 (매우 중요!!!)

- 프로그램의 각 기능별로 함수들을 분리해서 구현한다.
- 5월 5일(금) 23시 59분까지 가상대학에 업로드하도록 한다.
- 과제 코드는 **Ubuntu 16.04 LTS 환경에서 make 명령으로 컴파일 가능**하도록 작성한다.
- 과제 코드는 다음의 파일들은 하나의 압축파일로 묶어 **tar.gz** 파일 형식이나 표준 **zip** 파일 형식으로만 제출하도록 한다. 이때, 압축파일의 이름은 반드시 'OOOOOOOO_PROJ_01.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.