

PS0

Hello World with SFML

The main purpose of this assignment is for you to get up and running with your build environment. You'll write a bit of code.

1. Get your build environment set up

You must use [GCC \(g++\)](#) and [make](#) to configure and build your code. The reference compiler for the course is [g++ 7.3.0](#). It has support for C++ 11 and later.

For more information about Virtualbox see

<https://www.virtualbox.org/manual/ch01.html#hostsupport>

In this course, we will assume that you use the provided Ubuntu Linux virtual machine. If you prefer to use a Mac, or to configure your own Linux machine, please see the separate handouts on Blackboard for configuration hints; however, the instructor will not provide support for those environments.

We are using the C++ Simple Fast Media Library, SFML, www.sfml-dev.org. Any version 2.0 or later will work (as of 9/1/2018, the current version is 2.5.0).

2. Build the SFML Hello World code

The SFML “did I install everything correctly” code is about 20 lines long and results in your computer popping up a window like this:



If you're not using the provided VM, go to the tutorial for your platform -- e.g., <https://www.sfml-dev.org/tutorials/2.5/start-linux.php> -- and get the code running on your machine.

3. Extend the demo code.

Now that you've got things running, do something fun with SFML. In addition to the green circle, look at the documentation and get your window to show:

1. drawing an image sprite
2. make the image sprite move
3. make the image sprite respond to keystrokes
4. make it do something else

Use the documentation at <https://www.sfml-dev.org/documentation/2.5.0/>.

You probably want to make your window bigger than the 200x200 specified in the Hello World code. Also, no need for audio now (we will be using it later).

For #2 and #3 above, you can make x, y variables that represent the position of the sprite, and update them when a keystroke is detected. You're welcome to do something more complicated if you like.

The **`window.setFramerateLimit()`** method might be helpful if things move too fast (see "Controlling the framerate" at <https://www.sfml-dev.org/tutorials/2.5/window-window.php>).

Notes:

- It's correct to clear everything in the window and redraw stuff each time through the event loop. That's how SFML works.
- You reposition things (e.g. a sprite) by setting its position before you draw it. Check the sprite API docs.
- Use only relative paths to load your sprite image— e.g. `./sprite.png`, *not* `/home/fredm/ps0/sprite.png`.
- If you draw stuff outside the bounds of the window, you won't see it :)

Compiling program:

- When you compile your code, add the flags `-Wall -Werror -ansi -pedantic`. We will be using these to test your code, and your code must compile when these flags are used.
- To use C++11 features use flag `-std=c++11`

Make sure your code file is named **main.cpp**. You will submit this. Make sure your name is at the top of the file.

Make sure your sprite image file is named **sprite.png**. You will submit this.

When you are done, take a screenshot of just your SFML window, and save this image in a file named **screenshot.png**. You will submit this.

If you are new to Ubuntu, we highly recommend [Shutter](#) for doing screen grabs of windows or a selected area of the screen. Install with:

```
sudo add-apt-repository ppa:shutter/ppa
sudo apt-get update
sudo apt-get install shutter
```

The Print Screen key also works, but you'll have to crop the grab afterward to just show your SFML window.

4. Assignment readme File

Before submitting, fill out the requested info in the [ps0-readme.txt](#) file provided on Blackboard. Make sure you re-save it as a text only file. It must be named **ps0-readme.txt**.

5. Submit!

You will be submitting at least four files.

1. Your SFML demo program, named **main.cpp**.
2. Your sprite image file, named **sprite.png**. (If you made more than one sprite, you can name the others whatever you like.)
3. Your screen grab showing your code running, named **screenshot.png**.
4. Your completed **ps0-readme.txt** file.

The four files must be in a subdirectory named ps0. Then, **cd** to the directory containing the three files, and type the following:

```
$ ls
```

You should see the files:

```
main.cpp ps0-readme.txt screenshot.png sprite.png
```

Then, type the following commands to make a gzipped tar archive of the directory:

```
$ cd ..  
$ tar czvf my-submit.tar.gz ps0
```

You should see the following displayed:

```
ps0/  
ps0/sprite.png  
ps0/screenshot.png  
ps0/ps0-readme.txt  
ps0/main.cpp
```

Then, submit the file my-submit.tar.gz on Blackboard. (Note: you don't have to call it "my-submit.tar.gz". It would be better to name it with your real name and PS0, e.g., *Tom_Wilkes_ps0.tar.gz*. But it must be a gzipped tar of the correct directory.)

Grading rubric

Feature	Value	Comment
core implementation	6	full & correct implementation
		1 pt file name correct 1 pt for SFML window being displayed 1 pt for image 1 pt for image being in moving sprite 1 pt for sprite changing direction or somehow responding to keystrokes 1 pt for additional feature (explain in README)
screenshot	2	screenshot of just SFML window included
		-1 pt if it's a screen-grab of your entire monitor
tar.gz archive	2	all files packaged in .tar.gz file with correct directory structure
readme.txt	2	complete
Total	12	