ECE 270

Justin Newman

Quiz #6 - Part 1
Guessing Game

October 1, 2014

1. Statement of the Problem

The purpose of this program is to allow the user to enter letters in an attempt to guess a pre-defined mystery word, up to a pre-defined maximum number of attempts. For each guess the game must display how much of the mystery word has been guessed correctly. Based on whether the user has guessed the word correctly within the number of allowed guesses, the game will then display a win/lose message.

2. Description of solution

All program output is printed to the screen and a text file names quiz6part1.txt.

This program creates 2 arrays, one stores the mystery word as a string and the other stores how much of the word has been correctly guessed:

```
char word[ ]="dissonance”, display[strlen(word)]
```

The length of the display array is determined by the length of the mystery word, via the `strlen()` function.

By way of a while loop, the program allows the user to continue making guesses until they have correctly guessed the mystery word or used up all of their guesses.

```
while((correctCounter<10)&&(guessCounter<10))
      {
```

At the start of each guess opportunity, the program displays the current guess number:

```
guessCounter++;
printf("\n\nTry # %d",guessCounter);
fprintf(fp,"\n\nPTry # %d",guessCounter);

printf("\nGuess a letter: ");
scanf("\n%c",&guess);
fprintf(fp,"\nGuess a letter: %c",guess);
```

As the player makes their guesses, a **for** loop compares the guessed letter to each letter of the string in turn and increments the number of correct guesses for each letter that matches. Then the program assigns the correctly guessed letter to its proper position in the `display` array:

```
for (i=0;i<11;i++)
   {
      if ((guess==word[i])&&(display[i]=='-'))

           {
                display[i]=guess;
```

```
                    correctCounter=correctCounter+1;
                }
            }
```

After checking the guess against the mystery word and adjusting the display array and number of correct guesses accordingly, the program prints which letters have been correctly guessed.

3. Output and Testing

To test the program I ran it a series of times, attached are samples of a winning game and a losing game.

Win:

Word guessing game
You have 10 guesses

Try # 1
Guess a letter: d
d---------

Try # 2
Guess a letter: i
di--------

Try # 3
Guess a letter: s
diss------

Try # 4
Guess a letter: o
disso-----

Try # 5
Guess a letter: n
disson-n--

Try # 6
Guess a letter: a
dissonan--

Try # 7
Guess a letter: c
dissonanc-

Try # 8
Guess a letter: e
dissonance
Congrats, You WIN!!

Won in 8 guesses

Lose:
Word guessing game
You have 10 guesses

Try # 1
Guess a letter: d
d---------

Try # 2
Guess a letter: h
d---------

Try # 3
Guess a letter: g
d---------

Try # 4
Guess a letter: i
di--------

Try # 5
Guess a letter: f
di--------

Try # 6
Guess a letter: s
diss------

Try # 7
Guess a letter: n
diss-n-n--

Try # 8
Guess a letter: j
diss-n-n--

Try # 9

Guess a letter: l
diss-n-n--

Try # 10
Guess a letter: e
diss-n-n-e
Sorry, you lost

The mystery word was: dissonance

4. Code

```
1  /* Justin Newman
2  Quiz #6 Part 1:
3  Letter Guessing Game
4  ECE 270 9/28/14
5  */
6
7  #include<stdio.h>
8  #include<stdlib.h>
9  #include<string.h>
10
11 int main()
12     {
13
14         char word[ ]="dissonance", display[strlen(word)], guess;
15         int i=0, j=0, correctCounter=0, guessCounter=0;
16
17         FILE *fp;
18         fp=fopen("quiz6part1.txt","w");
19
20         printf("Word guessing game");
21         printf("\nYou have 10 guesses");
22
23         fprintf(fp,"Word guessing game");
24         fprintf(fp,"\nYou have 10 guesses");
25
26         for(j=0;j<strlen(word);j++)
27             {
28                 display[j]='-';
29             }
30         display[strlen(word)]='\0';
31
32         //game plays until the number of correct guesses = letters in the word
33         //or the number of guesses = the number of allowed guesses
34         while((correctCounter<10)&&(guessCounter<10))
35
36             {
37
38                 guessCounter++;
39
```

```
40                  printf("\n\nTry # %d",guessCounter);
41                  fprintf(fp,"\n\nPTry # %d",guessCounter);
42
43                  printf("\nGuess a letter: ");
44                  scanf("\n%c",&guess);
45                  fprintf(fp,"\nGuess a letter: %c",guess);
46
47                  for (i=0;i<11;i++)
48
49                      {
50
51
52                          if ((guess==word[i])&&(display[i]=='-'))
53
54                              {
55
56                                  display[i]=guess;
57                                  correctCounter=correctCounter+1;
58
59                              }
60
61                      }
62
63                      printf("\n%s",display);
64                      fprintf(fp,"\n%s",display);
65
66                  }
67
68          //Determines if game was won/lost and displays accordingly
69          if (correctCounter==10)
70              {
71                  printf("\nCongrats, You WIN!!");
72                  printf("\nWon in %d guesses", guessCounter);
73
74                  fprintf(fp,"\nCongrats, You WIN!!");
75                  fprintf(fp,"\n\nWon in %d guesses", guessCounter);
76              }
77          else
78              {
79                  printf("\nSorry, you lost");
80                  printf("\n\nThe mystery word was: %s",word);
81
82                  fprintf(fp,"\nSorry, you lost");
83                  fprintf(fp,"\n\nThe mystery word was: %s",word);
84              }
85          return 0;
86
87      }
```