//Maxwell Maia 21236277

#include <stdio.h>

#include <string.h>

#include <cctype>

#pragma warning(disable:4996)

typedef struct {

char description[100];

int n, s, e, w, in, out;

} location;

location locations[40];

int numLocations = 0;

char locationsFile[] = "adventure\_locations.txt";

FILE\* openFileForReading(char\* filename) {

FILE\* file\_ptr = fopen(filename, "r");

if (file\_ptr == NULL)

printf("Could not open %s\n", filename);

return file\_ptr;

}

bool readLocations() {

FILE\* file\_ptr = openFileForReading(locationsFile);

if (file\_ptr == NULL)

return false;

numLocations = 0;

int readHeaderLines = 0;

char line[200];

while (fgets(line, 99, file\_ptr) != NULL) {

if (readHeaderLines < 2)

readHeaderLines++;

else {

numLocations++;

location l;

int locNum;

sscanf(line, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%[^\t]\n", &locNum, &l.n, &l.s, &l.e, &l.w, &l.in, &l.out, l.description);

int len = strlen(l.description);

l.description[len - 1] = '\0'; // remove \n from the string

if (l.description[len - 2] == '\r')

l.description[len - 2] = '\0'; // aslo remove \r from the string

locations[numLocations] = l; // the 1st location is 1 (not 0) so we can use 0 to mean 'nothing' in adventure\_locations.txt

}

}

return true;

}

int loop()

{

int currentLocation = 4;

int lastKnownLocation = 0;

char userInput[20] = "";

printf("---------------------------------------\n\nIt's a bright day in Galway. You are %s.\nYou are ready to walk.\n> ", locations[currentLocation].description);

if (scanf("%s", userInput) == 1) //get user input

{

}

else {

printf("Error reading user input.\n");

}

for (int i = 0; i < strlen(userInput); i++) //sets userIn to lower case

userInput[i] = tolower(userInput[i]);

while (1)

{

//commands

if (strcmp(userInput, "quit") == 0)

{

printf("Bye!\n");

break;

}

if (strcmp(userInput, "help") == 0)

{

printf("\nI know these commands.\nn, s, e, w, in, out, look, help, quit.\n");

}

if (strcmp(userInput, "n") == 0) //go north

{

printf("You walk north.\n\n");

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].n;

printf("\n%s\n", locations[currentLocation].description);

}

if (strcmp(userInput, "e") == 0)

{

printf("You walk east.\n\n");

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].e;

printf("\n%s\n", locations[currentLocation].description);

}

if (strcmp(userInput, "s") == 0)

{

printf("You walk south.\n\n");

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].s;

printf("\n%s\n", locations[currentLocation].description);

}

if (strcmp(userInput, "w") == 0)

{

printf("You walk west.\n\n");

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].w;

printf("\n%s\n", locations[currentLocation].description);

}

if (strcmp(userInput, "in") == 0)

{

printf("You go inside.\n\n");

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].in;

printf("\n%s\n", locations[currentLocation].description);

}

if (strcmp(userInput, "out") == 0)

{

printf("You go outside.\n\n");

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].out;

printf("\n%s\n", locations[currentLocation].description);

}

if (strcmp(userInput, "look") == 0)

{

printf("\n%s\n", locations[currentLocation].description);

}

if (currentLocation == 0)

{

printf("But you can't go that way.\n\n");

currentLocation = lastKnownLocation;

printf("\n%s\n", locations[currentLocation].description);

}

printf("\nYou are ready to walk.\n> ");

if (scanf("%s", userInput) == 1) //get user input for next loop

{

}

else

{

printf("\nError reading in user input.");

}

for (int i = 0; i < strlen(userInput); i++) //sets userIn to lower case

userInput[i] = tolower(userInput[i]);

}

return 0;

}

int main() {

if (readLocations()) {

printf("Successfully read %d locations from file\n", numLocations);

loop();

}

}

