

EHB 110E Final Project Report

Alp Eren Kiyak, 040200100

08/01/2022

Introduction

Goal of this project is making a fast writing game with C language. The game gives some words depends on level. Words must be texted before certain time that reduces at each level. The time calculating with a bomb deployed from plane. Bomb, plane and city has ascii animation on terminal. If bomb hits and explodes city game overs. Figure 1.0 is from terminal and there is plane at top, game score at top left, the at bottom, there is a text linked with bomb paces around. And there is end game screen(figure1.1).



Figure0.1=Picture of Gameplay

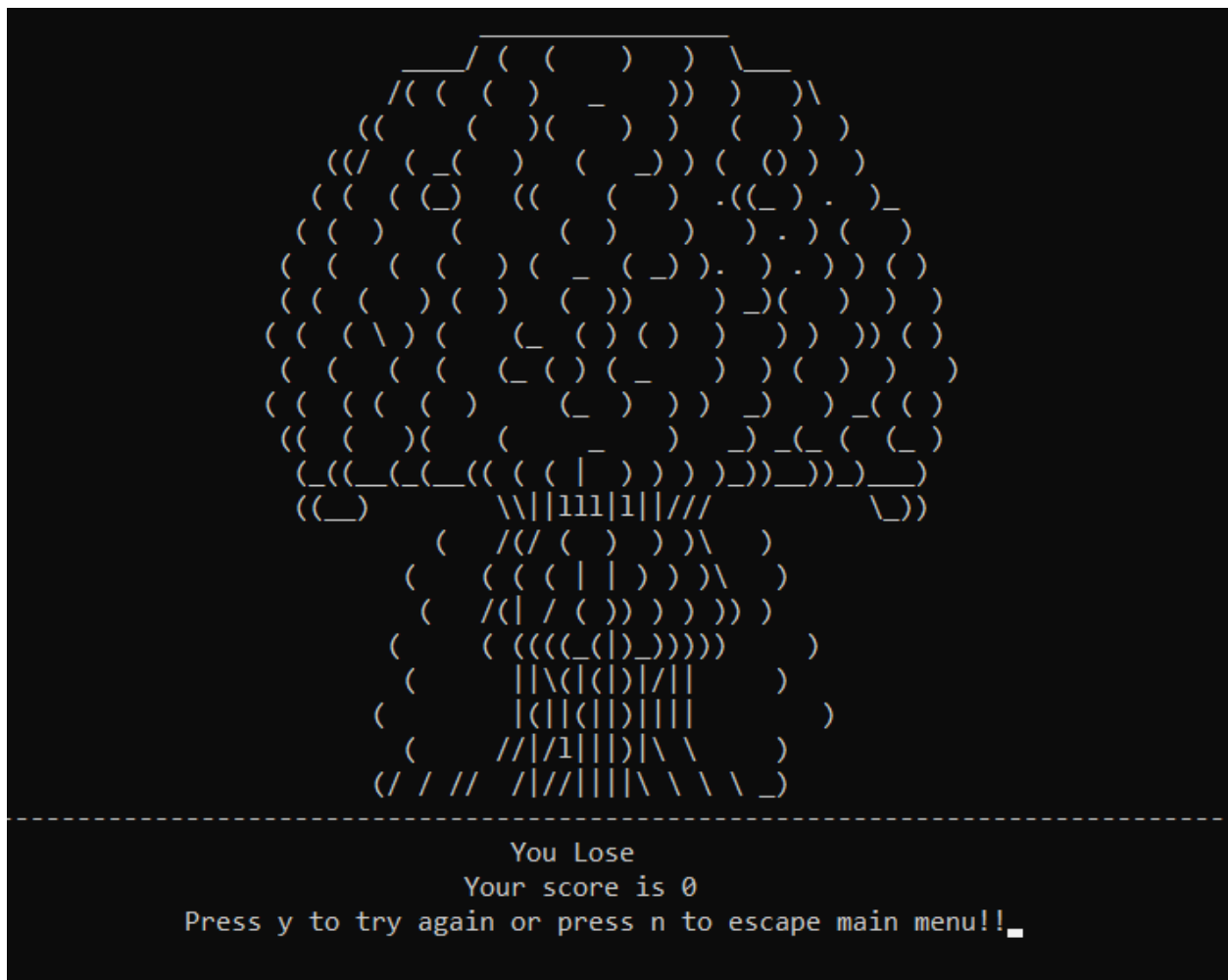


Figure 1.1=End game screen

Beside the gameplay there should be 2 type of menu for controlling game data. First one is main menu that have 4 option. First one 'New Game' option that starts new game. Second one is 'Load Game' that loads a saved from a file game. Third one is 'Save Game' that saves current game to a file. The last selection is 'Quit Game' that closes the application. Other