

# Lab 9: DS4Talker

*NOTE: This is a two week lab.*

## Objectives:

- Practice top-down program design, problem solving in C
- Practice strings and arrays in C

## Starting Point:

- [lab9.c](#)
- [wordslst.txt](#)

## Process:

### Creating a New Folder

Create a new folder named *lab9* in your *cpre185labs* folder on the U: drive. You will want to copy over *ds4rd.exe* to the *lab9* folder.

### Problem (Week 1)

People with speaking disabilities and limited mobility, such as famed physicist Stephen Hawking, often use speech synthesizers controlled by a simple input device. In this lab, we will write a tool that allows the user to generate sentences of text using the DualShock 4 as input. This could then feed a speech synthesizer program.

The skeleton code takes a filename as a command line argument. The file will contain a list of up to 100 words, one per line. Each word will not exceed 10 characters. For week one, your code should read every word from a file and print them on the screen. Sounds easy right?

The skeleton code also has two functions already prototyped for you. You just have to implement them. The functions are `trimws()` and `readWords()`.

The demo for week one is printing all of the words from the file and printing the number of words read. `Ncurses` is not needed for part one.



## Problem (Week 2)

Create a copy of your part one code and label it part two.

Edit your part two code so it presents the words in five columns ("%15s" perhaps) on the screen. To do this, you use the mvprintw function from the ncurses library.

**Do not use printf. It doesn't play nicely with ncurses.** More information about using mvprintw can be found in the development tips at the end of this lab.

Develop an input method so that a user can select one of the displayed words using the DualShock 4. You would be wise to use the joystick. Its values are found using the -j flag

```
./ds4rd.exe -d 054c:05c4 -D DS4_BT -t -b -j -bt
```

You should indicate the word currently selected by placing a character next to it using the draw char function used in DS4Maze.

Begin with an empty sentence at the bottom of the screen.

The TRIANGLE button will append a space to the current sentence followed by the selected word.

The X button will remove the last item (a word, punctuation, or letter) added. You need to be able to use the remove feature multiple times in a row. Think of it as an "undo" button in software you use on a daily basis. The X button also needs to remove everything in the last command (including spaces).

The SQUARE button will append the word but without adding a space.

**BONUS:** (5 points). The CIRCLE button will capitalize the first letter of the next word appended.

Update the current sentence at the bottom of the screen as words are added and deleted. The sentence can be limited to 80 characters.

A new sentence can be started by pressing down on the joystick. This should clear the bottom line of the screen.

**Be sure to demonstrate your program and get it checked off by your undergraduate TA.**

**NOTE: You can demo during lab time in week two of this lab.**

### **Development Tips**

- You will need to use ncurses as in DS4Maze. **Do not use printf. It doesn't play nicely with ncurses.** Instead of using printf, you should use the mvprintw function. Its prototype looks like: void mvprintw(int row, int col, char \* format, arg1, arg2, arg3...). Hence, you can use it like mvprintw(3, 10, "%d is an integer", 55) which will print 55 is an integer starting at row 3 in column 10.
- Be sure to call the refresh() function every time you want the screen to update.
- In order to use the X button, you may want to keep track of the lengths of the items in the sentence using an array.

### **Questions**

For this lab, answer these questions in the comments of your code.

1. Describe your process for reading every word from a file. What were some of the struggles you encountered?
2. Describe how you keep track of the word selected on the screen and how this interfaces with the DualShock 4. Is the interface reasonable?

### **Turn-In:**

One report for parts one and two. Due after the second week.

Demo Part 1 for a TA before your next lab.

There is a separate demo for parts one and two.