8주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20202106 이름: OSHIMA ASUKA

1. Flow chart



1. 각 함수의 기능

void InitTetris() : 테트리스를 수행하는데 필요한 기본적인 변수와 자료구조를 초기화한다.

void DrawOutline() : 블록이 떨이지는 테두리, 다음 블록을 보여주는 테두리 그리고 점수를 보여줄 테두리를 그린다.

int GetCommand() : 사용자부터 입력을 받아 그 입력을 구분하여 입력받은 명령을 리턴 한다.

int ProcessCommand(int command) : GetCommand()에서 입력받은 command를 input값으로 받아서 command별로 적절한 동작을 취한다.

void DrawField() : 테트리스의 필드를 그린다.

void PrintScore(int score) : 점수를 화면에 표시한다.

void DrawNextBlock(int \*nextBlock) : nextBlock의 정보를 바탕으로 화면에 다음 블록을 확인할 수 있는 상자 안에 다음 블록을 그린다.

void DrawBlock(int y, int x, int blockID,int blockRotate,char tile) : 정해진 위치에 블록을 그린다. Tile은 그리려고 하는 모양이다.

void DrawBox(int y,int x, int height, int width) : 상자의 왼쪽 상단의 위치 상자의 높이와 너비를 입력 받아서 입력받은 위치에 입력받은 크기를 맞게 상자를 그린다.

void play() : 게임을 플레이하기 위한 정보를 초기화하고, 게임을 플레이하기 위한 과정을 제어한다.

char menu() : 화면의 메뉴를 출력하고 사용자부터 입력을 받으면 그 입력 값을 리턴한다.

int CheckToMove(char f[HEIGHT][WIDTH],int currentBlock,int blockRotate, int blockY, int blockX) : 블록이 아래에 이동할 수 있는지 확인하여 이동가능하면 0을 불가능한다면 1을 리턴한다.

void DrawChange(char f[HEIGHT][WIDTH],int command,int currentBlock,int blockRotate, int blockY, int blockX) : 사용자부터 입력받은 command에 적절하게 블록을 업데이트한다.

void BlockDown(int sig) : 블록을 자동으로 이동시키는 함수다. CheckToMove함수로 블록을 아래에 이동할 수 있는지 확인한 후 이동 불가능한 경우 게임오버 flag를 1로 한다. 이동가능한 경우 블록을 필드에 추가하여 다음 블록을 그리고 블록의 위치 등 정보를 초기화한다.

void AddBlockToField(char f[HEIGHT][WIDTH],int currentBlock,int blockRotate, int blockY, int blockX): 현제 블록을 field에 추가한다.

int DeleteLine(char f[HEIGHT][WIDTH]) : 완전히 채워진 라인을 찾아 지우고 지운 라인수에 대한 점수를 계산하여 리턴한다.

1. pseudo code

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

    for i=0 to i<BLOCK\_HEIGHT do

        for j=0 to j<BLOCk\_WIDTH do

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

                if the position is movable then

                return 0

        done

    done

    return 1

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

    이전 불록 정보 찾음

    switch(command)::=

        case KEY\_UP : 회전 상태를 하나 전 상태로 되돌림 break

        case KEY\_DOWN: prevY-- break

        case KEY\_LEFT : prevX++ break

        case KEY\_RIGHT : prevX-- break

        default : break

    2. 이전 블록 정보를 지운다.

    for i=0 to i< BLOCK\_HEIGHT

        for j=0 to j<BLOCK\_WIDTH

                if(블록의 일부인 셀이면)

                printw(".");

    3. 새로운 블록 정보를 그린다.

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

    4 커서를 화면 밖으로 이동

void BlockDown(int sig)::=

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

        blockY++;

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

    else

        if blockY=-1 then gameOver

        else

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

            for i=0 to i<BLOCK\_NUM-1

             shift nextBlock array elements by 1 position to the left

            nextBlock[BLOCK\_NUM-1] <-rand()%7

            DrawNextBlock(nextBlock)

            reset

             blockRotate ,blockY,and blockX

            PrintScore(score)

            DrawField()

            timed\_out=0

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

    for i=0 to i< BLOCK\_HEIGHT

        for j=0 to j<BLOCK\_WIDTH

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

                f[blockY+i][blockX+j]<-1

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

    1. 필드를 탐색하여, 꽉 찬 구간이 있는지 탐색한다.

    for i=0 to i<HEIGHT  flag=1

        for j=0 to j<WIDTH

            if f[i][j]=0 flag=0

    2. 꽉 찬 구간이 있으면 해당 구간을 지운다. 즉, 해당 구간으로 필드값을 한칸씩 내린다.

        if(flag)

            for Y=i to  Y>0  Y--

                for X=0 to X<WIDTH

                      f[Y][X]=f[Y-1][X]

                for x=0 to WIDTH f[0][x]=0

            deletedLines++;

    return (deletedLines\*deletedLines\*100)