

CS 2110 Timed Lab 6: GBA ”Revealing Boo”

Farzam, Joshua, Maddie, Jim, Vivian

Spring 2019

Contents

1	Timed Lab Rules - Please Read	2
1.1	General Rules	2
1.2	Submission Rules	2
1.3	Is collaboration allowed?	3
2	Overview	3
2.1	Description	3
2.1.1	The Program	3
3	Instructions	4
3.1	drawBackgroundPortion	4
3.2	getNextLocation	4
3.3	drawPlayer	4
3.4	playerWithinBoxBounds	4
3.5	Testing your work	4
4	Important Notes	5
5	Rubric	5
5.1	Autograder	5
6	Deliverables	5

Please take the time to read the entire document before starting the assignment. It is your responsibility to follow the instructions and rules.

1 Timed Lab Rules - Please Read

1.1 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 be 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 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
 - Any compiler that outputs LC3 code
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.

1.2 Submission Rules

1. Follow the guidelines under the Deliverables section.
2. You are also responsible for ensuring that what you turned 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. 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.

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.

1.3 Is collaboration allowed?

Absolutely NOT. No collaboration is allowed for timed labs.

2 Overview

2.1 Description

For this timed lab you will be finishing up someone's GameBoy program!

2.1.1 The Program

We've given you a partially complete GameBoy program. In this program there's a nice background image, but parts of it are missing! Thankfully there is a helpful player, Mr. Boo, who will jump around the screen. If Mr. Boo hits a part of the screen that is missing, he should redraw **ONLY** that portion of the screen. Your job is to finish 4 functions needed for the program to run properly!



Figure 1. The current state of the GameBoy program (boo not drawn). Your job is to help Mr. Boo fill in the missing parts of the image!

3 Instructions

Take a look at `tl6.h` to familiarize yourself with the structs used in the game and the `playerCoordinates` array which will be used later on. Keep in mind that the required player and boxes will be initialized for you and your job is to simply implement the following methods:

1. `drawBackgroundPortion`
2. `getNextLocation`
3. `drawPlayer`
4. `playerWithinBoxBounds`

3.1 `drawBackgroundPortion`

This function takes in attributes of a `Box` (`row`, `col`, `width`, `height`) and the ENTIRE background image as parameters. Your job is to use **DMA** to draw only a portion of the background image. This portion should be a rectangle with the top left starting at (`row`, `col`) and with the corresponding width and height. The background is guaranteed to be `SCREEN_WIDTHxSCREEN_HEIGHT` which is `240x160`.

3.2 `getNextLocation`

This function takes in a counter as its only argument and returns the next location of the player "boo" from the `playerCoordinates` array. If the index is out of bounds of the array, return the coordinate `0,0`. **This array is of size `PLAYER_MOVES` and will be initialized with valid coordinates for you, do not modify this array.**

3.3 `drawPlayer`

Use `nextCoordinates` to get the next location of the player, passing in the number of times the player has been drawn so far, which **you should keep track of in this function**. After getting the `nextCoordinate` of the player, draw the player at this location using the `drawBoo` function. Look at the function comments or `gba.h` for `drawBoo`'s function prototype. Keep in mind that this function is already defined for you, your job is to call it with the correct coordinates and the already included image 'boo'. Please take a look at the important notes before beginning coding.

3.4 `playerWithinBoxBounds`

This is essentially checking for a collision between the player and the box. Make sure you cover every edge case in this method as the autograder checks for that! This function should return `1` if there is a collision and `0` otherwise.

3.5 Testing your work

Test your work using:

```
make emu
```

In the docker image, to see if the program is working properly.

You'll know it's working properly when Mr. Boo successfully jumps around the screen and whenever he hits a black part of the background, the background is filled in there.

Note: Just because your image looks correct does not necessarily mean it is correct. The grader on gradescope tests for all edge cases and ensures you followed our instructions.

4 Important Notes

1. Due to the nature of this GBA game, there might be cases when you get to the ending point of the game in the emulator (completing the image with tearing, etc.), however, your grade from the auto-grader is what we check for. Therefore you are required to make sure all tests cases pass in order to receive full credit.
2. Do not modify any other files besides `tl6.c` (although feel free to read the other files).
3. Do not alter the `#include` list or add any global variables. Any attempts to circumvent this will result in severe penalties. There is a test to check for this and it will show the line that it is failing on.
4. Feel free to remove the `UNUSED_PARAM` calls in each function (except the last `OPTIONAL` method if you are not going to use it). Also the initial return statements are solely for turning off compiler errors so you can see the initial state of the program.

5 Rubric

5.1 Autograder

There is **no** local autograder for this timed lab. Submit to gradescope to run the autograder on your submission (as always you have unlimited submissions during the timed lab period).

6 Deliverables

Please upload the following files to Gradescope:

1. `tl6.c`

Download and test your submission to make sure you submitted the right files