type of menu is pause menu you can only reach it whit esc key while started a game and playing it. The only difference between both is pause menu has 'Return Game' selection as fourth selection then there is 'Quit Game' selection at last. The figure 2.0 shows how each state looks like at menus. They can switched with up and down direction keys.



Figure 2.0='Save Game' selection of menu.

Implementation

At the beginning of the code. I gathered all variables and arrays about game's data in a structure. When application starts first it setups texts from a file then setups some array and variables to organizing data. First it checks length of string. And append them to levels according to length of string(Figure3.0).

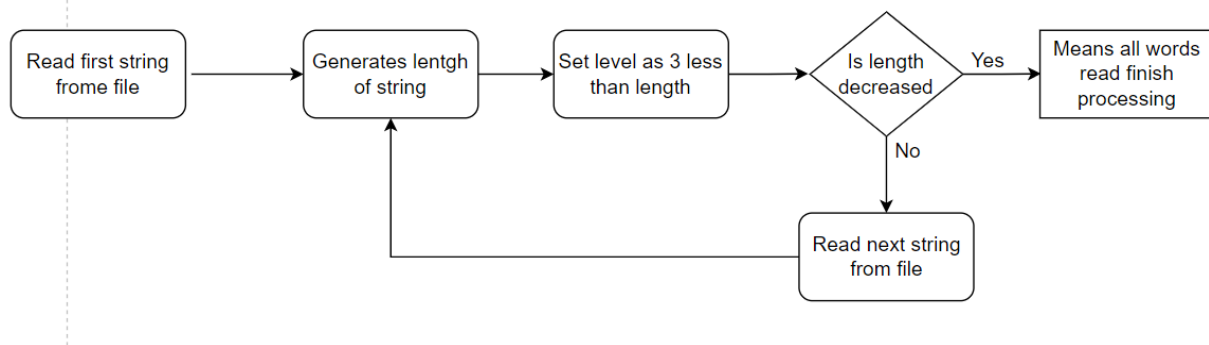


Figure 3.0=Flow chart of text setup.

After setup is done code generates a swapper between menu, gameplay, pause menu and end game screen. Swapper starts from main menu to switch buttons between starts new game, loads game, saves current data and exits from application. (Figure 3.1)

