



B1 - Unix & C Lab Seminar

B-CPE-100

Day 13

CSFML Discovery





Day 13

repository name: CPool_Day13_\$ACADEMICYEAR
repository rights: ramassage-tek
language: C



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).

Today you will discover graphical programming in C, and learn about CSFML

TASK 00: INSTALLING THE CSFML

First, you need to install the CSFML on your computer. To do so, please download and execute (as root) the `build_csFML.sh` script found alongside this subject.



This script installs all the required libraries in your work environment so that you do not have to worry too much about it.

TASK 01: OPENING A WINDOW

The first step for displaying an image in a window is the opening of said window.



Have a look at `sfRenderWindow` and its associated functions.

Open a 800x600 window, and keep it open!



TASK 02: DISPLAY A PIXEL

+ CREATING A PIXELS ARRAY

To be able to modify the pixels of your window you must first represent the pixels in the form of a pixels array.

Initialize a pixels array the size of your window.

+ CREATING A FRAMEBUFFER

It seems relevant to bring together the various components of our pixels array in one type.

Create a structure called **framebuffer** that contains a pixels array, the width and height of the array.

We will use a typedef for this structure named *framebuffer_t*.

Now create a **framebuffer_create** function that initializes your array of pixels and returns it. It must match the following prototype:

```
framebuffer_t *framebuffer_create(unsigned int width, unsigned int height);
```

+ DRAW PIXELS

Now that your framebuffer is created, change the color of some pixels and load them into a texture, then a sprite, to finally display them in your window.

Create a **put_pixel** function with the following prototype:

```
void put_pixel(t_framebuffer *framebuffer, unsigned int x, unsigned int y, sfColor color);
```

Display red pixels at positions (10;10), (100;100) and (250;400).

TASK 03: DISPLAY A SQUARE

Display a blue colored square of 10 pixels by 10 pixels at position (100;100), using a **draw_square** function respecting the following prototype:

```
void draw_square(t_framebuffer *framebuffer, sfVector2u position, unsigned int size, sfColor color);
```



Check **sfVector2u** in the documentation.

TASK 04: DISPLAY AN IMAGE

Now that you can modify the pixels by yourself, you need to display an already existing image from a file in your window to master the basics of CSFML!



The BMP file format is quite simple to use.