8주차 예비보고서

전공: 경제학과 학년: 4학년 학번: 20180501 이름: 김연수

1. flow chart

flow chart 1 :

도표이(가) 표시된 사진

자동 생성된 설명

flow chart 2 :

도표이(가) 표시된 사진

자동 생성된 설명

1. pseudo code for 5 functions

int CheckToMove {

int I, j;

for(i=0;i<4;i++) {

for(j=0;j<4;j++) {

if the current block cell is occupied {

if the new position is not within the field boundaries, then return 0;

if the new position is occupied by another block in field, then return 0

}

}

}

return 1;

}

----------------------------------------------------------------------------------

void DrawChange {

int prevY = blockY;

int prevX = blockX;

int prevRotate = blockRotate;

int i, j;

Determine the previous block position and rotation based on the command. get prevX, prevY, prevRotate value based on command.

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

for(j=0;j<4;j++){

if(block[currentBlock][prevRotate][i][j]==1 && i+prevY>=0){

Erase block in position (prevY + i + 1, prevX + j + 1)

}

}

Draw the new block with DrawBlock function

}

}

----------------------------------------------------------------------------------

void BlockDown(int sig){

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

if(blockY == -1) gameOver = 1;

add block to field with the function of addblocktofield.

Erase Line by DeleteLine function. and add the return value in score.

replace currentblock to nextblock and nextblock to rand number 0~6;

call drawnextblock

call printscore

initialize location of current block

call drawfield

} else {

blockY++;

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

}

timed\_out = 0;

}

----------------------------------------------------------------------------------

void AddBlockToField {

// Iterate through the 4x4 block matrix

for (i = 0; i < 4; i++) {

for (j = 0; j < 4; j++) {

//If the block has a non-zero value at the current position, add it to the field

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

Check if the block is within the field boundaries before adding it {

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

}

}

}

}

}

----------------------------------------------------------------------------------

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

int i, j, k;

int linesDeleted = 0;

int isLineFull;

// Iterate through each row in the field

for (i = 0; i < HEIGHT; i++) {

isLineFull = 1;

// Check if the current row is full

for (j = 0; j < WIDTH; j++) {

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

isLineFull = 0;

break;

}

}

// If the row is full, delete it and move rows above it down

if (isLineFull) {

linesDeleted++;

// Move all rows above the deleted row down by 1

for (k = i; k > 0; k--) {

for (j = 0; j < WIDTH; j++) {

f[k][j] = f[k - 1][j];

}

}

// Clear the top row

for (j = 0; j < WIDTH; j++) {

f[0][j] = 0;

}

}

}

return linesDeleted \* 100;

}