



9/12/2020

UCP Report: “Hot Bomb” Text Based Game

Muhammad Naufal Al Ghifari

ID: 90019280

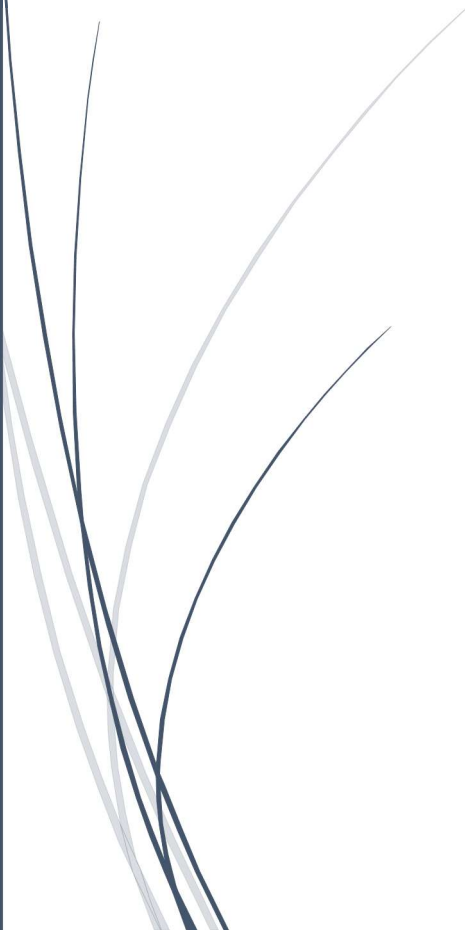


Table of Contents

1. Program Overview	1
2. Files and Functions.....	1
2.1 main.c.....	1
2.2 bombHeader.h	1
2.3 bombFunc.c.....	1
2.4 bombMainMenu.c	2
2.5 linkedList.c	2
2.6 linkedList.h	2
2.7 makefile.....	2
2.8 settings.txt	2
3. Gameplay and Demonstration	2
3.1 Setting up the game	2
3.2 Playing the game.....	3
3.3 Saving and loading a game.....	3
4. Discussion.....	3
4.1 General Issues	3
4.2 Save and Load features	3
4.3 User inputs	3

1. Program Overview

Hot Bomb is a turn-based text game with a fairly simple concept. There are 2 players and a bomb. The game handles players and bomb stats using structs, as for the moves, it stores it in a generic linked list. The game fits inside a little less than 1000 C code lines.

Game Features:

- Start a game
- Special Abilities
- Save Game
- Load Game

The game requires the following files:

- main.c
- bombHeader.h
- bombFunc.c
- bombMainMenu.c
- linkedList.c
- linkedList.h
- makefile
- settings.txt

The program uses the following libraries:

- stdio.h
- stdlib.h
- string.h
- ctype.h

2. Files and Functions

This section covers all files that is required by the game in order to run properly.

2.1 main.c

As any other C program, this game requires a main function, this function is stored in main.c. This function is responsible for creating structs, creating linked list, opening settings file and memory clean-ups. This function will prevent the game from running if the settings file is not given in the command line.

2.2 bombHeader.h

This is the “main” header file of this program as not only it contains declarations for most functions that composes the game, it also contains structs for the players, the bomb as well as the moves. This file does not contain any function.

2.3 bombFunc.c

This file contains the majority of functions of the game, these are some of the most important ones:

- readSettings() = This functions reads the settings file and assign the values to the correct structs. This function is capable of reading the values in settings file in any order.

- writeNewSettings() = This function writes to the settings file right before closing the program.
- startGame() = This function calls playersTurn() for each player's turn as long as both player's health are above 0.

2.4 bombMainMenu.c

BombMainMenu.c contains the main menu function that calls other functions of the game. The menu has been put to a separate file due to its length.

2.5 linkedList.c

Contains functions related to linkedlist, such as creating, adding a node, printing as well as freeing the list.

2.6 linkedList.h

Contains declarations for functions in linkedList.c as well as the linked list to save the game.

2.7 makefile

This file is responsible for compiling the various files of the game and make an executable. There are 2 versions of the game, the final product called "Hot_Bomb_Game" and a debug version which prints extra informations called "Hot_Bomb_Debug".

2.8 settings.txt

A text file containing player and bomb stats such as health, max health, damage, normal damage, as well as players special moves. Can be written in any order.

3. Gameplay and Demonstration

There are 2 players with various special abilities they can choose from (SKIP, HEAL, RESET, DOUBLE). Each player can hold the bomb for a desired period of time (between 1 to 5 seconds).

The bomb has a timer that reduces whenever a player holds the bomb, if said health reached zero, the bomb explodes and inflicts damage to the current holder. The first player hitting 0 health loses.

Each player can use their special move twice per game, the special moves are as follows:

- SKIP: Current player skips their move. In other words, holding the bomb for 0 sec.
- HEAL: This will fully heal current player to its maximum health.
- DOUBLE: Doubles the damage of the bomb once until it explodes.
- RESET: Resets the bomb timer to its max timer.

The players can save the game at any point by typing "3" when prompted, the program stops the current game and bring back the player to the main menu.

3.1 Setting up the game

To compile the program, type the following command:

make

Once compiled, run the game using the following command:

./HOT_BOMB_GAME settings.txt

(Note: if the settings file is not provided, the game will not run)

Next, the player will be shown the main menu, many things can be done from here such as starting a game, view and change settings, loading a game, and quitting the game. Players can view the game settings by typing:

2

The settings will be displayed, and the program asks if they wish to change any settings. Once done, the main menu will be displayed again.

3.2 Playing the game

To start a new game, type:

1

Each player will have their turn, here the playing player can either make a move by typing "1", use special move by typing "2" or save the game by typing "3". Once the game ends, the program will congratulate the winner, e.g. **"Player1 WINS!!!!!!"**

3.3 Saving and loading a game

Should the player wish to save a game, they may type "3" and they will be prompted to enter the name of the save file, the program will then save to the specified file.

To load a game, the player can type "3" in the main menu and input their save file and the program will run based on the provided save file.

4. Discussion

4.1 General Issues

Various issues have been encountered throughout the development of this game. Other than quite expected incompatible types or struct related errors. Surprisingly, most of them are actually logical error. Furthermore, the biggest mistake was using individual variables instead of structs to store players and bomb stats. This made the functions requiring too many parameters to pass around.

4.2 Save and Load features

Save feature was quite straight forward to develop, a linked list is created and then filled with every move from both players, then print the list to a file chosen by the user. However, loading the saved game was surprisingly very complicated. At first, loading the game by playing all previous moves from the save file was attempted, this was found to be very complicated.

The way the program loads a saved game is by reading the saved file which itself is very similar to a settings file, in fact the same function (readSettings) is used for both reading the settings as well as reading the save file. Logged moves in the linked list is printed to the file although it serves no real purpose. A copy of settings.txt can be found in the bottom part of bombHeader.h (commented text).

4.3 User inputs

Prompting the user for moves and other user input was quite a challenge. Placing them under a single data type seemed to be the most viable option. Thus, instead of typing the number of seconds to hold the bomb, "special" or "save", the user can type "1", "2" or "3" to make a special move or save the game. This simplified this part of the program quite significantly.