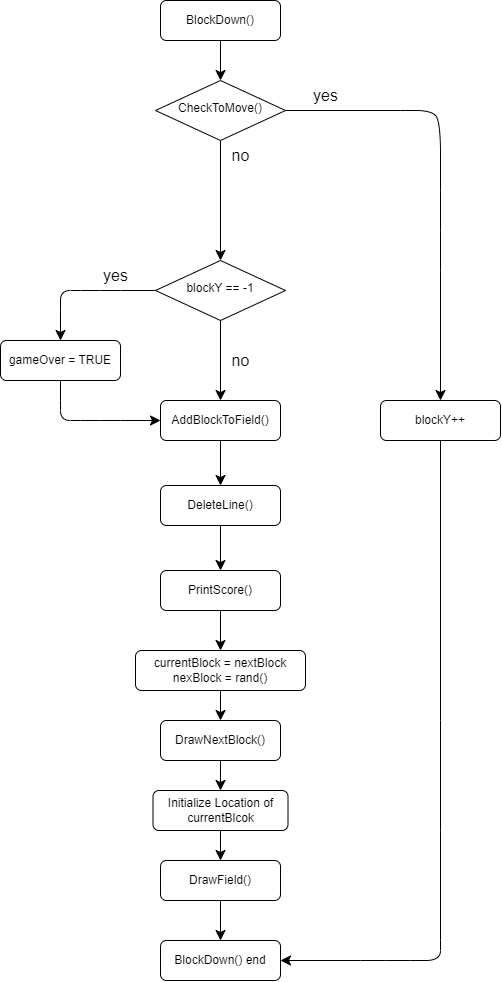
8주차 예비보고서

전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현

**1.**

도표이(가) 표시된 사진

자동 생성된 설명

menu()함수는 화면에 1~4번까지의 메뉴를 출력하고, 사용자가 번호를 입력하면 그 값을 반환한다. InitTetris() 함수는 게임을 수행하기 위해 변수와 자료구조, 그리고 화면을 초기화한다. DrawOutline() 함수는 필드의 테두리, 다음 블록을 그리는 상자의 테두리, 그리고 점수 상자의 테두리를 그린다. DrawNextBlock() 함수는 다음 블록을 그리는 상자 안에 4x4 크기의 다음 블록을 그린다. DrawField() 함수는 필드 부분을 화면에 그리는 함수이다. PrintScore() 함수는 점수를 화면에 출력한다. DrawBlock() 함수는 블록의 좌표, 모양, 회전수를 입력받아 해당 블록을 그린다. InitTetris() 함수에서 DrawOutline(), DrawField(), DrawBlock(), DrawNextBlock(), PrintScore() 함수는 호출한다. GetCommand() 함수는 사용자의 입력을 구분하여 입력 받은 명령을 반환한다. 이 때 입력은 key\_up, key\_down, key\_left, key\_right, q로 총 5가지이다. processCommand() 함수는 GetCommand() 함수의 반환값을 input으로 받아서 그에 해당하는 명령을 수행하는 함수이다. input이 q이면 게임을 종료하고, key\_up이면 시계 반대방향으로 블록을 회전시킬 수 있는지 CheckToMove() 함수를 통해 확인한다. key\_down이면 CheckToMove() 함수를 이용해서 블록을 아래로 이동할 수 있는지 확인한다. key\_left와 key\_right는 각각 블록이 왼쪽이나 오른쪽으로 이동할 수 있는지 체크한다. CheckToMove() 함수는 블록이 예정된 위치로 이동할 수 있는지 체크하는 함수이다. DrawChange() 함수는 위치, 회전수 등 변경된 정보를 통해 블록을 다시 그리는 함수이다. BlockDown() 함수는 현재의 블록을 한 칸씩 아래로 이동시키는 함수이다. CheckToMove() 함수를 통해 아래로 이동 가능한지 여부를 판단하고, 이동이 불가능하다면 블록의 y좌표가 -1인지 확인하고 -1이라면 gameOver을 true로 하고, 아니라면 AddBlockToField() 함수를 실행한다. AddBlockToField() 함수는 필드의 위치에 현재 블록을 쌓는 함수이다. DeleteLine() 함수는 완전히 채워진 라인이 있으면 그 라인을 지우고, 점수를 계산하여 반환하는 함수이다.

**2.**

**CheckToMove()**

for i=0 to 4, i++

for j=0 to 4, j++

if block[currentBlock][blockRotate][i][j]==1 then

if blockX+j >= WIDTH-1 or blockY+I >= HEIGHT-1 or blockX+j <= 0

then return 0

else if field[i+blockY][j+blockX]==1 then return 0

return 1

DrawChange()

switch command

case QUIT

break

case KEY\_UP

for j=0 to 4, j++

for i=0 to 4, i++

if field[j+blockY][i+blockX] == 1 then

field[j + blockY][i + blockX] = 0

DrawField();

blockRotate = (blockRotate + 1) % 4;

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

break

case KEY\_DOWN

for j=0 to 4, j++

for i=0 to 4, i++

if field[j+blockY][i+blockX] == 1 then

field[j + blockY][i + blockX] = 0;

DrawField();

blockY++;

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

break

case KEY\_RIGHT

for j=0 to 4, j++

for i=0 to 4, i++

if field[j+blockY][i+blockX] == 1 then

field[j + blockY][i + blockX] = 0;

DrawField();

blockX++;

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

break

case KEY\_LEFT

for j=0 to 4, j++

for i=0 to 4, i++

if field[j+blockY][i+blockX] == 1 then

field[j + blockY][i + blockX] = 0;

DrawField();

blockX--;

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

break

BlockDown()

if CheckToMove() == 1 then blockY++

else

if blockY == -1 then gameOver = 1

AddBlockToField()

DeleteLine()

nextBlock[0] = nexBlock[1]

nextBlock[1] = rand()%7

blockRotate=0, blockY=1 blockX=WIDTH/2-2

DrawNextBlock()

PrintScore()

DrawField()

AddBlockToField()

for i=0 to 4, i++

for j=0 to 4, j++

if block[currentBlock][blockRotate][i][j] == 1

then field[blockY+i][blockX+j] = 1

DeleteLine()

int flag, line

for i=0 to HEIGHT

flag = 0

for j=0 to WIDTH

if field[i][j] == 0 then break

flag++

if flag==WIDTH

then line++

for p=I to 1, p--

for q=0 to WIDTH, q++

field[p][q] = field[p-1][q]

return line\*line\*100