PS0: Hello World with SFML

Due: Wednesday, May 18, 11:59pm

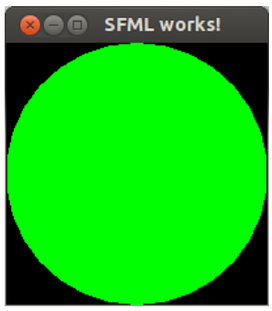
The main purpose of this assignment is for you to get up and running with your build environment. You'll write a bit of code.

1 Get your build environment set up

Use the setup document to get Linux setup and all of the packages that you will need this semester installed. You must use **GCC** and **Make** to configure and build your code. The reference compiler for the course is `g++` 9.3.0. It has support for C++ 17. For Project 0, a makefile has been provided. For future assignments, you will need to provide your own makefile.

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:



Go to the tutorial,`https://www.sfml-dev.org/tutorials/2.4/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. Draw a sprite (the technical term for a movable image that makes up a larger image).
2. Make the sprite move.
3. Make the sprite respond to keystrokes.
4. Add another feature of your choice.

Use the documentation at `https://www.sfml-dev.org/documentation/2.4.2/`.

You probably want to make your window bigger than the 200×200 specified in the Hello World code. You don't need to do audio now (we will use audio in a later project).

For #2 and #3 above, you can make x and y variables that represent the position of the sprite and update them when a keystroke is detected. You are 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/2.4/window-window.php`).

3.1 Notes

- It's correct to clear everything in the window and redraw stuff each time through the even 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* an absolute path (e.g. "`/home/dalyj/ps0/sprite.png`").
- If you draw stuff outside the bounds of the window, you won't see it. :) $(0, 0)$ is in the upper-left corner and positive x and y coordinates move right and down.

3.2 Compiling program

- When you compile your code, add the flags "`-Wall -Werror -pedantic`". We will be using these to test your code and your code must compile when these flags are used.
- To use newer C++ features use the flag `-std=c++17`.
- The provided makefile has these flags already set. You can run the makefile by typing `make`.

Make sure yiour 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, I highly recommend Shutter for doing screen grabs or a selected area of the screen. Install with:

```
sudo apt-get --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 afterwards to just show your SFML window.

4 Additional course participation

Please do the following:

- Join the course discussion on Discord (`https://discord.gg/7Kr9ZftK6w`). Be sure to give yourself the Computing IV role so you can see our channel.
- Review the Academic Integrity discussion at the bottom of the syllabus. Follow the link to the Univeristy Policy on Academic Integrity and answer the following question: There are six examples of academic misconduct, labeled (a) through (f). Other than (a), "Seeks to claim the work or efforts of another without authorization or citation," which of these do you think would most apply to this class, and why? Write approximately 100 words. Note: there is no single correct answer to this. I am looking for your opinion.

5 Assignment readme file

Before submitting, fill out info in the `Readme-ps0.md` file. Make sure to re-save it as a markdown file. It must be named `Readme-ps0.md`.

6 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 `Readme-ps0.md` file.
5. The provided makefile.

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

```
$ ls
```

You should see the five files:

```
Makefile    main.cpp    Readme-ps0.md   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/Readme-ps0.md
ps0/main.cpp
ps0/Makefile
```

You **must not** submit the intermediate `.o` file or compiled `ps0` program. Doing so is unnecessary (since we can build them from the other files) and increases the submission size by a lot. You can use the command `make clean` to remove these files before creating your archive.

Then submit the file `my-submit.tar.gz` to the autograder. (Note: you don't have to call it `my-submit.tar.gz`. If would be better to name it with your real name and PS0, e.g. `dalyj-ps0.tar.gz`, but it must be a gzipped tarball of the correct directory.)

Submit your tarball to Blackboard.

7 Grading rubric

Feature	Points	Comment
Core Implementation	7	Full & Correct Implementation
	1	File names correct
	1	SFML Window is displayed
	1	Circle shape is displayed
	1	Image is displayed
	1	Image can move
	1	Program reacts to keystrokes
	1	Implements an additional feature and describes in Readme
Screenshot	2	Screenshot of just SFML window
	(-1)	Screengrab of the entire monitor
Tarball	2	All files in `.tar.gz` archive
	1	All files included
	1	In the `ps0` directory
	(-1)	`.o` files are submitted
Readme	2	Complete
	1	Complete project description
	1	Thoughtful answer to the integrity reading.
Total	13	