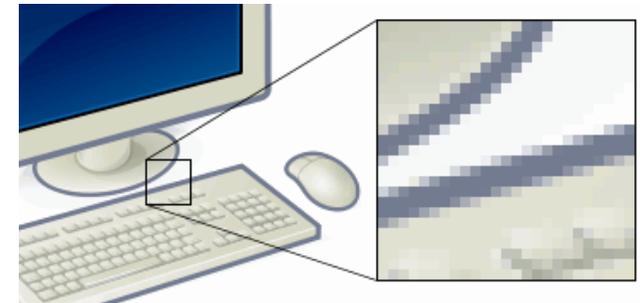
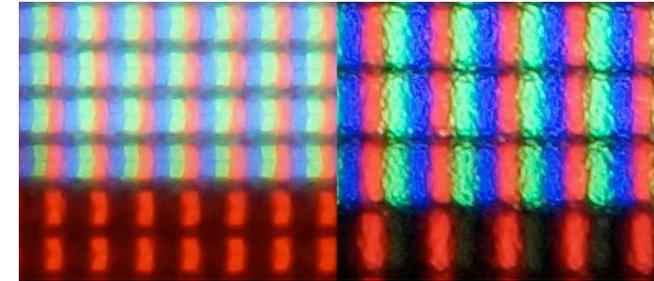


- 3 lumières (rouge, verte et bleue) couvrant un carré
 - couvrant un carré (ou presque)
 - en Matrice les uns a coté des autres
 - intensité des 3 lumières contrôlable
- Suffisamment petites
 - pour ne pas voir leur forme carrée
 - pour ne pas distinguer les lumières
- Attention
 - les lumières ne sont pas de l'encre
 - abrégé p ou px

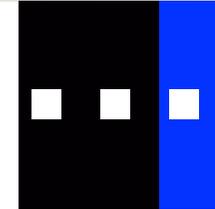


Matrice de pixels

R G B

- noir+noir+bleu = bleu foncée?

- Mélange des couleurs
- Aussi vrai pour pixels adjacents



- Rouge+Vert+Bleu = Blanc !

- Technologies

- LCD=LEDs+masque
- CRT= faisceau d'électron

- Conséquences

- LCD=meilleur contraste local, noir moins noir
- CRT= meilleur noir, plus lumineux... mais moins bon contraste

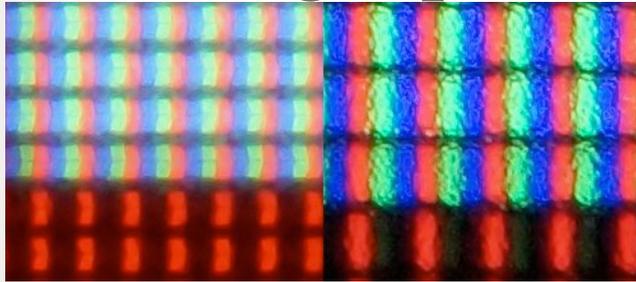
- Contraste (papier + encre noir=10000:1)

- Ex 400:1 différence de luminosité entre plus sombre et plus clair
- Si blanc 2x plus blanc = 800:1 mais est-ce les blancs le pb ?

