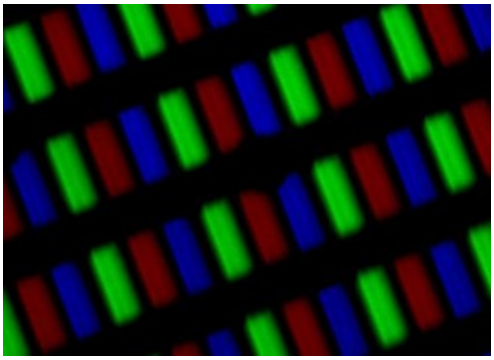
Activité - La photographie numérique

Première partie : Analyse d'un écran

I. Les pixels



À l'aide d'une loupe binoculaire on a observé la surface d'un écran d'un iPhone5S. Voici ci-contre le résultat obtenu.
(source : B. Castagnetto, académie de Strasbourg)

Le plus petit élément physique de l'écran s'appelle un pixel (contraction de picture element). On désigne généralement un pixel comme la réunion de trois sous-pixels de couleur rouge, vert, bleu.

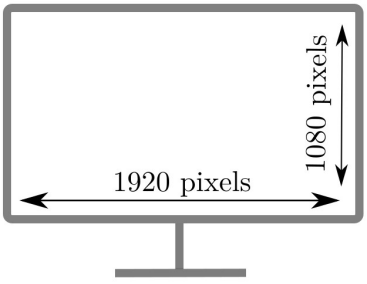
1) Grâce au site http://www.ostralo.net/3_animations/swf/couleurs_ecran.swf, observez comment le réglage de l'intensité des 3 sous-pixels rouge, vert et bleu peut générer une multitude de couleurs : vous pouvez bouger les curseurs de couleurs et zoomer sur les pixels qui s'allument.
La perception «uniforme» pour l'œil, qui ne distingue pas (sauf à la loupe) le détail des sous-pixels mais simplement une couleur globale, est due au pouvoir de résolution limité de l'œil humain.

2) Combien de couleurs différentes peuvent-elles être générées par ces 3 pixels, sachant que chaque sous-pixel (rouge, vert ou bleu) possède 256 nuances possibles ?
……
……
……

I. Définition et résolution

Deux notions (très souvent confondues) sont à connaître :

La définition d'un écran : c'est le nombre de pixels qui composent l'écran. Attention, on ne parle pas ici des sous-pixels rouge, vert, bleu, mais bien du pixel global créé par la réunion de ces trois sous-pixels. Cette définition est par exemple donnée sous la forme 1920x1080.



Cela signifie que l'écran comporte 2 073 600 pixels ($1920 \times 1080 = 2\,073\,600$) répartis uniformément sur toute sa surface.
On dit que 1920x1080 est la définition native de l'écran.

Lorsque dans les réglages de votre ordinateur on vous propose de changer la « résolution » (confusion de vocabulaire!), en la baissant par exemple à 1280x720, les 2 073 600 pixels de cet écran restent bien présents mais se regroupent pour simuler un écran comportant moins de pixels : la qualité d'affichage se trouve alors dégradée.

La résolution d'un écran : généralement exprimée en dpi (dot per inch) ou ppi (pixel per inch) : Elle mesure donc le nombre de pixels disponibles sur une longueur de 1 pouce, soit environ 2,54 cm. Plus ce nombre est élevé, plus la taille des pixels est réduite, et plus l'image paraît donc précise.

1. Les expressions « HD », « Full HD » et « 4K » sont des normes relatives à des définitions d'écran. Recherchez les valeurs de ces définitions.

HD :……

Full HD : ……

4K :……

2. La définition d'un téléphone portable Samsung S4mini (bas de gamme) est 540x960 et les dimensions de son écran sont de 5,3cm sur 9,5cm.
Déterminer la résolution de son écran :
……
……
……
……
……
……
……
……
……
……

3. On donne ci-dessous les spécifications d'un téléviseur 4K (4K signifie que sa définition, et non sa résolution comme indiqué sur la fiche produit, est égale à 3840x2160 pixels).

