2020년도 2학기 컴퓨터공학설계및실험I

6주차 예비 보고서

20170175 김태안

1. 실습 목적

누구에게나 친숙하고 유명한 게임인 테트리스(tetris)를 구현한다. 실험 1주차에서는 제공된 프레임(frame) 프로그램을 바탕으로, 블록의 이동, 블록의 회전, 블록을 필드에 쌓기, 줄 삭제, 점수 계산, 블록 미리 보여주기로 구성된 기본적인 기능들을 갖는 테트리스 게임을 구현하고, ncurses 라이브러리, 디버깅 방법, makefile을 만드는 방법 등을 익히도록 한다.

1. 관련 이론

**ncurses 라이브러리**

Ncureses는 터미널을 제어하는 라이브러리인 curse의 발전된 버전으로, 터미널에서의 입출력과 관련된 많은 기능을 가지고 있다. 만들어진 1995년부터 리눅스 커널, 맥OS, MinGW 등 다양한 환경에 포함되었다. 오픈소스는 아니지만 무료로 사용 가능하다.

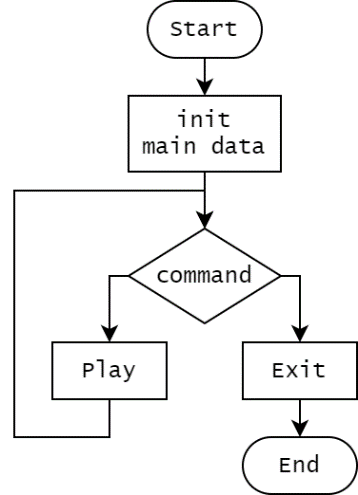
Ncurses에는 터미널에서의 입출력과 관련된 많은 기능이 있는데, 그 중 하나는 터미널에 색이 있는 결과를 출력하는 것이다. color 변수를 설정해 둔 뒤, start\_color와 같은 명령어를 사용하면 이후의 출력 결과는 기존의 흑백 결과가 아닌 빨강, 파랑 등의 색을 가진 결과가 출력된다.

입력과 관련한 Ncurses의 기능 중 하나는 사용자의 방향키를 포함한 입력을 받아 명령을 수행하는 것이다. 기존 터미널에서 키보드로 입력할 수 있는 명령은 숫자와 문자열에 그쳤지만, Ncurses의 wgetch를 통해 방향키와 같은 키의 입력도 받아 신호로 사용할 수 있다. 이번 실습에서는 이를 통해 방향키를 사용해 플레이가 가능한 테트리스를 구현할 것이다.[1]

