2020년도 2학기 컴퓨터공학설계및실험Ⅰ

6주차 기본 테트리스 프로그램 예비보고서

20161663 허재성

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 파이브러리, 디버깅 방법, makefile을 만드는 방법 등을 익히도록 한다.

2. 관련 이론

테트리스 프로젝트에서는 ncurses 라이브러리를 사용한다. ncurses (new curses)는 프로그래머가 텍스트 사용자 인터페이스(TUI)를 터미널에 독립적인 방식으로 기록할 수 있도록 API를 제공하는 프로그래밍 라이브러리이다. 단말 에뮬레티어에서 실행하는 GUI와 같은 응용 소프트웨어(Application Software)를 개발하는데 사용되는 툴킷으로 원격 셸(shell)을 사용할 때 겪을 수 있는 레이턴시(Latency, 지연시간)를 줄이기 위한 화면의 변화도 최적화한다.[1]

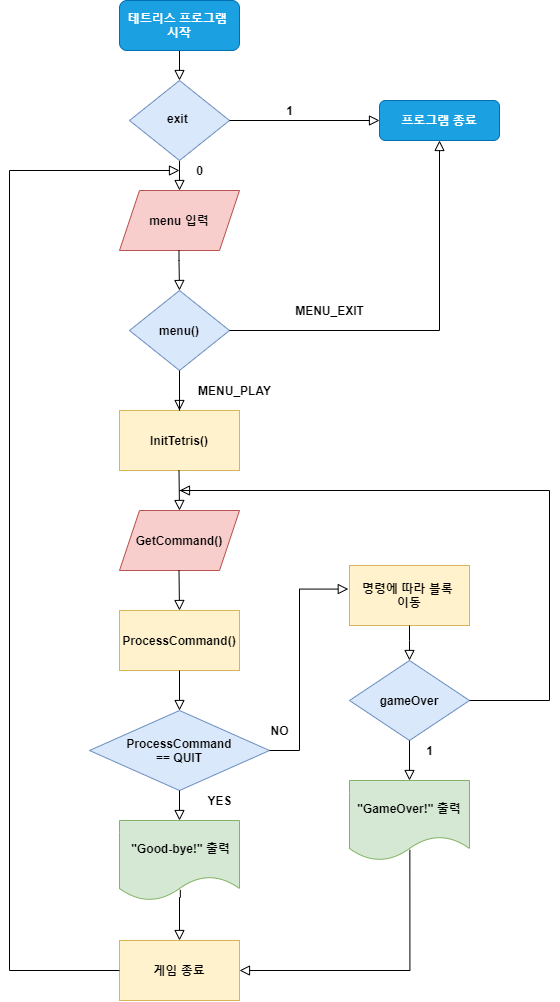
단말 에뮬레이터는 터미널 에뮬레이터(terminal emulator)라곧 하며 몇 가지 다른 디스플레이 구조를 갖춘 덤(dumb) 비디오 단말기를 가상으로 구현하는 프로그램을 말한다.[2] 단말 에뮬레이터의 대표적인 예로 리눅스의 콘솔과 macOS의 터미널이 있다.

텍스트 사용자 인터페이스(TUI)는 명령 줄 사용자 인터페이스(CLI)와 구분되는 인터페이스로 그래픽 사용자 인터페이스(GUI)처럼 화면 전반을 사용하며 마우스와 다른 입력을 수락할 수 있다는 점에서 CLI와 차이가 있다. 색을 사용할 수도 있으며 괘선 문자인 ┌, 등의 특수한 그래프 문자를 사용하여 화면을 구성할 수도 있다.[3]

위의 내용을 종합해 보면 프로그래머가 ncurses 라이브러리를 사용해 CLI 터미널(실습에서 사용하는 리눅스 터미널)에서 TUI처럼 기록할 수 있도록 한다. 이를 통해 테트리스 프로그램과 같이 GUI 요소가 가미된 프로그램을 단말 에뮬레이터에서 실행할 수 있다. 간단한 예로 앞으로 개발할 테트리스 프로그램의 필드(field)의 외곽선을 그리는데 ncurses 라이브러리에서 제공하는 addch 함수를 사용하여 ncurses에서 제공하는 ┌, ┐, └ , ┘ 등의 문자를 커서 위치에 출력할 수 있다. CLI 환경에서는 사용할 수 없는 문자들로 ncurses 라이브러리를 포함하여 사용 가능하다. 또 다른 예로 ncurse 라이브러리에서 제공하는 move 함수로 커서를 이동시키고 해당 커서에서 원하는 작업을 수행할 수 있다. 전달 인자로 이동할 위치의 y좌표와 x좌표를 y, x 순으로 입력한다.