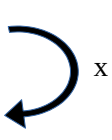
Type d'écran	OLED
Taille de l'écran	55 pouces
Diagonale de l'écran	140 cm
Format de l'écran	16/9
Ecran large	Oui
Ecran incurvé	Non
Compatible 3D	Non
Résolution	3840 x 2160 pixels
Design	Ultra slim

La taille d'écran de 55 pouces correspond à la longueur de sa diagonale en pouce.
Le format 16/9 signifie que la longueur et la largeur de l'écran sont proportionnelles à un écran de longueur 16 pouces et de largeur 9 pouces.

a) Déterminer la diagonale d'un écran de longueur 16 pouces et de largeur 9 pouces.
……
……
……
……
……
……
……
……
……

b) Compléter le tableau de proportionnalité suivant :

	Diagonale	Longueur	Largeur
Mesures (en pouce) sur un écran 16 pouces sur 9 pouces			
Mesures (en pouce) sur l'écran 4K considéré			



c) Calculez la résolution de cet écran de télévision. Comparez avec les résolutions de l'écran du téléphone Samsung S4mini.

……
……
……
……
……
……

d) Pourquoi les fabricants de téléviseurs ne font-ils pas de la résolution de leurs écrans un argument publicitaire ?

……
……
……
……
……
……

Deuxième partie : Création d'une image numérique

I. Création d'une image avec Python

1. Ouvrir EduPython et recopier dans l'éditeur le code ci-contre.

```
from PIL import Image
img = Image.new("RGB", (8,8))

for x in range(8):
    for y in range(8):
        img.putpixel((x,y),(255,255,255))

img.save("carreblanc.jpg")
```

Explications du code :

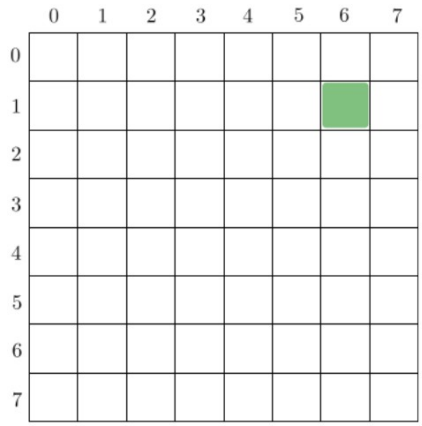
• *ligne 1 : on importe le module Image de la bibliothèque PIL, qui permet la manipulation d'images.*
• *ligne 2 : on crée une (très petite) image de 8 pixels sur 8, dont les pixels seront codés en RGB : (Red Green Blue)*
• *lignes 4 et 5 : double boucle permettant de faire prendre au couple (x,y) successivement toutes les valeurs entre (0,0) et (7,7).*
• *ligne 6 : on colorie le pixel de coordonnées (x,y) en blanc. Cette instruction étant située à l'intérieur de la double boucle, ce sont tous les pixels de l'image qui seront donc coloriés en blanc.*
• *ligne 9 : l'image est sauvegardée sous le nom « carreblanc.png».*

Première image :
2. Enregistrer ce fichier dans votre dossier *photo* en le nommant *carreblanc.py*
3. Exécuter le programme en cliquant dans EduPython sur le triangle vert.
4. Aller voir dans votre dossier *photo* : un fichier image vient d'être créé, il s'agit d'un carré blanc.

Deuxième image :
5. Modifier le programme Python pour créer un carré rouge. L'image devra se nommer *carrerouge* et le programme python *carrerouge.py*

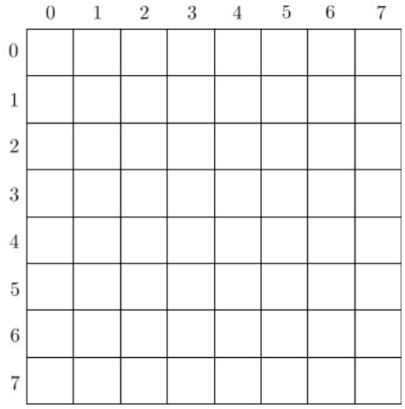
Troisième image :
L'image créée peut être vue comme un quadrillage, chaque pixel représentant une case de ce quadrillage.
L'ajout de la ligne de code *img.putpixel((6,1),(0,255,0))* juste avant la création de l'image permet par exemple de colorier en vert le pixel de coordonnées (6 ; 1) comme ci-dessous.

```
from PIL import Image
img = Image.new("RGB", (8,8))

for x in range(8):
    for y in range(8):
        img.putpixel((x,y),(255,255,255))

img.putpixel((6,1),(0,255,0))

img.save("quadrille.jpg")
```



6. Effectuer les modifications et penser à modifier le nom de l'image et à sauvegarder le fichier python sous le nom *quadrillage* avant d'exécuter le programme.

