CS 2110

Timed Lab 5

Due Date and Time

Day: Wednesday, November 9th
Time: Before the end of your assigned lab section

Policy

Submission

TURN IN THIS ASSIGNMENT ELECTRONICALLY USING T-SQUARE

SUBMISSIONS WHICH ARE LATE WILL NOT BE ACCEPTED.

EMAIL SUBMISSIONS WILL NOT BE ACCEPTED UNDER ANY CIRCUMSTANCES!

IF YOU FORGET TO HIT THE SUBMIT BUTTON YOU WILL GET A ZERO.

Questions

If you are unsure of what questions mean, the TA's will clarify them to the best of their ability. We will not be able to answer any questions about how to reach a solution to the timed lab questions. You should know how by now!

What's Allowed

- The assignment files
- Your previous homework and lab submissions
- Your mind
- Blank paper for scratch work

What's Not Allowed

- The Internet (except the T-Square Assignment page to submit)
- Any resource on T-Square that is not given in the assignment.
- Textbook or notes on paper or saved on your computer.
- Dropbox (If your hard drive crashes we will let you retake it).
- Email/IM
- Contact in any form with any other person besides TAs

If you have any questions on what you may not use then assume you can't use it and ask a TA.

Other Restrictions

1. You may not leave the classroom until we have verified that you have submitted the lab. If you leave the classroom without submitting you will receive a zero.

2. **YOU MUST SUBMIT BY THE END OF YOUR LAB PERIOD**. Bear in mind that the clock on your computer may be a few minutes slow. You are supposed to have a full class period to work, and we are letting you use the 10 minutes between classes to make sure you have submitted your work. **WE WILL NOT ACCEPT LATE SUBMISSIONS**, be they 1 second or 1 hour late.

3. The timed lab has been configured to accept one submission. If you accidentally submit or submit the wrong version flag one of the TAs and we will reopen submission for you.

Violations

Failure to follow these rules will be in violation of the Georgia Tech Honor Code **AND YOU WILL RECEIVE A ZERO** and you will be reported to Bill and the Office of Student Integrity.

We take cheating and using of unauthorized resources **VERY SERIOUSLY** and you will be in serious trouble if you are caught.

Remember

1. There is partial credit given, and some of it is just following the directions.

2. We allow you to use your homework assignment.

3. Please don't get stressed out during a timed lab. You have plenty of time; however, use your time effectively

4. Again, remember: Don't get stressed. Partial credit will be given for things you have done correctly. Do the best you can!

5. If you don't know something at least TRY. Do not just walk out of the lab or submit an empty file. Partial credit!

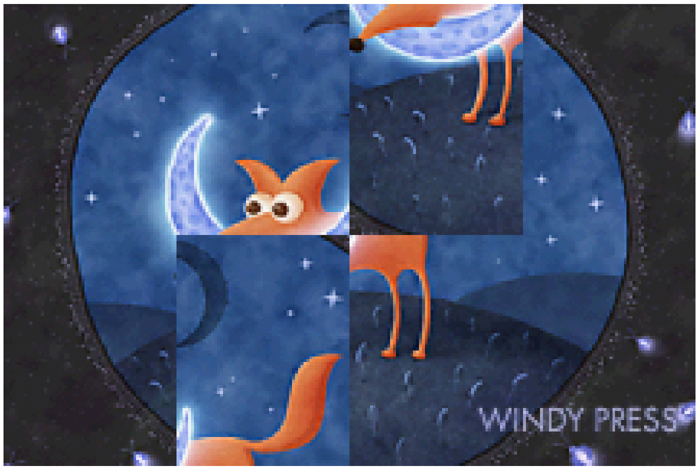
6. Remember what you can and can't use. If you don't know, then don't use it and ask a TA if you can use it. If we catch you with unauthorized resources we will give you a zero, so better to be safe than sorry.

The Assignment

Overview

One of our TA's was playing a picture puzzle game, but he got bored at the last minute and left the puzzle unfinished. Your job is to help him finish it.

You will be given an image like this:



Notice that two pieces of the image are swapped. We will call these pieces "cells". You need to complete the puzzle by displaying the image with the two cells swapped back correctly on the GBA screen.

