

CS 2261 ColLab:

Mode 0

Provided Files

- main.c
- myLib.c
- myLib.h
- collab.bmp

Files to Edit/Add

- main.c
- myLib.c
- collab.bmp
- collab.c
- collab.h
- Your Makefile

Instructions

In this lab, you will be completing several different TODOs, which will, piece by piece, complete a scrollable background in Mode 0. 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 – Exporting the tiles

For us to be able to see anything in Mode 0, we're gonna need to have some tiles.

- TODO 1.0: Open collab.bmp in Usenti.
- TODO 1.1: Go to Image > Export, keep the default name, and make sure the type is GBA Source. Make sure it is exporting them to your folder.
- TODO 1.2: In the Export popup, make sure the type is 8bpp tiles, and make sure that Pal is checked and Map is not checked (just for this lab).
- TODO 1.3: Add the new .c file to your Makefile.
- TODO 1.4: At the top of main.c, include collab.h.

At this point, we haven't added them to your code, so compiling will not do anything useful.

TODO 2 – Setting up Mode 0 and Displaying Tiles

We want to be able to actually see them, so let's set that up.

- TODO 2.0: In main.c, in the initialize function, setup the Display Control Register.
- TODO 2.1: In Mode 0, you also need to set up that background's control register. So do that for background 0.
- TODO 2.2: Since Mode 0 uses indexed color, load your tiles' palette (which was exported along with them) to the PALETTE.
- TODO 2.3: We need to actually load our tiles into memory. So scroll down a bit and use DMANow to load them into the correct character block. Make sure it is the same character block where you told background 0 to find them.
- TODO 2.4: For them to show up, you need to load a map into memory. For this lab, we made an array for this. In every other time in the future, Usenti will make this for us. For now, though, load this map that we created into the correct screenblock. Make sure this is the same screenblock where you told background 0 to find it.

Compile and run. You should see a green circle in the top left corner of the screen. If not, then you either exported wrong or did not load the tiles or map into memory correctly. Fix this before continuing.

TODO 3 – Scrolling

The real benefit of Mode 0 is that it is easy to scroll backgrounds, so let's make that happen. We made two variables, hOff (for horizontal offset) and vOff (for vertical offset) that we will use to update the actual offsets of the background.

- TODO 3.0: In game(), make the arrow keys change these two variables. When horizontal offset increases, the screen scrolls to the right. When vertical offset increases, the screen scrolls to the left.
- TODO 3.1: These variables won't do anything unless we actually update the background 0 offset registers using them. So do that.

Compile and run. You should be able to use the arrow keys to scroll around. If you scroll far enough, the background will repeat and you will see the circle again. If not, fix this before going further.

TODO 4 – Customizing the map

You need to make your own custom tiles and map now for practice.

- TODO 4.0: In collab.bmp, add your own custom tile. Make sure it fits within a tile grid space. You can add as many colors to the palette as you want for this. Re-export collab.bmp with this new tile.
- TODO 4.1: In the initialize function, customize the map by adding your new tile, as well as adding in the other tiles from collab.bmp. For full credit, your map

needs every tile from collab.bmp as well as your new one. You also must flip at least one tile vertically and at least one horizontally. Finally, you have to make a larger “image” that uses multiple tiles to create the shape.

Compile and run. You should be able to scroll around and see your background repeat. If this all works, submit your lab.

Submission Instructions

Don't submit anything. This was just an in-class exercise this year.