7. Colorier quelques cases (quatre ou cinq) du quadrillage ci-dessous et apporter les modifications au programme précédent puis l'exécuter.



Quatrième image :

8. Ouvrir le programme python *carreblanc.py* et l'enregistrer sous le nom *lignes*.
9. Modifier le programme pour qu'il créée une image blanche de taille 300x200 et rajouter avant la dernière ligne de code la ligne suivante :

```
for x in range(50,150):
    img.putpixel((x,20),(156,61,207))
```

10. Analyser bien cette ligne de code ci-dessus et modifier le programme pour tracer trois ou quatre lignes de couleur différentes à des emplacements différents et dont une fait toute la largeur de l'image.

II. Création d'une photographie numérique

L'étude matérielle de la composition des écrans nous a permis de créer une image en spécifiant (grâce au langage Python dans notre cas) la valeur de chacun de ses pixels.

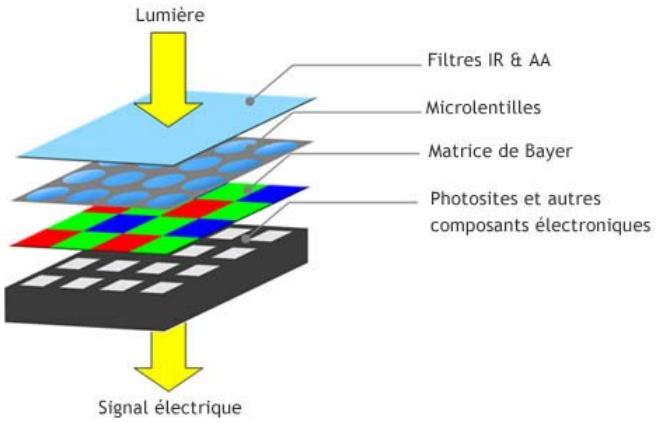
Mais l'ambition d'une photographie est autre : il s'agit de capturer puis retranscrire le plus fidèlement possible l'image formée sur notre rétine par la lumière extérieure.

À l'époque de la photographie argentique, des réactions chimiques successives permettaient de fixer sur du papier la lumière capturée par l'objectif de l'appareil photo.

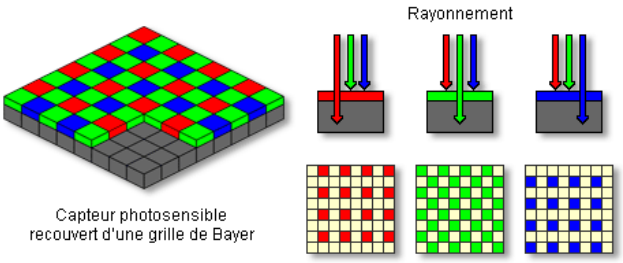
La photographie numérique consiste à convertir en signaux numériques cette lumière capturée par l'objectif.

1) Capteur et matrice de Bayer

Comme découvert précédemment, le principe physique de fonctionnement d'un écran impose qu'il reçoive une information décomposée en niveaux de rouge, de vert et de bleu. Le procédé technique fondamental de la photographie numérique est donc la décomposition de la lumière visible suivant ces trois composantes : c'est le rôle de la matrice de Bayer.



La lumière extérieure, après avoir été filtrée (filtres InfraRouge et Anti-Aliasing), va être concentrée par des microlentilles avant de passer dans la matrice de Bayer.



