9주차 결과보고서

전공 : 컴퓨터공학 학년 : 2학년 학번 : 20151616 이름 : 최승환

1. 실습시간에 작성한 프로그램의 함수들이 예비보고서에서 작성한 각 구현 함수들의 pseudo code와 어떻게 달라졌는지 설명하고, 각 함수에 대한 시간 및 공간 복잡도를 보이시오(각 함수의 시간 및 공간복잡도를 구할 때, 어떤 변수에 의존하는지를 판단해아한다).

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

// user code

int xcnt,ycnt;

for(ycnt=0;ycnt<4;++ycnt)

{

for(xcnt=0;xcnt<4;++xcnt)

{

if(block[currentBlock][blockRotate][ycnt][xcnt] == 1)

{

if(f[blockY+ycnt][blockX+xcnt] == 1)

{

return 0;

}

if(blockY+ycnt>HEIGHT-1)

{

return 0;

}

if(blockX+xcnt>WIDTH-1)

{

return 0;

}

if(blockX+xcnt<0)

{

return 0;

}

}

}

}

return 1;

}

pseudo code와 같이 해당 좌표에서 이동 할 수 있는지 판단한다. 모든 필드를 확인하는 것이 아니라 자신의 위치에서만 판단한다. 따로 오른쪽으로 이동 가능한지, 왼쪽으로 이동 가능한지, 아래로 이동 가능한지를 판단하는 것이아니라 해당 좌표가 주어졌을 때 그 좌표로 이동이 가능한지 여부만 체크한다. 따라서 현 좌표에서 block의 좌표가 1인 곳의 위치만 확인하고, 이 좌표에 필드의 좌표값이 1인지만 확인하면 상수시간만에 이동 가능 여부를 확인할 수 있다. 따라서 시간 복잡도는 O(1)이다.

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

// user code

DrawOutline();

DrawField();

DrawBlockWithFeatures(blockY,blockX,currentBlock,blockRotate);

}

입력된 커맨드에 따라 블록의 위치를 변경한다. 그리고 입력과 상관없이 그림자를 구하며, 그림자와 블록의 잔상을 지우며 DrawBlockWithFeatures함수를 호출하여 블록을 새로 그린다. 이 경우 DrawBlock 함수와 DrawShadow 함수를 호출한다. 그림자를 그릴 때에 CheckToMove 함수를 호출하며 아래로 이동이 불가능할 때 까지 호출을 하므로 최악의 경우 시간복잡도가 O(HEIGHT)만큼의 시간이 걸린다. 나머지는 상수 시간이 걸린다. DrawBlock함수도 상수 시간이므로 총 시간 복잡도는 O(HEIGHT)이다.

void BlockDown(int sig){

// user code

if(CheckToMove(field,nextBlock[0],blockRotate,blockY+1,blockX))

{

++blockY;

DrawChange(field,0,nextBlock[0],blockRotate,blockY,blockX);

}

else

{

if(blockY == -1)

{

gameOver=TRUE;

}

else

{

AddBlockToField(field,nextBlock[0],blockRotate,blockY,blockX);

score += DeleteLine(field);

nextBlock[0] = nextBlock[1];

nextBlock[1] = nextBlock[2];

nextBlock[2] = rand()%7;

blockY=-1;

blockX=WIDTH/2-2;

DrawNextBlock(nextBlock);

}

}

PrintScore(score);

timed\_out=0;

}

CheckToMove로 blockY+1, blockX로 이동 가능한지 확인하며 이동이 가능 할 경우 DrawChange를 call하며 그렇지 않은 경우엔 AddBlockToField와 DeleteLine을 call한다.

이 경우 field에 현 블록을 추가해 좌표값을 바꿔준다. 그리고 DeleteLine 함수를 이용해 모두 채워진 줄이 있는지 확인한 후 이를 지운다. AddBlockToField의 시간복잡도는 상수 시간이지만 DeleteLine에서 모든 필드를 탐색하며 블록이 전부 채워진 라인을 구해야 하므로 이때 O(HEIGHT\*WIDTH)만큼의 시간이 걸린다. 나머지 함수들 또한 상수시간이므로 총 시간 복잡도는 O(HEIGHT\*WIDTH)이고 공간 복잡도는 상수이다.

2. 테트리스 프로젝트 1주차 숙제 문제를 해결하기 위한 pseudo code를 기술하고, 작성한 pseudo code의 시간 및 공간 복잡도를 보이시오.

void DrawOutline(){

int i,j;;

DrawField();

DrawBlockWithFeatures(blockY,blockX,nextBlock[0],blockRotate);

/\* 블럭이 떨어지는 공간의 태두리를 그린다.\*/

DrawBox(0,0,HEIGHT,WIDTH);

/\* next block을 보여주는 공간의 태두리를 그린다.\*/

move(2,WIDTH+10);

printw("NEXT BLOCK");

DrawBox(3,WIDTH+10,4,8);

DrawBox(3,WIDTH+23,4,8);

/\* score를 보여주는 공간의 태두리를 그린다.\*/

move(9,WIDTH+10);

printw("SCORE");

DrawBox(10,WIDTH+10,1,8);

}

DrawShadow와 DrawNextBlock을 수정했기 때문에 이를 담을 박스또한 DrawOutline에서 그려줘야 한다.

void DrawShadow(int y, int x, int blockID,int blockRotate){

int cnt;

for(cnt=0;;++cnt)

{

if(!CheckToMove(field,blockID,blockRotate,y+cnt,x))

{

DrawBlock(y+cnt-1,x,blockID,blockRotate,'/');

return;

}

}

// user code

}

void DrawBlockWithFeatures(int y,int x,int blockID,int blockRotate)

{

DrawBlock(y,x,blockID,blockRotate,' ');

DrawShadow(y,x,blockID,blockRotate);

}