

CS 2110 Timed Lab 6

Direct Memory Access

Austin Adams, Lauren Chen, Cem Gokmen, and Preston Olds

Fall 2018

ASSIGNMENT DESCRIPTION BEGINS ON PAGE 4

Contents

1	Before You Begin	2
2	Timed Lab Rules - Please Read	2
2.1	General Rules	2
2.2	Submission Rules	2
2.3	Is collaboration allowed?	3
3	Overview	4
3.1	Provided Files	4
3.2	Running the Animation	4
3.3	Completing the Animation (t106.c)	4
4	Instructions	5
4.1	Getting the Correct Animation Frame	5
4.2	Drawing the Images	5
5	Testing Your Work	6
6	Rubric	6
7	Deliverables	6

1 Before You Begin

Please take the time to read the entire document before starting the assignment. We have made some important updates, and it is your responsibility to follow the instructions and rules.

2 Timed Lab Rules - Please Read

2.1 General Rules

1. You are allowed to submit this timed lab starting at the moment the assignment is released, until you are checked off by your TA as you leave the recitation classroom. Gradescope submissions will remain open until 7:15 pm - but you are not allowed to submit after you leave the recitation classroom under any circumstances. **Submitting or resubmitting the assignment after you leave the classroom is a violation of the honor code - doing so will automatically incur a zero on the assignment and might result in you being referred to the Office of Student Integrity.**
2. Make sure to give your TA your Buzzcard before beginning the Timed Lab, and to pick it up and get checked off before you leave. **Students who leave the recitation classroom without getting checked off or submit after getting checked off will receive a zero.**
3. Although you may ask TAs for clarification, you are ultimately responsible for what you submit. **The information provided in this Timed Lab document takes precedence.** If in doubt, please make sure to indicate any conflicting information to your TAs.
4. Resources you are allowed to use during the timed lab:
 - Assignment files
 - Previous homework and lab submissions
 - Your mind
 - Blank paper for scratch work (please ask for permission from your TAs if you want to take paper from your bag during the Timed Lab)
5. Resources you are **NOT** allowed to use:
 - The Internet (except for submissions)
 - Any resources that are not given in the assignment
 - Textbook or notes on paper or saved on your computer
 - Email/messaging
 - Contact in any form with any other person besides TAs
6. **Before you start, make sure to close every application on your computer.** Banned resources, if found to be open during the Timed Lab period, will be considered a violation of the Timed Lab rules.
7. We reserve the right to monitor the classroom during the Timed Lab period using cameras, packet capture software, and other means.

2.2 Submission Rules

1. Follow the guidelines under the Deliverables section.

2. You are also responsible for ensuring that what you turn in is what you meant to turn in. After submitting, you should be sure to download your submission into a brand new folder and test if it works. There are no excuses if you submit the wrong files; what you turn in is what we grade. In addition, your assignment must be turned in via Gradescope. Under no circumstances whatsoever will we accept any email submission of an assignment. Note: if you were granted an extension, you will still turn in the assignment over Gradescope.
3. Do not submit links to files. We will not grade assignments submitted this way as it is easy to change the files after the submission period ends.

2.3 Is collaboration allowed?

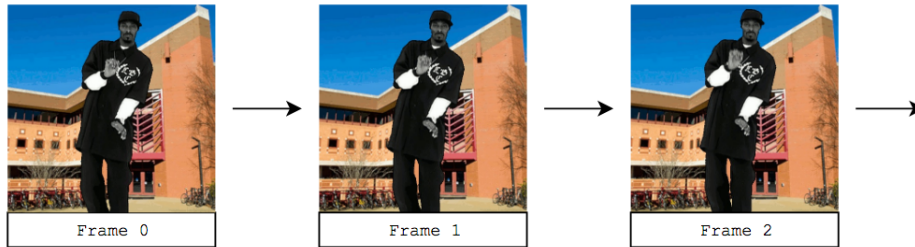
Absolutely NOT. No collaboration is allowed for timed labs.

ASSIGNMENT DESCRIPTION BEGINS HERE

3 Overview

For this assignment, you will complete five functions to draw an animation with 60 frames on the GBA screen. The animation has been provided as an array of `unsigned short` in `snoopframes.c`. We have also provided for your reference `snoopframes.bmp`, which displays the array in `snoopframes.c`.

The animation contained in the array has been provided for your reference as (`snoop-inputanimation.gif`). After completing the functions correctly, the animation should display as `snoop-outputexpected.gif`.



Note: In the `snoopframes` array, frames 0 → 14 are not flipped at all, frames 15 → 29 are flipped horizontally only, frames 30 → 44 are flipped both ways, and frames 45 → 59 are flipped vertically only.