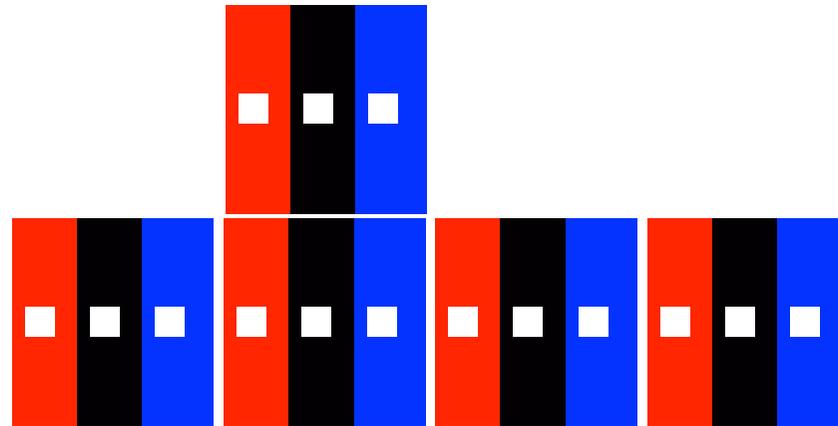


Couleur des pixels

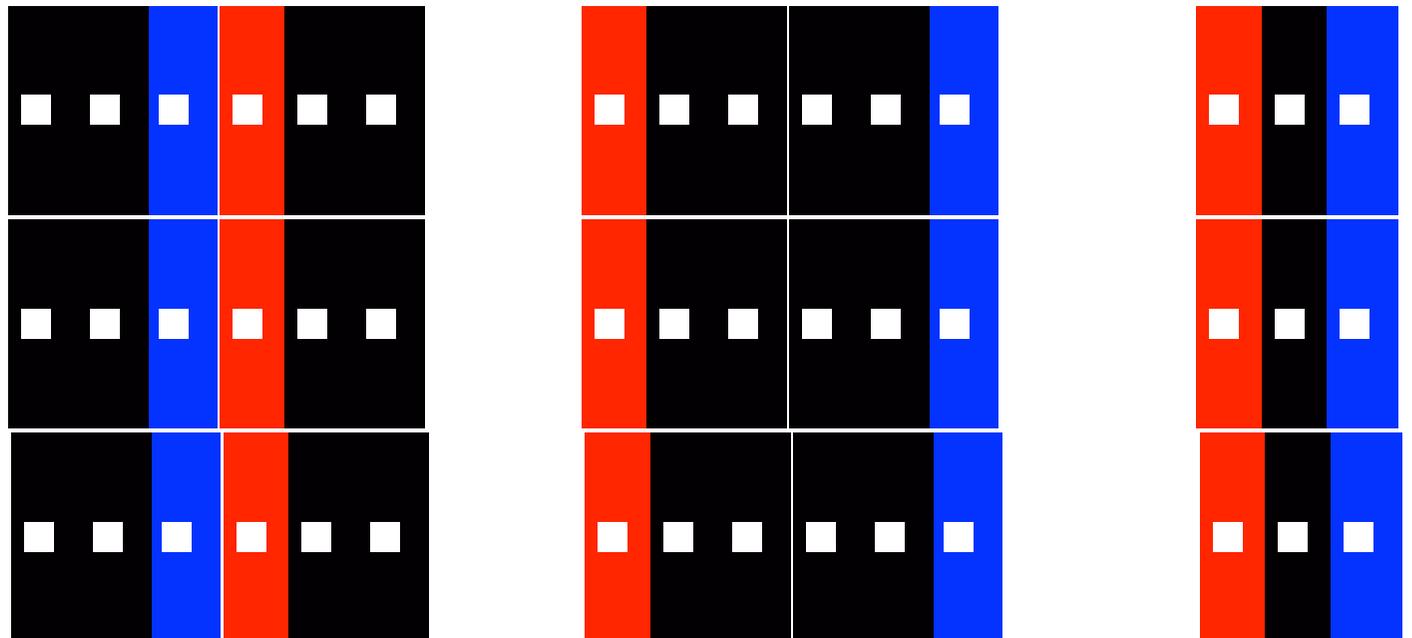
- Pixel rouge, pixel blanc, pixel magenta



