HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING
BBM104

Prepared by Burak YILMAZ
Number : 21627868
E-Mail : burak040898@gmail.com
Subject : Assignment 1

MAIN GOAL

In this assignment, our main goal is to use two dimensional arrays and dynamic memory allocation. Also understanding more about pointers.

PROBLEM

Implementing a game which is runned with three inputs file(characters.txt, commands.txt,output.txt).This game is based on a
fantastic adventure board game and runs on a given multi-dimensional array. There will be two sides (monsters, heroes) and they will
attack to each other. After given commands are executed, my program should print related outputs into a text file such as current
situation of the map and status of the characters. Also, these two sides have got specialties like damage point and health point moreover
only heroes gain xp point during the game if they kill a monster while attacking.

SOLUTION

First I divided the problem into small pieces so it is easier to find solutions and handle with each small problems. Firstly ˝ focused on
how to store the data then ˝ thought how to execute each command. I decided to use functions for both of them. Finally, I have to decide how
to create 2D map. Then I decided to use malloc and pointers for that. After I found all the steps I had to do I started to write.

STRUCTURES

I used different 5 structures one of them for initializing map two of them for store commands and two of them for store characters but I
didn't separate characters into two structs I behave it like one struct so it is easier to keep in mind. First struct is CHARACTER and it contains
type, name, damage, health, xp, pos_x and pos_y. Second struct is CHARACTERS struct and it contains character_count and CHARACTER **characters
Third struct is COMMAND struct and it contains two arrays(cmd and params).Fourth struct is COMMANDS struct and it contains
cmd_count and COMMAND **commands.And the last struct is MAP struct and it contains rows,columns and CHARACTER ***character_data.
For storing characters I used malloc dynamic array and size of this dynamic array is the total size of CHARACTERS struct. For storing commands I also used malloc dynamic array and size of this dynamic array is the total size of COMMANDS struct. For map again I used malloc dynamic array and size of
this dynamic array is multiplication of (rows*columns) and size of CHARACTER**.

ALGORITHM

Firstly I read two input files. After reading I store the data in a dynamic arrays. That’s why first I need to count how many lines in each file
that's why I read each file two times first get the line and second for storing the data.After that I opened the output file to write the
correct data. After storing the characters and commands I initialize the 2D map and put the characters correct locations. Then I finished
the first part after this ˝ started to write functions for each command. I will explain fuctions below.

FUNCTIONS

CHARACTERS *read_character_file:

-This function reads the first input until the pointer returns null and stored the data in dynamic array. If it cannot read it gives an error.
It contains another function the name of other function is get_line function i will explain below.

COMMANDS *read_commands_file:

-This function reads the first input until the pointer returns null and stored the data in dynamic array. If it cannot read it gives an error.
It contains another function the name of other function is get_line function.It almost same with the above function the only difference
is the size of dynamic arrays allocated on the memory.

FILE *open_output_file:

-It opens an the third command argument in writing mode.

int get_file_line_count:

-It counts the lines in each file and returns the integer value of total lines.

char *get_line:

-It holds the adress of the current line and returns it until the newline character or EOF comes.

MAP *init_map:

-It allocates 2D dynamic array from the memory and initializing the map.

void put_characters:

-It puts the characters on the correct location of 2D array.

void attack:

-It provides attack command. It compares the parameter with the type if the commands 'attack monsters' then it ignores the heroes. Also
it checks few thinks like the location on the map and it finds the locations which can attackers attack. It contains ternary operations
min and max which i defined above the main. Min and max for determining the attackers max and min coordinates to attack. Finally it checks

the healts of the each type at the end of the function because if all the monsters or all the heroes are dead it gives an error and exit
from the program.

void move_characters:

-It moves the characters on the correct location on the map. It is for only heroes and it checks many things like if the location is occupied
by another character it gives an error or if the character which is trying to move is dead then it gives another error. Also, it checks the
borders of the map because characters only move on the map we define if they try to reach out of the map it gives an error "there is a wall"

void print_character_status:

-It is writing the status of the type which is given and it writes the output file for the preferred type. If the preferred type is monster
it only shows health if it is heroes it shows health and xp points for each.

void print_map:

-It shows the current map status if the location is empty on the map it shows "." if it is occupied it shows the first letter of the name.

void exit_program:

-It frees the allocated memory.

COMMENTS

-When I first read the assignment paper I assumed it was going to be easy but forgot we are writing in C language that's why it took
a lot of time and sometimes it got harder than I thought otherwise I consolidate pointers and functions also I learnt new knowledge
about structs how to use them efficiently and also learnt how to use malloc and calloc and free.