3.1 Provided Files

Several files are provided for running the animation:

`main.c` : Calls your code to draw the animation. You don't need to touch this.

`gba.c` : Library functions for GBA and Mode 3

`gba.h` : Header file for GBA and Mode 3, as well as function prototypes

`dma.h` : Header file for DMA

`uint.h` : Header file for integer typedefs

`font.c` : A font array used to draw text

The animation frames in `snoopframes.c` are accompanied by the header file `snoopframes.h`.

3.2 Running the Animation

Please use the provided `Makefile` to start the program with `make emu`.

Once the animation has begun, you can manipulate it with the following input:

Button A (Z) : Play and pause the animation

Button L (A) : If paused, step one frame backward

Button R (S) : If paused, step one frame forward

3.3 Completing the Animation (`t106.c`)

You must implement the following five functions in `t106.c`:

1. `getPointerToCurrentFrame`
2. `drawSquareImage`
3. `drawSquareImageFlippedHorizontal`
4. `drawSquareImageFlippedVertical`
5. `drawSquareImageFlippedBoth`

Note: The only file you need to edit is `t106.c`.

4 Instructions

4.1 Getting the Correct Animation Frame

The animation frames are consecutive to one another in the `snoopframes` array. Observe that each frame is 160×160 (i.e. 25,600) pixels, there are 60 frames in the animation, and `snoopframes` has 1,536,000 unsigned shorts: $(160 \times 160) \times 60 = 1,536,000$.

Open `snoop-inputimage.bmp` to see the image this array was created from.

The function `getPointerToCurrentFrame` takes as parameters: a pointer to the beginning of the frames array (`frames`), the dimension of each square frame (`img_dimension`), and the current frame (`count`).

You must calculate and return a pointer to the beginning of the correct frame.

4.2 Drawing the Images

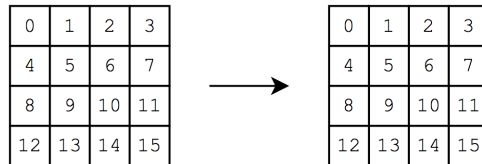
Some of the frames in the animation have been flipped horizontally, vertically, or both ways. To properly draw the animation, you will need to implement four functions with Direct Memory Access (DMA).

Note: Calling the correct functions on the correct frames is taken care of for you in `main.c`, provided you've correctly implemented `getPointerToCurrentFrame`.

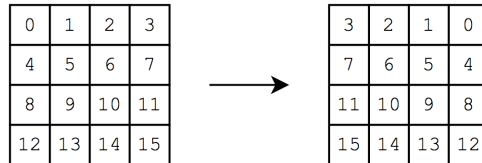
All four of these functions will draw the `image` starting at the top left corner of the screen. Each `image` has an `img_dimension` which is the size of its width, the same as its height. Remember, **you must use DMA!**

For more details on implementation please read the comments provided in `t106.c`.

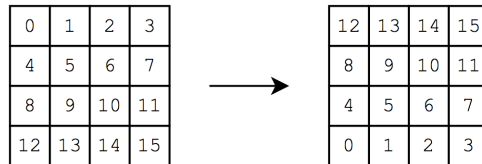
1. `drawSquareImage(const u16 *image, int img_dimension):`



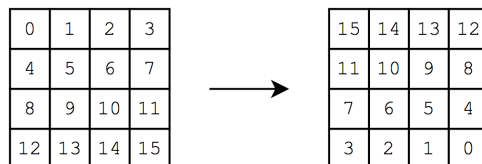
2. `drawSquareImageFlippedHorizontal(const u16 *image, int img_dimension):`



3. `drawSquareImageFlippedVertical(const u16 *image, int img_dimension):`



4. `drawSquareImageFlippedBoth(const u16 *image, int img_dimension):`



5 Testing Your Work

To debug your work, run `make emu` in the directory and step through frame-by-frame to check how your functions are drawing to the screen.

Uploading `t106.c` to the Gradescope assignment will check your assignment against our tests. You may resubmit your work as many times as needed, until you sign out and leave the classroom.

6 Rubric

The output of the Gradescope autograder is an approximation of your score on this assignment. The tool is provided so you can evaluate whether your submission fulfills the assignment expectations.

However, we reserve the right to run additional tests, fewer tests, different tests, or potentially change individual tests – your final score will be determined by your instructors, and there is no guarantee your score will correlate with the tester output.

7 Deliverables

Please upload the following file to the assignment on Gradescope:

1. `t106.c`

Do NOT upload an archive; upload the file individually.

Be sure to check your Gradescope test score before you leave the room.