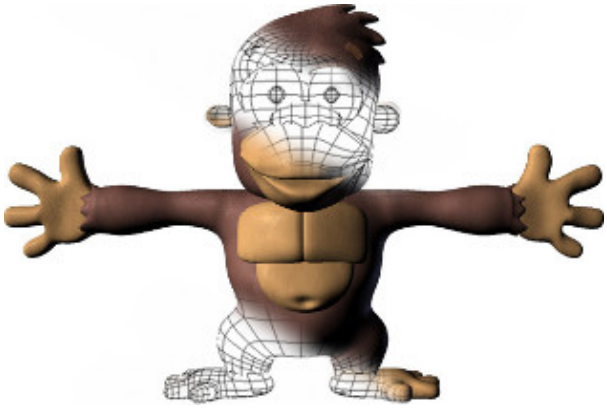
B1 - C Graphical Programming

B-MUL-100

CSFML Initiation

C Graphical Programming Bootstrap



1.1

OPENING A WINDOW

The goal of this Bootstrap is to display your first images in a window.
The first step, obviously, is to open this very window.
In order to do this, look at **sfRenderWindow** and associated functions, like `sfRenderWindow_create`.

```
CSFML_GRAPHICS_API sfRenderWindow* sfRenderWindow_create (sfVideoMode mode,
                                                          const char * title,
                                                          sfUint32 style,
                                                          const sfContextSettings * settings
                                                         )
```

Construct a new render window.

Parameters

mode	Video mode to use
title	Title of the window
style	Window style
settings	Creation settings (pass NULL to use default values)

Once you read the documentation about these functions, open an 800x600 window.

The point of this exercise is not just to open a window, but also to keep it open!

DISPLAYING PIXELS

+ CREATE A PIXEL ARRAY

To modify the pixels of your window, one must represent them in memory as an array of pixels.
Initialize a pixel array the size of your window

+ CREATE A FRAMEBUFFER

It seems relevant to bring together the various components of our array of pixels in one type.
Create a structure, called *framebuffer*, that contains a pixel array, the width and height of the array.
Turn it into a new *framebuffer_t* type.

```
struct framebuffer {
        unsigned int width;
        unsigned int height;
        sfUint8 *pixels;
};

typedef struct framebuffer framebuffer_t;
```

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

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

+ DRAW PIXELS

Now, change the color of some pixels from your pixel array and load them into a texture, then a sprite, to display them in your window.
Your function must have the following prototype:

```
void my_put_pixel(framebuffer_t *buffer, unsigned int x, unsigned int y, sfColor color);
```

Use this function to display red pixels at positions (10 ; 10), (100 ; 100), and (250 ; 400).

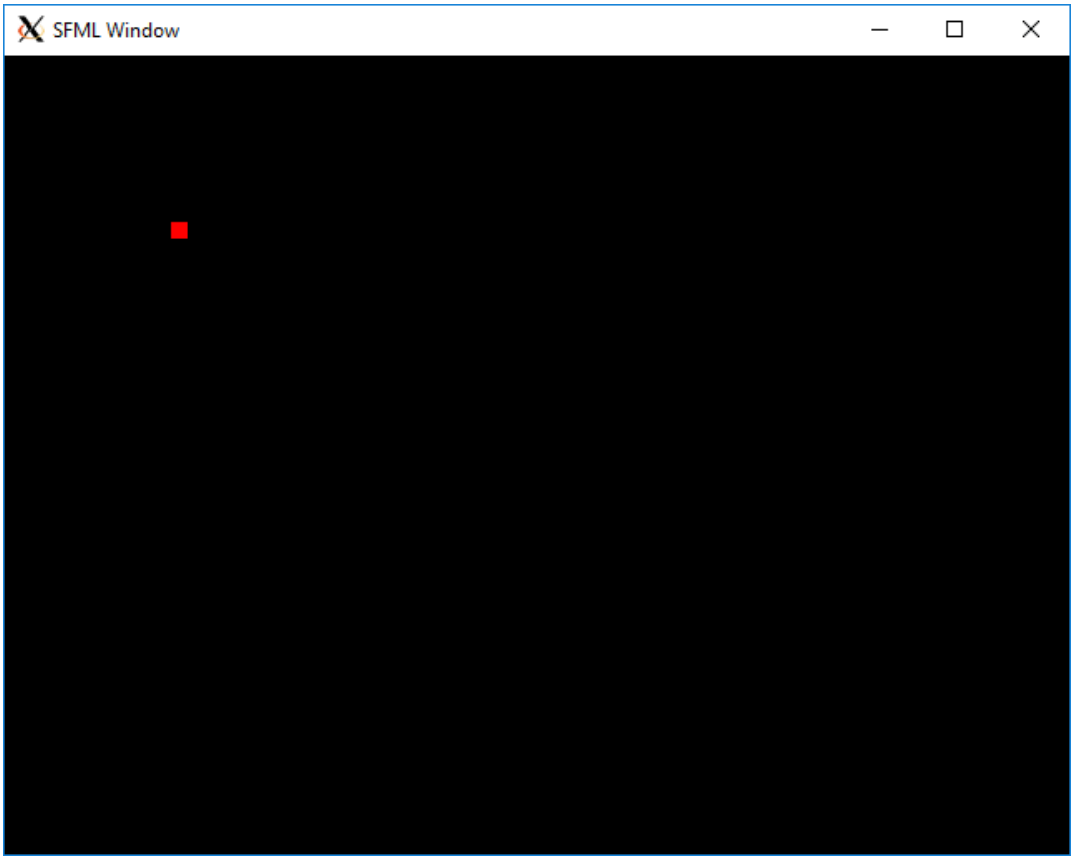
sfColor...

