1. 프로그램 구성  
   Sokoban 게임을 리눅스 환경에서 C언어로 구현하였다. Ncurses 라이브러리를 이용한 UI를 제공한다.   
   Sokoban 게임은, 벽과 길로 구성된 평면의 맵에서 블록들을 제한시간 내에 목적지까지 모두 밀어서 위치시키는 것을 목적으로 하며, 이동횟수가 적을수록 높은 점수를 부여한다. 각각 다른 맵인 12개의 스테이지로 구성된다.  
   유저 플레이 모드, 데모 플레이 모드, 도움말과 랭킹 보기 기능으로 구성되어있다.
2. 실행 방법  
   리눅스 환경에서 make 명령어를 통해 컴파일 된 실행파일 run 파일을 실행한다.
3. 매크로 (Macro)

|  |  |  |
| --- | --- | --- |
| **매크로(Macro)** | | |
| 1 | WIDTH | 윈도우 창의 폭 |
| 2 | HEIGHT | 윈도우 창의 높이 |
| 3 | PREV | 실행취소 |
| 4 | NEXT | 다음 스테이지로 이동 |
| 5 | BACK | 이전 스테이지로 이동 |
| 6 | NOTHING | 10 |
| 7 | QUIT | 종료(‘q’) |
| 8 | BOX | 맵 상에서 옮겨야 할 박스 (2) |
| 9 | WALL | 맵 상에서 막힌 벽 (1) |
| 10 | WAY | 맵 상에서 이동 가능한 경로 (0) |
| 11 | DESTINATION | 맵 상에서 박스를 가져다 놓아야 할 지점 (3) |
| 12 | DSTNBOX | 맵 상에서 박스가 놓인 목적지 |
| 13 | UP | 유저 키 입력 (arrow up) |
| 14 | DOWN | 유저 키 입력 (arrow down) |
| 15 | LEFT | 유저 키 입력 (arrow left) |
| 16 | RIGHT | 유저 키 입력 (arrow right) |
| 17 | NAMELEN | 랭킹 이름 문자열의 최대 길이 |
| 18 | MAX\_ELEMENTS | 실행취소에 필요한 상태 스택의 최대 크기 |

