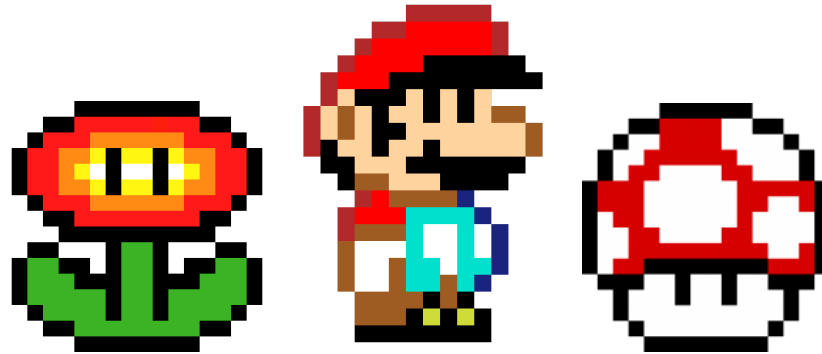




# HOMEWORK 01:

## Looping Animation



### Purpose:

- Creating a dynamic animation using VSync
- Drawing and erasing pixels in Mode 3

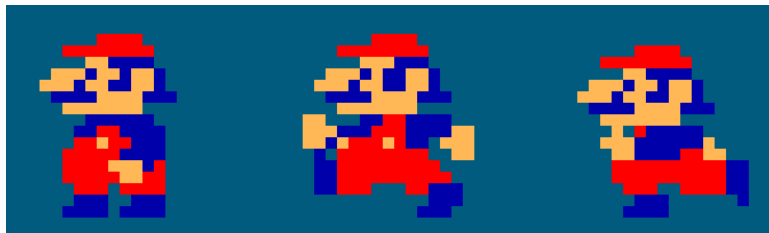
---

## Instructions

Before you can make games, you need to get a solid grasp on displaying imagery to the screen. To that end, you will create a looping animation of several frames (at least 4) in Mode 3. You will also implement button controls to handle various playback functions.

## Theme

The theme for this homework is **Mario**! Implement anything in the world of Mario, including any characters or objects like mushrooms or goombas.



3 frame walk animation for Mario. Note: your animation needs to have at least **4** distinct frames

We plan to select the best submissions and make a compilation of them to be shown during recitation and potentially posted to the class Canvas page. If you don't mind your submission being potentially put into that compilation, **please comment on your submission saying something like "I'm okay with my submission being shown to**



**the class”. If you want to be anonymous, please include that in your comment as well by saying something like “I want to remain anonymous”.**

Submissions will be selected on creativity, complexity, and visual appeal.

**There is no prize for being selected other than pride.** You can get a perfect grade on this assignment by meeting the criteria in the Requirements section. The primary purpose of this assignment is to help familiarize you with drawing in Mode 3 — making something artistically pleasing is secondary!

## Button Controls

You should implement the following controls:

- **A:** Advance to the next frame.
  - If there is no next frame, this button should cause the program to return to the first frame in the sequence.
- **B:** Return to the previous frame.
  - If there is no previous frame, this button should cause the program to show the last frame in the sequence.
- **Start:** Toggle looping through frames.
  - If the animation is not looping, it should start looping.
  - If the animation is already looping, it should stop looping.
- **Select:** Toggle the animation playing forwards/backwards.
  - Any time select is pressed, the animation should begin playing in the opposite direction it was previously.
    - Note: If start has not been pressed yet, it should begin to loop backwards.
  - When the animation is playing forwards and reaches the last frame, it should display the first frame in the sequence next.
  - Similarly, when the animation is playing backwards and reaches the first frame, it should display the last frame in the sequence next.
- **Left:** Skip to the first frame of the animation.
  - This does not need to pause the animation if it is already playing.
- **Right:** Skip to the last frame of the animation.
  - This does not need to pause the animation if it is already playing.

## Extra Credit

You may also implement these controls for bonus points (these are **not** required to receive full credit for this assignment):

- **Up:** Increase the playback speed of the animation (+2.5pts)
  - There should be a maximum speed and at least 5 total different speed levels.



