TAWS 2

TAWS2 is, as you may have guessed, the 2nd game in a series of "Text Adventures Without Swords". TAWS2 establishes that aside from the name you can expect very little similarities between the games in the TAWS series. Like TAWS1, everything you need to play the game exists within the game due to a help menu that is easy to find.

In TAWS2 you are attempting to navigate a grid "roguelike" style using the number pad for direction preprogramming your path before pressing enter. In that way this could be called a roguelike practice program. The way it is written, you gain more points for shorter paths. As a challenge to the reader, change it so that you have to take the longest path you can to the exit to net more points, receiving 0 points for a 9999 path.

TAWS2 was written by "Entar".

```
/* taws2.c listing begins: */

#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define EMPTY 0
#define MINE 1
#define START 2
#define FINISH 3

int grid[5][5];
long score=0;
unsigned long timestart;

int IsANumber(char c) {
  if (c == '0' || c == '1' || c == '2' || c == '3' || c == '4'
  || c == '5' || c == '6' || c == '7'|| c == '8' || c == '9')
    return 1;
  return 0;
}

int CheckPath (const char *path) {
  int pos[2]={0,0};
  int i=0, num;
  char c[2]="0\0";
  while (i<strlen(path)) {
    c[0] = path[i];
    if (!IsANumber(c[0])) {
      i++;
      continue;
    }
    num = atoi(c);
    if (num == 1 || num == 4 || num == 7)
      pos[0]--;
    if (num == 3 || num == 6 || num == 9)
      pos[0]++;
    if (num == 1 || num == 2 || num == 3)
      pos[1]--;
    if (num == 7 || num == 8 || num == 9)
      pos[1]++;

    if (pos[0] < 0 || pos[0] > 4 || pos[1] < 0 || pos[1] > 4)
      return MINE; // went out of bounds, invalid.
    if (grid[pos[0]][pos[1]] == FINISH)
      return FINISH;
    if (grid[pos[0]][pos[1]] == MINE)
      return MINE;

    i++;
  }
  return MINE;
}

int CheckGrid(void) {
  return 1;
}

void ConstructGrid (void) {
  int x, y, num;
  do {
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
    for (x=0; x < 5; x++)
      for (y=0; y < 5; y++)
        grid[x][y]=EMPTY;

    grid[0][0] = START; // start
    grid[4][4] = FINISH; // finish

    num=0;
    while (num < 6) {
      x = rand()%5;
      y = rand()%5;
      if (grid[x][y] == EMPTY) {
        grid[x][y] = MINE;
        num++;
      }
    }
  } while (CheckGrid() == 0);
}

char GetSym (int type) {
  if (type == START)
    return 'S';
  else if (type == FINISH)
    return 'F';
  else if (type == MINE)
    return '*';
  else
    return ' ';
}

void DisplayGrid (void) {
  printf("_____________________\n");
  printf("| %c | %c | %c | %c | %c |\n",
    GetSym(grid[0][4]),GetSym(grid[1][4]),GetSym(grid[2][4]),
    GetSym(grid[3][4]),GetSym(grid[4][4]));
  printf("---------------------\n");
  printf("| %c | %c | %c | %c | %c |\n",
    GetSym(grid[0][3]),GetSym(grid[1][3]),GetSym(grid[2][3]),
    GetSym(grid[3][3]),GetSym(grid[4][3]));
  printf("---------------------\n");
  printf("| %c | %c | %c | %c | %c |\n",
    GetSym(grid[0][2]),GetSym(grid[1][2]),GetSym(grid[2][2]),
    GetSym(grid[3][2]),GetSym(grid[4][2]));
  printf("---------------------\n");
  printf("| %c | %c | %c | %c | %c |\n",
    GetSym(grid[0][1]),GetSym(grid[1][1]),GetSym(grid[2][1]),
    GetSym(grid[3][1]),GetSym(grid[4][1]));
  printf("---------------------\n");
  printf("| %c | %c | %c | %c | %c |\n",
    GetSym(grid[0][0]),GetSym(grid[1][0]),GetSym(grid[2][0]),
    GetSym(grid[3][0]),GetSym(grid[4][0]));
  printf("---------------------\n");
  printf("\n");
}

int main (int args, char *argc[]) {
  FILE *f;
  char buf[64];
  char temp[64];
  char path[64];
  int add;
  long oldscore;
  unsigned int oldtime;
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
  unsigned int start = time(NULL);
  unsigned int spent;
  srand(time(NULL));
  printf("\n");
  printf("************************************************************\n");
  printf("Welcome to TAWS (Text Adventure Without Swords) volume 2!\n");