Drawing a square

Let's display a blue square of 10 pixels by 10 pixels at the position (100 ; 100).
Create a *my_draw_square* function with the following prototype:

```
void my_draw_square(framebuffer_t *buffer, sfVector2u position,
                                                unsigned int size, sfColor color);
```

Here is the result you should get:



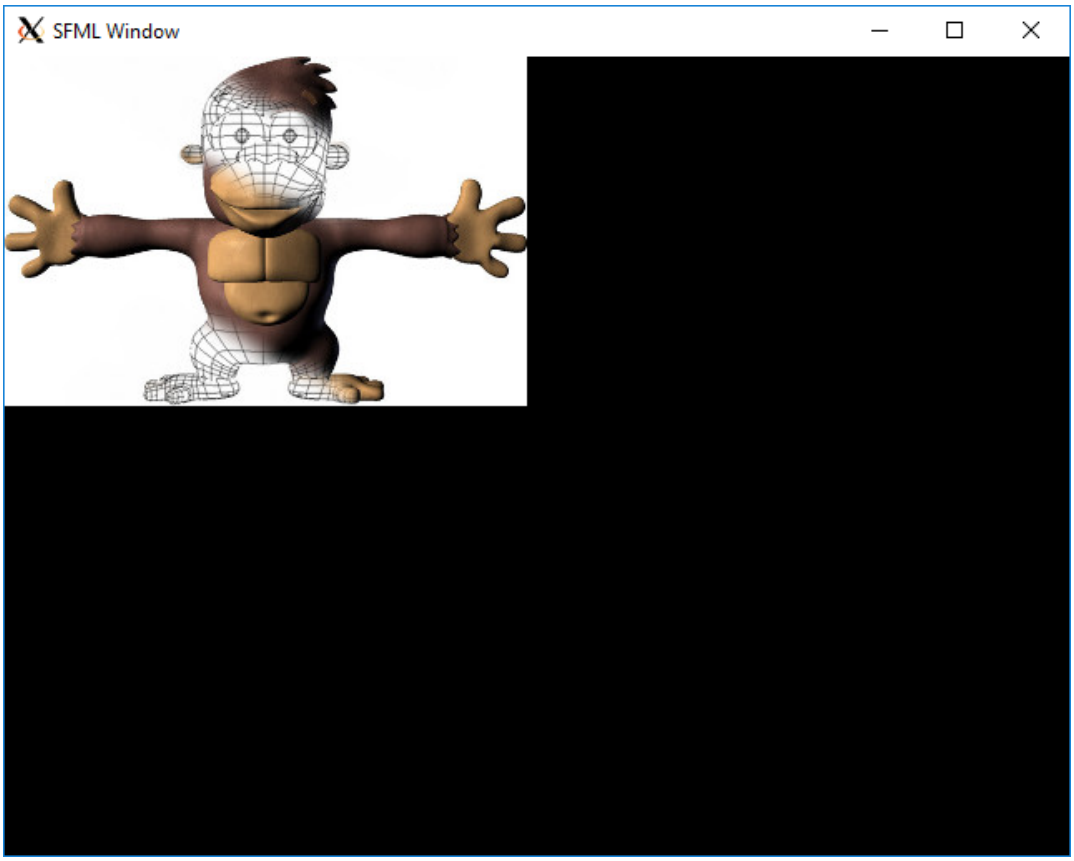
According to you, where should look for some information about sfVector2u??

DISPLAYING AN IMAGE

To complete this initiation, you need to display an existing image from a file in your window.

You should have already found that you need to refer to the *sfTexture_createFromFile* function.

Here is an example of the result of an image loaded in this way:



GOING FURTHER

If you are done with the previous exercises, take some time to check CSFML functions.
Test them and implement some nice features:

- drawing more shapes (circles,…),
- displaying several images,
- moving shapes or images,
- adding some sound,
- building a full layer-based computer-aided architectural design software,
- …