CS 2261 Homework 04: Full Mode 3 Game

Instructions

In this assignment, you will be making a complex game in Mode 3 like the last assignment, but this time adding in states, text, and DMA to create a much more complete game. **You are free to extend the game you already made for the previous assignment, but you are welcome to create something new**. As a reminder, some suggested games along the complexity level we expect are:

- Breakout, with at least 3 rows of 5 blocks (Easy)
- Tetris (Hard)
- Simple flash games (ex. http://www.ferryhalim.com/orisinal/
- Simple Neopets games (http://www.neopets.com/games/)
- Old Atari games, like Asteroids (http://www.freeasteroids.org)

Read the following requirements very carefully. We will be grading these more harshly that in the previous assignment, so keep in mind that if you extend your previous submission, you may have to improve several of its qualities. This should not be difficult, as DMA allows you to draw way more per frame anyway. Your game must have the following:

- At least one struct
- At least one array
- Pooling
- Meaningful text
  - It can't just be present; it needs to be relevant to the game
- Non-static text
  - This means it changes while you are looking at it (erased then redrawn)
    - Score that visibly updates during the game is a good example
- The following states: Start, Game, Pause, Win/Lose
  - You can have either a Win state, a Lose state, or both.
  - Each state (except game) must be labeled with text.
- DMA used correctly for `fillScreen()` and `drawRect()`.
- At least five moving objects
- At least three buttons used for input

(continued on the next page)

- Collision that matters
    - Something must happen whenever two different objects hit each other
- A readme.txt file
    - An instruction manual (of sorts) that tells a player how to play your game
- Only a *minimal* amount of flicker.

Your code must have the following:

- Multiple .c files (more than just `main.c` and `myLib.c`)
- At least two .h files
- Good organization (see tips below)
- Meaningful comments

Tips

- Start early. Never underestimate how long it takes to make a game.
- **Do not draw text every frame**. It takes a while to draw, so only update it when it needs to update
    - For score, only redraw when the score is different than before.
- 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 function should not be very long.
    - Having `update()` and `draw()` functions that you call in `main()` is helpful.
    - Make sure the order takes into account waiting for vblank at the correct times to minimize flicker.
- Build upon the `myLib.c` and `myLib.h` files from previous assignments (especially Lab05).

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 T-Square. Name your submission HW04_FirstnameLastname, for example: “HW04_OpoChano.zip”.