1. 전역변수

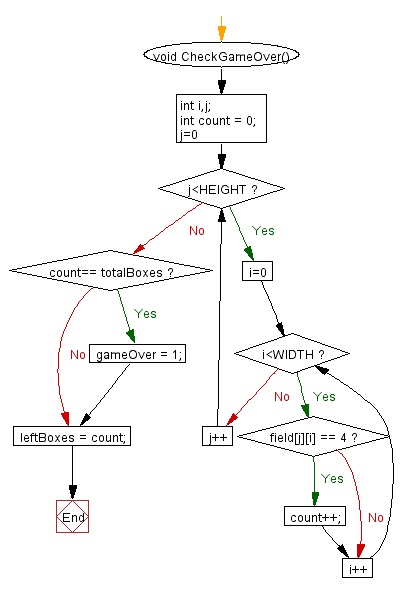
|  |  |  |
| --- | --- | --- |
| **전역변수** | | |
| **sokoban.h** | | |
| 1 | field[][] | char형 배열, 맵을 저장하기 위한 것 |
| 2 | score | int형 변수, 사용자의 이동횟수를 체크, 게임종료시 랭킹을 저장할 때 점수계산 방식에 따라 값이 변환됨 |
| 3 | cursor | typedef 변수, 현재의 커서 위치(x,y)를 저장. |
| 4 | gameOver | int형 변수, 게임의 종료 혹은 반복 여부를 나타내기 위한 flag |
| 5 | leftBoxes | int형 변수, 현재 남아있는 상자의 수를 나타냄 |
| 6 | inittime | 초기시간 |
| 7 | nowtime | 현재시간 |
| 8 | deltatime | 경과시간 |
| 9 | num\_per | 랭킹 리스트 아이템의 개수 |
| 10 | Node | 랭킹을 저장하는 Linked List의 Node 구조체. 점수(score), 이름(name), 그리고 다음 node를 가리키는 node포인터를 저장한다. |
| 11 | \*ptr, \*head, \*pre, \*t | Linked List에 쓰이는 Node 포인터 변수 |
| 12 | gameOver | CheckGameOver() 함수에서 쓰이는 flag변수. 현재 상태를 체크하여 게임 종료 조건을 만족할 시 1값으로 설정된다. |
| 13 | timeOver | StageLimitTime이 0이 되면 게임을 종료하도록 1로 설정된다. |
| 14 | leftBoxes | 맵 상에 목적지에 옮겨지지 않은 박스의 수 |
| 15 | initTime | 초기 시각 |
| 16 | nowTime | 현재 시각 |
| 17 | deltaTime | 경과 시간 |
| 18 | stageLimitTime | 각 맵에 지정된 제한시간 |
| 19 | STACK[] | 실행취소에 필요한, 맵의 현재 상태를 저장하는 구조체의 스택. Field[][] 배열과 cursor구조체의 정보를 저장 |
| 20 | top | 스택의 top인덱스 |
| 21 | mapDatas[] | Typedef 구조체 Map 형식의 변수. 12개의 맵의 정보를 저장 - Field[][] 배열, box(박스의 수), LimitTime(스테이지 제한시간) |
| 22 | Stage | 현재 스테이지 번호 |
| 23 | demoPflag | 값이 0일 때 유저 플레이 모드, 1일 때 데모 플레이 모드를 실행한다. |
| 24 | demoCount | demo[] 의 index |
| 25 | demo[][] | 데모 플레이를 위해 미리 지정된 스테이지 별 명령의 배열 |

