Assignment 2 – Hangman

Jivaj Brar

CSE 13S – Fall 2023

Purpose

The purpose is to recreate the game Hangman in the programming language C.

How to Use the Program

The user will input a secret word, which the other user will type in letters that will either reveal the letters or say that it was wrong, add a body part to the hangman, and write that that letter is eliminated.

Program Design

There will be a loop that waits for either the user to guess the word correctly or run out of guesses. There will be a loop for the input of the letters, and it will determine the value of the input as if it were a correct or wrong answer.

Data Structures

I will use an array to log the letters that have been guessed. I will have to access the array that has the hangman templates, as the letters are guessed wrong. Print won or loss depending on what happens.

Algorithms

The algorithm will be to have an outer loop that has not done and another loop for the guesses.

```
While (not done)
      While guess:
         Correct guess -> add letter
         Incorrect guess -> add letter to eliminated and add to hangman
         If done:


      Break
 If win:
        Print you win
If loss:
        Print you lost
```

Function Descriptions

The following functions input and allow you to play the game.
- Inner loop for guesses and giving whether the guess was correct or incorrect.
- Functions for determining if the given secret message follows the correct input criteria.
- Outer loop for the game done = true /false

Result

I created the game Hangman and code functions as desired. A user can choose the secret and pick inputs to see whether they guess correctly. If correct, will show the letter in its proper place. If incorrect, places the letter in the Eliminated. If you guess all of the letters, you win, and if you run out of lives, the game ends and prints the loss statement.
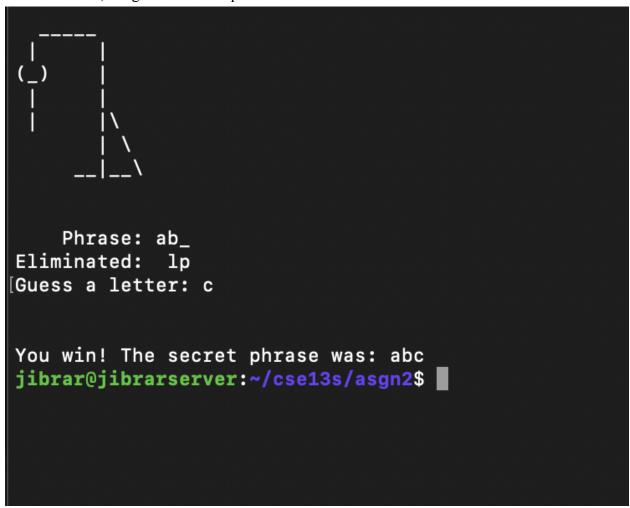

Figure 1: Code running properly

```c
        char* read = strchr(secret, guess);
        //printf("GUESSES:  %s", guesses);
        // guess loop
        if (read != NULL && (int)strlen(secret) > correctguesses && LOSING_MISTAKE > lives_used) {
            for (int j = 0; j < (int)strlen(secret); j++) {
                /*
                bool searchguesses = true;
                for (int inner = 0; inner < (int)sizeof(guesses); inner++) {
                    if( guess == guesses[inner]) {
                        searchguesses = false;
                        }
                }*/
                if (guess == secret[j]) {
                    p_input[j] = true;
                    correctguesses++;
                    //guesses[26] = p_input[j];
                    //guesscount++;
                    //printf("GUESSES:  %s", guesses);
                    }}
            }
            else {
                //printf("failed in read else");
                lives_used++;
                eliminated[lives_used] = guess;
                //guesses[guesscount] = guesses[guesscount] + p_input[j];
                //guesscount++;
            }

        //printf("CORRECT GUESSES: %d\n", correctguesses);
        //printf("strlen: %d\n", (int)strlen(secret));
        if ((int)strlen(secret) == correctguesses || LOSING_MISTAKE == lives_used){
            //printf("makes it in here");
                gamedone = false;
            }

    } while (gamedone); //ends game depending on the condition


    // win or losing arguments:

    //lost
    if (correctguesses != (int)strlen(secret)) {
        printf("\nYou lose! The secret phrase was: %s\n", secret);


        }
    // won
    else {
        printf("\nYou win! The secret phrase was: %s\n", secret);
        }
    return 0;
```

Figure 2: Image of the code