Design Report

Travelling Salesman Problem Game!

Course Code: COS10009

Course Name: Introduction to Programming

Name: Ahmed Tarek AbdelRaouf Bahloul
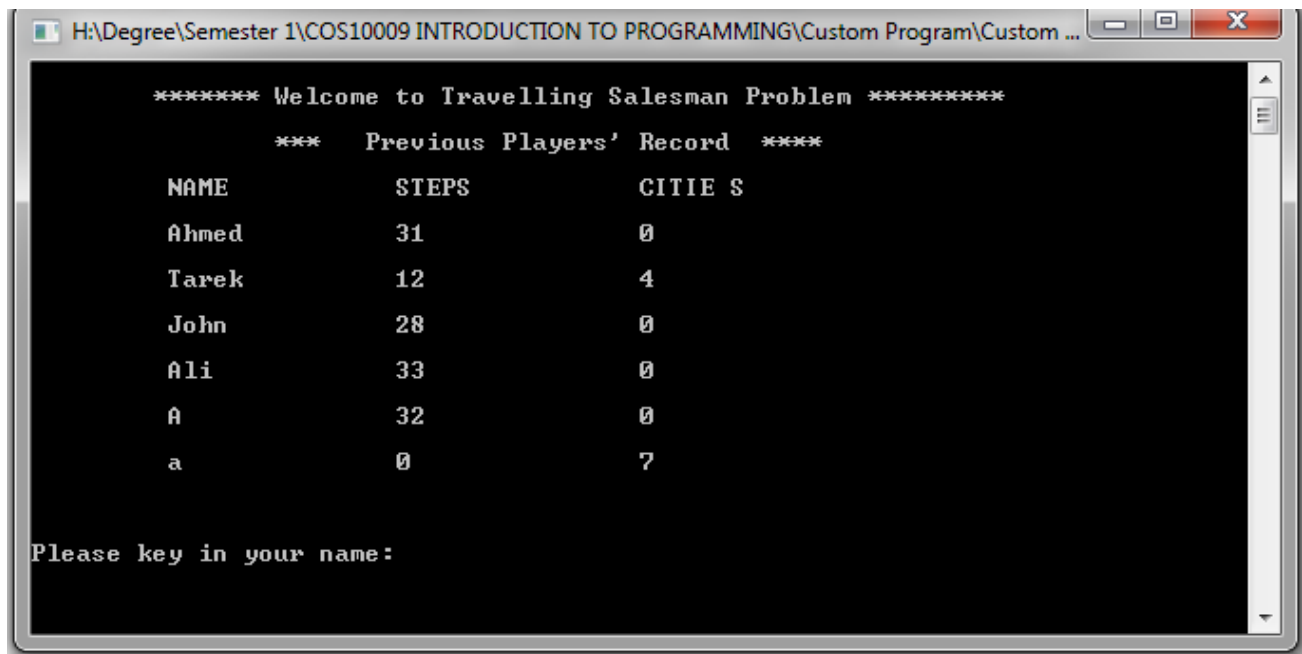
Student Id: 100089038

User Manual

Background of the game:

The traveling salesman problem (TSP) is an algorithmic problem tasked with finding the shortest route between a set of points and locations that must be visited. The salesman 's goal is to keep the distance travelled as low as possible.
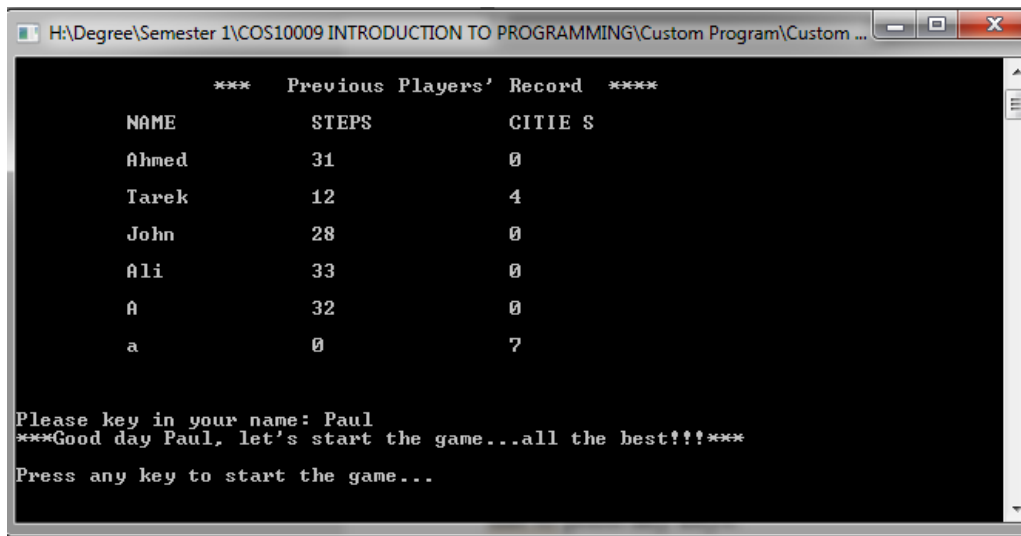
How to play:

First, the player will be in the main menu. The player will be able to see the previous players' records of the game. To proceed, the player will have to key in their first name on the 'Please key in your name:' area and press the Enter key to continue.

