



# LAB 08:

## Complex Movement, Collision Maps and Multiple Backgrounds

### Provided Files

- Example.gba
- main.c
- lab08lib.c
- lab08lib.h
- game.c
- game.h
- house.bmp
- house.c
- house.h
- spritesheet.bmp
- spritesheet.c
- spritesheet.h
- collisionmap.bmp
- foreground.bmp
- foreground.c
- foreground.h

### Files to Edit/Add

- main.c
- game.c
- collisionmap.c
- collisionmap.h
- Makefile
- .vscode
  - tasks.json

---

### Instructions

In this lab, you will be completing several different TODOs, which will, piece by piece, add complex sprite movement. 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 1.0 – Complex Camera Movement

We want our Pikachu to walk only on the screen, and have the camera follow it without showing something outside of the world.

### TODO 1.0

- In game.c, in the updatePlayer() function, add in the sprite and camera movement.

### TODO 1.1

- In game.c in the drawPlayer() function, set Pikachu's screen row and column in the shadowOAM
  - Hint: the screen row is the world row minus the background vertical offset; the screen col is the world col minus the background horizontal offset.

Build and run. Your Pikachu should now be able to walk around and see the entire map, but not off of the edge. The Pikachu should always be in the middle of the screen unless at the edge of the map (see the Example.gba).

## TODO 2.0 – Collision Map

Now we want our Pikachu to treat the visuals of the map as if they were actual barriers to movement.

### TODO 2.0

- The collision map has been created for you. Open collisionmap.bmp in Usenti and export it. Remember that we want to check the colors of each pixel, so take that into account when choosing export settings. As such, export the collision map as bitmap(GBA) and select 8 bpp. Check the Pal box.
  - Note: Make sure that black is on index 0, and white is on index 1 in your palette!

### TODO 2.1

- Include collisionmap.h in game.c.
- At the top of game.c, declare an unsigned char \* collisionMap and assign it the value collisionmapBitmap
  - **If you get a warning from this line in the terminal when building, you need to fix it or you will lose points!**
  - Make sure to explicitly cast collisionmapBitmap to the right type.

### TODO 2.2

- In lab08lib.c, write the collisionCheck function.

### TODO 2.3

- Update your movement code to only allow Pikachu to move if the collision map allows it (is white) in the areas that you need to check. Your Pikachu should not



be able to walk over the house or the bushes. The Pikachu should be able to walk between the house and the bushes on the right and top, in both directions.

- **You must use the collisionCheck function you wrote in TODO 2.2!**
- Extra: Consider how you might write additional collisionCheck functions for specific situations; for example, if you wanted to check if the player is touching a given area at all, you could write a function which returns the largest collision map value at the four corners of the player (that way touching it with only one corner counts as being inside it.) If your player was larger than some of the collider regions, they would be able to “engulf” them since none of the corners would touch. Consider how you might adjust your collisionCheck function to account for this as well. **You do not need to worry about any of this for this lab, but it is food for thought moving forward!**

### TODO 3.0 – Foreground Overlay

Now we will add a foreground to our game that stays stationary on top of our moving background! You can look at the foreground by opening foreground.bmp. The green color is in our transparent index (index 0 of the palette), so we will not see it in our game. We also already merged the palettes for you, but reference the Merging Palettes pdf on Canvas > Files > Recitations > Usenti for more details on how to do this!

#### TODO 3.0

- The foreground has been created and exported for you. Include foreground.h in main.c.

#### TODO 3.1

- In our initialize function, tell the Display Control Register (REG\_DISPCTL) to also enable background 0 for our foreground.

#### TODO 3.2

- Set up background 0's control register. The BG will display a 240x160 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 3.3

- 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 it.

#### TODO 3.4

- 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.

#### TODO 3.5

- Set up background 0's hOff and vOff registers.



- Tip: We want the top left corner of the screen to be at the top left corner of the foreground at all times!
- 

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

- Your Project.gba has the same behavior as the Example.gba provided for you. The Example.gba Pikachu is light pink, but Project.gba Pikachu should be yellow.
- 

### Tips:

- Follow each TODO in order, and only move forward if everything is correct.
  - Review recitation materials: Canvas > Files > Recitation Materials
- 

### Submission Instructions:

Ensure that **cleaning** and building/running your project still gives the expected results. **Please reference the last page of Lab02.pdf for instructions on how to perform a "clean" command.**

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\_LastnameFirstname, for example:

"Lab08\_KetchumAsh.zip"

It is your responsibility to ensure that all the appropriate files have been submitted, and that your submitted zip can be opened and everything cleans, builds, and runs as expected.