

CS 2261 Lab 07:

Multiple Backgrounds

Provided Files

- main.c
- myLib.c
- myLib.h
- skyTiles.bmp
- trees.bmp

Files to Edit/Add

- main.c
- myLib.c
- lab06.bmp
- skyTiles.c
- skyTiles.h
- trees.bmp
- trees.c
- trees.h
- Your Makefile

Instructions

In this lab, you will be completing several different TODOs, which will, piece by piece, display multiple simultaneously-appearing scrollable backgrounds 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 skyTiles.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 Lab07 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.

- TODO1.4: At the top of main.c, include skyTiles.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 1.
- 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 1 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 1 to find it.

Compile and run. You should see an orange 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 skyTiles.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 skyTiles.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 skyTiles.bmp. For full credit, your map needs every tile from skyTiles.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.

TODO 5 – Exporting trees.bmp

Let’s set up the other background now.

- TODO 5.0: Open trees.bmp in Usenti.
- TODO 5.1: This background will appear the same time as the other, so they have to use the same palette. First, we want to get these colors onto the palette of the one that we are loading (the palette of furtherTrees). So go to Image > Export, and save as type “Palette (.pal).” You can export just the 16 colors that you need.
- TODO 5.2: We want the other image to include these newly exported colors. So open up furtherTrees.bmp again, and go to Image > Import, then import the palette you just exported. Put them on the second row of the furtherTrees.bmp palette (destination 16). Now, for the GBA to see this, save the image and re-export it (as GBA Source, not Palette). These new colors will be part of what loadPalette is loading.
- TODO 5.3: Now we need to tell trees.bmp to use the second row. Open it back up, then go to Palette > Swap. Swap the first 16 colors with the second 16 colors (source should be the beginning of the first row, destination should be the beginning of the second row, and the count should be the number of colors being swapped). When you hit “Swap,” you should see it happening in the palette in Usenti.
- TODO 5.4: Export this image the same way you exported the last one.
- TODO 5.5: Add the new .c file to your Makefile.

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

TODO 6 – Displaying trees

We want to be able to actually see the new background, so let’s set that up.

- TODO 6.0: At the top of main.c, include the .h file for the image you just exported.

- TODO 6.1: In the initialize function, tell the Display Control Register to also enable background 0 for our new map.
- TODO 6.2: Set up background 0's control register. This is a 512x256 pixel background, so think about what size to tell it and where to put the tiles and map (save enough room for them, but waste no space). Remember that since the size is bigger than the last time, you may need more space.
- TODO 6.3: Use DMANow to load the tiles into the correct character block. Make sure it is the same character block where you told background 0 to find them.
- TODO 6.4: Use DMANow to load the map into the correct screen block. Make sure it is the same screen block where you told background 0 to find it.

Compile and run. You should see your trees map on top of your skyTiles map, and be able to scroll both with the left and right arrow keys. If not, fix this before continuing.

TODO 7 – Parallax

skyTiles should look like it is farther away. However, we aren't currently creating that illusion. In real life, when you move, things that are farther away look like they are moving more slowly. At the bottom of game(), find where the background offset registers are being updated. Change the line that updates the offset for skyTiles so that it moves more slowly. Hint: For every two pixels that trees moves, it should move one. Use your Pre-Algebra skills.

Compile and run. When you scroll, skyTiles should move more slowly, and create the illusion of depth. If so, zip up your files and submit.