After pressing the Enter key, it will show a greeting message. To proceed, the player has to press any keys.

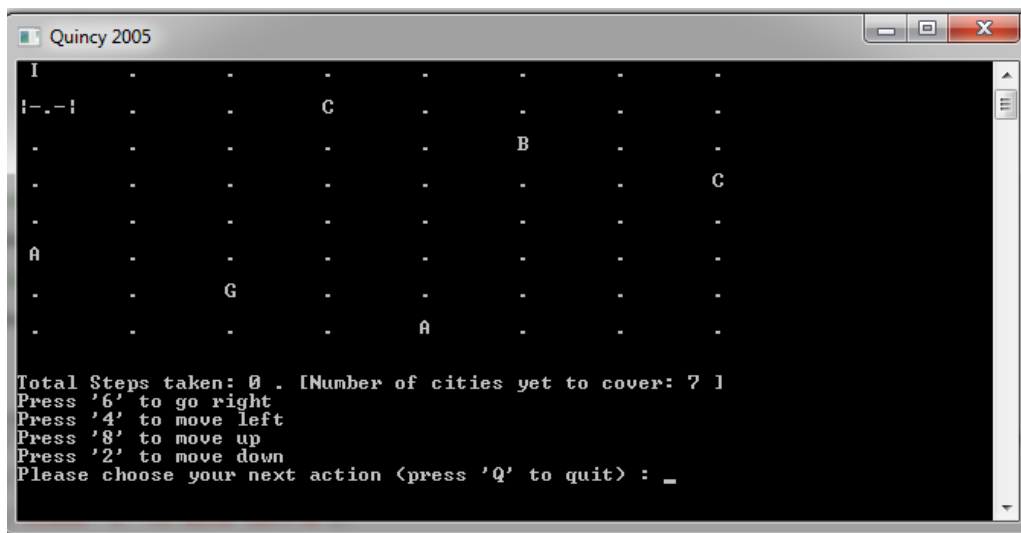
```
H:\Degree\Semester 1\COS10009 INTRODUCTION TO PROGRAMMING\Custom Program\Custom ...

            ***    Previous Players' Record    ****
        NAME              STEPS           CITIE S
        Ahmed             31              0
        Tarek             12              4
        John              28              0
        Ali               33              0
        A                 32              0
        a                 0               7


Please key in your name: Paul
***Good day Paul, let's start the game...all the best!!!***

Press any key to start the game...
```

After entering the Enter key, the player will now be in the in-game window. There are 8 rows and 8 columns. Random letters are in each grid. The user only can choose to move RIGHT or move LEFT or move DOWN or move UP. Each movement will be counted and the steps will be updated. The user can look at their current steps at the section which says 'Total Steps taken:' while playing. The aim of the game is to get as many points as the player can from the starting area until the number of Cities become 0.


```
Quincy 2005

 I         .        .        .        .        .        .        .
!-.-!      .        .        C        .        .        .        .
 .         .        .        .        .        B        .        .
 .         .        .        .        .        .        .        C
 .         .        .        .        .        .        .        .
 A         .        .        .        .        .        .        .
 .         .        G        .        .        .        .        .
 .         .        .        .        A        .        .        .

Total Steps taken: 0 . [Number of cities yet to cover: 7 ]
Press '6' to go right
Press '4' to move left
Press '8' to move up
Press '2' to move down
Please choose your next action (press 'Q' to quit) : _
```

Buttons that players can USE:

- ➢ Press '6' to go right
- ➢ Press '4' to move left
- ➢ Press '8' to move up
- ➢ Press '2' to move down
- ➢ Press 'Q' to quit

```
Total Steps taken: 0 . [Number of cities yet to cover: 7 ]
Press '6' to go right
Press '4' to move left
Press '8' to move up
Press '2' to move down
Please choose your next action (press 'Q' to quit) : _
```

Note:

You do not have to press the enter key after typing any of the keys such as 6, 4, 8, 2, 'Q'/'q'. It runs by its own.