Le rôle de la matrice de Bayer est de séparer la lumière, en laissant passer sa composante rouge, verte, ou bleue. À noter que la composante verte est deux fois plus représentée que les autres couleurs. Cela est dû au fait que l'œil humain est naturellement plus sensible au vert qu'aux autres couleurs. Il faut donc artificiellement favoriser la lumière verte lors de sa captation.
Sous cette matrice de Bayer se situent les photosites, qui vont convertir la lumière reçue (donc rouge, verte ou bleue) en signal électrique, plus ou moins important suivant la puissance de la lumière.
Cette conversion de l'information lumineuse en information électrique est l'étape essentielle de la prise d'une photographie numérique.

Les photosites jouent un rôle dans la captation de la lumière, à l'intérieur du capteur de l'appareil photo numérique, alors que les pixels de l'écran servent à reproduire cette lumière.
Lorsque les fabricants d'appareil photo ou de smartphones communiquent sur le nombre de mégapixels, ils font référence à la définition maximale (en nombre de pixels) que pourra avoir l'image une fois affichée. Ce nombre de mégapixels n'est pas égal au nombre de photosites. En effet, des procédés algorithmiques permettent maintenant de deviner de nouveaux pixels (on parle d'interpolation) non captés par les photosites.

En vous aidant de cet article , répondez aux questions suivantes :

a) Comment a évolué le nombre de méga-pixels entre l'iPhone 6 et l'iPhone 6S ?

……
b) Quelle a été la conséquence de cette évolution ?

……
c) Dans le même temps, qu'ont décidé de faire Samsung et Google ?

……
d) Quel est l'avantage à avoir de plus grands photosites dans un capteur ?

……

e) Quel changement la matrice de Bayer subit-elle avec la technologie BritCell développée par Samsung ?

……
……

2) les algorithmes de traitement des photographies numériques

La qualité des photographies prises par les appareils photo numériques ou les smartphones augmente d'année en année.
Il devient de plus en plus facile de réaliser une photographie qui satisfait nos attentes. Si des progrès ont eu lieu dans le domaine de l'optique, c'est essentiellement aux progrès fulgurants des algorithmes de traitement d'images que l'on doit la satisfaction d'une photographie réussie.
Les algorithmes présentés ci-dessous peuvent être utilisés en post-traitement de photographie (sur un ordinateur avec un logiciel dédié), par le biais d'un filtre appliqué sur un réseau social, ou même de manière automatique lors de la prise de vue, lorsque ces algorithmes sont implémentés dans l'appareil photo numérique.

► algorithme ①

Fusion automatique de plusieurs photographies pour ne garder que des visages souriants

► algorithme ②

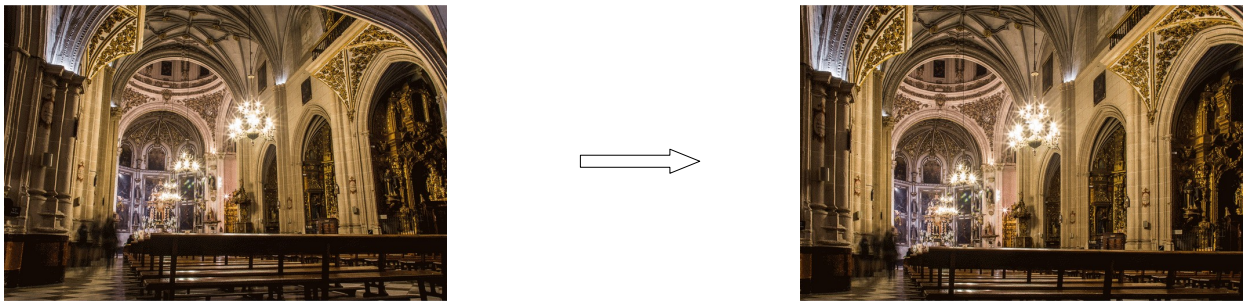
Effet Bokeh : modification artificielle de la profondeur de champ. Appelé « mode Portrait » sur iOS

► **algorithme ③**

Focus stacking : plusieurs photos de profondeurs de champs différentes sont fusionnées pour que le premier plan et l'arrière-plan soient nets en même temps

► **algorithme ④**

correction de la distorsion



Classez ces algorithmes dans le tableau suivant :

