8주차 예비보고서

전공: 신문방송학과 학년: 3학년 학번: 20191150 이름: 전현길

1. 테트리스 frame 프로그램 파일을 미리 읽어보고(부록 1 ncurses 라이브러리 포함), 테트리스 게임의 flow chart를 자세히 작성하시오. 그리고 테트리스 게임을 구성하는 각 함수의 기능에 대해서 설명하시오.

int CheckToMove (char f[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX)
블록이 움직일 수 있는지 확인한다. 매개변수로 현재 필드의 상태, 현재 블록의 모양, 회전수, 블록이 이동하게 될 y/x좌표 또는 현재 y/x좌표를 입력받는다. 이동이 가능하면 1, 불가능하면 0을 반환한다. 필드에 쌓인 블록과 겹치거나 필드 밖으로 빠져나갈 경우 이동이 불가능하다.

void DrawChange (char f[HEIGHT][WIDTH], int command, int currentBlock, int blockRotate, int blockY, int blockX)
user의 명령어 입력으로 변경된 블록을 다시 그린다. 1) 블록이 움직이기 전의 정보를 입력받은 뒤, 해당 위치에 있는 블록을 삭제한다. 2) 명령어와 현재 위치 정보를 통해 새로운 블록의 정보를 계산한다. 3) 새로운 블록을 DrawBlock() 함수를 호출해 표시한다. 4) move() 함수로 커서를 필드 밖으로 초기화한다.

void BlockDown(int sig)
블록이 단위 시간(=1초)마다 한 칸씩 내려가게 한다. 1) CheckToMove() 함수를 불러와 블록을 한 칸 내릴 수 있는지 살펴본다. 2) 내릴 수 있다면 한 칸 내리고 함수를 종료한다, 내려갈 수 없다면 블록의 y좌표를 확인한다. 3-1) 블록의 y좌표가 -1이라면, gameover flag를 set하고 함수를 종료, 프로그램을 종료한다. 3-2) 블록의 y좌표가 -1이 아니라면, 블록을 field에 쌓는다. 4) field의 한 줄이 다 채워졌다면 DeleteLine()으로 삭제한 뒤 점수를 갱신, PrintScore()로 출력한다. 5) 다음 블록의 정보를 현재 블록으로 불러온 뒤, DrawNextBlock()으로 다음 블록을 생성한다. 6) 현재 블록을 초기화하고, DrawField()로 동작을 종료한다.

void AddBlockToField (char f[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX)
블록이 내려갈 수 없을 때 필드에 합친다. CheckToMove() 함수에서 이미 이동 가능 여부를 확인했으므로, 추가적으로 확인할 필요 없이 블록을 field에 추가한다. currentBlock[i][j] == 1일 때 field[blockY + i][blockX + j]를 1로 변경한다.

int DeleteLine(char f[HEIGHT][WIDTH])
필드에서 완전히 채워진 라인을 찾은 뒤 삭제하고, 점수를 계산해 반환한다. 1) 필드의 위에서부터 아래로 이중 loop 탐색하여 완전히 채워진 line을 찾는다. 2) 완전히 채워진 line이 있다면, 지워진 라인 수를 증가시키고, 필드에서 라인을 삭제한 뒤 위의 라인을 모두 아래로 내린다. 3) 필드의 끝까지 전부 탐색했다면 지워진 라인 수를 통해 점수를 계산해 반환한다.

char menu()
게임 화면에 메뉴를 출력한다.

void play()
테트리스 게임을 시작한다.

void DrawField()
게임이 진행되기 위한 field를 그린다.

void DrawNextBlock(int *nextBlock)
다음으로 나타날 테트리스 블록을 그린다.

void PrintScore(int score)
화면에 현재 점수를 출력한다.

void DrawOutline()
게임 화면 내 인터페이스 박스를 출력한다.

int GetCommand()
사용자 입력을 명령어로 받아들여 반환한다.

int ProcessCommand(int command)
command를 받아들여 적절히 처리한다.

void DrawBox(int y, int x, int height, int width)
화면의 (y, x) 좌표에 입력된 크기의 박스를 그린다.

void DrawBlock(int y, int x, int blockID, int blockRotate, char tile)
(y, x) 좌표에 블록을 char tile을 이용해 그린다.

void InitTetris()
변수를 초기화하고 게임의 최초 화면(UI, 필드, 현재/다음 블록, 점수)을 출력한다.

void gotoyx(int y, int x)
커서를 입력된 (y, x) 좌표로 이동시킨다. ncurses 라이브러리 move() 함수를 이용한다.

void createRankList()
rank file을 입력받아 랭킹 목록을 구성한다.