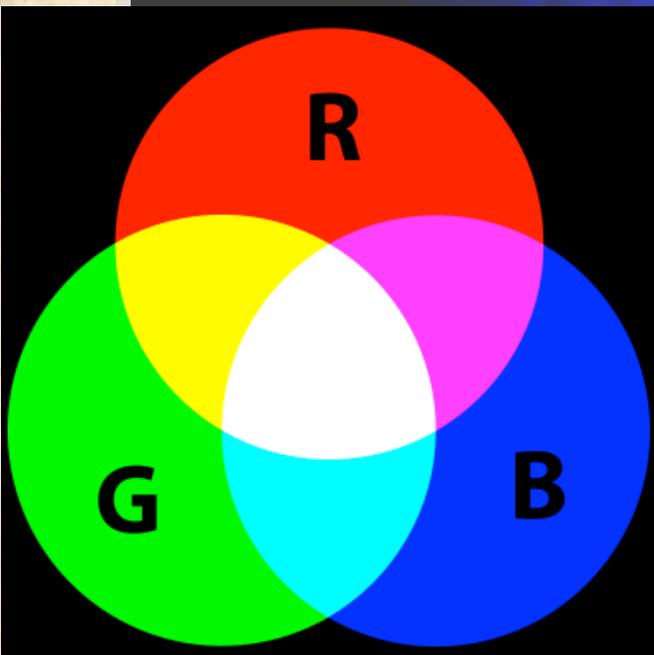
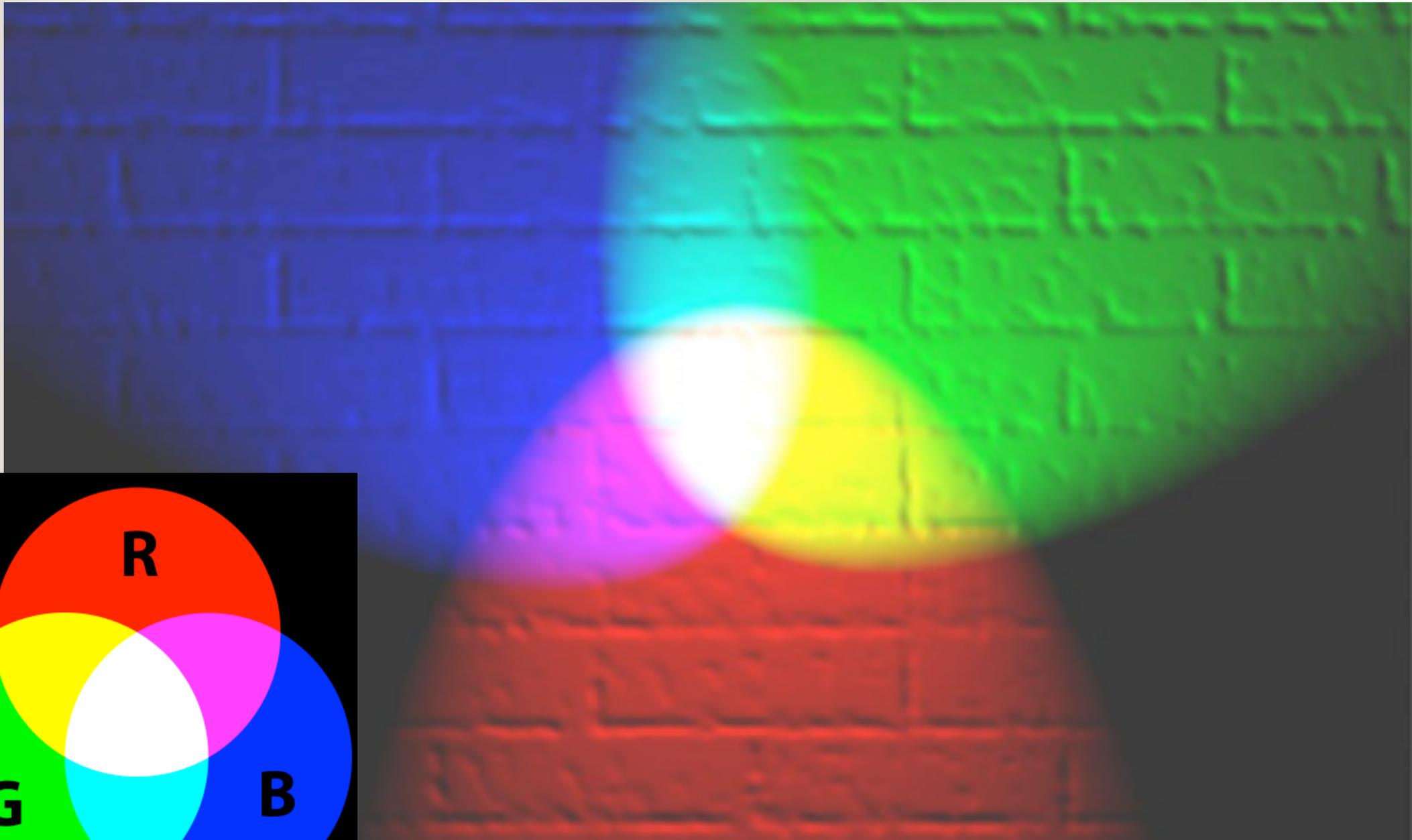
- Horizontal magenta :



