

CS 2261 Lab 03:

Arrays and Swap

Provided Files

- `main.c`
- `myLib.c`
- `myLib.h`
- `coords.c`
- `coords.h`

Files to Edit/Add

- `main.c`
- Your Makefile

Instructions

In this lab, you will be completing several different `TODO`s, which will, piece by piece, complete a simple cactus jumping game (although the term “game” only loosely applies to this one). Your code may not compile until you complete an entire `TODO` block, at which point the game should compile with a new component of the final outcome (unless otherwise specified).

- **TODO 1 – Cacti**

- First, we want to create the cacti. They will sit on the top of the sand, and will move to the right (the movement has already been written).
- TODO 1.0: Find the `cactusCols` and `oldCactusCols` arrays in `main.c`. These will be the locations of the cacti. Notice that we don't have variables for the lengths of these, so we're going to find a way to calculate the length when we need it.
- TODO 1.1: Find the `drawCacti` function. Complete this by calling the already-written `eraseCactus` function with each column in your `oldCactusCols` array (to erase them) and the `drawCactus` function for each column in your `cactusCols` array (to draw them).
 - Hint: you will need to calculate (not hard code) the length of the arrays. Since they are preallocated globally, what tool can we use to find the length?
- TODO 1.2: Call the `drawCacti` function in the main while loop to draw your cacti each frame of the game.
- Compile and run. If you did everything correctly, you should see two equally-spaced cacti moving to the right, and, when they hit the right side of the screen, reset to the left.

- **TODO 2 – Player**

- The game wouldn't be much fun without a player, so let's draw one.
- TODO 2.0: Open `coords.c` and examine the contents. For this lab (and this lab only), we have opted to draw the player by having arrays of coordinates (row arrays that match up to column arrays), and setting a pixel at each of those row-column pairs. Note that there is a macro defining the length of the coordinate arrays, which we may need later.
- TODO 2.1: In `main.c`, find the `drawPerson` function. It takes an array of rows and a same-length array of columns, along with the length of those arrays (remember: there would otherwise be no way to determine this), as well as a color. Complete this function so that it sets a pixel at each corresponding row-column pair in the arrays to the specified color.
 - Hint: don't overthink this. All you're doing is iterating over the length of the arrays and calling `setPixel` for each `i`.
- TODO 2.2: In the main while loop, after `waitForVBlank`, call `drawPerson` with the `invisiblePerson` arrays using the `SKYCOLOR` to erase the person's previous location (this will make sense later).
- TODO 2.3: A few lines later, call `drawPerson` with the `visiblePerson` arrays using `WHITE` to draw the person's current location.
- Compile and run. You should see our new player character on top of the sand, getting repeatedly smacked by cacti.

- **TODO 3 – Swap**

- For the purposes of this lab, we're going to have our player jump by swapping the coordinates of the `visiblePerson` with those of the `invisiblePerson` (who is currently safely above the cacti).
- **TODO 3.0:** In `main.c`, write a new function called `swap` that returns `void` and swaps the value of two integers passed in as parameters. Don't forget to write a prototype at the top of `main.c`.
 - **Note:** this function may NOT exploit the fact that the things we will be swapping (in later `TODOs`) are statically allocated and/or in arrays. It must work for any two integers.
 - **Hint:** think about what you will need to actually take in as arguments to make this work
- **TODO 3.1:** In our main while loop, if the Start button is pressed, utilize your new `swap` function so that each element of `invisiblePersonRows` is swapped with each corresponding element of `visiblePersonRows`, and then the same for `invisiblePersonCols` and `visiblePersonCols`.
- **Compile and run.** If everything is working right, every time you press the Start button, the player character alternates between a place of safety above the cacti and back to where they started, standing on the sand.

- **TODO 4 – More Cacti**

- Now that everything is working, let's make it more exciting by adding more cacti.
- **TODO 4.0:** Change the initialization of the `cactusCols` and the `oldCactusCols` arrays so that the starting locations are 0, 73, and 147.
- **Compile and run.** If you coded `TODO 1.1` correctly, this should be all it took to add a third cactus in the cycle. If you don't see this, revisit `TODO 1.1` to make sure it works for any length of the arrays. If it does all work, submit your lab.

Submission Instructions

Zip up your entire project folder, including all source files, the Makefile, and everything produced during compilation (including the `.gba` file). Submit this zip on Canvas. Name your submission `Lab03_FirstnameLastname`, for example: "`Lab03_TraskUlgo.zip`".