3. 실습 방법

구현할 테트리스 프로그램의 순서도(Flow Chart)는 다음과 같다.



구현되어 있는 함수는 다음과 같다.

(1) void InitTetris()

프로그램이 시작되고 메뉴에서 MENU\_PLAY를 선택하면 호출되는 play 함수에서 InitTetris 함수가 호출된다. InitTetris 함수에서는 테트리스 게임 관련 사항을 초기화한다. 테트리스 게임을 플레이할 수 있게 테트리스 필드를 0으로 초기화하여 필드를 그릴 때 필드가 .으로 채워져 있도록 한다. 그리고 게임 시작 후 처음 나타나는 현재 블록과 다음 블록을 rand() 함수를 호출한 후 7의 나머지로, 임의의 블록을 구한다. 현재 블록의 회전 상태를 0으로, 초기 위치 (y, x)를 필드의 최상단 가운데인 (-1, WIDTH/2-2)로 설정하고 score를 0으로 초기화한다. 게임의 종료 여부를 나타내는 변수 gameOver와 BlockDown 함수에서 사용되는 시간 초과를 나타내는 변수 timed\_out를 false인 0으로 초기화한다. 테트리스 필드와 다음 블록 칸(NEXT BLOCK), 점수판(SCORE)의 윤곽을 DrawOutline 함수를 호출해 그리고, DrawField 함수를 호출해 필드를 그린 후, DrawBlock 함수와 DrawNextBlock 함수를 호출해 각각 필드에 현재 블록을, 다음 블록 칸에 다음 블록을 그린다. PrintScore 함수를 호출해 초기화된 점수 0을 출력한다.

Input: 없음.

Output: 없음

(2) void DrawOutline()

테트리스 게임에서 실제로 게임이 진행되는 필드(field), 다음 블록을 보여주는 NEXT BLOCK, 점수를 보여주는 SCORE 상자의 테두리를 그려주는 함수이다. 테두리 선을 그리기 위해서 따로 정의한 DrawBox 함수를 호출한다. NEXT BLOCK 상자와, SCORE 상자를 그릴 때 ncurses에서 제공하는 move 함수를 이용해 커서를 원하는 위치에 옮긴 후, 커서의 위치에 printw 함수를 이용해 문자열 “NEXT BLOCK”과 “SCORE”을 출력한다.

Input: 없음

Output: 없음

(3) int GetCommand()

ncurses에서 제공하는 함수인 wgetch를 이용해 사용자가 키보드로 명령을 입력하길 기다리다가 입력한 명령을 command에 저장한다. command에 따라 입력을 구분하여 입력받은 명령을 구분한다. 유효한 명령은 총 다섯가지로 방향키 상 키를 입력한 KEY\_UP, 하 키를 입력한 KEY\_DOWN, 왼쪽 키를 입력한 KEY\_LEFT, 오른쪽 키를 입력한 KEY\_RIGHT, 대소문자 구분 없이 ‘Q’를 입력한 QUIT이 있다. 유효하지 않은 명령은 NOTHING으로 처리한다. 구분한 명령을 반환한다.

Input: 없음.