- Verticales: bleu&rouge ou rouge&bleu ou magenta



Couleurs additives RGB

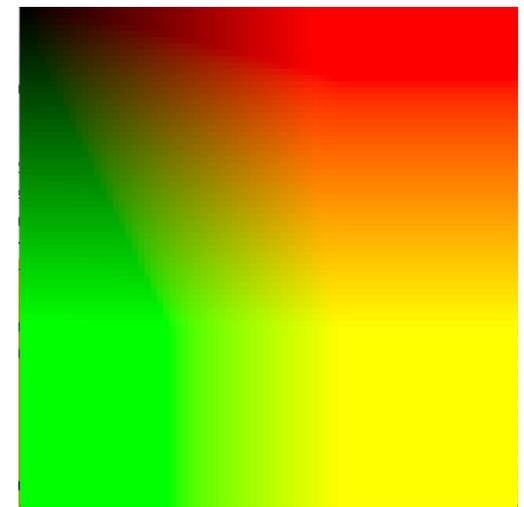
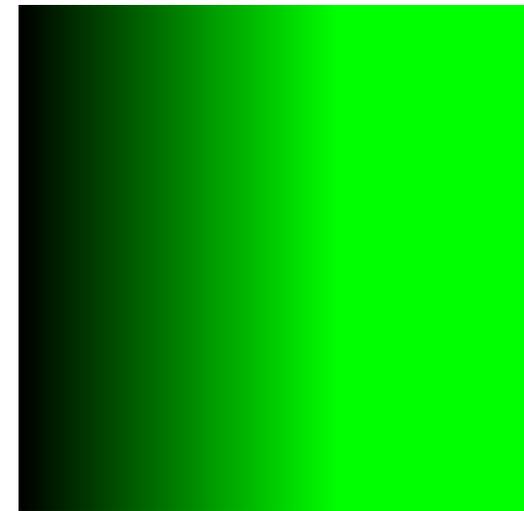


■ Pixel par pixel

```
void draw() {  
  background(0, 0, 128);  
  loadPixels();  
  for (int j = 0; j < height; j++) {  
    for (int i = 0; i < width; i++) {  
      pixels[j*width+i] = color(0, i, 0);  
    }  
  }  
  updatePixels();  
}
```

