CS 2261 Lab 02:
Input and Collision

Provided Files

- `main.c`
- `myLib.c`
- `myLib.h`

Files to Edit/Add

- `main.c`
- `myLib.c`
- `myLib.h`
- Your Makefile

Instructions

In this lab, you will be completing several different TODOs, which will, piece by piece, make an interactive rectangle game.

After you download and unzip the files, add your Makefile, add the SOURCES with it, and then compile and run it. At this point, you should see just a blank black screen. Complete the TODOs on order, paying close attention to the instructions.

TODO #1 - drawRect

- Open `myLib.c` and complete the `drawRect` function.
- In `main.c`,find and uncomment UNCOMMENT#1 to see the fruits of your labor. You should see a green rectangle bounce around the screen, and a blue one sit unmoving in its path.

TODO #2

- Head back to `myLib.c`, find `fillScreen` (right beneath `drawRect`) and complete it.
- You may only use a single loop (one for-loop) to complete this one.
- Hint: the index of the top-right pixel is 239, and the index of the leftmost pixel in the next row is 240. Use this concept to your advantage.

- Enter main.c and uncomment UNCOMMENT#2. The background of your game should now be a lovely shade of cyan.

TODO #3

- This one requires a lot of moving parts to work, so it is broken into three parts.
- TODO #3.0
    - Open `myLib.h`, find the button macros, and complete them.
    - For `BUTTON_PRESSED`, assume that `buttons` and `oldButtons` have been set up appropriately.
- TODO #3.3
    - Since `buttons` and `oldButtons` *haven't* been set up correctly, head on back to `main.c` and initialize them in the `initialize` function.
    - They have already been declared in `main.c` (and in `myLib.h` as `extern`), so you don't have to worry about that part this time, but you will when you code things yourself for your homework.
- TODO #3.6
    - They still won't do anything unless you update them each frame, so do that in the main while-loop.
- After you have completed these, find `UNCOMMENT #3` in the `update` function and do the thing.
- Run it, and now you can press Start to toggle the background between cyan and a fabulous bright yellow.
    - Holding down the Start button for a long time should have the same effect as just tapping it a single time. If not, you did one of these three `TODO`s incorrectly.

TODO #4

- Now that you can take button input, find this `TODO` in the update function and make it so that the blue rectangle can move up, down, left, and right if you press the corresponding arrow key.
    - This time, it should move for as long as the key is held, meaning that if you tap it once quickly, it will barely move, but if you press it for several seconds, it will move a lot. If not, fix that.
- Compile and run, and you should be able to move the blue rectangle around as you please.

TODO #5

- This one will likely take you the longest, but is also the most important.

- TODO #5.0
    - At the bottom of `myLib.c`, implement the `collision` function. It takes in the positions and dimensions of two rectangles, and returns a 1 if they are colliding, or a 0 if not.
    - Hint: draw lots of pictures. Using graph paper, or drawing a grid yourself, is extremely useful. This function should work regardless of the size or velocity of the rectangle (if one is moving).
- TODO #5.5
    - Now that you have it, use it in the `update` function to make the green rectangle reverse direction when it hits the blue rectangle.
    - The green rectangle will just reverse direction, not properly reflect like it does with the walls.
        - Reflecting properly requires some additional code you won't have time for in lab, but may need for your homework.
        - Again: your collisions in your homework should not look like this ugly collision here. It will be penalized.
    - If both rectangles are moving, the collision will glitch. Don't worry about this for now.
        - If you want to fix this, ask in office hours.
    - Compile and run this, verify that it runs correctly, and then you are done.

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 Lab02_FirstameLastname, for example:"Lab02_TerenaAdare.zip".