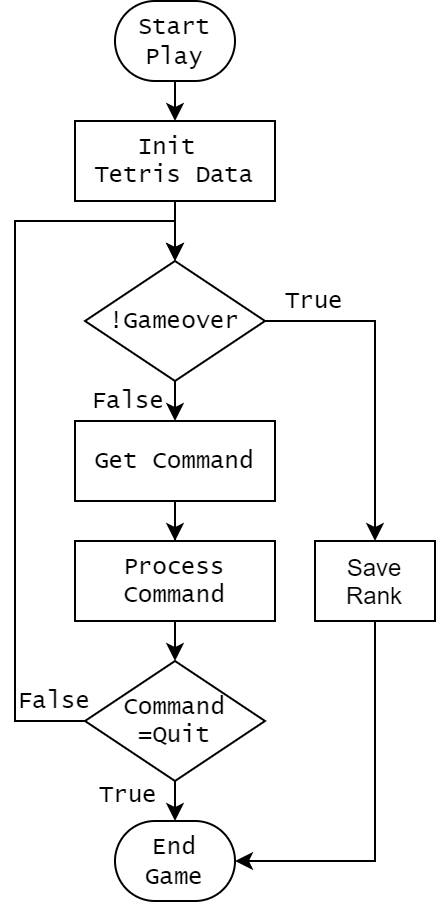
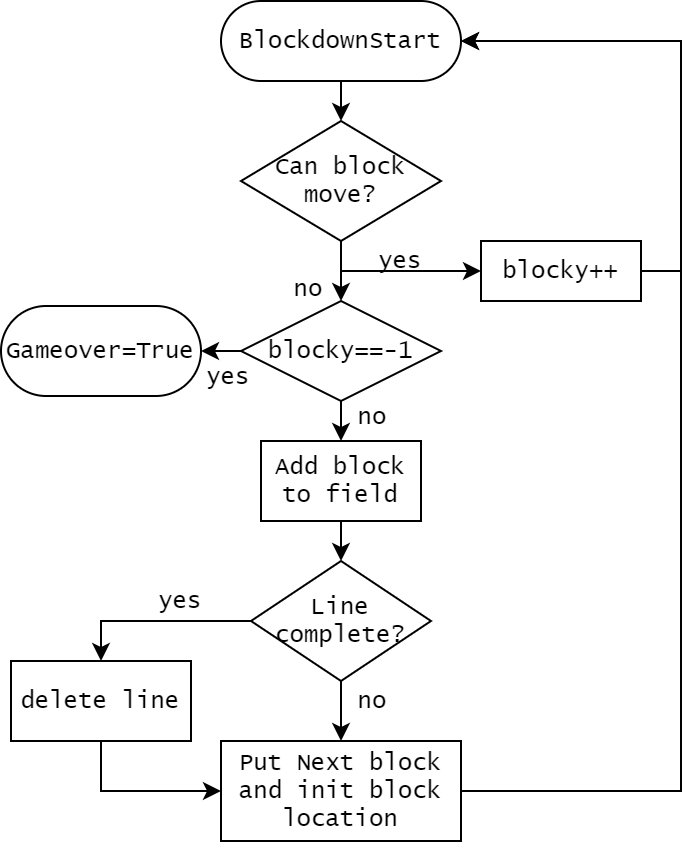
C에서 ncurse를 사용하기 위해서는 #include <ncurese.h> 명령어를 통해 라이브러리를 불러와야 한다. 그리고 컴파일할 때, -lncurese 옵션을 통해 ncurse를 링크해야 정상적으로 사용이 가능하다.[2]

1. 실습 방법

**테트리스 프로그램의 알고리즘**



테트리스 프로그램은 제일 먼저 사용자에서 메뉴를 보여준다. 메뉴는 게임 시작, 점수 확인, 모드 변경, 종료의 4가지로 이루어져 있다. 사용자가 종료 커맨드를 입력하지 않는 이상 프로그램을 작동하며 다른 3가지의 동작이 끝나더라도 다시 메인으로 돌아와 다음 명령을 기다린다.

제일 중요한 기능인 play는 두 함수의 독립된 실행으로 작동한다. 먼저 play 함수는 사용자의 입력을 담당한다. 사용자가 입력하는 방향키에 따라 떨어지는 블록을 이동시키거나 회전시킨다. 종료 커맨드가 입력되거나 Gameover 플래그가 참이면 함수를 점수를 등록하고 함수를 종료해 게임을 종료한다.

BlockDown 커맨드는 떨어지는 블록을 담당한다. 1초마다 호출되는 이 함수는 게임이 종료되기 전까지 매초 실행된다. 실행될 때 마다 블록이 한 칸씩 떨어진다. 블록이 바닥에 닿아 더 이상 움직일 수 없다면 새 블록을 추가해 떨어트린다. 이때, 쌓인 블록이 한 줄을 완성했다면 줄을 지우고 점수를 추가한다. 블록이 맨 위에 닿아 더 이상 떨어질 곳이 없다면 Gameover 플래그를 켜 게임 종료를 알린다.

**구현된 함수**

**int main();**

프로그램의 시작으로 커맨드를 입력 받아 메뉴에 맞는 함수를 실행한다.

**char menu();**

메뉴를 출력한다.

**void play();**

테트리스 게임을 실행한다. 동작 방식과 역할은 위의 설명과 같다.

**void InitTetris();**

초기화해야 할 변수를 초기화 한다.