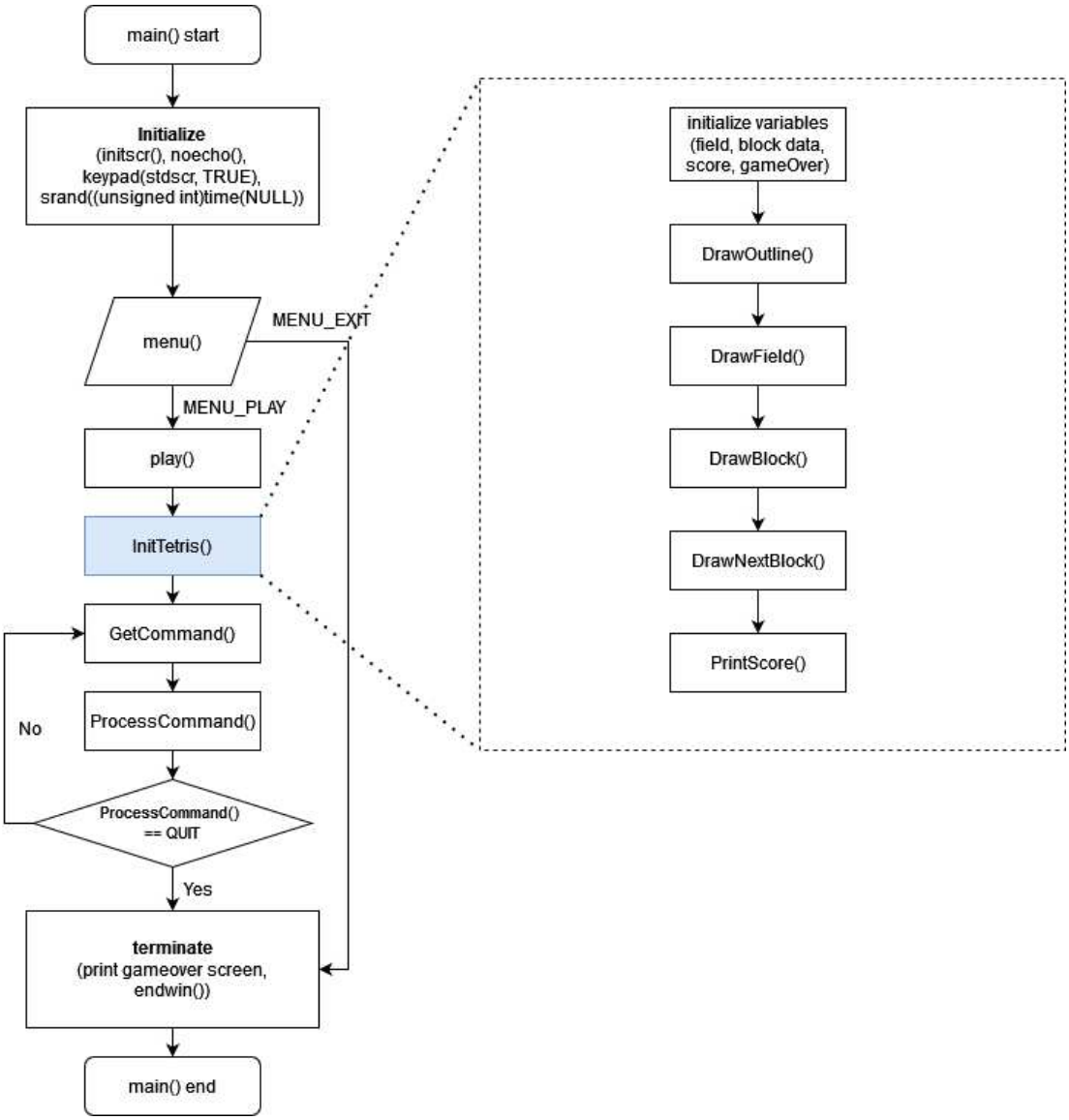
void rank()
화면에 랭킹 목록을 출력한다.

