

8주차 예비보고서

전공: 컴퓨터공학과

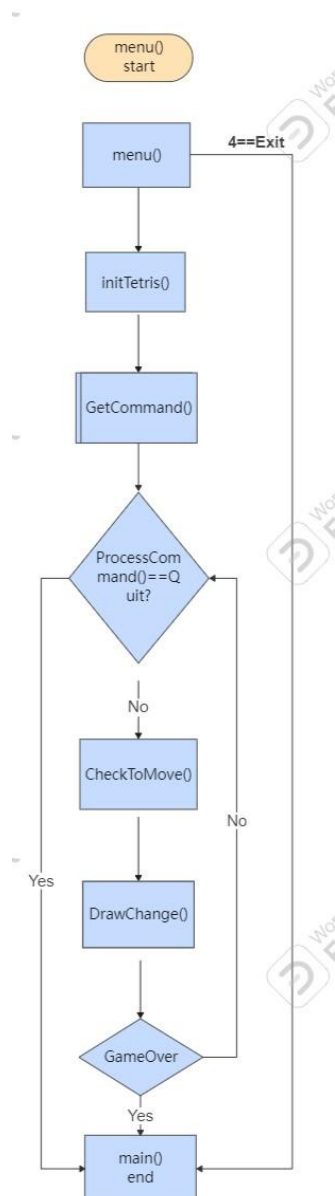
학년: 2학년

학번: 20231632

이름:

Jumagul Alua

게임의 전반적인 flow chart



main(): 게임을 시작하거나 종료할 때까지 반복적으로 게임을 실행하는 메인 루프이다.

InitTetris(): 다음 블록을 설정하며, 게임 상태를 초기화한다.

DrawOutline(): 테트리스 필드, 다음 블록 보여주는 창, 그리고 점수를 표시하는 창 등의 외곽선을 그린다.

GetCommand(): 사용자 입력을 받아와서 해당 입력에 맞는 명령을 반환한다.

ProcessCommand(): 사용자의 입력에 따라 블록을 이동시키거나 회전시키는 등의 동작을 한다.

DrawField(): 현재 필드의 상태에 따라 블록이 있는 부분은 채워진 칸으로, 없는 부분은 비어있는 칸으로 표시한다.

PrintScore(): 현재 점수를 출력한다

DrawNextBlock(): 다음에 나올 블록을 표시한다.

DrawBlock(): 현재 블록을 그려서 화면에 표시한다.

DrawBox(): 상자 모양의 테두리를 그린다.

play(): 사용자의 입력을 받아 수행하고, 게임이 종료될 때까지 반복 실행된다.

menu(): 게임 메뉴를 표시하고 사용자 입력을 받아 반환한다.

CheckToMove(): 블록이 이동 가능한지 여부를 확인한다.

DrawChange(): 블록의 변화를 그린다.

BlockDown(): 블록을 아래로 움직이는 함수이다.

AddBlockToField(): 블록을 필드에 추가한다.

DeleteLine(): 가득 찬 줄을 지우고 필드를 정리한다.

5가지 함수들의 pseudo code:

1. CheckToMove():

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

for i = 0 to BLOCK_HEIGHT:

```

        for j = 0 to BLOCK_WIDTH:

            if block[currentBlock][blockRotate][i][j]가 1이고 (i+blockY, j+blockX) 필드를 벗어남 or
이미 블록이 있다면

                return 0

        return 1

```

2. DrawChange():

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

```

preRot = blockRotate

preblX = blockX

preblY = blockY

flag = 1

    case KEY_UP

        preRot← (preRot + 3) % 4

    case KEY_DOWN

        preblY ← preblY - 1

    case KEY_LEFT

        preblX ← preblX + 1

    case KEY_RIGHT

        preblX ← preblX - 1

    default

        break

```

```

for i=0 to 3

```

```

    for j=0 to 3

```

```
if block[currentBlock][preRot][i][j] = 1
```

기존의 그림자를 지운다

```
DrawBlockWithFeatures(blockY, blockX, currentBlock, blockRotate)
```

3. BlockDown()

```
BlockDown(sig)
```

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

```
for i=0 to BLOCK_HEIGHT
```

```
for i=0 to BLOCK_WIDTH
```

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

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

```
if blockY ≤ -1
```

```
gameOver = 1
```

```
return
```

```
score += AddBlockToField(field, nextBlock[0], blockRotate, blockY, blockX)
```

```
score += DeletetLine(field)
```

```
nextBlock[0] ← nextBlock[1]
```

```
nextBlock[1] ← nextBlock[2]
```

```
nextBlock[2] = rand() % 7
```

```
blockRotate = 0
```

```
blockY = -1
```

```
blockX = WIDTH / 2 - 2
```

```
DrawField()
```

```
DrawNextBlock(nextBlock)
```

```
PrintScore(score)
```

4. AddBlockToField

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

```
for i=0 to 3
    for j=0 to 3
        if block[currentBlock][blockRotate][i][j] = 1
            f[blockY + i][blockX + j] = 1
```

5. DeleteLine():

DeleteLine(char f[HEIGHT][WIDTH])

```
line_cnt = 0
for i from 0 to HEIGHT-1
    for j from 0 to WIDTH-1
        if f[i][j] = 0
            isFullLine ← false
            break
    if isFullLine
        line_cnt ++
for k=j down to 1
    for j=0 to WIDTH-1
        f[k][j] = f[k-1][j]
return line_cnt^2 * 100
```