  printf("************************************************************\n");
  printf("\n");
  ConstructGrid();
  DisplayGrid();
  timestart = time(NULL);
  while (1) {
    printf("\n");
    printf("Enter command here: ");
    scanf("%s", buf);
    if (!strcmp(buf, "show"))
      DisplayGrid();
    else if (buf[0] == '5' || !strcmp(buf, "new")) {
      ConstructGrid();
      DisplayGrid();
      timestart = time(NULL);
    } else if (!strcmp(buf, "score"))
      printf("Score: %li\n", score);
    else if (!strcmp(buf, "help")) {
      printf("The object of TAWS v2 is to enter a sequence\n");
      printf("of numbers (on the numpad) that matches a path to\n");
      printf("the finish (F) from the start (S). Avoid mines (*).\n");
      printf("For example, 8 moves up, 9 moves up and right, and 2 moves "
        "down.\n");
      printf("-- Commands --\n");
      printf("#####: Type a series of numbers and hit enter to enter a path.\n");
      printf("'score': Show your current score.\n");
      printf("'new' or '5': Generate a new board.\n");
      printf("'show': Display current board.\n");
      printf("'help': Show this help.\n");
      printf("'highscore': See a saved highscore.\n");
      printf("'exit': Exit the game, but why would you want that?\n");
      printf("--   Tips   --\n");
      printf("The shorter your path and faster you do it, the more points you "
        "get!\n");
    } else if (!strcmp(buf, "TAWS")) {
      printf("TAWS is most commonly known as Text Adventure Without Swords.\n");
      printf("However, it has other meanings as well, including:\n");
      printf("Timed Avoidance Withstanding Simulation\n");
      printf("Torrential Admission of Watery Soup\n");
      printf("Timetable Antithesis Waking Somebody\n");
      printf("Top Ankle Wise Surrender\n");
      printf("... or just TAWS.\n");
    }
    else if (!strcmp(buf, "highscore")) {
      printf("Please enter a name: ");
      scanf("%s", temp);
      sprintf(path, "%s.sav", temp);
      f = fopen(path, "rb");
      if (f) {
        fread(&oldscore, sizeof(long), 1, f);
        fread(&oldtime, sizeof(unsigned int), 1, f);
        fclose(f);
        printf("Old score for %s: %li in %u seconds.\n",
          temp, oldscore, oldtime);
      } else
        printf("Failed to open save file %s.\n", path);
    }
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
    else if (!strcmp(buf, "quit") || !strcmp(buf, "exit"))
      break;
    else if (IsANumber(buf[0])) {
      int result = CheckPath(buf);
      if(result == FINISH) {
        float timediff = (float)(time(NULL)-timestart);
        if (timediff > 0)
          add = 50-(strlen(buf)-4)*4
            + (int)(50.0f*(float)strlen(buf)/(float)(time(NULL)-timestart));
        else
          add = 50-(strlen(buf)-4)*4 + (int)(50.0f*(float)strlen(buf));
        printf("Score: %li+%i=%li\n", score, add, score+add);
        score += add;
      } else {
        printf("You didn't make it to the finish!\n");
        printf("Score: %li-250=%li\n", score, score-250);
        score -= 250;
      }
      ConstructGrid();
      DisplayGrid();
      timestart = time(NULL);
    }
    else
      printf("Come on, type something real (try 'help').\n");
  }
  spent = time(NULL)-start;
  printf("You got %li points in %u minutes and %u seconds.  Good job!\n", score,
(spent/60), (spent%60));
  printf("Please enter your name: ");
  scanf("%s", buf);
  sprintf(temp, "%s.sav", buf);
  f = fopen(temp, "rb");
  if (f) {
    fread(&oldscore, sizeof(long), 1, f);
    fread(&oldtime, sizeof(unsigned int), 1, f);
    fclose(f);
  } else {
    oldscore = 0;
    oldtime = 0;
  }
  if (score > oldscore) {
    f = fopen(temp, "wb");
    if (f) {
      printf("New personal highscore: %li in %u seconds!\n", score, spent);
      fwrite(&score, sizeof(long), 1, f);
      fwrite(&spent, sizeof(unsigned int), 1, f);
    }
  }
  if (oldscore > 0)
    printf("Old personal highscore: %li in %u seconds.\n", oldscore, oldtime);

  printf("Well wasn't that fun? See you next time!\n\n");
  return 0;
}
```