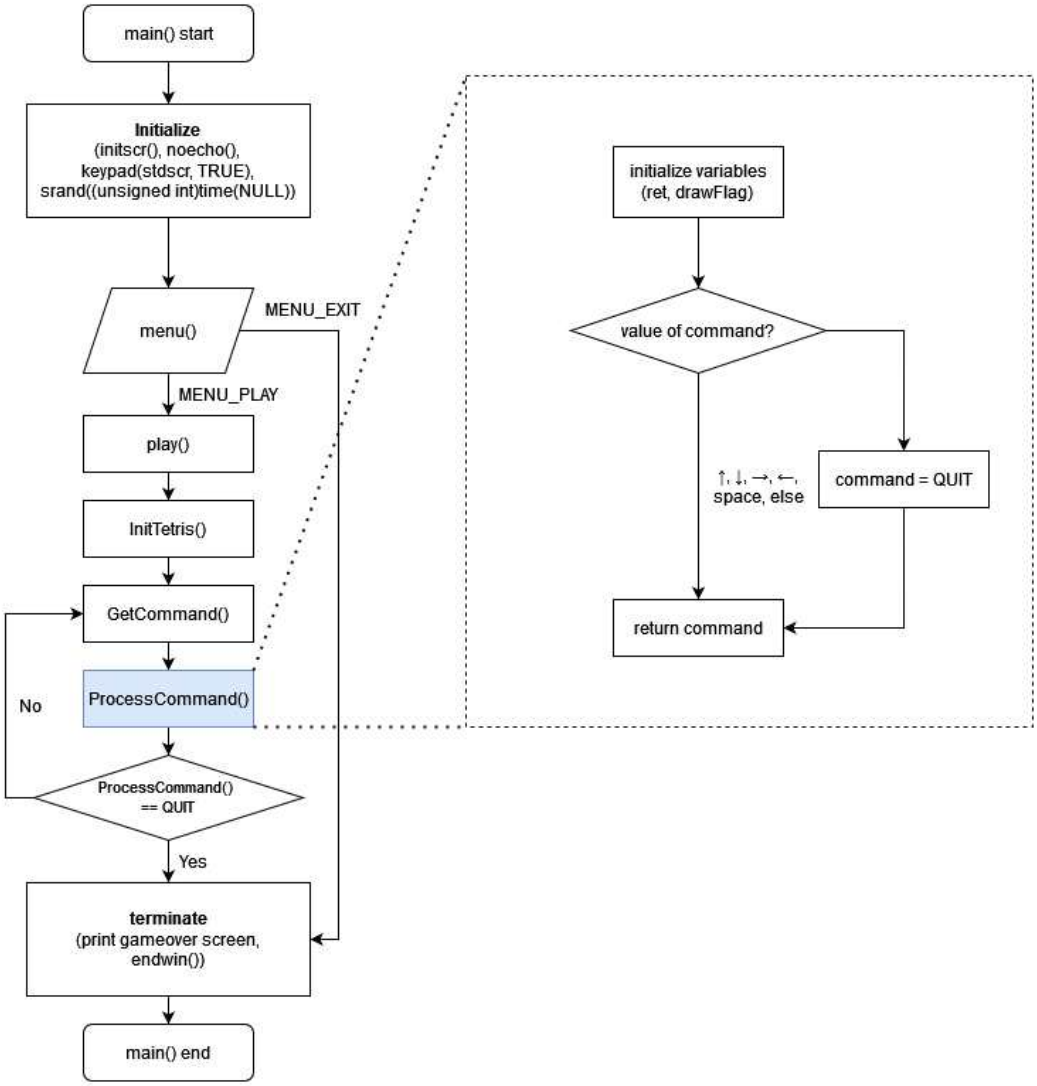
void writeRankFile()
rank file을 생성한다.