1. 함수설명

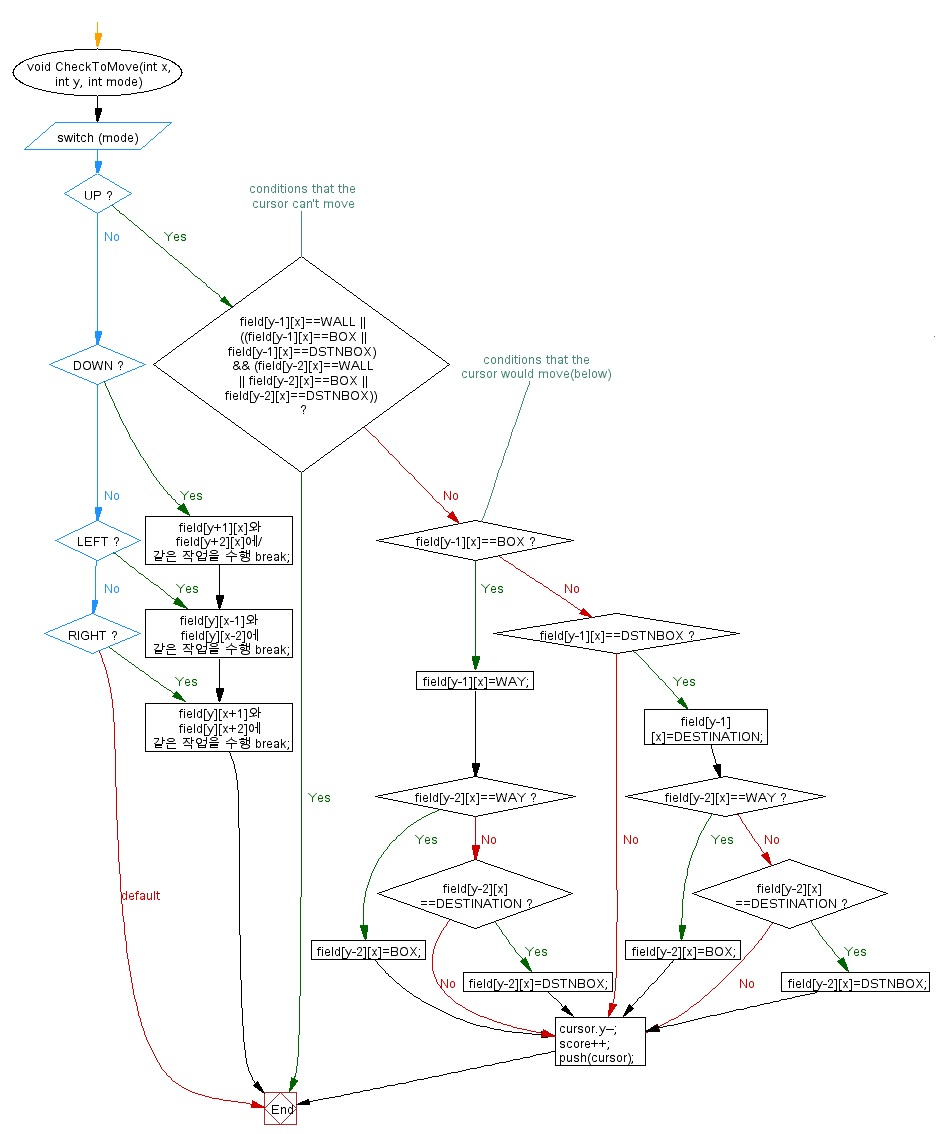
|  |  |  |
| --- | --- | --- |
| **함수설명** | | |
| **sokoban.h** | | |
| 1 | 이름 | DrawBox(); |
| 파라미터 | inty : 세로 시작 위치  intx : 가로 시작 위치  intheight : 세로 크기  intwidth : 가로 크기 |
| 리턴값 | None |
| 기능 | 시작 위치와 가로 세로 크기를 입력 받아 사각형의 테두리를 그린다. |
| 2 | 이름 | InitGame() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 게임 플레이를 위한 변수와 초기 맵 상태 생성 |
| 3 | 이름 | DrawOutline() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 게임 플레이에서 변하지 않는 UI생성 |
| 4 | 이름 | GetCommand() |
| 파라미터 | X |
| 리턴값 | int command |
| 기능 | 유저 플레이 모드의 경우 키 입력을, 데모 플레이 모드의 경우 demo[][]에 저장되어있는 command를순차적으로 로드 |
| 5 | 이름 | ProcessCommand() |
| 파라미터 | int command |
| 리턴값 | int ret |
| 기능 | GetCommand() 에서 넘겨받은 command의 전처리(커서이동 조건, 종료조건, 스테이지이동 조건, 실행취소 조건 판별) |
| 6 | 이름 | CheckToMove() |
| 파라미터 | int x, int y, int mode |
| 리턴값 | X |
| 기능 | ProcessCommand 에서 커서 이동 조건일 때, 유저가 입력한 화살표 방향으로 이동 가능한지 체크, 커서와 박스를 이동시키도록 field[][]의 값을 변경한다. |
| 7 | 이름 | Color() |
| 파라미터 | int a |
| 리턴값 | X |
| 기능 | Field[][] 에 저장된 값에 따라 색을 칠한다. |
| 8 | 이름 | DrawField() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 커서를 움직여 field의 값에 따라 맵의 색을 칠하는 color() 함수를 통해 맵을 그린다. |
