

CS 2261 Lab 00:

Installation

Provided Files

- InstallationFiles
 1. SublimeTextSetup.txt
 2. devkitARM
- SampleProject
 1. Main.c
 2. Makefile

Files to Edit/Add

- Makefile

Instructions

In this lab, you will be installing the software to write, compile, and run GBA games for this class. It is broken up into various parts. If any part does not produce the expected outcome, alert a TA and fix the problem before continuing.

- **Part One – Compiler**

- ❖ This is the biggest difference between the Windows and Mac installations, since the Mac folk (usually) already have a C Compiler installed. Let's check that.

1. OpenTerminal(Applications>UtilitiesorApplications>Accessories).
2. Type "gcc" and press Enter. You should see something like the following:

```
lawn-128-61-53-109:~ YourName$ gcc
clang: error: no input files
```

3. Then try typing "make" and pressing Enter.You should see the following:

```
lawn-128-61-53-109:~ YourName$ make
make: *** No targets specified and no makefile
found. Stop.
```

4. If you see those things, you can skip to Part Two. If you see "command not found" or something like that, you need to obtain XCode Command Line Tools or reinstall GCC and Make.
 - Sometimes Terminal will automatically ask you to install Command Line Tools. If it does, just follow the instructions to do so.
 5. If it does not ask you to install the Command Line Tools, type "xcode-select --install" and press Enter.
 6. Allow it to finish, then close Terminal.
 7. Open Terminal again, type "gcc", then press Enter. If you see what you expected to see earlier, then the tools have installed correctly. If not, alert a TA.
- ❖ Congrats! You can now compile C code.

- **Part Two – devkitARM**

- ❖ DevkitARM is your development kit for the Game Boy Advance. Without it, we can compile C code, but not actually format the result for the Gameboy. So let's fix that.
- ❖ A zip file titled "devkitARM.bz2" has been included in this project folder. Download and unzip it in a folder somewhere that you will not move or delete until this class is over. Make note of this location, because you will have to find it in a later step.

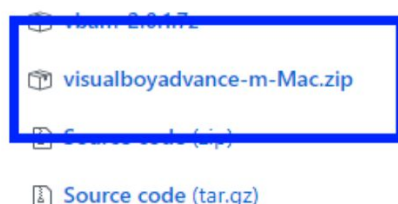
- **Part Three – VisualBoyAdvance-M**

- ❖ VisualBoyAdvance-M is your emulator. Since we can now compile and format our code, it would be useless if we couldn't actually run it. So let's fix that.
 - Note: if you already have a GBA emulator that you are comfortable with, I still highly recommend you use this one for this class. It has some special features that will come in handy for debugging.

1. Go to the project website, <https://github.com/visualboyadvance-m/visualboyadvance-m/releases>

 Zachbacon released this on Oct 11, 2017 · 11 commits to master since this release

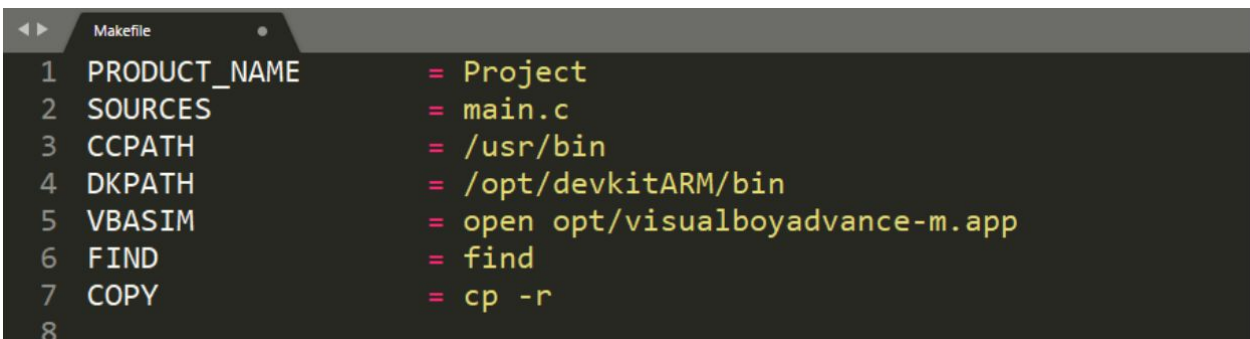
Assets



2. Click on “visualboyadvance-m-Mac.zip” to download it.
3. Unzip the file (if that wasn’t done automatically) and move the.app file somewhere that you will not move or delete until this class is over. The same folder where you put devkitARM will usually suffice. Make note of this location, because you will have to find it in a later step.

