

## 컴퓨터공학실험1 6주차 예비보고서

전공: 컴퓨터공학과

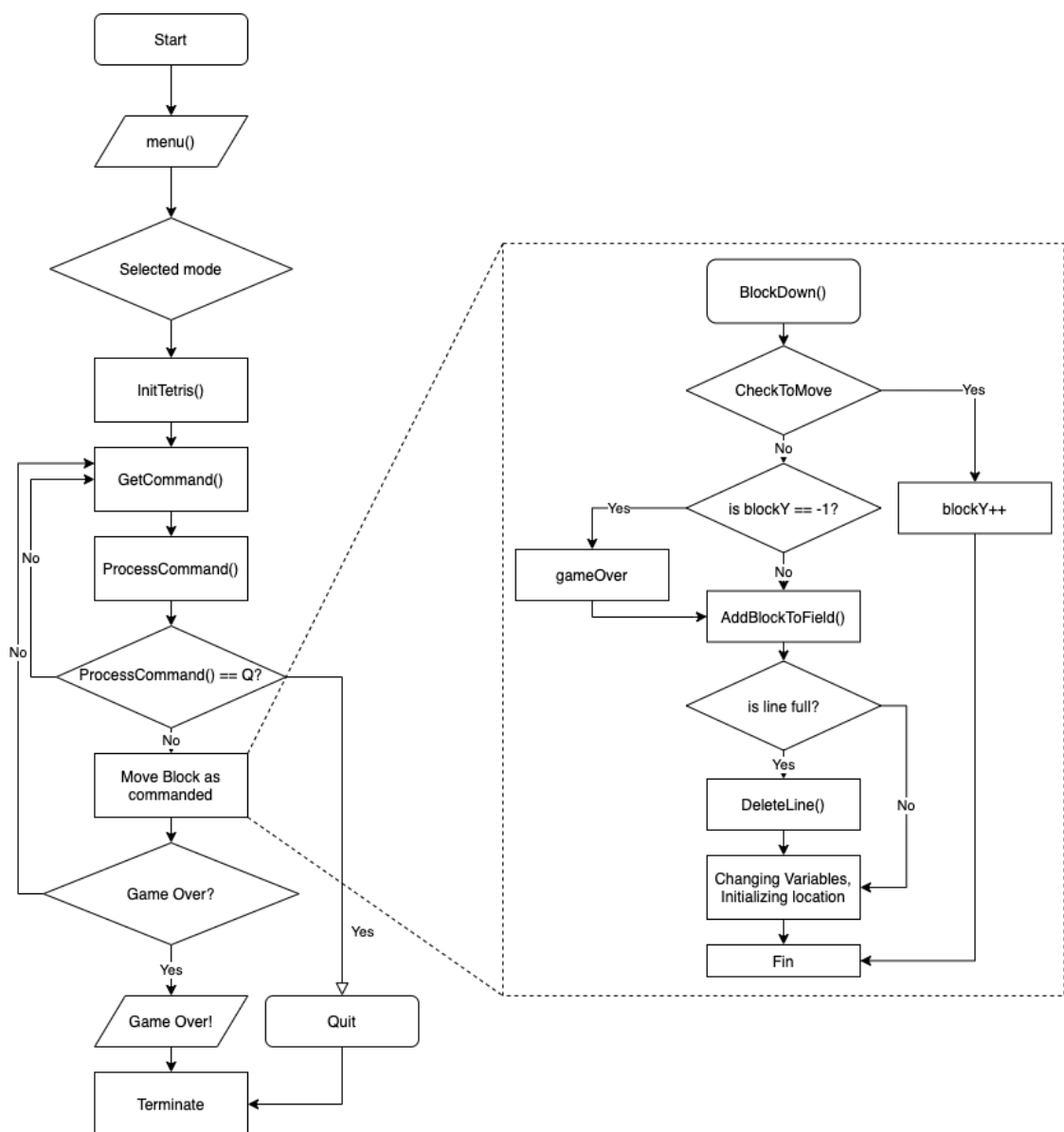
학년: 2

학번: 20191559

이름: 강상원

- 테트리스 frame 프로그램 파일을 미리 읽어보고 (6주차 강의자료 부록1. Ncurses 라이브러리 포함), 테트리스 게임의 flow chart를 자세히 작성하시오. 그리고 테트리스 게임을 구성하는 각 함수의 기능에 대해서 설명해보시오.

