Images

In this tutorial we will see how to upload image from file and how to create image and save it to file and much more.

Image Is a stand-alone representation of the object or scene.

sf::Image class

sf::Image is a class for loading, manipulating and saving Images. The class provides functions to load, read, write and save pixels as well as many other useful functions. sf::Image is an abstraction to manipulate Images as a 2D arrays of pixels.

Load Image from a file

- To load the image from file we use Image::loadFromFile( ) function.
- The most common way of loading an Image from a file. The following code exits the main( ) function if the image is not loaded correctly. This is a safe way to prevent any future bugs from occurring in the code.

```
sf::Image image;
//
    if (!image.loadFromFile("axe.png"))
        return -1;
// Note- axe.png is in my current directory, if your image is in the other directory
// then copy the whole path of your image and paste in the loadfromfile() function.
```

- Loading image from a file is an effective way than creating images if you want to use an image which is already available on the machine. SFML supports the following file formats: **bmp, png, tga, gif, psd, hdr, pic, and jpg** (progressive JPEG is not supported).
- If we try to load an image with a different file format, or the given file does not exist, Image::loadFromFile() returns false and prints a “Unable to open file” message in the console.

Create Image

We can create Image In two ways :

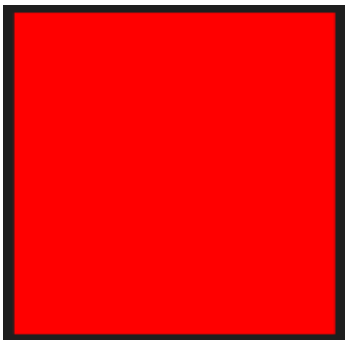
- By Create function
- By Array of Pixels

Create( ) function

```
sf::Image image;
image.create(50, 50, sf::Color::Red);
```

- Create 50x50 sized Image with color- red.
- The first two arrguments represents the width and height of the Image.
- The third arrgument represents the color of the image, By default the color is set to black.

  The preceeding code will crosspond to these :



Array of pixels

Creating a image by array of pixels means that we are creating an 1D array of type Uint8, The Uint8 type used to represent the RGBA format. where one pixel contains only one color. The sf::Color class has four data members where each member consumes 1-byte of memory space to represent the single component of a color and group of 4-byte represented as a whole color and this group of 4-bytes is the final color of the Image. The four data members of Uint8 are- **Uint8 r(red)**, **Uint8g(green)**, **Uint8 b(blue)**, **Uint8a (alpha),** alpha represents the *opacity and *transparency of image, by default the value of alpha is 255(100% opacity), if alpha value is 0 it means transparency is 100%.

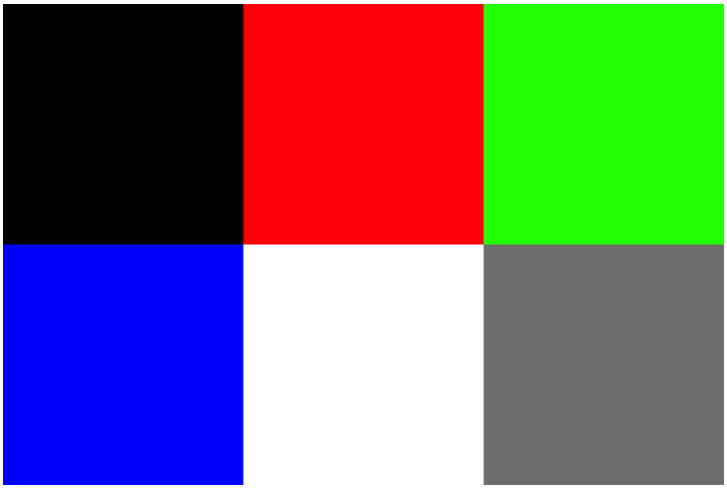
```
const unsigned int width = 3;
const unsigned int height = 2;
```

```
// Total pixel (imagine as area covered by the image) of the image will be 3x2 = 6 pixels
// Now to color this pixel we have to multiply extra 4 bytes

sf::Uint8 pixels[width * height * 4] = {   // total 24 bytes and each byte represents the single
                                          //  color of a pixel
//  red, green, blue, alpha
    0,   0,   0,   255, // black
    255, 0,   0,   255, // red
    0,   255, 0,   255, // green
    0,   0,   255, 255, // blue
    255, 255, 255, 255, // white
    128, 128, 128, 255, // gray
};

image.create(width, height, pixels);
```

The preceeding code will crosspond to these :



Save Image

saveToFile( )

save the image to a file on disk the format of the image is automatically deduced from the extension. The supported image formats are bmp, png, tga and jpg. The destination file is overwritten if it already exists. This function fails if the image is empty.

To save Image into a current directory

```
if (!image.saveToFile("result.png"));
        return -1;
```

To save Image into a another directory

```
if (!image.saveToFile("..\\specify_the_directory\\result.png"));
        return -1;
```

To know more about functions related to image then follow the below link :)

sf::Image Class Reference (SFML / Learn / 2.5.1 Documentation)

Class for loading, manipulating and saving images. sf::Image is an abstraction to manipulate images as bidimensional arrays of pixels. The class provides functions to load, read, write and save pixels, as well as

https://www.sfml-dev.org/documentation/2.5.1/classsf_1_1Image.php