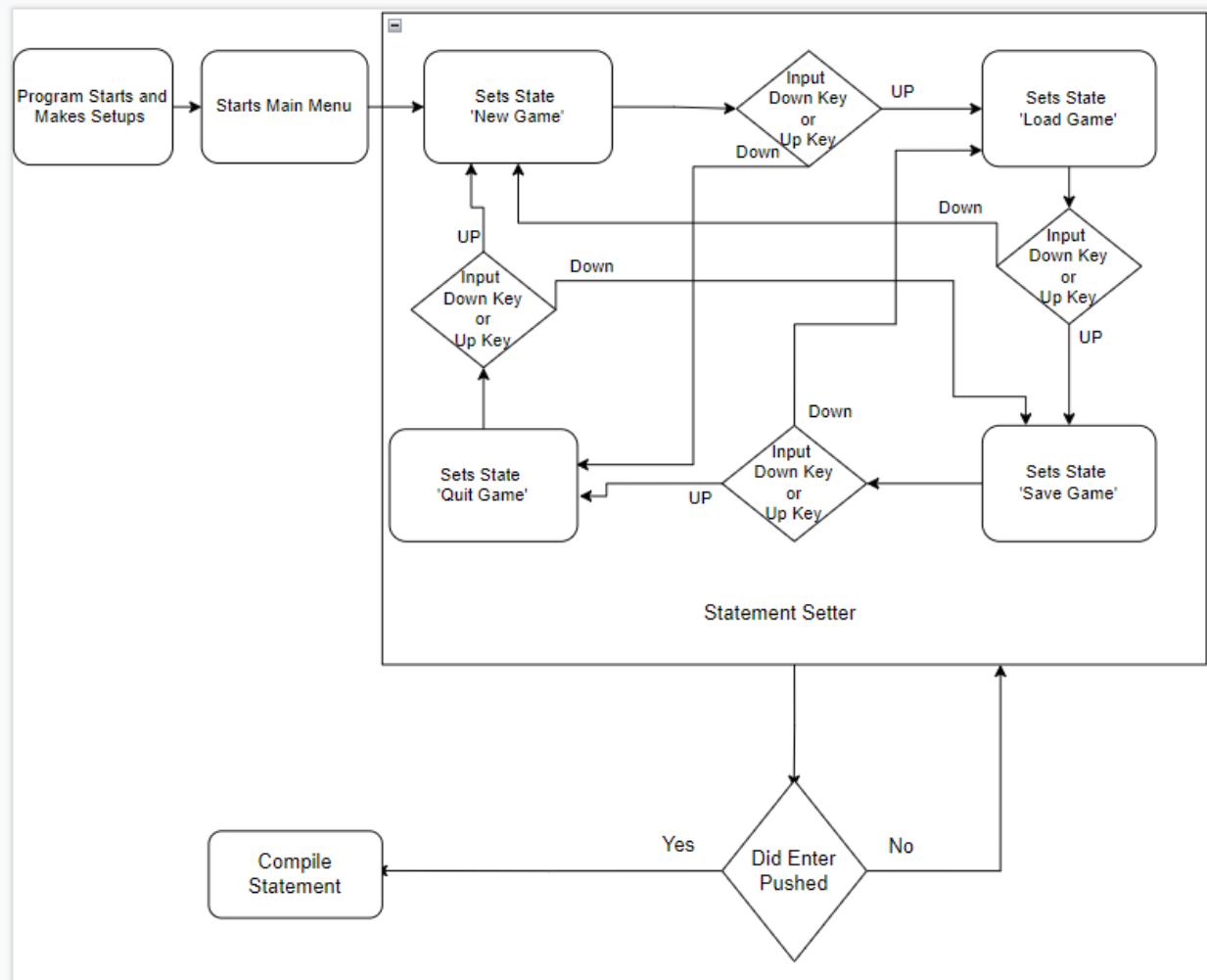


Figure 3.1= Follow chart of Main Menu.

Main menu's animations are made with delays that checks keyboard while waiting clock pass. So it can read inputs from keyboard and while in main menu it checks enter key, up arrow key and down key. It prints the structures so fast and waits like 2.8 second so while it processing it acts like always listening the keyboard because of processing speed. And it's already the same at the gameplay but gameplay can have little lag on it at first levels because if player texts faster than a delay then it can execute two or more before printing current situation. Pause Menu has already the same structure with main menu but Pause Menu has one more state from main menu with the same architecture (Figure 3.2).

