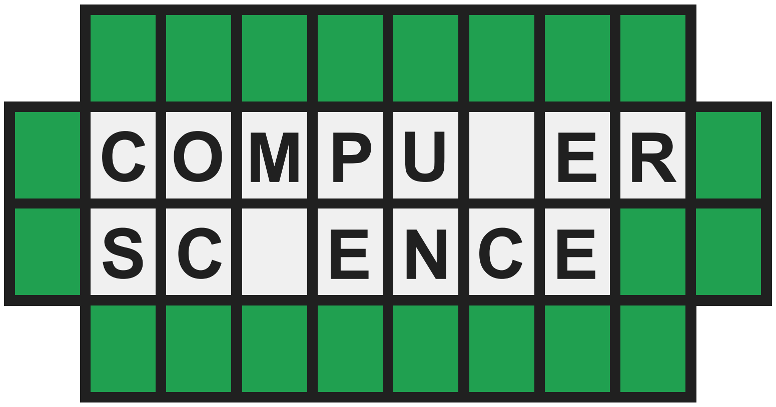
California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Fall 2017 – Project 1 – *Guess My Phrase*

Overview



We have all watched shows like "Wheel of Fortune". …or, perhaps, played the classic game Hangman. In both games, the player must guess a word or phrase by guessing letters. Getting a letter correct, uncovers part of the puzzle.

For this project, you are going to create something very simular – and fun!

Gameplay

After the first player enters a word, you will clear the screen. The second player will attempt to guess letters. As soon as all the letters are uncovered, the game ends.

Use your imagination. You can base your game on a fun theme. For example:

- Rick and Morty
- Politics
- A movie – comedy, sci-fi, horror, etc…
- A video game
- Television program
- Book
- etc…

Due Date

The assignment is due the **last** day of Dead Week.

Tips

1. ***Use incremental design!*** Write your program in parts – get it to work a little bit more each time you assembly it. Never attempt to write the entire solution in just one take.... Take your time!

2. Make use of the library functions! Look into LengthCString and ScanChar.

3. You need two buffers! This solution is impossible without both.

4. Make sure you understand all your labs before attempting this project.

5. Fill the "display" buffer with zeroes when you declare it.

Sample Program

Here is a sample program. The player 1's input is printed in **red**. Player 2's input is displayed in **blue**.

```
Welcome to Mystery Word!

Player 1, enter a word: orville
```

Screen is cleared

```
Your secret word: -------

Guess a letter: l
Your secret word: ----ll-

Guess a letter: o
Your secret word: o---ll-

Guess a letter: s
Your secret word: o---ll-

Guess a letter: r
Your secret word: or--ll-

Guess a letter: v
Your secret word: orv-ll-

Guess a letter: t
Your secret word: orv-ll-

Guess a letter: e
Your secret word: orv-lle

Guess a letter: i
Your secret word: orville

Congrats! You guessed the secret word!
```

Reading and Storing Text

To read text from the keyboard, please read about the ScanCString subroutine in the CSC35 Library. You will need to create a buffer large enough to hold the phrase.

The example below creates a buffer (space) of 30 characters.

```
MyAwesomeBuffer:
   .space 30
```

Requirements

YOU MUST DO YOUR OWN WORK. DO NOT ASK OTHER STUDENTS FOR HELP.

If you ask for help, both you and the student who helped you will receive a 0. Based on the severity, I might have to go to the University.

1.	Display the title (of your choice)	(5 points)
2.	Read the word from the keyboard	(15 points)
3.	Clear the screen	(10 points)
4.	Loop until all letters are guessed	(30 points)
5.	Print the display string (with the missing characters) each time in the loop	(15 points)
6.	When they are correct, display message congratulating them	(10 points)
7.	Proper formatting. Labels are not indented. All instructions are indented the same	(10 points)
8.	Comments	(5 points)

Pseudocode

The following is the basic logic of the game in Visual Basic-like pseudocode.

```
Print your program's name
Scan the answer from the keyboard.
Implement a For Loop to fill a buffer for the display string with the correct number of dashes.
Do
    Input a guess letter            #Look at the CSC35 library
    Loop through each letter of the answer
        If the answer[n] == guess letter
            display[n] == guess letter          #This uncovers the letter
        end if
    end loop
    Print the display string (the one with dashes)
loop while no dashes exist in the display (this will require another loop)
```

Submitting Your Project

Run Alpine by typing the following and, then, enter your username and password.

```
alpine
```

To submit your lab, send the source file (not a.out or the object file) to:

```
dcook@csus.edu
```

Extra Credit

Color – 5 points

Make use of color to enhance your game. The color must be meaningful – don't just set the color at the beginning of the program.

Player can lose – 10 points

Keep track of how many time the user guesses. After a set number of wrong guesses, the game should stop and they lose. It's up to you have many wrong guesses they can have.

Display a "hangman" graphic – 10 points

Each time the user guesses a wrong letter, show a graphic (ASCII art naturally), that is similar to Hangman. You can display anything you want. The solution to this type of problem can be solved using multiple string buffers or very clever loops. Use your imagination!

For example, you can show an ASCII worm that keep getting longer.

```
====:)
```

Or, perhaps, a progress bar:

```
--------------------
********            | Overheat
--------------------
```

Case Insensitivity – 10 points

For 10 points of extra credit, modify your program so the game in not case sensitive. In other words, if Player 1 enters the text "Hello", the "H" should be matched by both 'h' and 'H'.

How to Connect from Home

Step 1 – Windows

The three servers that we use to do our labs cannot be accessed from off campus – at least directly. To connect these computers, first connect to Athena using Putty.

```
athena.csus.edu
```

Step 1 – Macintosh

Open the Terminal program. This is the same UNIX prompt that you get when you connect to Athena. Mac-OS X is a version of UNIX. Neat! Once at the prompt, type the following where ***username*** is your ECS username.

```
ssh username@athena.csus.edu
```

Step 2 – Secure Shell to SP1, SP2, or SP3

Once you are connected, you need to Secure Shell (SSH) to the SP computers. Basically, you will connect to Athena and it will connect you to the SP computer. Type the following at the UNIX prompt (this example uses SP2).

```
ssh sp2
```

You will have to enter your password again and (maybe) have to manually type "yes".

UNIX Commands

Editing

Action	Command	Notes
Edit File	`nano` *filename*	"Nano" is an easy to use text editor.
E-Mail	`alpine`	"Alpine" is text-based e-mail application. You will e-mail your assignments it.
Assemble File	`as -o` *objectfile asmfile*	Don't mix up the *objectfile* and *asmfile* fields. It will destroy your program!
Link File	`ld -o` *exefile objectfiles*	Link and create an executable file from one (or more) object files

Folder Navigation

Action	Command	Description
Change current folder	`cd` *foldername*	"Changes Directory"
Go to parent folder	`cd ..`	Think of it as the "back button".
Show current folder	`pwd`	Gives a file path
List files	`ls`	Lists the files in current directory.

File Organization

Action	Command	Description
Create folder	**`mkdir`** *foldername*	Folders are called directories in UNIX.
Copy file	**`cp`** *oldfile newfile*	Make a copy of an existing file
Move file	**`mv`** *filename foldername*	Moves a file to a destination folder
Rename file	**`mv`** *oldname newname*	Note: same command as "move".
Delete file	**`rm`** *filename*	Remove (delete) a file. There is **no** undo.