8주차 예비보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20211558 이름 : 윤준서

**1. Flow Chart**

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

**2. 함수 설명**

**- char menu() :** 화면에 메뉴를 출력한다.

**- void play() :** tetris 게임의 play를 수행한다

**- void DrawField(int currentBlock, int blockRotate, int blockX, int blockY)**

**:** tetris를 play하는 field를 그린다.

**- void DrawNextBlock(int \*nextblock) :** next block을 화면에 그린다.

**- void PrintScore(int score) :** score를 출력한다.

**- void DrawOutline() :** 화면의 기본 테두리를 그린다.

**- int GetCommand() :** key 입력에 대한 command를 반환한다.

**- int ProcessCommand(int command) :** command에 대한 처리를 수행한다.

**- void DrawBox(int y, int x, int height, int width) :** 화면에 box를 그린다.

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

**:** (y, x)좌표에 block을 character tile로 채워서 그린다.

**- int CheckToMove(char filed[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX) :** block이 움직일 수 있는지 확인한다.

**- void DrawChange(char field[HEIGHT][WIDTH], int command, int currentBlock, int blockRotate, int blockY, int blockX) :** 이전 block을 지우고 다시 그린다.

**- void BlockDown(int sig) :** 매초마다 block을 한 칸 씩 내린다.

**- void AddBlockToField(char field[HEIGHT][WIDTH], int currentBlcok,int blockRotate, int blockY, int blockX) :** block을 field에 채운다.

**- int DeleteLine(char field[HEIGHT][WIDTH]) :** 채워진 맨 아래 가로줄을 삭제한다.

**3. Pseudo Code**

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

for(i=0; i<=3; i++) {

for(j=0; j<=3; j++) {

if(block[currentBlock][blockRotate][i][j] == 1) x = blockX+j; y = blockY+i;

if(((0 <= x < WIDTH) && (0 <= y < HEIGHT)) != TRUE) return TRUE;

if(f[y][x] == 1) return FALSE;

return TRUE; }

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

if(command == up) blockRotate = (blockRotate + 3) % 4;

else if(command == down) blockY -= 1;

else if(command = left) blockX += 1;

else if(command = right) blockX -= 1;

DrawBlock(blockY, blockX, currentBlock, blockRotate, '\0'); }

**void BlockDown(int sig)** {

if CheckToMove(f, currentBlock, blockRotate, blockY + 1, blockX) blockY += blockY + 1;

DrawChagne(f, command, currentBlock, blockRotate);

else

if(blockY == -1) GameOver = TRUE;

else

score += DeleteLine(field);

for(i=0; i<=1; i++) nextBlock[i] = nextBlock[i + 1];

nextBlock[2] = rand(0~6); DrawNextBlock(nextBlock);

blockY = -1; blockX = WIDTH/2 - 2; blockRotate = 0;

PrintScore(score);

DrawField();

timed\_out = TRUE; }

**void AddBlockToField(char field[HEIGHT][WIDTH], int currentBlcok,int blockRotate, int blockY, int blockX)** {

for(int i=0; i<=3; i++) {

for(int j=0; j<=3; j++) {

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

if((0 <= i + blockY < HEIGHT) && (0 <= j + blockX < WIDTH))

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

}

}

return TRUE; }

**int DeleteLine(char field[HEIGHT][WIDTH])** {

er\_line\_cnt = 0; all\_fit = FALSE;

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

all\_fit = TRUE;

for(j=0; j<WIDTH< j++) if(f[i][j] == 0) all\_fit = FALSE;

if(all\_fit == TRUE)

er\_line\_cnt = er\_line\_cnt + 1;

for(y=i; y<=1; y--) for(x=0; x<WIDTH< x++) f[y][x] = f[y-1][x];

for(x=0; x<WIDTH; x++) f[0][x] = 0;

i -= 1;

}

return er\_line\_cnt \* er\_ line\_cnt \* 100; }