➤ flow chart



➤ 구성하는 함수 기능 설명

- void InitTetris(): 변수 초기화 후 초기 화면 그리기
- void DrawOutline(): 테트리스 field, 다음 블록 칸, 점수 확인 창 등의 테두리를 그린다.
- int GetCommand(): 키보드 입력을 받아 명령으로 변환한다. (Q, 방향 키 4개, Space)
- int ProcessCommand(int command): 명령 실행. 명령이 QUIT인 경우 0, 아닐 경우 1을 반환한다. (command에 대한 처리)
- void DrawField(): 테트리스 play 하기 위한 field를 그린다.
- void PrintScore(int score): 점수 표시.
- void DrawNextBlock(int \*nextBlock): 다음 블록 칸에 다음에 나올 블록 정보를 표시.
- void DrawBlock(int y, int x, int blockID, int blockRotate, char tile): (x, y) 좌표에 character tile의 블록을 그린다.
- void DrawBox(int y, int x, int height, int width): 해당 좌표에 직사각형을 그린다.
- void play(): 게임 시작.
- char menu(): 메뉴를 그린다.
- int CheckToMove(char f[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX): 명령 종류에 따라 블록을 이동할 수 있는지 여부 판단.
- void Delete\_Block(int ypos, int xpos, int currentBlock, int blockRotate): 지나간 블록을 지우고 다시 "."으로 채운다.
- void DrawChange(char f[HEIGHT][WIDTH], int command, int currentBlock, int blockRotate, int blockY, int blockX): 명령에 의해 바뀐 부분만 field에서 업데이트.
- void BlockDown(int sig): 블록을 한 칸씩 밑으로 내린다. 더 이상 내릴 수 없을 경우 field에 고정하며, 다음 블록을 사용 시작한다.
- void AddBlockToField(char f[HEIGHT][WIDTH], int currentBlock, int blockRotate, int blockY, int blockX): 특정 좌표에 현재 블록을 고정.
- int DeleteLine(char f[HEIGHT][WIDTH]): 꽉 찬 (완성된)줄이 있는지 체크하고, 있는 경우 그 줄을 지우고 스코어를 증가시킨다. (지운 line 개수)^2 \* 100

나중에 구현해야 될 함수

- void DrawShadow(int y, int x, int blockID, int blockRotate): 블록이 맨 밑에 떨어질 위치를 미리 보여 주는 그림자를 그린다.
- void createRankList(): 랭킹 정보 만들기.
- void rank(): 랭킹 기록 출력.
- void writeRankFile(): 랭킹이 저장된 파일 생성.
- void newRank(int score): 새 랭킹 정보를 추가.
- int DrawRecommend(int y, int x, int blockID, int blockRotate): 추천 블록 배치 계산.
- void recommendedPlay(): 추천 블록 배치 방식으로 게임 자동 진행

2. 실습시간에 구현 할 5가지 함수들에 대한 간단한 의사 코드(pseudo code)를 제시하시오.

```
CheckToMove(f, currentBlock, blockRotate, blockY, blockX)
    for i = 0 to 3
        for j = 0 to 3
            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
```

```
DrawChange(f, command, currentBlock, blockRotate, blockY, blockX)
    if (command == KEY_UP) blockRotate = (blockRotate+3)%4
    if (command == KEY_DOWN) blockY -= 1
    if (command == KEY_LEFT) blockX += 1
    if (command == KEY_RIGHT) blockX -= 1
    //ERASE BLOCK by Calling Delete_Block function
    //Draw new block
    DrawBlock(blockY, blockX, currentBlock, blockRotate, ' ')
```

```
BlockDown(sig)
    if CheckToMove(f, currentBlock, blockRotate, blockY +1, blockX)
        blockY += blockY + 1
        DRAWCHANGE(f, command, currentBlock, blockRotate, blockY, blockX)
    else
        if blockY == -1
            gameOver = TRUE
        else
            score += DeleteLine(field)
            for i = 0 to 1
                nextBlock[i] = nextBlock[i+1]
            nextBlock[2] = (Random integer in 0 … 6)
            DrawNextBlock(nextBlock)
            blockY = -1
            blockX = WIDTH / 2 - 2
            blockRotate = 0
            PrintScore(score)
            DrawField()
    timed_out = TRUE
```

```

AddBlockToField(f, currentBlock, blockRotate, blockY, blockX)
    for i = 0 to 3
        for j = 0 to 3
            if block[currentBlock][blockRotate][i][j]
                if (0 ≤ i+blockY < HEIGHT && 0 ≤ j+blockX < WIDTH)
                    f[i+blockY][j+blockX] = 1
    return TRUE

```

```

DeleteLine(f)
er_line_cnt = 0
all_fit = FALSE
for i = 0 to HEIGHT - 1
    all_fit = TRUE
    for j = 0 to WIDTH - 1
        if f[i][j] == 0
            all_fit = FALSE
    if all_fit == TRUE
        er_line_cnt = er_line_cnt + 1
        for y = i downto 1
            for x = 0 to WIDTH - 1
                f[y][x] = f[y-1][x]
        for x = 0 to WIDTH - 1
            f[0][x] = 0
        i = i - 1
return er_line_cnt * er_line_cnt * 100

```