Output: 사용자가 입력한 명령을 저장한 command 변수

(4) int ProcessCommand(int command)

GetCommand에서 입력 받은 command를 입력으로 받아서 command에 따라 적절한 동적을 하는 함수이다. 지역변수 ret는 1로, drawFlag는 false인 0으로 초기화되어 있다. command가 QUIT일 경우, ret는 QUIT으로 값을 변경하고, 그 외의 경우에는 1을 유지한다. 입력 command의 값에 따라 동작이 구분되는데, command가 QUIT일 경우 ret 값을 1에서 헤더 파일 tetris.h에 정의된 QUIT으로 변경 후 반환한다. Command가 QUIT이 아닐 경우, KEY\_UP, KEY\_DOWN, KEY\_RIGHT, KEY\_LEFT에 따라 해당 명령대로 블록을 움직일 수 있는지 체크한다. 체크하는 데에는 따로 정의된 CheckToMove 함수를 호출하여 체크한다. KEY\_UP일 경우, 현재 블록이 회전 가능한 지 체크 후, 회전이 가능하면 전역 변수 blockRotate 값을 갱신한다. KEY\_DOWN일 경우, 현재 블록이 한 칸 내려갈 수 있는지 확인 후, 가능할 경우 blockY 값을 1 증가시킨다. KEY\_RIGHT의 경우, 현재 블록이 오른쪽으로 한 칸 이동할 수 있는 지 체크 후, blockX 값을 1 증가시키고, 반대로 KEY\_LEFT의 경우 왼쪽으로 이동할 수 있는 지 체크 후, 1을 감소시킨다. 체크 결과 블록의 회전 및 이동이 가능하면 CheckToMove 함수는 1을 반환하고, 불가능하면 0을 반환하며 그 값은 drawFlag에 저장된다. drawFlag가 1일 경우 현재 블록의 회전 및 이동, 즉 변경이 발생했다는 의미이므로 따로 정의된 함수 DrawChange를 호출하여 해당 변경사항을 반영하여 현재 블록을 다시 그리고 ret 값을 반환한다.

Input: int command – GetCommand() 함수에서 입력받은 명령을 나타내는 변수

Output: int형 변수 ret로 초기값은 1이며, 입력받은 command가 QUIT일 경우에는 QUIT 반환, 그렇지 않을 경우 1 반환

(5) void DrawField()

테트리스 게임에서 현재 블록이 움직이는 테트리스 필드를 그리는 함수이다. 이중 for문을 이용하여 구현되어 있다. 실제로 필드를 그리는 함수는 ncurses에서 제공하는 함수인 printw 함수로 해당 함수를 이용하기 위해 move 함수로 각 행의 첫번째 위치로 이동한다. move로 이동할 때에는 filed가 화면에서는 (1,1)부터 시작하므로 그에 맞게 작성되어 있다. 이중 for문으로 2차원 배열 filed[HEIGHT][WIDTH]의 각 원소를 확인하여, 원소가 0일 경우, 블록이 없음을 의미하므로 printw 함수로 “.”을 출력하고, 1일 경우, 블록을 의미하는 “ “(공백)을 출력한다. 공백을 그냥 출력하면 보이지 않으므로, attron(A\_REVERSE) 함수를 호출하여 색 반전 옵션을 킨 후, 공백을 출력하여 공백이 하얗게 출력되어 보일 수 있게 한 후, 출력을 마치면 attroff(A\_REVERSE) 함수를 호출하여 색 반전 옵션을 끈다.

Input: 없음

Output: 없음

(6) void PrintScore(int score)

점수판인 SCORE 상자에 현재 점수를 출력하는 함수이다. 출력에는 printw 함수를 이용하며, move 함수로 상자 안으로 커서를 이동시킨 후 입력으로 받은 전역 변수인 현재 점수 score를 출력한다.

Input: 없음

Output: 없음

(7) void DrawNextBlock(int \*nextBlock)

입력 매개변수로 전역 변수로 선언한 배열 nextBlock에 접근해 다음 블록의 정보를 얻는다. nextBlock[1]에 저장된 값이 다음 블록의 모양을 나타내는 번호이다. 다음 블록의 정보를 알아낸 후, 이중 for문을 이용해 NEXT BLOCK 상자에 4\*4 크기의 다음 블록을 그린다. move 함수로 NEXT BLOCK 상자 안으로 이동하여 printw 함수로 출력한다. 이중 for문으로 다음 블록의 4\*4 정보를 확인하여 공백인 부분은 printw 함수로 공백 문자 “ “를 색 반전 없이 출력하고, 블록인 부분은 DrawField 함수와 같이 attron, attroff 함수를 이용하여 색 반전하여 공백 문자를 출력한다.