| 9 | 이름 | checkGameOver() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | Field[][] 값들을 검사하여 모든 destination에 블록이 있을 경우 게임 오버 조건(gameOver)을 1로 설정한다. |
| 10 | 이름 | PrintScore() |
| 파라미터 | int score |
| 리턴값 | X |
| 기능 | UI에 남은 박스의 수, 움직인 회수, 남은 시간을 표시 |
| 11 | 이름 | DrawBox() |
| 파라미터 | int y, int x, int height, int width |
| 리턴값 | X |
| 기능 | 게임 플레이 시 맵을 둘러싸는 박스를 그린다. |
| 12 | 이름 | push() |
| 파라미터 | struct cursor\_ c |
| 리턴값 | X |
| 기능 | 유효한 이동 명령이 있을 때 마다 스택에 전 상태를 저장 |
| 13 | 이름 | pop() |
| 파라미터 | X |
| 리턴값 | struct cursor\_ c |
| 기능 | 실행취소 명령 실행 시 스택에 저장된 상태를 불러와 적용한다. |
| 14 | 이름 | showHelp() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 도움말을 표시한다. |
| 15 | 이름 | createRankList() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 게임 실행 시, 종료 후 rank를 저장하기 위해 rank.txt 파일로부터 저장된 리스트를 불러와 linked list에 저장한다. |
| 16 | 이름 | rank() |
| 파라미터 | X |
| 리턴값 | x |
| 기능 | 랭킹을 화면에 출력한다. |
| 17 | 이름 | writeRankFile() |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 스테이지 종료 후 rank.txt. 파일에 name과 score를 write한다. |
| 18 | 이름 | newRank() |
| 파라미터 | int score |
| 리턴값 | X |
| 기능 | 스테이지 종료 후, score와 입력 받은 name과 정보를 새로운 rank 노드에 추가한다. |
| 19 | 이름 | Freeall |
| 파라미터 | X |
| 리턴값 | X |
| 기능 | 랭킹에 이용된 linked list 에 할당된 메모리를 free |