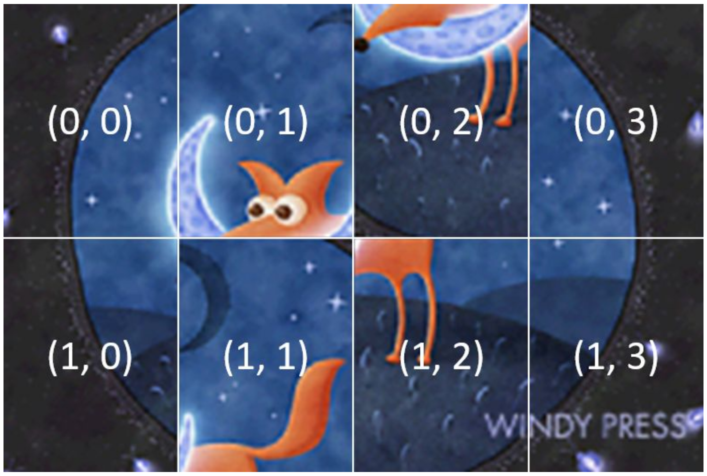
In the code, you will notice that we have given you some values:

cellA and **cellB** are the two swapped cells. These variables are of type **CELL**, as defined in *myLib.h*:

```
typedef struct {
    int row;
    int col;
} CELL;
```

NUM_ROW_CELL and **NUM_COL_CELL** tell you how many cells there are per row and per column.

For the first image, there are 4 cells per row and 2 per column, and cells (0,2) and (1,1) have been swapped.



Each cell is identified by its (row, col)

So how do we swap the cells? First you need to draw the image as is onto the GBA screen. The image size will always be 240x160 (full screen), so you need not make your function work for images of variable size. Complete **drawFullScreenImage()** and use this function in main().

The next step is to draw the cells from the source image in their correct locations. You will draw cell A in the source image over cell B and vice versa. To do this, you need to complete **drawCell()**, which is a function that takes a srcCell from the image and copy it to the dstCell in the video buffer. Once done, call this function in main().

Note: We will test you with two images. You can assume that the images are always of size 240x160 and only two cells are swapped in any image.

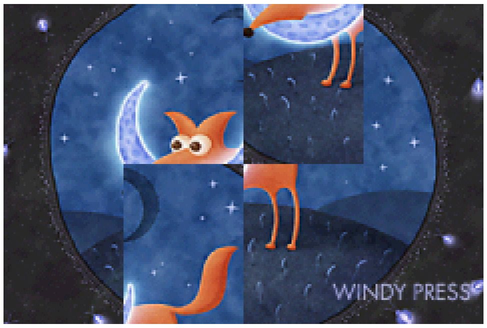
Requirements

- Do not modify the original image array. All you need to do is display the image correctly.
- You must use DMA for this timed lab. If you set each pixel individually, you will receive no points. You will also need to use DMA efficiently.
- You must follow the steps laid out above. Do not make any extra functions.

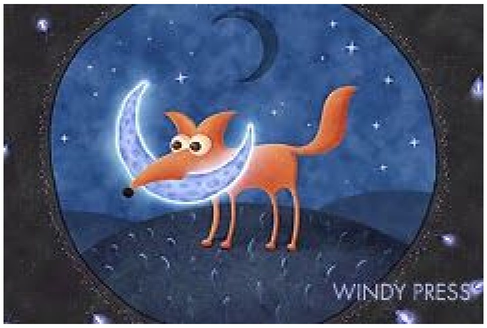
Testing

Run "make vba" to test your code. Your code should draw the image on the right. To test the second image, uncomment the block of code under *Image 2* (and comment out the block of code under Image 1).

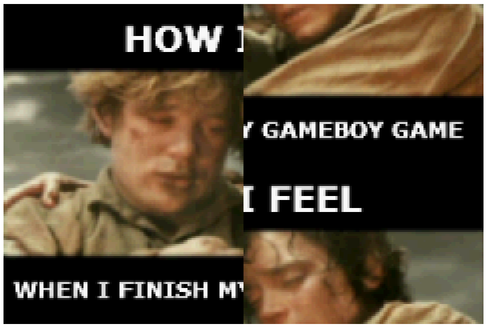
Given image



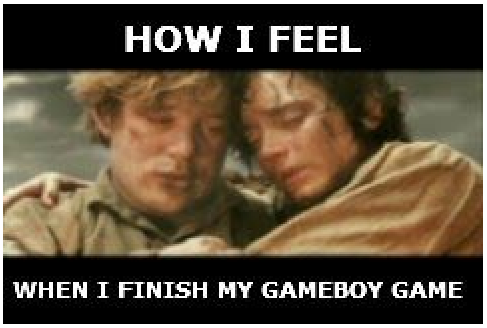
Expected GBA screen



Given image



Expected GBA screen



Implementation Hints

- drawCell: Think about how you calculate the offset for the top-left pixel of a cell. You should already know how to calculate the offset if you know the x and y coordinates of a pixel. Think about how you can convert (cell_row, cell_col) into (x, y), and then into an offset.

Deliverables

1. main.c

Note if your file is not named this you may lose points!

You may submit only the files listed above. We will not accept any internet links we want the files above and only these files!

Check over your submission after you submit it. If you submit the wrong file and leave the lab I will not be happy and we will grade what you submit so please check over what you submitted after you submit it!

Have fun and good luck!