8주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20191612 이름: 윤기웅

**1.**

* InitTetris: 이 함수는 테트리스 게임을 초기화한다. 게임 필드를 비우고, 다음 블록을 무작위로 선택하며, 초기 위치와 회전 상태, 점수, 게임 오버 상태, 타임아웃 상태를 설정한다. 또한, 게임 화면의 윤곽, 필드, 다음 블록, 점수를 그린다.
* DrawOutline: 이 함수는 게임 화면의 윤곽을 그린다. 블록이 떨어지는 주 게임 필드, 다음 블록을 보여주는 영역, 점수 표시 영역의 윤곽을 그림으로써 게임의 시각적 틀을 제공한다.
* GetCommand: 이 함수는 사용자 입력을 받아 해당 입력에 따라 동작을 식별하는 명령을 반환한다. 화살표 키, 스페이스 바, 'q' 또는 'Q' 키 등의 입력을 처리하여 게임 내에서 특정 동작을 수행할 명령을 식별한다.
* ProcessCommand: 이 함수는 GetCommand 함수에서 받은 명령을 처리한다. 사용자 입력에 기반하여 블록 회전, 좌우 이동, 하강, 게임 종료 등의 동작을 수행한다. 동작이 가능한 경우에만 이를 실행하고 화면에 결과를 업데이트한다. 이 함수는 게임의 핵심 로직을 담당한다.
* DrawField : 게임 필드를 그립니다. 이 함수는 HEIGHT와 WIDTH를 기반으로 게임 판의 각 위치에 대해 반복하며, 위치에 블록이 존재하는 경우(값이 1인 경우)는 특정 스타일로, 없는 경우(값이 0인 경우)는 다른 스타일로 화면에 출력합니다.
* PrintScore: 이 함수는 사용자의 점수를 화면에 표시합니다. 이 함수는 점수를 오른쪽 정렬로 표시하기 위해 지정된 위치에 점수를 출력합니다.
* DrawNextBlock: 이 함수는 다음에 나올 블록을 보여주는 영역에 블록을 그립니다. 이 함수는 nextBlock 배열을 사용하여 어떤 블록을 그릴지 결정하고, 해당 블록의 모양에 따라 지정된 위치에 블록을 그립니다.
* DrawBlock: 이 함수는 게임 필드에 블록을 그립니다. 이 함수는 주어진 블록 ID, 회전 상태, 위치를 기반으로 블록을 그립니다. 블록의 각 부분이 실제 게임 필드에 그려지며, 특정 스타일을 적용하여 블록을 표시합니다.
* DrawBox: 이 함수는 지정된 위치에 상자를 그립니다. 이 함수는 상자의 가로 세로 길이와 위치를 기반으로 상자의 테두리를 그리며, 상자의 모서리에는 특별한 문자를 사용하여 상자의 모양을 완성합니다. 이 함수는 게임의 UI 요소를 구성하는 데 사용됩니다.
* Play : 이 함수는 테트리스 게임의 주요 게임 루프를 실행하는 함수이다. 이 함수는 게임이 시작될 때 초기화 작업을 수행하고, 사용자의 입력을 받아 처리하며, 게임의 종료 조건을 확인한다.

1. clear() 함수를 호출하여 화면을 초기화한다.
2. BlockDown 함수를 SIGALRM 시그널의 핸들러로 설정한다. 이는 일정 시간이 지날 때마다 블록을 자동으로 아래로 이동시키는 역할을 한다.
3. InitTetris 함수를 호출하여 테트리스 게임을 초기화한다. 게임 필드를 비우고, 다음 블록을 선택하며, 점수와 게임 상태를 초기화하는 등의 작업을 수행한다.
4. do-while 루프를 통해 게임 루프를 실행한다. 루프 내에서는 다음과 같은 작업을 수행한다:
   * timed\_out 변수가 0이면 alarm(1)을 호출하여 1초 후에 SIGALRM 시그널을 발생시키고 timed\_out을 1로 설정한다. 이는 블록을 자동으로 내리는 기능을 구현하기 위함이다.
   * GetCommand 함수를 호출하여 사용자의 입력을 받아온다.
   * ProcessCommand 함수를 호출하여 사용자의 입력(명령)을 처리한다. 만약 사용자가 게임 종료 명령(QUIT)을 내렸다면, 중앙에 "Good-bye!!" 메시지를 출력하고 함수에서 반환하여 게임을 종료한다.