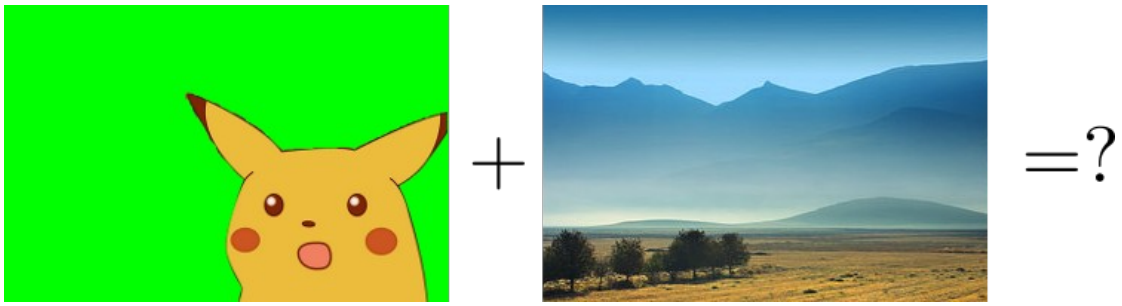
Algorithmes essayant de reproduire le plus fidèlement possible une réalité	Algorithmes essayant d'imiter un effet artistique de la photo argentique	Algorithmes produisant une photo d'une situation qui n'a jamais existé

3) Manipulation d'images avec Python

Aujourd'hui, vous êtes familiers des images de tournages vidéo réalisés devant un fond vert. En post-production, ces pixels verts sont remplacés par des pixels d'une autre vidéo, donnant une incrustation parfaite du personnage dans un nouveau décor.

L'objectif de cette activité est d'enlever le fond vert de la première image, et de le remplacer par le paysage de la deuxième image.

Les deux images ont la même taille : 300x200.



1) Vous trouverez dans le disque dur *groupe*, dossier *photo*, un dossier nommé *fusion* contenant ces deux images. Copier et coller ce dossier dans votre dossier *photo* sur votre disque dur *perso*.

Analyse de la première image :

Programme 5 :

À l'intérieur de votre dossier *fusion* contenant les fichiers surprised_pika.bmp et paysage.jpg, créez puis exécutez un fichier que vous nommerez detourage.py qui contiendra les lignes suivantes :

```
from PIL import Image

img1=Image.open("surprised_pika.bmp")

p=img1.getpixel((0,0))

print(p)
```

Explications du code :
• ligne 1 : on importe le module Image de la bibliothèque PIL, qui permet la manipulation d'images.
• ligne 2 : on ouvre l'image surprised_pika.bmp. Dans toute la suite du programme, elle sera désignée par le nom img1.
• lignes 3 : on stocke dans la variable p la valeur du premier pixel en haut à gauche (pixel de coordonnées (0 ;0)).
• ligne 4 : on affiche la valeur de p.

2) Observer le résultat donné par Python. Vous paraît-il correct ?…….

Le but maintenant est de parcourir tous les pixels de l'image pour tester s'ils sont verts ou non, c'est-à-dire que le programme devra parcourir tous les pixels de gauche à droite (donc pour x allant de 0 à 299) et tous les pixels de haut en bas (donc pour y allant de 0 à 199).
Nous allons créer une variable a qui vaudra 0 dans un premier temps et qui servira à compter les pixels verts en ajoutant 1 à a si le pixel rencontré est vert.

Algorithme (en langage naturel)	Programme (en langage Python)
. Importation du module Image de la bibliothèque PIL . Ouverture de l'image *surprised_pika.bmp* nommée dans la suite du programme *img1* . Création d'une variable a valant 0 . Pour x allant de 0 à 299 : . Pour y allant de 0 à 199 : . Stockage de la valeur du pixel de coordonnées (x ; y) dans une variable *p* . Si ce pixel est vert alors on stocke dans la variable a la somme de 1 et de a . . On affiche la valeur finale de a	

3) Compléter la partie *Programme en langage Python* du tableau ci-dessus. N'hésitez-pas à vous aider des programmes précédents déjà rédigés lors de ce thème. **(Pour tester si le pixel est vert, utiliser un double == dans la ligne de la boucle *if*)**

Programme 6 :

4) Ouvrir un nouveau module dans Pyhton et écrire votre programme puis le tester. L'enregistrer sous le nom *nbr_pixelsverts.py* Vous devriez trouver 44 749 pixels verts.