Input: 블록 정보를 가지고 있는 전역적으로 선언된 1차원 정수형 배열 nextBlock

Output: 없음.

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

필드에 현재 블록을 그리는 함수이다. 2차원 배열 field의 (y, x) 인덱스부터 시작하여 블록을 그린다. 2차원 배열은 왼쪽 상단의 인덱스가 [0][0]이지만, field는 화면 상에서 왼쪽 상단이 (1, 1)이므로 이를 감안하여 move 함수로 커서를 이동시킨 후, 입력받은 char형 변수 tile을 색 반전하여 출력하여 블록을 채운다. 출력하기 전에 블록의 칸이 1이고, 해당 블록 칸에 대응하는 필드가 범위를 벗어나지 않는지 확인 후, 출력하여 그린다.

Input: 2차원 배열에 그릴 위치 인덱스 y, x, 현재 블록의 번호 blockID, 현재 블록의 회전 상태 blockRotate, 블록을 채우기 위해 출력할 문자 tile

Output 없음.

(9) void DrawBox(int y, int x, int height, int width)

왼쪽 상단의 좌표가 (y, x)이고 높이가 height, 너비가 width인 상자를 그리는 함수이다. ncurses에서 제공하는 함수 addch를 이용해 테두리를 표현하는 데 쓰이는 문자를 출력한다. addch 함수의 입력은 ncurses에서 정의된 character인 chtype으로 DrawBox 함수에서는 ACS\_ULCORNER, ACS\_HLINE, ACS\_URCORNER, ACS\_VLINE, ACS\_LLCORNER, ACS\_LRCORNER 문자를 사용한다.

Input: 그리려는 상자의 왼쪽 상단 좌표 (y, x)의 y, x, 상자의 높이 height, 너비 ,width

Output: 없음

(10) void play()

테트리스 게임을 플레이하기 위해 기본적인 정보를 초기화하고 테트리스 게임 플레이의 전반적인 과정을 제어하는 함수이다. do-while 반복문을 돌며 gameOver가 1이 될 때까지 GetCommand 함수를 호출해 명령을 입력받아 command에 저장한다. ProcessCommand 함수에 command를 전다하여 반환값이 QUIT일 경우, 강제 종료를 알리는 상자와 문장 “Good-bye!!”를 띄운다. 그 상태에서 사용자가 아무 키나 누르면 함수가 종료된다. ProcessCommand 함수가 command를 입력받아 명령을 처리하고 gameOver가 될 경우, 게임 오버를 알리는 상자와 문장 “GameOver!!”를 출력 후 종료한다.

Input: 없음

Output: 없음

(11) char menu()

프로그램이 시작될 시 출력되는 메뉴 화면으로 사용자가 선택할 수 있는 메뉴가 해당 번호와 함께 출력된다. 사용자가 해당하는 번호를 누를 때까지 기다렸다가 사용자가 입력한 문자를 반환한다.

Input: 없음

Output: 사용자가 입력한 문자

4. 기타

ncurses 라이브러리를 사용하기 위해서는 헤더 파일 ncurses.h 파일을 포함해야 한다.

|  |
| --- |
| #include <ncurses.h> |

컴파일 시 ncurses 라이브러리를 링크하기 위해서는 gcc로 컴파일할 때, 다음과 같이 -l 옵션을 줘야 한다.

|  |
| --- |
| gcc -lncurses prog.c |

일반적인 C 프로그램에서의 출력 함수인 printf()와 같은 함수는 화면에 출력하기 위해 stdout을 사용하지만, ncurses 라이브러리를 이용하여 화면에 출력할 때에는 ncurses에서 제공하는 stdscr이라는 스크린에 출력한다.[4]

테트리스 프로그램의 main 함수에서 호출되는 ncurses 라이브러리 함수는 다음과 같다.