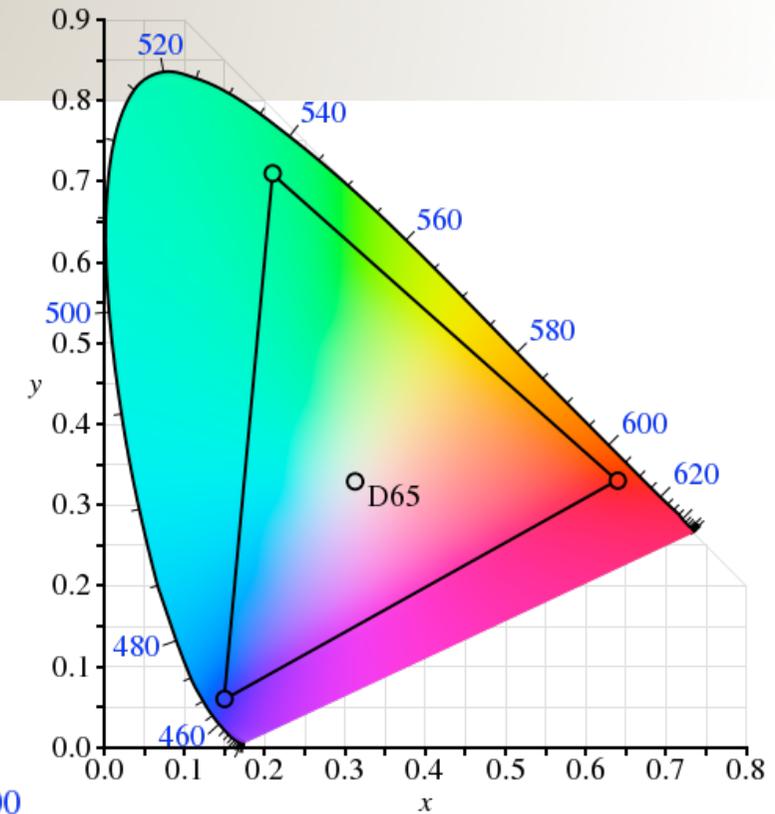
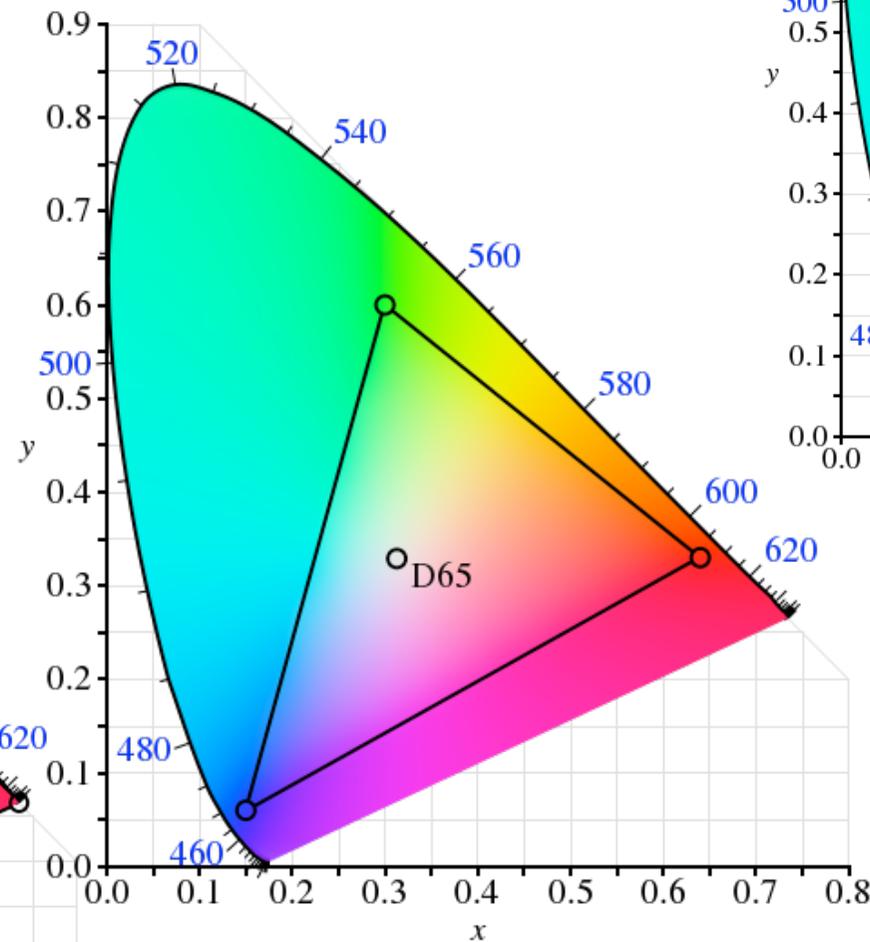
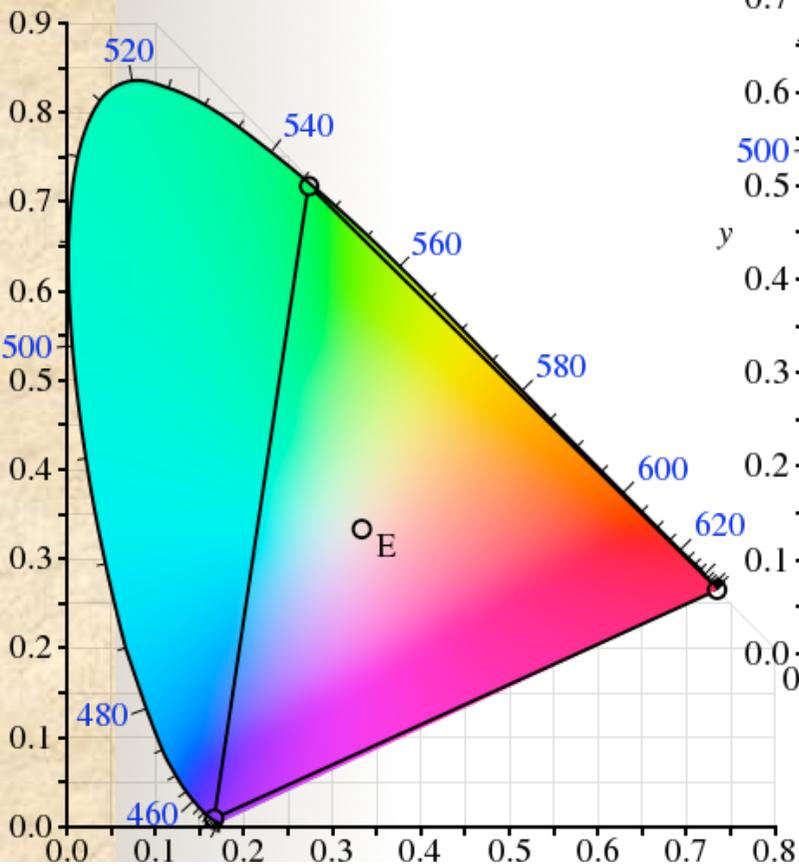
```
void setup() {  
  size(400, 400);  
}
```

