



# LAB 08:

## Backgrounds

### Provided Files

- `main.c`
- `myLib.c`
- `myLib.h`
- `grass.bmp`
- `clouds.bmp`

### Files to Edit/Add

- `main.c`
- `grass.bmp`
- `grass.c`
- `grass.h`
- `clouds.bmp`
- `clouds.c`
- `clouds.h`
- `Makefile`
- `.vscode`
  - `tasks.json`

---

### Instructions

In this lab, you'll be completing several different TODOs, which will, piece by piece display multiple, simultaneously appearing, scrollable backgrounds in Mode 0. This is an example of how you can find a background image on a site like OpenGameArt to create a dynamic background! In this case, separate images (one of grass, and one of clouds) were used. 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 clouds.bmp**

Let's set up one background first before moving on to any others. This will be the further background.

- TODO 1.0



- Open clouds.bmp in Useti
- TODO 1.1
  - The palette and image have been set up for you (the palette is reduced to a single row of colors, the transparent color is at index 0, and the image is an appropriate size).
  - Go to Image > Export, keep the default name, and make sure the type is **GBA source**. Make sure that it is exporting to your Lab08 folder.
- TODO 1.2
  - Since we are using multiple backgrounds in this lab, the easiest way to do this is by using:
    - 4BPP tiles (so that their individual palettes can fit onto a single master palette, the key here is that we are utilizing the transparent color for each row)
  - Your **export settings** should be as follows:
    - 4BPP
    - Map is checked, sbb selected
    - Pal is checked

## TODO 2.0 - Setting up Mode 0 and displaying clouds

Now we are going to set up the background so that we can actually see it.

- 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 (REG\_DISPCTL)
    - We want to use **Mode 0**
    - Enable **background 1**
      - We are not enabling background 0 since we want this image to be behind the next background we will load in later.
    - Look at myLib.h for macros to help you set this up
- TODO 2.2
  - Load your tiles' palette (which was exported along with it) into the PALETTE
- TODO 2.3
  - Set up **background 1**'s control register
    - The background is **512x160** pixels (think about what **size** to tell the GBA that it is and where to put the **tiles and map**)
      - Remember, we need to set up the charblock to tell the tiles where to go and screenblocks to tell where the map will go
      - Look at myLib.h for the necessary macros for background sizes, background registers, and background charblock and screenblock
- TODO 2.4



- Use DMANow to load the **tiles** into the correct **character block**.  
Make sure it is the SAME character block that you told background 1 to find them in the previous TODO.
  - This is referring to the actual CHARBLOCK struct
  - Check myLib.h under // Character and Screen Blocks
- TODO 2.5
  - Use DMANow to load the **map** into the correct **screenblock**. Make sure it is the same screenblock you told background 1 to find it in TODO 2.3.
    - This is referring to the actual SCREENBLOCK struct
    - Check myLib.h under // Character and Screen Blocks
  - Build and run. You should see your clouds map, and be able to scroll it with the left and right arrow keys (the map should loop as you move in one direction). If not, fix this before continuing.

### TODO 3.0 - Exporting grass.bmp

Let's set up the closer background now.

- TODO 3.0
  - Open grass.bmp in Usenti
- TODO 3.1
  - This background will appear at the same time as the other background, so they have to share the same master palette.
  - First, we want to get these colors onto the palette of the one that we have already loaded in (the palette of clouds).
  - Go to Image > Export and save the type as **Palette (.pal)**.
    - You can export just the 16 colors that you need (start: 0, count 16)
- TODO 3.2
  - Now, we need to tell grass.bmp to use the second row of the "master" palette since we know that the other background is using the first row.
  - Go to Palette > Swap
    - Swap the first row of 16 colors with the second row of 16 colors
      - **First Index:** beginning of the first row (0 in this case)
      - **Second Index:** beginning of the second row (16 in this case)
      - **Count:** number of colors being swapped (16 since each row has 16 colors)
    - When you hit swap, you should see the entire row of colors this background image is using move to the second row.
- TODO 3.3
  - Export this image the same way you exported the last one:
    - Image > Export



- **GBA source**
  - 4BPP
  - Map is checked, sbb selected
  - Palette checked
- **TODO 3.4**
  - Now, we've told this new background image to use the second row of colors within a "master palette". In order to create this palette, we need to merge the colors into our previous background's palette.
  - Open **clouds.bmp** again and go to Image > Import
    - Import the **palette (.pal)** we exported in TODO 3.1
      - Remember, this is just the set of 16 colors that the background was using
    - We want to put them on the second row of the clouds.bmp palette
      - **Source:** 0 (from this palette we are importing, start at index 0)
      - **Destination:** 16 (to this palette we are importing to, start placing it at index 16 -- first spot of the 2nd row)
      - **Count:** 16 (16 colors)
      - Note: You should see both rows have their OWN transparent color at the first index of their rows (magenta). This is necessary in order to be able to see layered backgrounds.
  - For the GBA to get these changes, we need to **re-export**.
    - Export as: GBA source, with the same export settings as before.
    - The new colors will now be a part of the palette we were DMAing before.
  - If you build and run right now, you should not see any changes. If you see that your image is using different colors now, then you have not merged the palettes correctly. Fix this before continuing.

## TODO 4 - Displaying grass

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 grass
- **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. The BG will display a **256x160** pixel 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, and make sure it doesn't overwrite background 1).
- **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 screenblock where you told **background 0** to find it.
  - Build and run. You should see both the grass and clouds maps (grass layered on top of clouds) and be able to scroll both with the left and right keys. They will move together. If not, fix this before continuing.

### TODO 5 - Parallax

The clouds image should look like it's farther away. However, we aren't currently creating that illusion. In real life, when you move, the 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 the **clouds** background so that it moves more slowly.
  - **HINT:** for every *4 pixels* that grass moves, clouds should only move one (clouds moves a quarter as fast as grass)
    - Use your Pre-algebra skills.
- Build and run. When you scroll, clouds should move more slowly, and create the illusion of depth (motion parallax).

## You will know if it runs correctly if you:

- Can see both backgrounds simultaneously.
- DO NOT see the transparent color on either background (magenta).
- Can scroll the backgrounds using the left/right keys and see the illusion of clouds moving slower than grass.

## Tips

- Review recitation materials on how to merge palettes
  - First follow the steps in this PDF
  - If you need an example, go to Canvas > Recitation Materials > Usenti > Merging Palettes Usenti.zip
- Review recitation materials on charblocks and screenblocks
  - Canvas > Recitation Materials > Mode0-CharblocksAndScreenBlocks.png
  - Canvas > Recitation Materials > Mode0BG\_Demo.zip
- Look at the `myLib.h` for any macros relating to backgrounds, palette, charblocks, screenblocks



## 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 Lab08\_FirstnameLastname, for example: "Lab08\_ReiAyanami.zip".