8주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20221532 이름: 김민기

1. Diagram

   Description automatically generated

Menu함수 실행 -> initTetris함수로 변수 초기화 -> GetCommant함수로 값 입력받음 -> processCommand함수로 입력값 처리 -> 입력값이 ‘q’이면 종료 후 menu 실행

Diagram

Description automatically generated

1초마다 blockDown실행 -> 밑으로 움직일 수 있는지 CheckToMove함수로 판단 -> 가능하면 블록 한 칸 밑으로/ 불가능하면 블록이 맨위인지 판단하여 gameover 출력 -> field에 블록저장 -> 채워진 줄이 있으면 없앤 후 점수 증가시킴 -> 현재 블록과 다음 블록 변경

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 테트리스의 변수 초기화 및 초기작업 실행

\* input : none

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void InitTetris();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 키 입력받아 command에 저장

\* input : none

\* return : (int) 입력받은 command

\* KEY\_DOWN : 방향키 아래

\* KEY\_UP : 방향키 위

\* KEY\_RIGHT: 방향키 오른쪽

\* KEY\_LEFT : 방향키 왼쪽

\* ' ' : Space bar

\* 'q'/'Q' : quit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int GetCommand();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* command에 따라 변수 변경 후 블록 조정

\* input : (int) GetCommand로 받은 command

\* return : (int) quit에 대한 입력을 받을 경우 0,

\* 그외의 경우 1을 return한다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int ProcessCommand(int command);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 블록이 1초마다 내려가게 함

\* 내려갈 수 없으면,

\* 블럭을 field에 합친다.

\* 완전이 채워진 line을 지운다.

\* next block을 current block으로 바꿔주고

\* block의 좌표를 초기화 한다.

\* 다음 블럭을 화면에 그리고 갱신된 score를

\* 화면에 display한다.

\* input : (int) sig

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void BlockDown(int sig);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* command에 따른 블록의 움직임이 가능한지 판단

\* input : (char[][]) 블럭의 움직임을 확인할 필드

\* (int) 현재 블럭의 모양 ID

\* (int) 블럭의 회전 횟수

\* (int) 블럭의 Y좌표

\* (int) 블럭의 X좌표

\* return : (int) 입력에 대한 블럭 움직임이 가능하면 1

\* 가능하지 않으면 0을 return 한다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* command에 의해 변경된 블록을 그리고 이전 블록은 지움

\* input : (char[][]) command의해 바뀐 블럭을 확인할 필드

\* (int) 바뀌기 전 모양을 알기 위해 command를 입력으로 받는다.

\* (int) 현재 블럭의 모양 ID

\* (int) 블럭의 회전 횟수

\* (int) 블럭의 Y좌표

\* (int) 블럭의 X좌표

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* field 그려줌

\* input : none

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void DrawField();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 더 이상 내려가지 못하는 블록을 field에 저장

\* input : (char[][]) 블럭을 쌓을 필드

\* (int) 현재 블럭의 모양 ID

\* (int) 블럭의 회전 횟수

\* (int) 블럭의 Y좌표

\* (int) 블럭의 X좌표

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 모두 채워진 line을 찾아 지운 후 점수를 증가시킴

\* input : (char[][]) 완전히 채워진 line을 확인할 필드

\* return : (int) 삭제된 라인의 갯수에 대한 점수

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 다음에 나올 블록을 화면 오른쪽에 출력

\* input : (int\*) 블럭의 모양에 대한 ID 배열

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void DrawNextBlock(int \*nextBlock);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 점수를 화면 오른쪽 아래에 출력

\* input : (int) 출력할 점수

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void PrintScore(int score);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* (y, x) 좌표에 인자로 받은 블록을 그림

\* input : (int) 그리고자 하는 박스의 왼쪽 상단모서리의 y 좌표

\* (int) 왼쪽 상단 모서리의 x 좌표

\* (int) 블록의 모양

\* (int) 블록의 회전 횟수

\* (char) 블록을 그릴 패턴 모양

\* return : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void DrawBlock(int y, int x, int blockID,int blockRotate,char tile);

2.

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

for(i, j 4번씩 반복)

블록이 field를 벗어나면 0, 아니면 1리턴

}

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

command로 변경되기 전의 변수 값 저장

for(i, j 4번씩 반복)

이전 블록 삭제

새 블록 그림

}

void BlockDown(int sig){

//강의자료 p26-27의 플로우차트를 참고한다.

if(블록이 아래로 움직일 수 있음){

아래로 움직인 후 블록 그림

}

else {

if(blockY == -1) {

gameOver를 1로 설정

}

Field 에 블록 저장

DeleteLine함수 실행 후 리턴값 score에 저장

PrintScore함수 실행

nextBlock변경

field 그림

}

Alarm함수 실행

}

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

for(i, j 4번씩 반복)

블록 field에 저장

}

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

for(HEIGHT, WIDTH만큼 i, j 반복){

}

if(한줄이 모두 채워져 있음) {

해당 줄부터 한 칸씩 밑으로 내림

맨 윗줄은 0으로 설정

Score증가

}

}

Score 리턴

}