**void DrawOutline();**

테트리스 게임의 아웃라인을 출력한다.

**void DrawField();**

테트리스 블록이 움직이고 쌓이는 게임 필드를 그린다. 필드는 쌓여 있는 블록을 포함한다.

**void DrawNextBlock(int \*nextBlock);**

다음에 떨어질 블록을 미리 보여준다. 다음 블록은 다른 함수에서 필요할 때 랜덤하게 설정되며, 이 값을 입력 받는다.

**void PrintScore(int score);**

현재 점수를 입력 받아 출력한다.

**void DrawBox(int y,int x, int height, int width);**

좌표를 입력받고, 좌표에 맞는 위치에 ■을 출력해 칸을 그린다.

**void DrawBlock(int y, int x, int blockID,int blockRotate,char tile);**

테트리스 블록의 번호와 좌표를 입력 받아 해당 위치에 출력한다. 출력할 모양 (■, ///, XXX 등)을 입력 받아 그에 맞게 출력한다.

**void gotoyx(int y, int x);**

화면에 어떤 값을 출력하기 위해 해당 좌표로 커서를 이동한다.

**int GetCommand();**

사용자가 입력하는 커맨드를 받아 정수로 변환해 반환한다.

**int ProcessCommand(int command);**

GetCommand를 통해 받은 커맨드에 맞는 행동을 수행한다. 커맨드 중 quit에 대한 입력은 0을, 그 외의 경우는 1을 반환한다.

**구현할 함수의 수도코드**

|  |
| --- |
| int CheckToMove(char field[HEIGHT][WIDTH], int currentBlock, int blockRotate)  if not (놓으려는 자리에 이미 블록이 있는 경우 && height<= y && x<0 && x>=WIDTH)  return 1;  else return 0; |
| void DrawChange(char field[HEIGHT][WIDTH], int command, int currentBlock, int blockRotate, int blocky, int blockX)  switch(command)  case 1: if (CheckToMove(변경값) == 1) draw new, erase old  case 2: if (CheckToMove(변경값) == 1) draw new, erase old  …. |
| int BlockDown(int sig)  if (y != -1 && CheckToMove == 0)  AddBlockToField;  DeleteLine  새 NextBlock 설정  else if (y = -1 && CheckToMove == 0)  gameover = 1  else  블록 한칸 내리기 |
| int AddBlockToField(char field[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blocky, int blockX)  블록 하나 추가  if (블록 채워져 있음)  field[blockY+i][blockX+j] = 1; |
| int DeleteLine(char field[HEIGHT][WIDTH])  for문 통해 줄 체크  if (모든 줄 이 채워짐)  라인 지우기  한 줄 내리기  return 점수 |

1. 기타

**Cureses 라이브러리**

앞서서 Ncureses(New Cureses)는 Curses의 발전된 버전이라고 언급했다. Curses도 마찬가지로 유닉스의 라이브러리이다. 터미널 환경에서 vim을 실행할 때를 생각해보자. 터미널은 유지되는 상태로 vim이 터미널에 출력된다. 그리고, 기존에 터미널에서 사용하는 커맨드가 아닌 vim 만의 명령어를 입력 받는다. 예를 들어 :qw는 터미널에서는 아무 역할이 없는 입력이지만 vim에서는 저장을 하고 프로그램을 종료한다. 터미널에서 응용 프로그램을 실행하기 위해 텍스트 사용자 인터페이스를 사용한다. 이를 위한 라이브러리가 cureses이다.[3]

1. 참고 문헌
2. Dickey, Thomas E. “What Is NCURSES?” *NCURSES - Frequently Asked Questions (FAQ)*, 11 July 2020, invisible-island.net/ncurses/ncurses.faq.html.
3. 서강대학교 컴퓨터공학과 교수진, “부록 6: ncurses 라이브러리 사용법” *컴퓨터공학 설계 및 실험 I*, 서강대학교 공학부 컴퓨터공학과, 2012
4. Matloff, Norman*. Introduction to the Unix Curses Library*. Department of Computer Science University of California at Davis, 2011.