- **Part Four – Makefile**

- ❖ Now that we have installed everything, we need to set up the compilation process so that we don’t have to run 1,000 commands every time we want to compile our code. C uses Makefiles for this. I have provided one for you, but it needs to know where to find the things you just installed.
1. In the provided Sample Project, find `Makefile` and open with a text editor (like Sublime Text). Near the beginning, you will see three lines that end in “=”.
 2. After the “=” next to “CCPATH”, type the path to your Command Line Tools (this is almost always “/usr/bin”, so type that if you are unsure). If there are any spaces in the path, surround it in quotes.
 3. After the “=” next to “DKPATH”, type the exact path to the `devkitARM` folder, and add “/bin” to the end. If there are any spaces in the path, surround it in quotes.
 4. After the “=” next to “VBASIM”, type “open”, then a space, then the exact path to `visualboyadvance-m.app`. If there are any spaces in the path, surround it in quotes.
 5. Make sure the result looks something like the following picture.



```
1 PRODUCT_NAME      = Project
2 SOURCES            = main.c
3 CCPATH             = /usr/bin
4 DKPATH             = /opt/devkitARM/bin
5 VBASIM             = open opt/visualboyadvance-m.app
6 FIND              = find
7 COPY              = cp -r
8
```

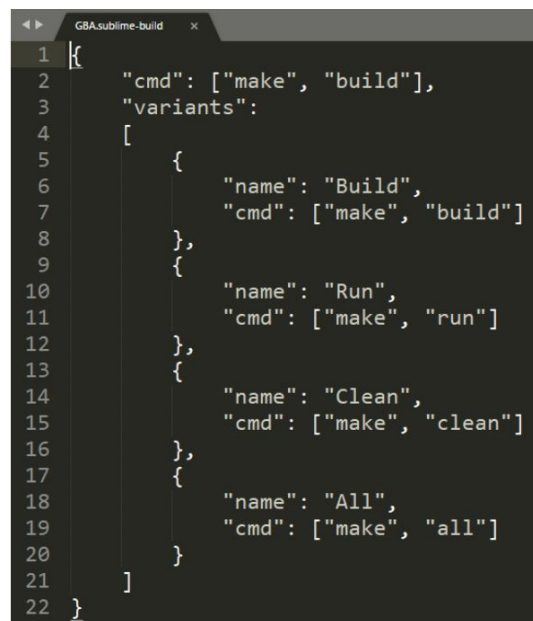
6. Save and close the file.
- ❖ Congrats. We now officially have everything it takes to simply compile, format, and run GBA games. Now let’s make it simpler.

- **Part Five – Sublime Text**

- ❖ We can make the process even more streamlined by incorporating the

compilation process into our text editor. For this class, we highly encourage Sublime Text. Even if you have Sublime Text installed, do not skip these steps.

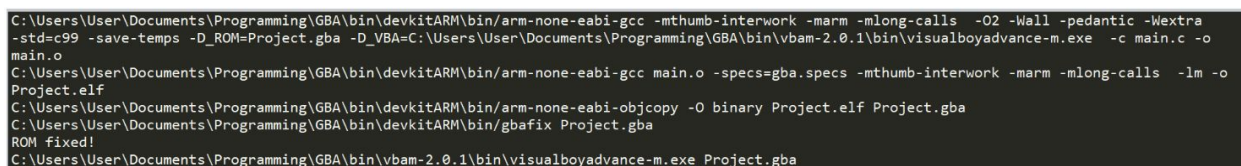
1. If you don't have SublimeText installed, download it from the project website, <https://www.sublimetext.com/3>. I don't need to walk you through this installation process; it's user-friendly enough.
2. Once you have SublimeText installed, open it, then go to Tools, then Build System, then click "New Build System..."
3. It will open a new file. Delete everything in it, then paste in everything from the provided SublimeTextSetup.txt.



```
1 {
2   "cmd": ["make", "build"],
3   "variants":
4   [
5     {
6       "name": "Build",
7       "cmd": ["make", "build"]
8     },
9     {
10      "name": "Run",
11      "cmd": ["make", "run"]
12    },
13    {
14      "name": "Clean",
15      "cmd": ["make", "clean"]
16    },
17    {
18      "name": "All",
19      "cmd": ["make", "all"]
20    }
21  ]
22 }
```