After finishing the game, a congratulation message will appear. Moreover, their name as well as their steps will be recorded after it reaches the ending area.
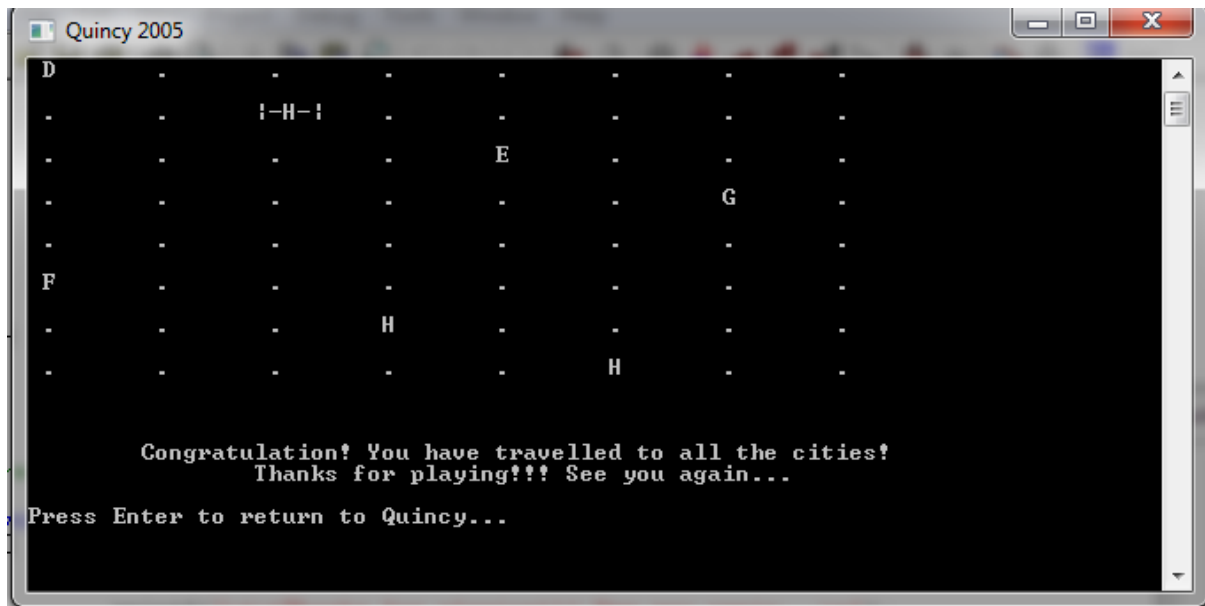
Short Descriptions

#define NROW 8

#define NCOL 8

Function: Define how many rows and columns we have.

void inputName(char name[])

Function: This function allows the player to key in his/her name to the game. The name will then be used to print a greeting message before proceeding to the in-game window. Moreover, the name will be used again in the writefile() function.

void initialize(char a[NROW][NCOL])

Function: This function initialize the array.

void disp_arr(char a[NROW][NCOL])

Function: initialize this array with specified values. Moreover, this function will then be used again to print a board function.

char rndch(void)

Function: This function allows to generate a random alphabet in the array.

void init(char a[NROW][NCOL])

Function: This function allows to generate a '.' in the array.

int playerAction(int *steps, int *citiescount)

Function: This function allows the user to give his/her input for the direction that the user wants to go (6 or 4 or 8 or 2). The function also allows the user to quit the game halfway with the input 'Q '. The function returns the choice.

void loopchoice(char a[NROW][NCOL], char choice, int *row, int *col, int *steps, int *citiescount)

Function: This function is used to loop the other functions as well as this function if the user gives the wrong input.

void endgame()

Function: This function is to print the congratulation message to the user when he/she wins.

void writefile(char name[20],int steps, int citiescount)

Function: This function is used to write information of the user such as user name, the steps taken by the user and number of cities yet to cover to the text file named playerinfo.txt.

void readfile()

Function: This function is to read the text file as well as printing the information of the file to the game.

int main()

Function: All the functions are in the main to make the game complete.

Flowchat for main()

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
┌───────────────────────────────────────────────────────┐
│              char ar[NROW][NCOL];                      │
│  int x, y, n=1, row=0,col=0, steps=0, citiescount = 7; │
│              char name[20], choice;                    │
│         srand((unsigned int)time(NULL));               │
│                  initialize(ar);                       │
└───────────────────────────────────────────────────────┘
                           │
                           ▼
      ╱──────────────────────────────────────────╲
     ╱   printf("\n\t******* Welcome to Travelling  ╲
    ╱      Salesman Problem! *********\n\n");         ╲
   ╱      printf("\t\t***   Previous Players' Record    ╲
   ╲          ****\n \n");                             ╱
    ╲    printf("\t NAME\t\tSTEPS\t\tCITIE S \n\n");  ╱
     ╲──────────────────────────────────────────────╱
                           │
                           ▼
              ┌───────────────────────┐
              │      readfile();      │
              │   inputName(name);    │
              └───────────────────────┘
                           │
                           ▼
              ┌───────────────────────┐
              │      getchar();       │
              │    system("cls");     │
              └───────────────────────┘
                           │
                           ▼
              ┌───────────────────────┐
              │       init(ar);       │
              │     disp_arr(ar);     │
              └───────────────────────┘
                           │
```

Yes

choice = playerAction(&steps, &citiescount);
loopchoice(ar, choice, &row, &col, &steps, &citiescount);
if(citiescount==0)
endgame();
break;

while(choice!='Q');

printf("\t\tThanks for playing!!! See you again...\n");

writefile(name, steps, citiescount);
return 0;

End