CS 2261 Lab 08: Backgrounds

Provided Files

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

Files to Edit/Add

- main.c
- trees.bmp
- trees.c
- trees.h
- furtherTrees.bmp
- furtherTrees.c
- furtherTrees.h
- Your Makefile
- Your tasks.json

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 furtherTrees.bmp

Let's set up one background first before moving on to any others.

- TODO 1.0: Open *furtherTrees.bmp* in Usenti.

- TODO 1.1: The palette and image have been set up for you (for this step). Go to Image > Export, keep the default name, and make sure the type is GBA Source. Make

sure it is exporting them to your Lab08 folder.

● TODO 1.2: Since we are using multiple backgrounds in this lab, the easiest way is to use 4bpp tiles so they can fit on the same palette. In the Export popup, make sure the type is 4bpp tiles, and make sure that Map (with sbb selected) and Pal are both checked.

TODO 2 – Setting up Mode 0 and Displaying *furtherTrees*
We want to be able to actually see it, so let's set that up.

● TODO 2.0: At the top of main.c, include the .h file for the image you just exported.

● TODO 2.1: In the initialize function, set up the Display Control Register. We want to use Mode 0, and also enable background 1 for our image (not 0, because we want furtherTrees to be behind the next background we add). Look at myLib.h for macros to help you set this up!

● TODO 2.2: Load your tiles' palette (which was exported along with them) to the PALETTE.

● TODO 2.3: Set up background 1's control register. This is a 256x256 pixel background, so think about what size to tell it and where to put the tiles and map.

● TODO 2.4: Use DMANow to load the tiles into the correct character block. Make sure it is the same character block where you told background 1 to find them.

● TODO 2.5: Use DMANow to load the map into the correct screen block. Make sure it is the same screen block where you told background 1 to find it. Compile and run. You should see your furtherTrees map, and be able to scroll it with the left and right arrow keys. If not, fix this before continuing.

TODO 3 – Exporting trees.bmp
Let's set up the other background now.

● TODO 3.0: Open *trees.bmp* in Usenti.

● TODO 3.1: 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 (start: 0; count: 16).

● TODO 3.2: 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 (source: 0; destination 16; count: 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 3.3: Now we need to tell trees.bmp to use the second row of the palette we just merged together in 3.2. Open back trees.bmp, 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 3.4: Export this image the same way you exported the last one.

TODO 4 – Displaying trees
We want to be able to actually see the new background, so let’s set that up.

● TODO 4.0: At the top of main.c, include the .h file for the image you just exported.

● TODO 4.1: In the initialize function, tell the Display Control Register to also enable background 0 for our new map.

● TODO 4.2: Set up background 0’s control register. This BG will display a 512 pixel wide, 256 pixel tall background, so think about what size this translates to **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 4.3: 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 4.4: 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 furtherTrees map, and be able to scroll both with the left and right arrow keys. If not, fix this before continuing.

TODO 5 – Parallax
The *furtherTrees* image 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. Implement that here.

TODO 5.0: At the bottom of game(), find where the background offset registers are being updated. Change the line that updates the offset for furtherTrees so that it moves more slowly. Hint: For every two pixels that trees move, furtherTrees should move one. Use your Pre-Algebra skills. Compile and run. When you scroll, furtherTrees should move more slowly, and create the illusion of depth. If so, zip up your files and submit.

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 Lab07_FirstnameLastname, for example:“Lab07_RonSwanson.zip”.