**-------------------------------------------------------------------------**

**1. CheckToMove:** 이 함수는 주어진 블록이 게임판에서 움직일 수 있는지 여부를 판단하는 함수다. 입력으로 받은 블록의 현재 상태 (위치, 회전 상태 등)와 게임판의 상태를 기반으로, 블록이 다음 위치로 이동하거나 회전할 수 있는지를 체크한다. 이동이나 회전이 가능하면 true(또는 1), 그렇지 않으면 false(또는 0)을 반환한다.

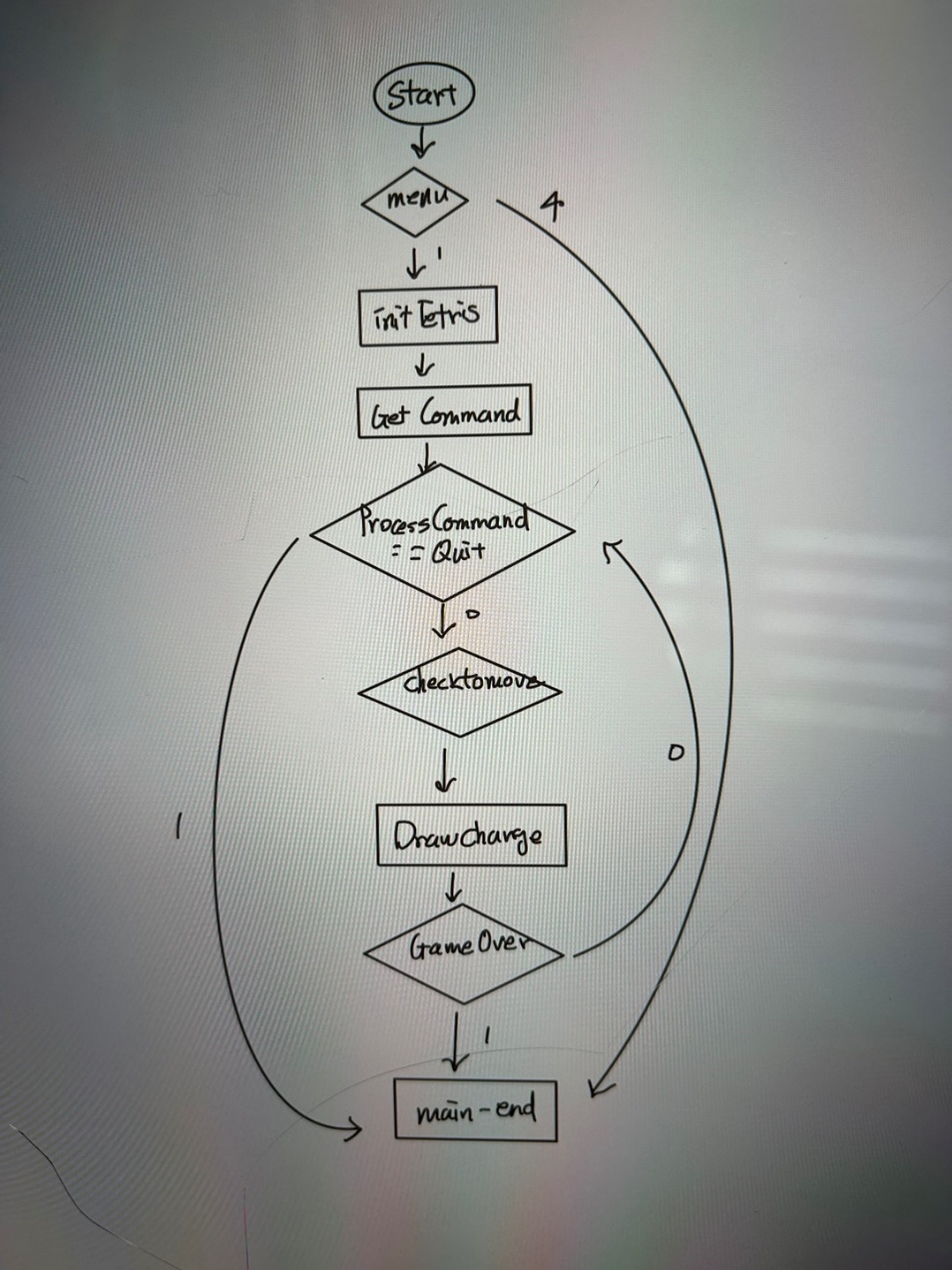
**2. DrawChange :** DrawChange 함수는 사용자가 블록을 이동하거나 회전시켰을 때, 게임판에 그 변경사항을 반영하여 화면에 그리는 역할을 한다. 이 함수는 먼저 이전 블록의 정보를 지우고, 새로운 위치나 회전 상태에 해당하는 블록의 정보를 게임판에 그린다. 이 과정에서 DrawBlock 함수를 사용하여 실제로 블록을 화면에 그린다.