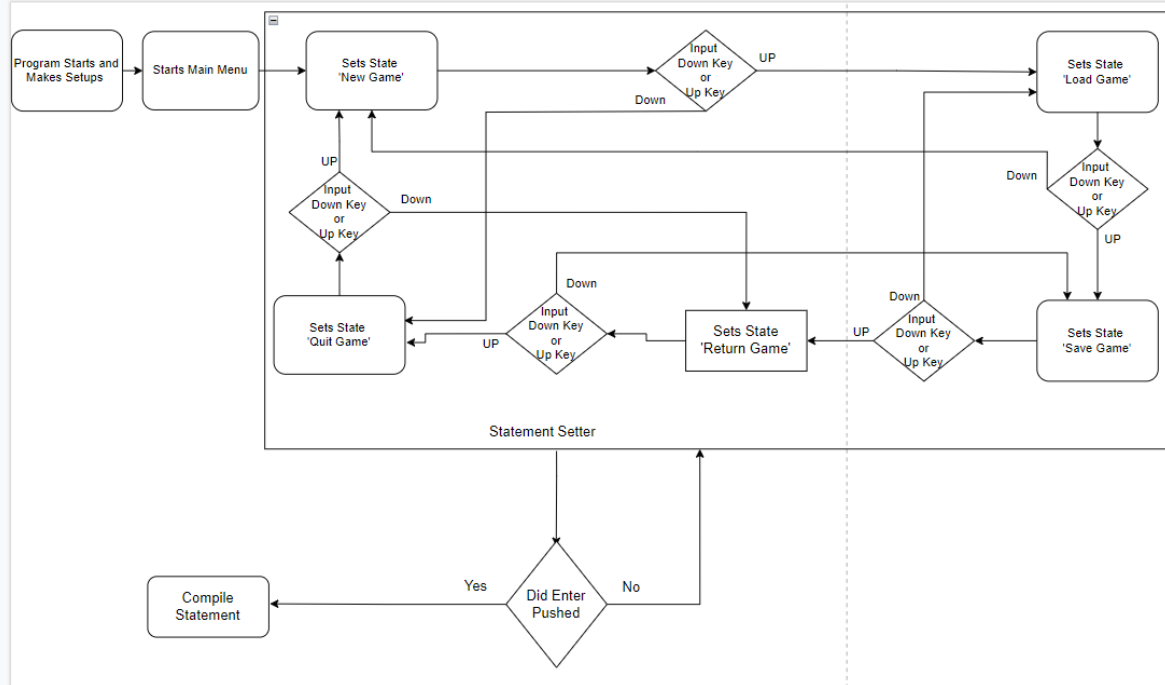


Figure 3.2 = Follow chart of Pause Menu

Gameplay looks like have different architecture but actually it's same with others. But gameplay has more if statement because gaming. It checks how much of string texted and while checking also checks for 'esc' button to pause game. Follow chart of gameplay is shows at figure 3.3. Application is linked with this follow charts. At last there is block diagram of application (Figure 3.4).

