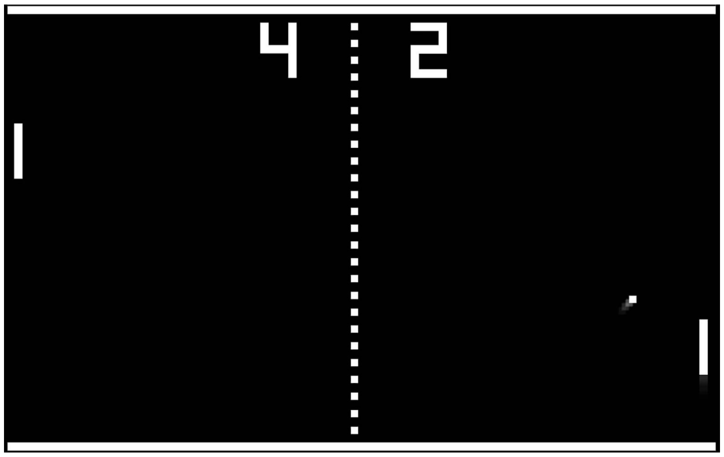
HOMEWORK 02

Mode 3: Pong



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 "**Pong**" in Mode 3. 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:

> There are two different paddles on opposite sides of the screen that can be controlled through user input. A ball moves around the screen and bounces off of the paddles. If a player misses the ball, the opposing player gets a point. The game ends when a player reaches 3 points, and that player wins!
>
> To get an idea of what the gameplay should look like, watch this video: https://www.youtube.com/watch?v=fiShX2pTz9A&ab_channel=andys-arcade. Of course your game is not expected to look exactly like the original pong (it can if you'd like), but the basic gameplay should still be recreated.

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:

- Two paddles that can collide with the ball and move meaningfully through *intuitive* user input controls (e.g. up/down & A/B)
    - If the ball collides with a paddle OR the top or bottom of the screen, it must “bounce off” and change direction. **Note**: If you decide to change the orientation of the paddles to be at the top and bottom of the screen, then the ball should bounce off of the left and right sides of the screen
- If the ball is missed by a paddle and hits the edge of the screen, the opposing player gets a point
- An end state
    - The game should end once any of the two players reaches 3 points
    - 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
- A **readme.txt** file
    - An instruction manual (of sorts) that tells a player how to play your game
    - Include which buttons are used for controlling the paddles
- 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:
        - Add a tally to the screen to indicate how many points each player currently has
        - Draw the ball as something other than a rectangle/square

        - Change the end state so that it indicates which player won

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 myLib.c, 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 myLib.c and myLib.h files we've provided in labs.

Submission Instructions:

Compress your entire project folder, including all source files, the Makefile, and everything produced during compilation (including the .gba file) into a single .zip file. Submit this .zip on Canvas. Name your submission HW02_FirstnameLastname, for example: "HW02_TimmyTurner.zip".