

# Pixels et RVB (Codage des couleurs)

## I. Généralités

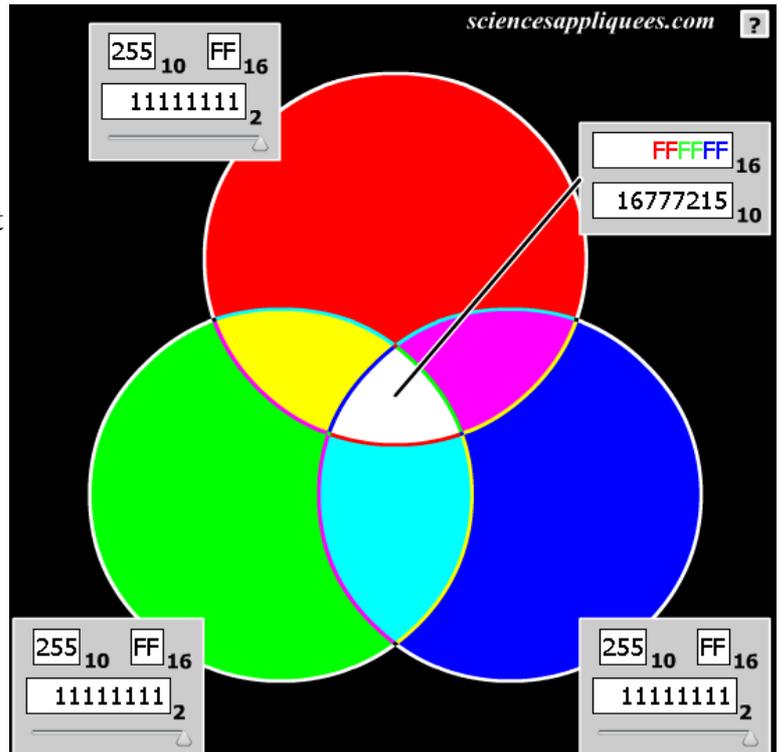
### I.1. RVB

Rouge vert bleu, abrégé RVB (ou RGB de l'anglais red green blue), est un format de codage des couleurs.

Ces trois couleurs sont les couleurs primaires en synthèse additive. Elles correspondent en fait à peu près aux trois longueurs d'ondes auxquelles répondent les trois types de cônes de l'œil humain. L'addition des trois donne du blanc pour l'œil humain.

Elles sont utilisées pour \_\_\_\_\_ afin d'obtenir toutes les couleurs visibles par l'homme.

Les couleurs en informatique sont codées sur \_\_\_\_\_



### I.2. Pixels

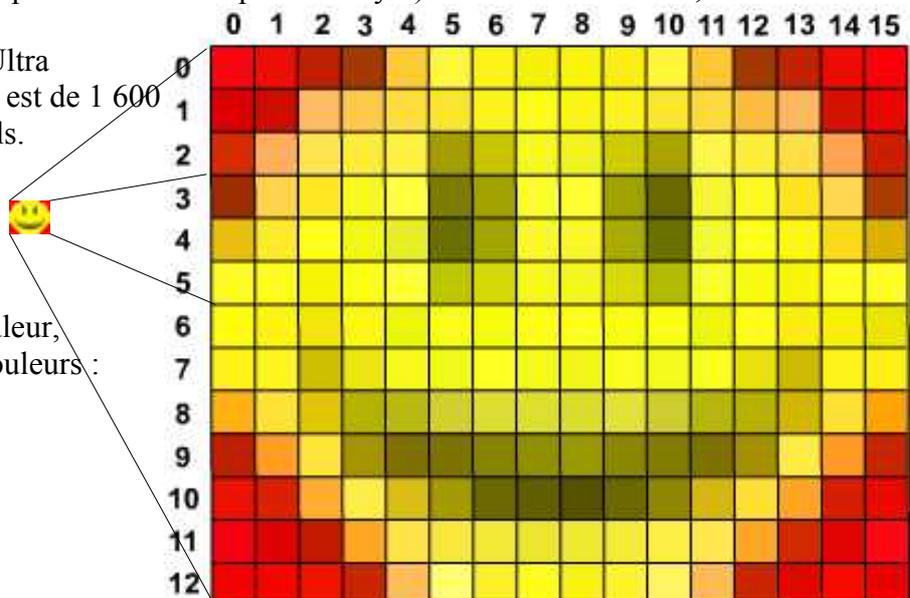
Le pixel, souvent abrégé px, est l'unité de base permettant de mesurer la définition d'une image numérique matricielle (tableau 2 dimensions). Son nom provient de la locution anglaise « picture element », qui signifie « élément d'image ».

Le pixel est l'unité minimale adressable par le contrôleur vidéo. C'est aussi l'unité utilisée pour spécifier les définitions d'affichage (largeur × hauteur) :

- la définition du VGA (« Video Graphics Array ») est de  $640 \times 480$ , soit 307 200 pixels ;
- la définition du Super-VGA, ou S-VGA (« Super Video Graphics Array ») est de  $800 \times 600$ , soit 480 000 pixels ;
- la définition du XGA (« Extended Graphics Array ») est de  $1\,024 \times 768$ , soit 786 432 pixels ;
- la définition du SXGA (« Super Extended Graphics Array ») est de  $1\,280 \times 1\,024$ , soit 1 310 720 pixels ;
- la définition du UXGA (« Ultra Extended Graphics Array ») est de  $1\,600 \times 1\,200$ , soit 1 920 000 pixels.

À chaque pixel est associée une couleur, usuellement décomposée en trois couleurs :

\_\_\_\_\_.



## II.Application

Nous allons charger une image et la modifier en AS3. Créer un nouveau projet et copier le programme ci-dessous. N'oubliez pas de tester au fur et à mesure de l'évolution de votre code.

```
public class Main extends Sprite
{
    //charge l'image directement dans le fichier swf (flash)
    [Embed (source = "stop.jpg")]
    private var image:Class;
    private var imageBitmap: Bitmap = new image();

    public function Main():void
    {
        if (stage) init();
        else addEventListener(Event.ADDED_TO_STAGE, init);
    }

    private function init(e:Event = null):void
    {
        removeEventListener(Event.ADDED_TO_STAGE, init);

        // entry point

        addChild(imageBitmap);
        //tester

        //Recupere la couleur en uint (entier long);
        var couleur1 : uint = imageBitmap.bitmapData.getPixel(10, 10);
        //Affiche la couleur en uint
        trace(couleur1);
        //tester

        //Affiche la couleur en hexa
        trace(couleur1.toString(16));
        //tester

        //affiche une ligne horizontale rouge en y=40;
        for (var i:int = 0; i < imageBitmap.width; i++)
        {
            imageBitmap.bitmapData.setPixel(i, 40, 0xFF0000);
        }
        //tester
    }
}
```



stop.jpg

## III.Travail à faire

- Récupérer la couleur du pixel x=50 et y= 70 ; \_\_\_\_\_
- Dessiner une ligne verticale verte en x = 80 ;
- Mettre les pixels pour x < 100 en violet ;
- Mettre les pixels pour y > 200 en jaune ;
- Remplacer tous les pixels rouges (supérieur à 0xDD0000) de l'image par la couleurs 0x00AA99.