실험 PRJ-1 1주차 기본 테트리스 프로그램 결과보고서

전공: 컴퓨터공학과 학년: 2 학번: 20211599 이름: 주현수

1. 이때 모든 시간복잡도는 예비보고서에서 붙인 코드가 아니라 제출한 실습코드를 바탕으로 하였다.

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

(예비보고서)

For I=0 to 3

For j=0 to 3

{if(block[currentBlock][blockRotate][i][j]==1)

{int y=blockY+j;

Int x=blockX+i}

If(y>=HEIGHT||x<0||x>=WIDTH|| f[y][x]==1)return 0;

return 1;

(실습)

실습에서는 3이라는 숫자 대신 BLOCK\_HEIGHT와 BLOCK\_WIDTH라는 고정상수를 사용하였다.

시간복잡도는 if(y>=HEIGHT || x<0 || x>=WIDTH || f[y][ x]==1 || y<0) 구문에 프로그래밍이 닿는 것을 변수로 두면, O(BLOCK\_HEIGHT\*BLOCK\_WIDTH)이다.

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

(예비보고서)

switch(command)

case KEY\_UP:

for i=0 to 3

for j=0 to 3

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

case KEY\_DOWN:

int a =blockY-1

for i=0 to 3

for j=0 to 3

f[a+i][blockX+j]=0

case KEY\_LEFT:

int b =blockX+1

for i=0 to 3

for j=0 to 3

f[blockY+i][b+j]=0

case KEY\_RIGHT:

int c =blockX-1

for i=0 to 3

for j=0 to 3

f[blockY+i][c+j]=0

DrawBlock(blockY,blockX,currentBlock,blockRotate,’ ‘)

(실습)

예비보고서의 수도코드에서는 각각의 경우를 switch로 나누고 각 경우에서 for문을 돌렸는데, 같은 작업을 switch의 경우마다 해야하는 불편함이 있어 x,y변수를 사용해 각각 blockX 와 blockY 의 값을 담고, switch문으로 변수를 조정한 후 마지막 한번 for문을 적었다.

시간복잡도는 if(block[currentBlock][blockRotate][i][j]==1) 구문 반복횟수를 변수로 잡으면 O(BLOCK\_HEIGHT\*BLOCK\_WIDTH)이다.

3) void BlockDown(int sig)

(예비보고서)

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

{blockY++

DrawChange(field,KEY\_DOWN,nextBlock[0],blockRotate,blockY,blockX)}

Else

{if(blockY==-1)

gameover=1;

else

{AddBlockToField(field,nextblock[0],blockRotate,blockRotate,blockY,blockX)

score=DeleteLine(field)

nextBlock[0]=nextBlock[1]

nextBlock[1]=rand()%7

blockY=-1

blockRotate=0

blockX=WIDTH/2-2;

DrawNextBlock(nextBlock)

PrintScore(score)

DrawField();}

(실습)

Timed\_out변수의 초기화를 해주었다. 나머지는 예비보고서에서 작성한 수도코드와 같다.

시간복잡도는 반복문이 없으므로 O(1)이다.

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

(예비보고서)

For i=0 to 3

For j=0 to 3

If(block[currentBlock][blockRotate][i][j]==1)

f[blockY+i][blockX +j]=1

(실습)

예비보고서에서 추가하지 않은 사항 [ if(0<=blockY+i && blockY+i<HEIGHT && 0<=bloc kX+j &&blockX+j<WIDTH) ]을 추가했다. 이때 이 if문이 돌아갈 횟수를 변수로 친다면 시간복잡도는 O(BLOCK\_HEIGHT\*BLOCK\_WIDTH)이다.

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

(예비보고서)

Int a=0

Int n=0;

For i=0 to HEIGHT-1

For j=0 to WIDTH-1

{If(f[i][j]==0)

{a=1;break;}

If(a==0)

{n++

For k=i-1; k>=0;k—

{for m=0;m<WIDTH;m++

{f[k+1][m]=f[k][m]}

Return n\*n\*100

(실습)

수도코드랑 실습코드랑 변수 이름빼고는 차이점이 없었다. 반복문이 돌아간 횟수를 기준으로 본다면 시간복잡도는 O(HEIGHT\*(WIDTH+WIDTH))이다.

2.

1) void DrawShadow

While(CheckToMove(field,blockID,blockRotate,y+1,x){

Y++;}

DrawBlock(y,x,blockID,blockRotate,’/’)

위에서부터 내려오면서 더 이상 움직이지 못할 거리까지 y를 키워주니까 y++이 실행될 최악의 시간복잡도의 경우 O(HEIGHT)가 된다.

2) void DrawBlockWithFeatures

DrawBlock

DrawShadow

두가지 함수를 부르는 함수이니 시간복잡도는 O(1) 이 된다.

3)void InitTeteris

(다른부분은 실습과 똑같이 유지하되, 다음 줄만 추가해준다.)

nextBlock[2]=rand()%7;

field[][]의 값을 초기화해주는 것을 기준으로 시간복잡도는 O(HEIGHT\*WIDTH)이다.

4)void DrawNextBlock

(다른부분은 실습과 똑같이 유지하되, 다음 줄만 추가해준다.)

for i in 0 to 3

move(10+I,WIDTH+13)

for j in 0 to 3

if(block[nextBlock[2]][0][i][j]==1){

attorn(A\_REVERSE)

printw(“ “)

attroff(A\_REVERSE);}

else printw(“ “);

for문 안에 들어있는 if구문이 돌아가는 횟수를 변수로 잡으면 이 시간복잡도는 O(3\*3+3\*3)이다. nextBlock[1]과 nextBlock[2]가 있기 때문에 3\*3이 두번 더해진다.

5) void BlockDown

(다른 부분은 다 똑같이 하고 아래 부분만 추가한다)

Score+=AddBlockToField~

nextBlock[1]=nextBlock[2]

nextBlock[2]=rand()%7

이는 1번에서 설명했듯 O(1)의 시간복잡도를 갖는다.

6) void AddBlockToField

(앞의 코드는 그대로 쓰되, 뒤에 코드를 추가한다.)

If(blockY+i==HEIGHT-1)

Touched++;

Else if(f[blockY+i+1][blockX+i]==1)

Touched++;

.

.

.

Return touched\*10

반복문을 기준으로 시간복잡도를 계산해보면 O(BLOCK\_HEIGHT\*BLOCK\_WIDTH)