Programme 7 :

L'expression ***img1.putpixel((32,156),(255,0,0))*** permet de colorier le pixel de coordonnées (32,156) en rouge.

5) Enregistrer votre programme précédent sous le nom *pikarouge.py*.
6) Supprimer toutes les occurrences à la variable *a* et modifier votre programme pour qu'il colore en rouge tous les pixels verts et qu'il enregistre l'image sous le nom *pikarouge.bmp*
7) Vérifier qu'il y a bien une image nommée *pikarouge.bmp* dans votre dossier *fusion* et qu'elle correspond bien à ce que l'on souhaitait.

Fusion des deux images :

Programme 8 :
8) Enregistrer votre programme précédent sous le nom *fusion.py*
9) Modifier votre programme pour qu'il remplace chaque pixel vert de l'image *surprised_pika.bmp* par le pixel situé au même endroit de l'image *paysage.jpg*. **Pour cela vous penserez à :**

- Ajouter une ligne de code permettant d'ouvrir dans Python l'image *paysage.jpg* que vous nommerez *img2*.
- Créer une variable *q* dans la boucle *if* qui permettra de stocker la valeur du pixel de coordonnée (*x* ; *y*) de l'image *paysage.jpg*
- Remplacer dans l'image 1 la valeur du pixel vert de coordonnée (*x* ; *y*) par le pixel de valeur *q*
- Enregistrer la nouvelle image sous le nom *pika_paysage.bmp*

10) Vérifier qu'il y a bien une nouvelle image nommée *pika_paysage.bmp* dans votre dossier *fusion* qui correspond bien à l'image souhaitée.

Pour aller plus… (si le professeur l'autorise. Sinon passer à la partie suivante)

Vous remarquez qu'il reste du vert autour de Pikachu, cela est du au fait que ces pixels n'ont pas tout à fait la valeur (0,255,0) mais une valeur très proche comme par exemple (10,250,6), pixel qui paraît vert mais qui n'est pas remplacé.

11) Remplacer la condition *p==(0,255,0)* par un encadrement de *p* admettant une marge d'erreur de 25 pixels pour chacune des trois couleurs.

Troisième partie : Les métadonnées d'une photographie numérique

I. Découverte des fichiers EXIF

Nous avons vu, notamment dans le thème *Web* et *Données* que des métadonnées peuvent se cacher dans une page web ou un document texte par exemple et contenant des données comme l'auteur ou la date de création de la page web ou du fichier texte.
Les photographies numériques peuvent également contenir de telles données enregistrées avec la photo sous le format EXIF pour *EXchangeable Image file Format.*

1. Copier et coller dans votre disque dur perso dans le dossier *photo* le dossier nommé *exif* se situant dans le disque dur *groupes*, dossier *photo*.
1. Grâce à l'outil en ligne http://exif.regex.info/exif.cgi, explorer les métadonnées de l'image ci-dessous. Vous la trouverez sous le nom *ville.jpg* dans votre disque dur dans le dossier que vous venez de récupérer.

Parcourir toutes les métadonnées de cette photo :
a) Avec quel appareil cette photo a été prise ?…….

b) A quelle date ?…….

c) Quelle est sa résolution ?…….

2. Faîtes de même avec l'image ci-dessous se trouvant au même emplacement que l'image précédente sur l'ordinateur :

Grâce aux métadonnées, utiliser Google map puis Google street view pour localiser et confirmer le lieu où a été prise cette photo :

……..

II. La question de la confidentialité et de la fiabilité des données

1. En faisant une recherche sur internet, supprimer les données EXIF de la photo *ville.jpg* se trouvant sur votre disque dur perso.
2. Est-il possible de modifier ces données EXIF ? Comment ?……..
3. Est-il possible de modifier les paramètres de son téléphone pour qu'il n'enregistre pas ces données ?
…….