- **Down:** Decrease the playback speed of the animation (+2.5pts)
  - There should be a minimum speed and at least 5 total different speed levels.

If both of these controls are fully implemented, you will receive 5 bonus points to your grade on this assignment. **If we cannot clearly tell that the speed is changing, then you will not receive points! Make sure to choose differences in speed that are visible, and choose a reasonable starting speed that allows us to properly see each frame of your animation.**

## Notes & Clarifications

- **The controls above refer to GBA buttons!** For example, A and B are *not* mapped to the A and B keys on your keyboard by default! Go to **Tools > Settings > Keyboard** to view or remap your control scheme on keyboard.
- For advancing and returning through animation frames, think of the current animation frame as a number ranging from [0, <number of frames>). You can then increase that number over time when the animation is playing or adjust it when the user presses different buttons. Importantly, if your current animation frame is the last one in the sequence, you need to reset to frame 0.
  - Similarly, if the animation is playing backwards and you need to move backwards from frame 0, you will need to instead change to the last frame of the animation.
- You **cannot** make an “animation” that is a bouncing ball or anything similar! We will not watch your program for 5 minutes waiting to see if it technically loops or not! Brickless Breakout will receive **0 points**.
- Mode 3 can only handle altering about 2,100 pixels per frame with the setPixel macro. If your animation is flickering or causing other visual artifacts, try reducing the size of your image.

---

## Requirements

- You must use the HW01 scaffold as a base for your assignment. This file includes macros and functions that we've completed in lectures, recitations, and labs. Note that some are not complete:
  - gba.h: BUTTON\_HELD, BUTTON\_PRESSED
  - gba.c: drawRectangle

You are required to fill these out.



- Follow **good code architecture**.
    - Not a bunch of set pixel calls in the main function.
  - The animation must **loop forever**.
    - The animation does not necessarily need to end where it started. You can just cut back to the first frame of the animation after the last frame. For example, you could have a cat jump off of a table, then have the animation restart with the cat back on the table again.
  - A **vertical sync function** must be used.
    - This function should only be called once in your entire project.
  - There must be at least **four meaningfully different frames** of animation.
    - This means at least two of:
      1. Shape
      2. Color
      3. Positionmust be different in each frame — just changing one of them is not enough!
  - There must be at least **three different colors**.
  - There must be a **minimal amount of flicker** (ask a TA if you are unsure!).
  - Your submission must adhere to the **theme** (ask a TA if you are unsure!).
  - All of the controls specified in the **Button Controls** section must work as stated.
- 

## Tips

- We ***highly recommend*** writing a separate function for every frame and calling these functions in an endless loop. You could use a variable to track the current animation frame, then call a different frame drawing function in your main while loop based on what the current value of that variable is.
- If you are having trouble with flicker, try drawing fewer pixels per frame. Don't draw a pixel unless you actually need to!
- If you need help getting started, drawing out a few frames on a 240x160 canvas in a simple paint editor or on graph paper may help.
  - You can use Microsoft Paint:
    - Open a new project, select "Resize".
    - Select "By Pixels".
    - Uncheck "Maintain Aspect Ratio".
    - Set "Horizontal" to 240 and "Vertical" to 160.
- Make sure your moving elements/shapes don't "wrap around" the GBA screen. In other words, check your boundaries!



- You will need to count GBA frames (remember, there are ~60 frames per second) as part of making the animation play over time. When this frame count reaches some value (determined by how fast the animation is playing) you should reset it to 0 and advance to the next animation frame.

---

## Submission Instructions

Ensure that **cleaning and building/running** your project still gives the expected results. Please reference the last page of this document 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 HW01\_LastnameFirstname, for example:

"HW01\_MartinetCharles.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.