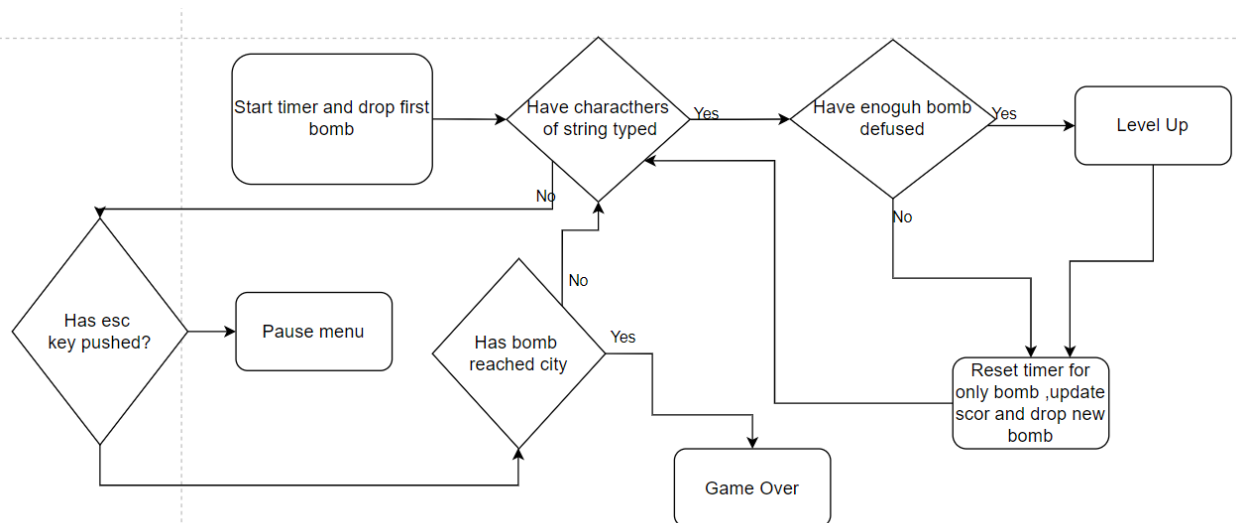


Figure 3.3= Follow chart of gameplay

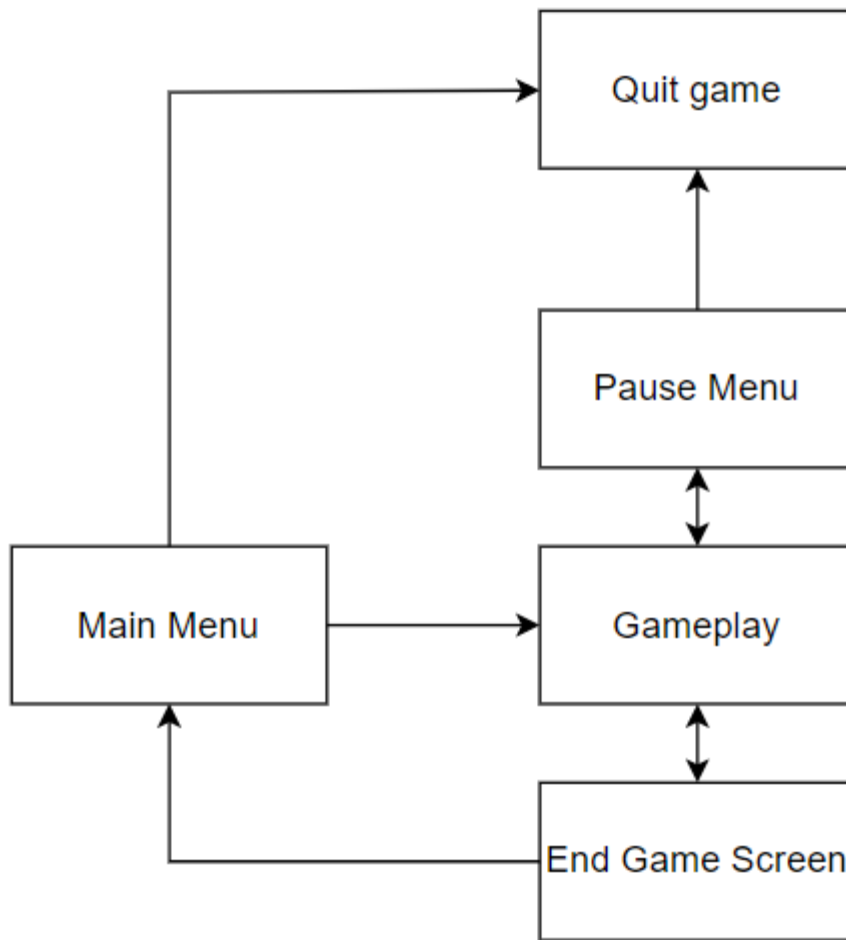


Figure 3.4= Block diagram of application.

Discussion

I made animations first. But after I implement the mechanics of gameplay. They didn't work properly and all lines were at main function. So I have to remake algorithm and rewrite them. Second I use more function and it was better. I think if I had enough time and permission I can add leveling more. For example each level can have their own time and you should survive till time ends. I wanted to add many other design features. I can make them as .txt file. For example it can be space age theme. I could also add more mechanic like city could fall part by part and you can till last part fall. There can be 2 player challenge also. But it will be more like race for writing. The player who texted the string first could send that bomb to other player. And while next word appear he can still defends his area with typing previous text. So he should double his speed. Or I don't know maybe he could see the next word before the enemy player to make the more fair. There was also saving and loading problems on file I searched like 1-1.5 hour to fix it. And taking words from file and organizing it was also challenging because you can't

estimate how many word could been in the file. So I just put a character at the end to avoid unknown values.