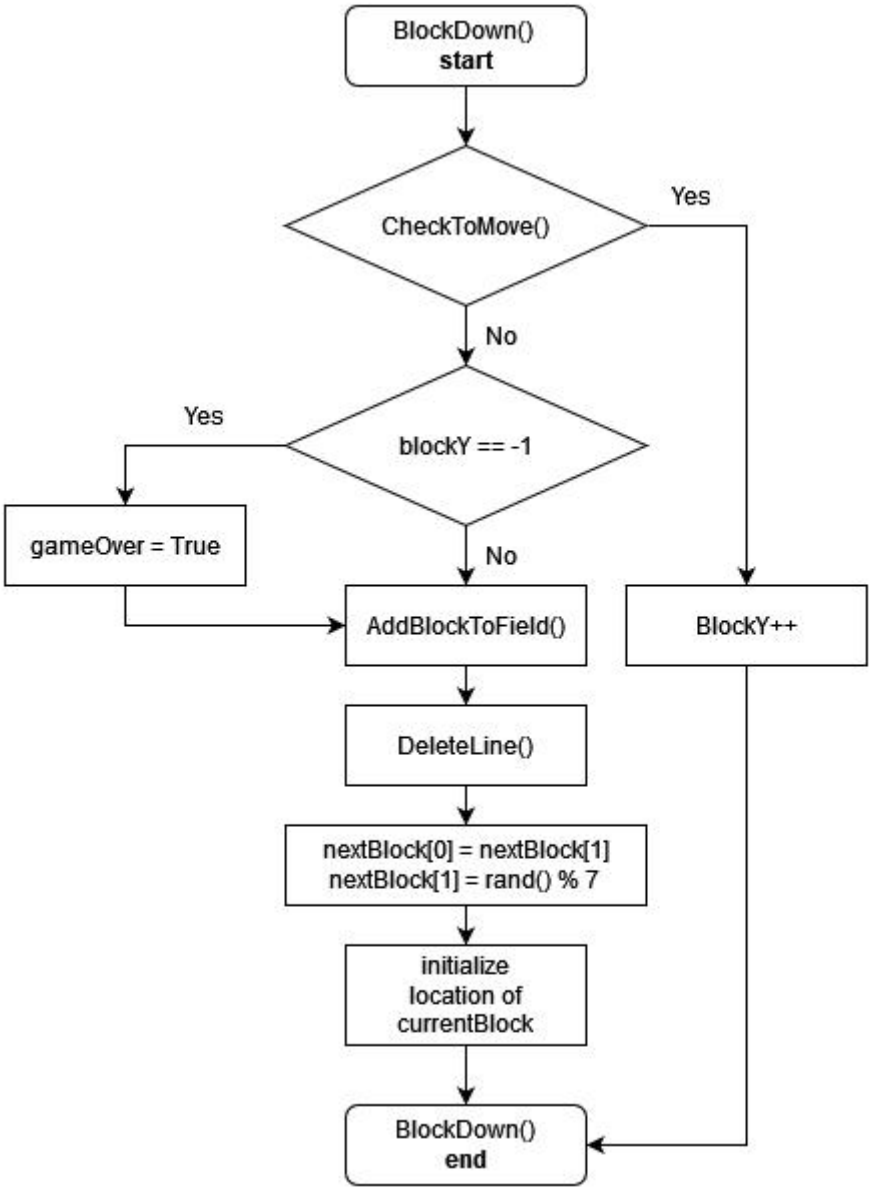
void newRank()
rank file에 새로운 랭킹 정보를 쓴다.

int recommend(RecNode* root)
추천 트리의 root를 입력받아 추천에 따를 때 얻을 수 있는 예상 점수를 반환한다.

void recommendPlay()
테트리스 게임의 recommendPlay 모드를 실행한다.