**3. BlockDown :** BlockDown 함수는 시그널 핸들러로 사용되며, 일정 시간 간격으로 자동으로 호출되어 현재 블록을 게임판에서 한 칸 아래로 이동시킨다. 만약 블록이 더 이상 아래로 이동할 수 없는 경우(다른 블록이 아래에 있거나 바닥에 도달한 경우), 블록을 게임판에 고정시키고 새로운 블록을 생성하는 등의 추가 작업을 수행한다.

**4. AddBlockToField :** 이 함수는 현재 블록을 게임판에 고정시키는 역할을 한다. 블록이 최종 위치에 도달했을 때, 해당 블록의 모양과 위치에 따라 게임판의 상태를 업데이트 한다. 이것은 게임판에서 블록이 차지하는 위치를 표시하는 데 이용된다.

**5. DeleteLine :** DeleteLine 함수는 게임판을 검사하여 한 줄이 완전히 채워진 경우 그 줄을 삭제하고, 위에 있는 줄들을 아래로 이동시키는 역할을 한다. 이 기능은 테트리스 게임에서 점수를 획득하는 주요 방법 중 하나다. 줄이 삭제될 때마다 사용자의 점수가 증가한다.

이 함수들은 테트리스 게임의 핵심적인 동작을 구현하는 데 필요하다. 사용자의 입력에 따라 블록을 이동하고 회전시키며, 게임판의 상태를 업데이트하고, 줄을 삭제하여 점수를 산출하는 등의 작업을 수행한다.



**2.**

### CheckToMove

1. 4x4 블록의 모든 셀에 대해 반복한다.
   * 만약 현재 셀에 블록이 있다면,
     + y는 현재 셀의 y좌표와 블록의 y위치를 더한 값이다.
     + x는 현재 셀의 x좌표와 블록의 x위치를 더한 값이다.
     + 만약 f[y][x]이 참이라면 (즉, 이미 블록이 존재한다면) 0을 반환한다.
     + 만약 y 또는 x가 필드를 벗어난다면 0을 반환한다.
2. 모든 검사를 통과하면 1을 반환한다.

### DrawChange

1. 이전 블록의 위치를 결정한다. 이는 사용자의 명령에 따라 달라진다.
   * 만약 오른쪽 키를 눌렀다면, x를 하나 감소시킨다.
   * 만약 왼쪽 키를 눌렀다면, x를 하나 증가시킨다.
   * 만약 아래 키를 눌렀다면, y를 하나 감소시킨다.
   * 만약 위 키를 눌렀다면, 블록의 회전 상태를 변경한다.
2. 이전 블록 정보를 지운다.
   * 4x4 블록의 모든 셀에 대해 반복한다.
     + 만약 해당 셀에 블록이 있고, y위치가 유효하다면,
       - 이전 위치로 이동하여 블록을 지운다.
3. 새로운 블록 정보를 그린다.
4. 커서를 필드 밖으로 이동시킨다.

### BlockDown

1. 블록을 아래로 이동시킬 수 있는지 확인한다.
   * 만약 이동 가능하다면,
     + 블록의 y위치를 증가시킨다.
     + DrawChange 함수를 호출하여 변경을 반영한다.
2. 만약 이동할 수 없다면,
   * 만약 블록의 y위치가 -1이라면, 게임 오버 상태를 참으로 설정한다.
   * 그렇지 않다면, 현재 블록을 필드에 추가한다.
   * 점수를 업데이트하고, 다음 블록을 설정한다.
   * 다음 블록을 그리고, 점수와 필드를 출력한다.
   * 블록의 위치와 회전 상태를 초기화한다.
3. timed\_out을 0으로 설정한다.

### AddBlockToField

1. 4x4 블록의 모든 셀에 대해 반복한다.
   * 만약 현재 셀에 블록이 있다면,
     + y와 x 위치를 계산한다.
     + 만약 y와 x가 필드 내에 있다면,
       - 해당 위치에 블록을 추가한다.

### DeleteLine

1. 필드의 모든 줄에 대해 반복한다.
   * 만약 한 줄이 꽉 차 있다면,
     + 꽉 찬 줄의 수를 증가시킨다.
     + 꽉 찬 줄을 삭제하고, 모든 줄을 한 칸씩 아래로 이동시킨다.
2. 점수를 계산하여 반환한다. 점수는 꽉 찬 줄의 수의 제곱에 100을 곱한 값이다.