A picture containing graphical user interface

Description automatically generated

/\*gavin skehan

21440824

22/03/22\*/

#include <stdio.h>

#include <string.h>

#include <stdbool.h>

#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 main() {

if (readLocations()) {

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

for (int i = 1; i <= numLocations; i++) {

printf("Location %d is %s", i, locations[i].description);

if (locations[i].n > 0)

printf(" North leads to %s\n", locations[locations[i].n].description);

else

printf(" From there you cannot go north.\n");

}

}

}