//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;

typedef struct {

char name[50];

int location;

char description[100];

} object;

object arrObjects[100];

location locations[40];

int numLocations = 0;

int numObjects = 0;

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

char objectsFile[] = "adventure\_objects.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'; // also 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;

}

bool readObjects()

{

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

if (file\_ptr == NULL)

return false;

numObjects = 0;

int readHeaderLines = 0;

char line[200];

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

{

if (readHeaderLines < 2)

{

readHeaderLines++;

}

else

{

object o;

// int objNum;

sscanf(line, "%[^\t]%d\t%[^\n]", &o.name, &o.location, &o.description);

int len = strlen(o.description);

o.description[len] = '\0';

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

{

o.description[len - 2] = '\0';

}

arrObjects[numObjects] = o;

numObjects++;

}

}

return true;

}

int checkForItems(int currentLocation) //scans the object array for objects that are at the new location. And prints the descriptions of what's there.

{

printf("\nObjects here:\n");

int printNone = 1;

for (int i = 0; i < numObjects; i++) //loop through each object

{

if (currentLocation == arrObjects[i].location) //check if the location of the object matches the location of the player.

{

printf("%s\n", arrObjects[i].name); //printf the object name if it is at the current location.

printNone = 0;

}

}

if (printNone) //print the word none if no objects are found at the location.

{

printf("none\n");

}

printf("\n");

return 0;

}

int loop()

{

int currentLocation = 4;

int lastKnownLocation = 0;

//int currentObject = 0;

char userInput[20] = "";

char tempWord[50] = "";

char word[50] = "";

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

checkForItems(currentLocation);

printf("What do you want to do?\n> ");

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);

checkForItems(currentLocation);

}

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

{

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

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].e;

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

checkForItems(currentLocation);

}

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

{

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

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].s;

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

checkForItems(currentLocation);

}

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

{

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

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].w;

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

checkForItems(currentLocation);

}

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

{

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

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].in;

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

checkForItems(currentLocation);

}

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

{

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

lastKnownLocation = currentLocation;

currentLocation = locations[currentLocation].out;

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

checkForItems(currentLocation);

}

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

{

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

}

if (strcmp(userInput, "take") == 0) //TAKE

{

printf("What do you want to take...\nEnter the amount of words you wish to enter? > ");

int numWords = 0;

scanf("%d", &numWords);

//scanf("%[^\n]s", temp); this doesnt work >:(

for (int i = 0; i < numWords; i++)

{

printf("Enter word %d > ", (i + 1));

scanf("%s", tempWord);

strcat(word, tempWord);

if (i < numWords - 1)

{

strcat(word, " ");

}

}

int printNone = 1;

for (int j = 0; j < numObjects; j++)

{

if (strcmp(word, arrObjects[j].name) == 0)

{

printf("You take %s.\n", word);

arrObjects[j].location = 0;

printNone = 0;

}

}

if (printNone)

{

printf("That object isn't here.");

}

strcpy(tempWord, "");

strcpy(word, "");

}

if (strcmp(userInput, "drop") == 0) //DROP

{

printf("What do you want to drop...\nEnter the amount of words you wish to enter? > ");

int numWords = 0;

scanf("%d", &numWords);

//scanf("%[^\n]s", temp); this doesnt work >:(

for (int i = 0; i < numWords; i++)

{

printf("Enter word %d > ", (i + 1));

scanf("%s", tempWord);

strcat(word, tempWord);

if (i < numWords - 1)

{

strcat(word, " ");

}

}

int printNone = 1;

for (int j = 0; j < numObjects; j++)

{

if (strcmp(word, arrObjects[j].name) == 0)

{

printf("You drop %s.\n", word);

arrObjects[j].location = currentLocation;

printNone = 0;

}

}

if (printNone)

{

printf("%sThat object isn't in your inventory.", word);

}

strcpy(tempWord, "");

strcpy(word, "");

}

if (strcmp(userInput, "examine") == 0) //EXAMINE

{

printf("What do you want to examine?\nEnter the amount of words you wish to enter? > ");

int numWords = 0;

scanf("%d", &numWords);

//scanf("%[^\n]s", temp); this doesnt work >:(

for (int i = 0; i < numWords; i++)

{

printf("Enter word %d > ", (i + 1));

scanf("%s", tempWord);

strcat(word, tempWord);

if (i < numWords - 1)

{

strcat(word, " ");

}

}

int printNone = 1;

for (int j = 0; j < numObjects; j++)

{

if (strcmp(word, arrObjects[j].name) == 0)

{

if (arrObjects[j].location == currentLocation || arrObjects[j].location == 0)

{

printf("You examine %s...\n", word);

printf("%s", arrObjects[j].description);

printNone = 0;

}

}

}

if (printNone)

{

printf("%s is not in the area or your inventory.\n", word);

}

strcpy(tempWord, "");

strcpy(word, "");

}

if (strcmp(userInput, "inventory") == 0) //INVENTORY

{

printf("Items in your inventory:\n");

int printNone = 1;

for (int i = 0; i < numObjects; i++)

{

if (arrObjects[i].location == 0)

{

printf("%s\n", arrObjects[i].name);

printNone = 0;

}

}

if (printNone)

{

printf("None... get some bruh.\n");

}

}

if (currentLocation == 0)

{

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

currentLocation = lastKnownLocation;

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

}

printf("\nWhat do you want to do?\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);

if (readObjects())

{

printf("Successfully read %d objects from file.\n", numObjects);

//printf("\n\n\nLOOK HERE\n%s\t%d\t%s\ndone", arrObjects[0].name, arrObjects[0].location, arrObjects[0].description);

loop();

}

}

}

