



# HOMework 02

## Mode 3: Tamagotchi



**Purpose:** To build a simple game in Mode 3 to further your understanding of: inputs to the GBA, collision detection and reaction, C basics, and drawing in Mode 3.

---

### Instructions:

For this homework, you will recreate the game “**Tamagotchi**” in Mode 3. Specifically, this ball minigame, with a few tweaks. Review the *Requirements* section on the succeeding page for an explicit list of what we expect as the base requirements. The basic idea is as follows:

You have a new pet! Inspiration for this lab came from this minigame: [https://youtu.be/bHN\\_M4IcDgM](https://youtu.be/bHN_M4IcDgM). Of course your game is not expected to look exactly like the ORIGINAL, but we have a few basic gameplay things to have.

Moreover, you are encouraged to be creative! Go outside of the requirements to add some flair to your game. **You will receive a maximum of 95 out of 100 points for meeting all of the base requirements. By adding your own flair, you may be awarded the 5 additional points needed to receive a 100.**



## Requirements:

Your *game* must have the following:

- A Tamagotchi that can move around (yes this is not like the original device, but we are making it more interesting for this GameBoy version)
  - You can design your Tamagotchi however you want, as long as it is reminiscent of the original Tamagotchi.
- Balls/objects that will fall and hit the Tamagotchi. **Collision** with one of these balls will cause the game to **reset**.
- Occasional **food** items will appear, which will cause your Tamagotchi to grow into a new sprite. **Three total!**
- An end state
  - The game should end after your Tamagotchi grows three times.
  - You do not have to have anything happen once you reach the end state, as long as it can be understood that you have reached an end to the game
  - **To clarify:** We need to be able to see **all three versions** of your pet, so after your third growth, eating a food will cause the end state.
  - Your end state can be as simple as fill screening with a certain color! But feel free to make it more interesting.
- A **readme.txt** file
  - An instruction manual (of sorts) that tells a player how to play your game
  - Include what each button does (controls)
- Only a **minimal amount of flicker**

Your *code* must have the following:

- **Multiple .c** files
- At least **one .h** file
- Good organization (see tips below)
- Meaningful comments

### *Flair (optional):*

- Add “flair” to your game in order to receive **up to an additional 5 points**
  - Some ideas:
    - Any other features from previous Tamagotchi games
    - Additional minigames for the Tamagotchi
    - Menu systems
    - Anything you can think of!



- Please **highlight to us what you added for flair** and how to access it (don't send your grading TA on a wild goose hunt to find your flair!)
  - Either in your readme.txt or in a submission comment

## Tips:

- **Start early.** Never underestimate how long it takes to make a game.
- For collision code, draw pictures. Graph paper is your friend!
- When splitting code between multiple files, put code that will be useful in multiple games in your lib.c file, and code specific to this game in main.c or other files. Those other files should be specific to a concept (collision, etc.).
- Organize your code into functions specific to what that code does. **Your main method should not be very long.** Use helper functions!
- Having update() and draw() functions that you call in main() is helpful.
- Make sure the order in which you call your functions takes into account waiting for vBlank at the correct times. This will help you to minimize flicker.
- Build upon the corresponding lib.c and .h files we've provided in labs to **create your own LIB .c and .h files** (you can name it whatever you want!).

---

## Submission Instructions:

Ensure that **cleaning** and building/running your project still gives the expected results.

**Please reference previous assignments for instructions on how to perform a "clean" command if you need clarification.**

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 **HW02\_LastnameFirstname**, for example:

"HW02\_DoeJohn.zip"

**This is switched from previous submissions!** (to reduce TA headaches 😊)

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.