main() start
Initialize
(initscr(), noecho(),
keypad(stdscr, TRUE),
srand((unsigned int)time(NULL))
menu()
MENU_EXIT
MENU_PLAY
play()
InitTetris()
GetCommand()
ProcessCommand()
No
ProcessCommand()
== QUIT
Yes
terminate
(print gameover screen,
endwin())
main() end
initialize variables
(field, block data,
score, gameOver)
DrawOutline()
DrawField()
DrawBlock()
DrawNextBlock()
PrintScore()


main() start
Initialize
(initscr(), noecho(),
keypad(stdscr, TRUE),
srand((unsigned int)time(NULL))
menu()
MENU_EXIT
MENU_PLAY
play()
InitTetris()
GetCommand()
ProcessCommand()
No
ProcessCommand()
== QUIT
Yes
terminate
(print gameover screen,
endwin())
main() end
initialize variables
(ret, drawFlag)
value of command?
↑, ↓, →, ←,
space, else
command = QUIT
return command



2. 실습시간에 구현할 5가지 함수들에 대한 간단한 pseudo code를 제시하시오.

int CheckToMove
(char f[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX)

```
for i = 0 to 3
  for j = 0 to 3
    if not (0 ≤ blockY + i < HEIGHT and 0 ≤ blockX + j < WIDTH)
      return FALSE;
    if (f[blockY + i][blockX + j] == 1)
      return FALSE;
return TRUE;
```

void DrawChange
(char f[HEIGHT][WIDTH], int command, int currentBlock, int blockRotate, int blockY, int blockX)

2중 for문을 통해 현재 블록의 이전 shape 정보, rotate 정보를 통해 확인했을 때 블록이 존재한다면 필드에 존재하는 블록이므로 삭제한다. 단, 이전 블록의 존재 여부를 필드 밖에서 찾을 수도 있으므로 해당 경우를 미리 제외한다.

```
for i = 0 to 3
  for j = 0 to 3
    if not (0 ≤ blockY + i < HEIGHT and 0 ≤ blockX + j < WIDTH)
      continue;
    if (block[currentBlock][blockRotate][i][j] == 1)
      f[blockY + i][blockX + j] = 0

DrawField()

if (command == KEY_UP)
  blockRotate = (blockRotate + 1) % 4
if (command == KEY_DOWN) blockY--
if (command == KEY_LEFT) blockX--
if (command == KEY_RIGHT) blockX++

DrawBlock(blockY, blockX, currentBlock, blockRotate, ' ')
```

void BlockDown(int sig)

```
if (CheckToMove(f, currentBlock, blockRotate, blockY, blockX) == TRUE)
  blockY++;
  DrawChange(f, KEY_DOWN, currentBlock, blockRotate, blockY, blockX)
else
  if (blockY == 1)
    gameOver = TRUE
  else
    AddBlockToField(f, currentBlock, blockRotate, blockY, blockX)
    score += DeleteLine(f)

    nextBlock[0] = nextBlock[1]
    nextBlock[1] = rand() % 7
    blockRotate = 0
    blockY = -1
    blockX = WIDTH / 2 - 2

    DrawNextBlock(nextBlock)
    PrintScore(score)
    DrawField()
```

void AddBlockToField
(char f[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX)

DrawChange()와 동일하게 블록이 필드 밖으로 나갈 경우를 미리 제외한다.

```
for i = 0 to 3
  for j = 0 to 3
    if not (0 ≤ blockY + i < HEIGHT and 0 ≤ blockX + j < WIDTH)
      continue;
    if (block[currentBlock][blockRotate[i][j]] == 1)
      f[blockY + i][blockX + j] = 1
```

int DeleteLine(char f[HEIGHT][WIDTH])

가득 찬 라인이 있을 경우, 해당 라인을 삭제하고 삭제한 라인의 위에 있는 라인을 모두 한 칸씩 내린다. 단, 가장 윗 줄의 경우 그보다 윗 줄이 없으므로 빈 줄로 따로 채워 주어야 한다.

```
del_line = 0
for i = 0 to HEIGHT - 1
  full_line = TRUE
  for j = 0 to WIDTH - 1
    if (f[i][j] == 0)
      full_line = FALSE
      break
  if (full_line == TRUE)
    del_line++
    for p = i downto 0
      for q = 0 to WIDTH - 1
        f[p][q] = f[p - 1][q]
    for q = 0 to WIDTH - 1
      f[0][q] = 0;
    i--;
return del_line * del_line * 100
```