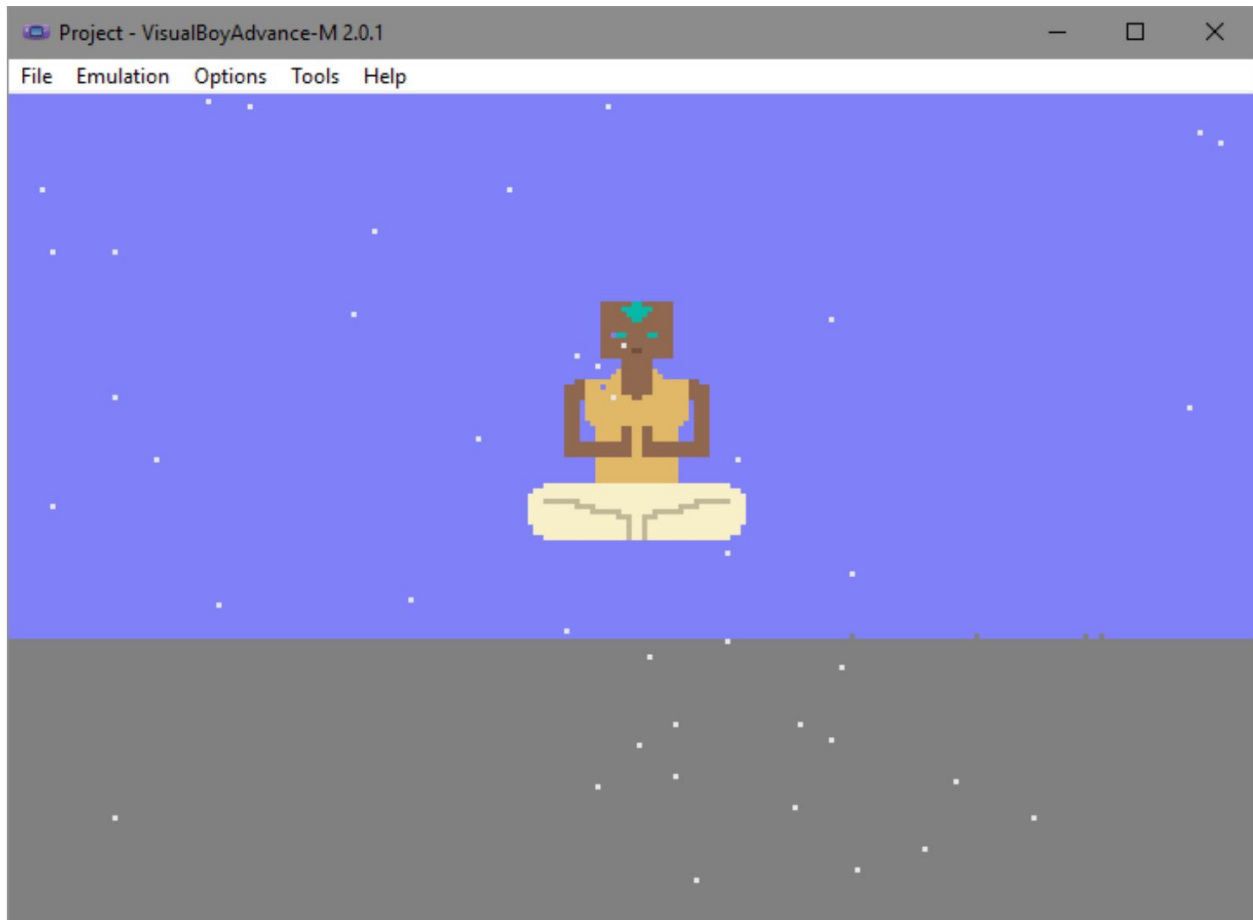
4. Save it in the folder it suggests when you hit save, and title it "GBA.sublime-build".
5. Close everything in Sublime Text, and then close Sublime Text.
6. Reopen Sublime Text, then use it to open Makefile in the Sample Project folder.
7. Go to Tools, then Build System, and make sure GBA is selected.
8. Go to Tools, then Build With, then select Run.

After completing all of these steps, you should see output in the console at the bottom of Sublime Text that looks something like the following picture.



```
C:\Users\User\Documents\Programming\GBA\bin\devkitARM\bin\arm-none-eabi-gcc -mthumb-interwork -marm -mlong-calls -O2 -Wall -pedantic -Wextra -std=c99 -save-temps -D_ROM=Project.gba -D_VBA=C:\Users\User\Documents\Programming\GBA\bin\vbam-2.0.1\bin\visualboyadvance-m.exe -c main.c -o main.o
C:\Users\User\Documents\Programming\GBA\bin\devkitARM\bin\arm-none-eabi-gcc main.o -specs=gba.specs -mthumb-interwork -marm -mlong-calls -lm -o Project.elf
C:\Users\User\Documents\Programming\GBA\bin\devkitARM\bin\arm-none-eabi-objcopy -O binary Project.elf Project.gba
C:\Users\User\Documents\Programming\GBA\bin\devkitARM\bin\gbafix Project.gba
ROM fixed!
C:\Users\User\Documents\Programming\GBA\bin\vbam-2.0.1\bin\visualboyadvance-m.exe Project.gba
```

You should also see a GBA game running that looks something like the following picture.



If so, you are done with the lab. If not, there is an issue somewhere.

If you are smart, you will install Wine now.

Submission Instructions

Zip up the entire sample 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 Lab00_FirstnameLastname, for example: "Lab00_ChianneConnelly.zip".