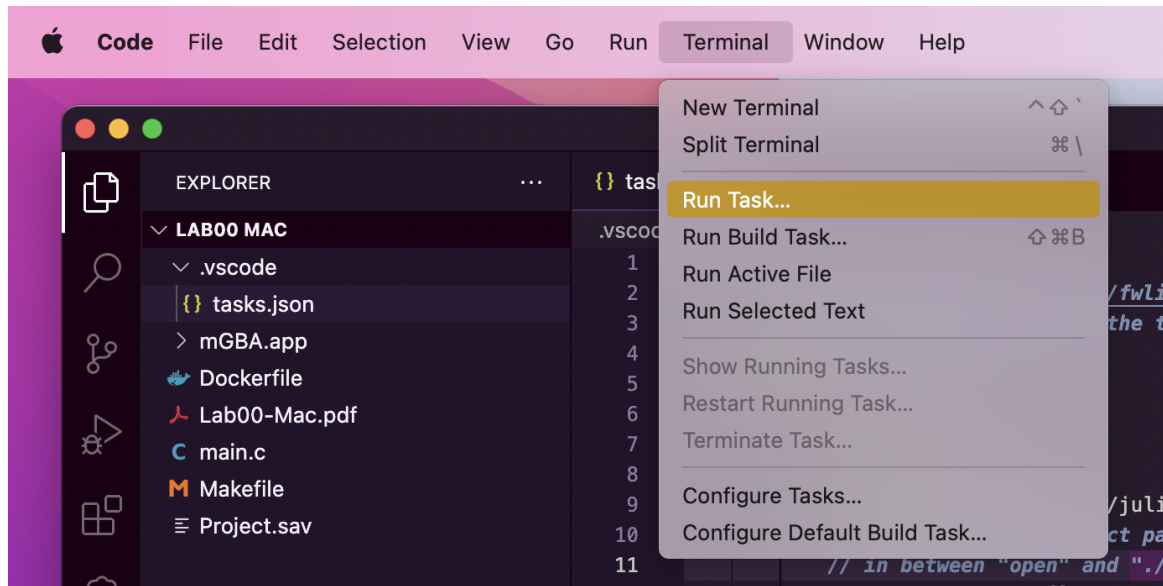
Please leave a comment on your submission which addresses **both**:

1. Whether you are okay with your submission being shared with the class.
2. Whether you wish to remain anonymous or to have your name shared along with the submission.

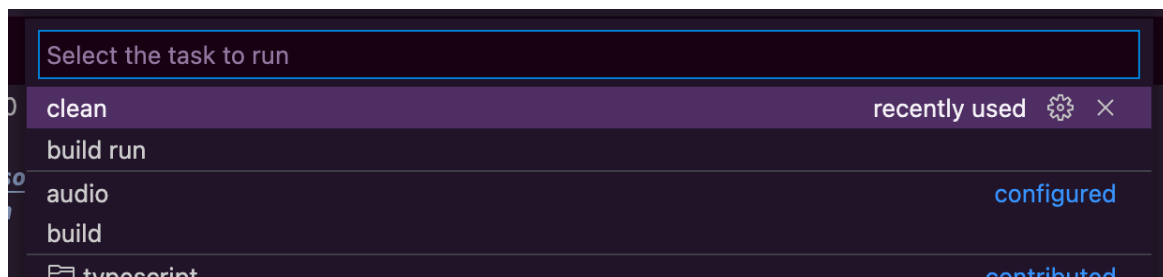
Example: "I am okay with my submission being shared with the class. I do not wish to remain anonymous."

## How to Clean

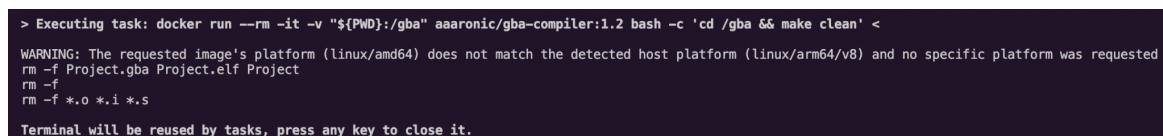
Navigate to the **Terminal** option at the top of your screen. Click on it, and then select **Run Task...**, as shown in the image below.



A dropdown menu will appear with the possible tasks to perform. Next, select **clean** from the options. Note that yours might be in a different order from mine, but the task should be there.



After selecting **clean**, something similar to the following should appear in your terminal tab (bottom of the screen).



If so, success! You have cleaned. You can now do **cmd/ctrl + shift + B** to build run (or click **Terminal**, then **Run Build Task...** or **Run Task...** and then **build run**).