CS2261 Media Device Architecture

Drawing a String

* Single characters are boring and not *that* useful
* What is a string, conceptually?
  + A sequence of characters
* What is a string, in C?
  + A constant array of characters, ending with NUL (ASCII value 0, typed ‘\0’)
  + Since its an array of characters, we can use our drawChar for the drawString

Steps to draw a string

1. Figure out which to draw, where to draw it, and what color
   * Function parameters
2. Iterate through the array of characters
   * While-loop, since we don’t necessarily know the length
3. For each character, draw it
   * Using drawChar
4. If this character is the last one, stop the loop
   * If the next character is NUL (\0), this character is the last one
5. If there is another character, increment the column to draw it
   * Otherwise they’d all be drawn on top of each other

Calling the drawString function

* Our function takes in a location, character array, and a color.
  + void drawString(int row, int col, char \*str, unsigned short color);
* str points to the beginning (first character) in the array
  + Since array variables are just specialized pointers, you just pass in the array
* Writing a string with double quotes automatically makes it an array of characters, ending with NUL. No casting is needed.
* Ex. char str[2];
  + str[0] = ‘a’;
  + str[1] = ‘\0’; // Without this, drawString may loop forever
* Ex. char str2[6] = “hello”; // length is 6 because \0 is added
* drawString(80, 40, str2, WHITE); // Works
* drawString(80, 40, “hello”, WHITE); // Also works

Easy string formatting

* We want to be able to print variables and formatted things easily
  + printf won’t work, because it prints to the console, which we don’t have
* Instead, use **sprintf**
  + A library function in stdlib.h
  + Returns void
  + Works just like printf, but it has an extra parameter at the beginning
    - The character array to save the result to, instead of printing
  + There’s a ton you can do with it. Google it.
* char buffer[41]; // The longest you can ever draw (6\*40=240)
  + a char is 6 columns long and the total screen width is 240, so you can draw only 40 characters plus the “/0” null char at the end
* sprintf(buffer, “The score is %d for %s”, score, “Player 1”);
  + note that you can reuse buffer and reset it
* drawString(76, 40, buffer, BLACK);

Why can you not use DMA for text?

* DMA copies blindly
  + You tell it how big of a chunk, how many, where from, and where to
  + You cannot have it do any sort of logic for you
* Drawing text requires logic
* Because you can’t use DMA, drawing text is slow
  + If at all possible, avoid drawing text every frame
  + One or two characters? Sure.
  + One or two words? Absolutely not.