1. 순서도

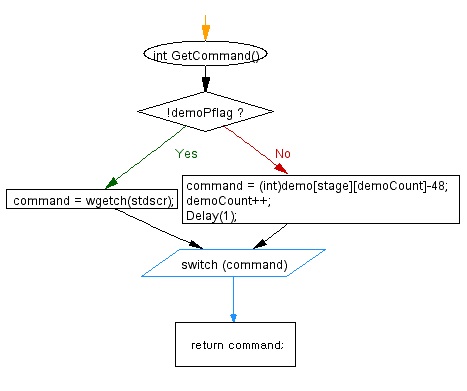
* CheckGameOver



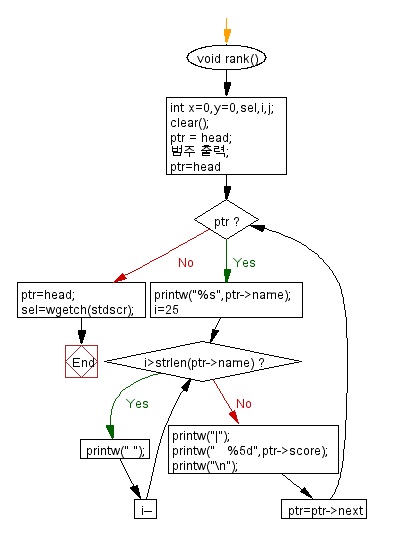
* CheckToMove



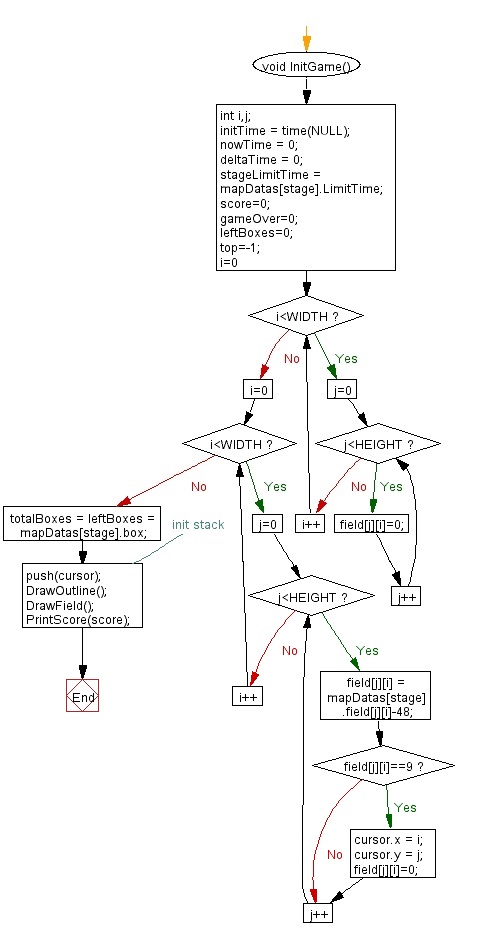
* GetCommand



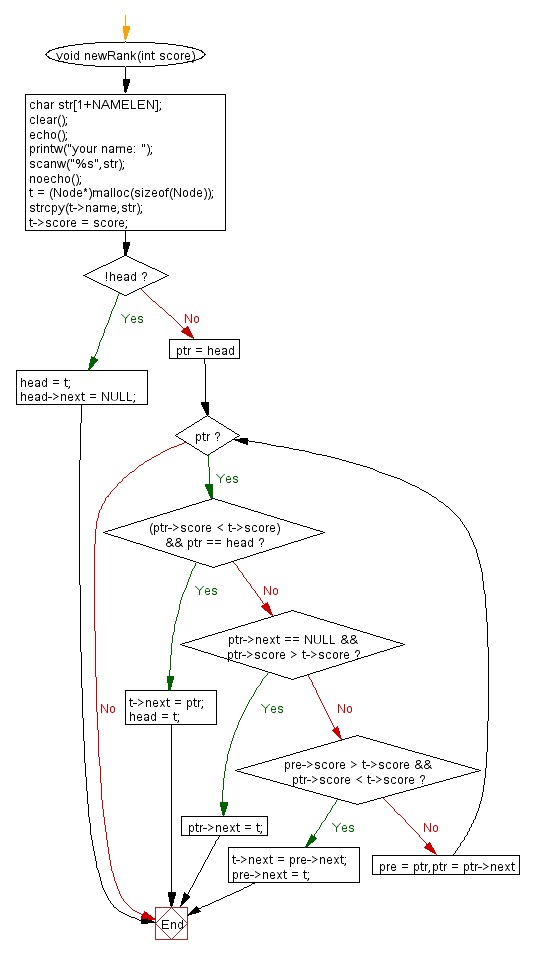
* rank



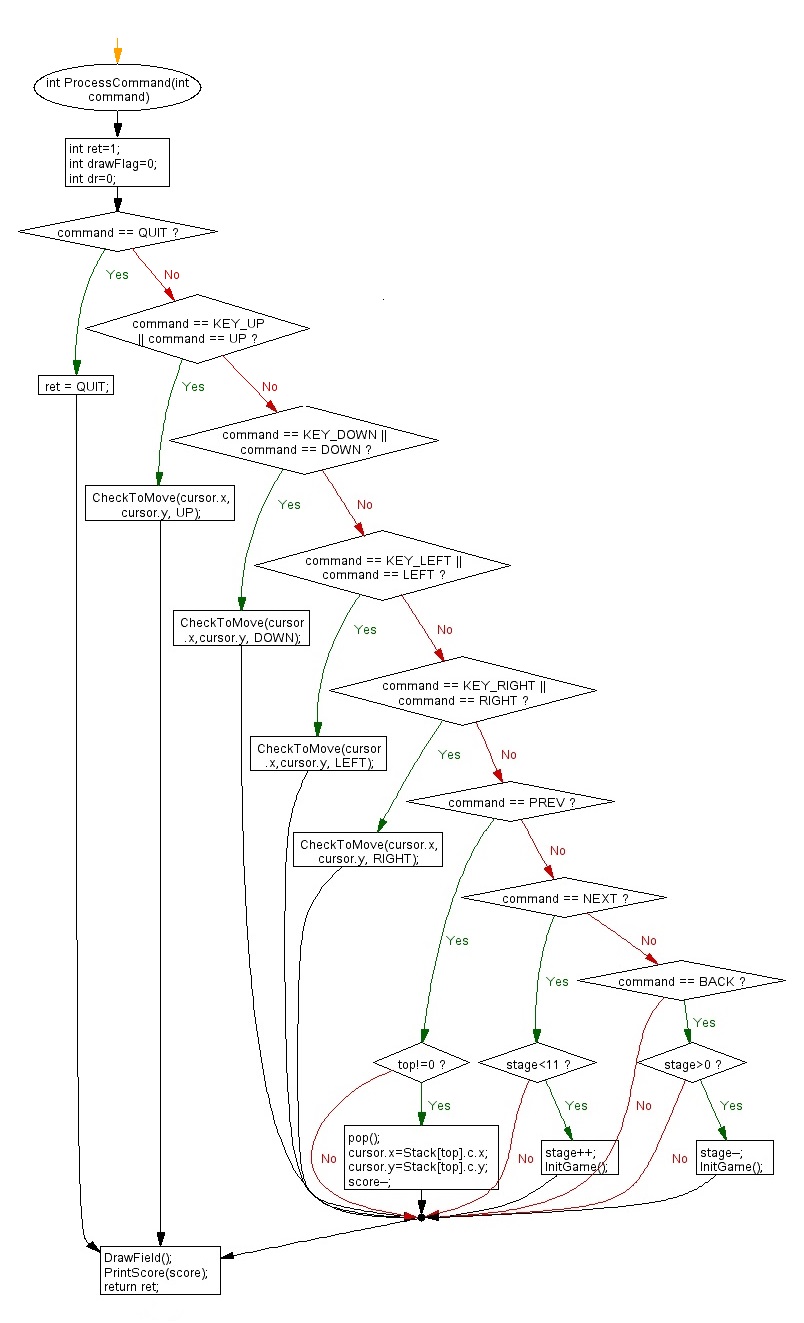