|  |
| --- |
| (1) WINDOW\* initscr(void)  ncurses의 자료구조를 초기화하는 함수로 ncurses를 사용하기 위해 반드시 호출해야 하는 함수이다. main 함수에서 호출한 후, ncurses에서 제공하는 다른 함수들을 사용한다.  (2) int endwin(void)  initscr 함수를 호출하기 전의 상태로 모드를 바꾸어 주고 커서를 좌측 하단으로 이동시킨다. 프로그램에서 ncurses를 종료하기 전에는 endwin 함수를 호출해야 한다.  (3) int echo(void)  사용자에 의해 타이핑된 입력이 화면에 출력되지 않도록 하는 함수이다. 테트리스 프로그램에서 사용자가 입력한 명령키가 화면에 보이지 않게 하기 위해 호출한다.  (4) int keypad(WINDOW\* win, int bf)  입력 파라미터인 bf가 TRUE이면, 윈도우 win에서 입력 받을 시, stdin에서 제공하지 않는 키패드와 특수문자의 입력이 가능해진다. 테트리스 게임에서 방향키 입력이 가능해지기 위해 TRUE를 전달한다. 테트리스 프로그램에서는 표준 스크린인 stdscr을 인자로 넣어준다.  (5) int clear(void)  화면에 그려져 있는 모든 내용을 지워준다. 테트리스 게임이 종료되고 메뉴로 넘어갈 때, 테트리스 필드와 다음 블록 상자, 점수 상자를 지우기 위해 호출한다. |

기타 함수에서 호출되는 ncurses 라이브러리 함수는 다음과 같다.

|  |
| --- |
| (1) int move(int y, int x)  커서를 윈도우 스크린의 좌표 (y,x)로 이동시킨다. 스크린에 출력을 원하는 위치에 하기 위해서 move 함수로 해당 위치로 커서를 이동시킨 후 출력한다.  (2) printw(WINDOW\* win, const char\* format [ , argument, … ] )  format 매개변수로 서식화된 일련의 argument를 ncurses에서 출력 시 사용되는 screen인 stdscr 상에서 출력하는 함수로서 출력 형식은 printf()와 같다. 스크린의 원하는 위치에 출력하기 위해 위의 move 함수를 호출하여 커서를 원하는 위치에 이동시킨 후, printw 함수를 이용해 출력한다.  (3) int attron(int attr) / int attroff(int attr)  attron, attroff 사이에 출력되는 문자의 색깔을 attr 모드로 출력한다. 테트리스 프로그램에서는 블록을 표현하기 위해 공백 문자를 색 반전하여 출력하는데 이를 위해 attron 함수와 attroff 함수에 A\_REVERSE 옵션을 줘 색을 반전시키고 attron, attroff 사이에 printw 함수로 색 반전된 공백 문자를 출력한다.  (4) int addch(chtype ch)  문자 ch를 스크린의 현재 커서 위치에 출력하는 함수로 chtype은 ncurses에서 정의된 character이다. 테트리스 게임에서 상자의 윤곽선을 그리는데 사용된다.  (5) char wgetch(WINDOW\* win)  윈도우 win에서 키보드 입력을 받아 입력받은 character를 반환한다.  (6) char getch(void)  키보드 입력을 받아 입력받은 character를 반환한다. |

5. 참고 문헌

[1] 위키백과, “ncurses”, <https://ko.wikipedia.org/wiki/Ncurses>

[2] 위키백과 “단말 에뮬레이터”

<https://ko.wikipedia.org/wiki/%EB%8B%A8%EB%A7%90_%EC%97%90%EB%AE%AC%EB%A0%88%EC%9D%B4%ED%84%B0>

[3] 위키백과 “텍스트 사용자 인터페이스”

<https://ko.wikipedia.org/wiki/%ED%85%8D%EC%8A%A4%ED%8A%B8_%EC%82%AC%EC%9A%A9%EC%9E%90_%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4>

[4] 서강대학교 컴퓨터 공학과 교수진, “컴퓨터공학 설계 및 실험 I”, *서강대학교 공학부 컴퓨터공학과*, 2012