* InitGame



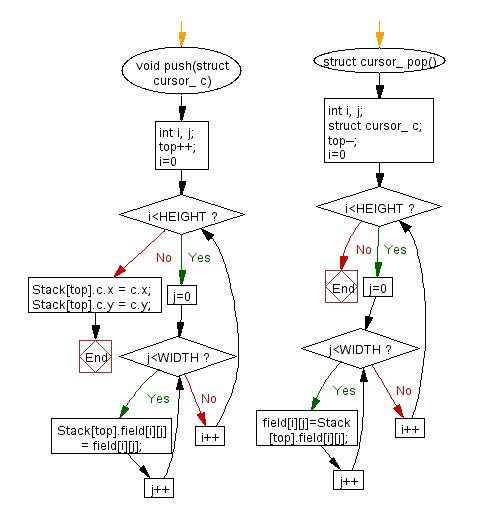
* newRank



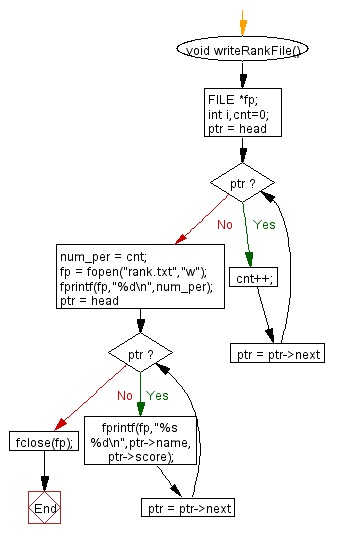
* ProcessCommand



* push, pop



* writeRankFile



1. 추가한 기능

- 사용자의 이동에 대한 되돌리기 기능

- Demo Play

- Score계산식 : ((1000/이동횟수)+(남은시간))\*(난이도) (난이도 = stage+1)