- Vous ne comprenez rien à ce code ?
 - C'est normal !!!
- Essayons-le ensemble
 - posons nous les bonnes questions
- Tout ne sera pas révélé aujourd'hui !



Modèle RGB

Tout n'est pas modélisable !



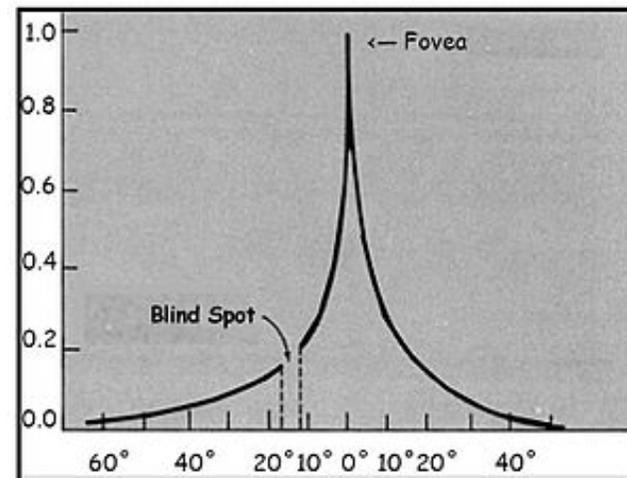
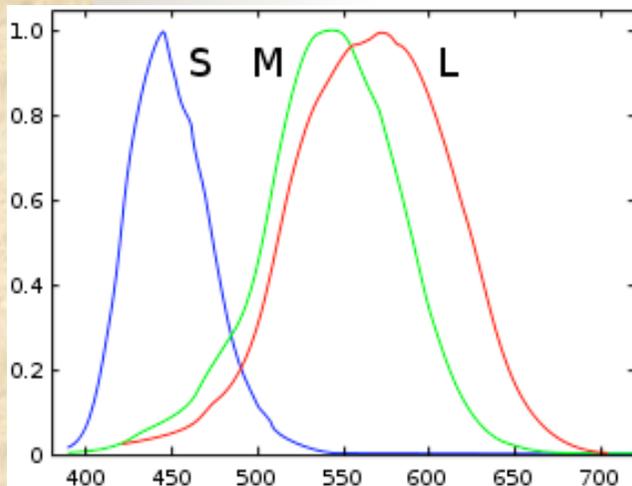
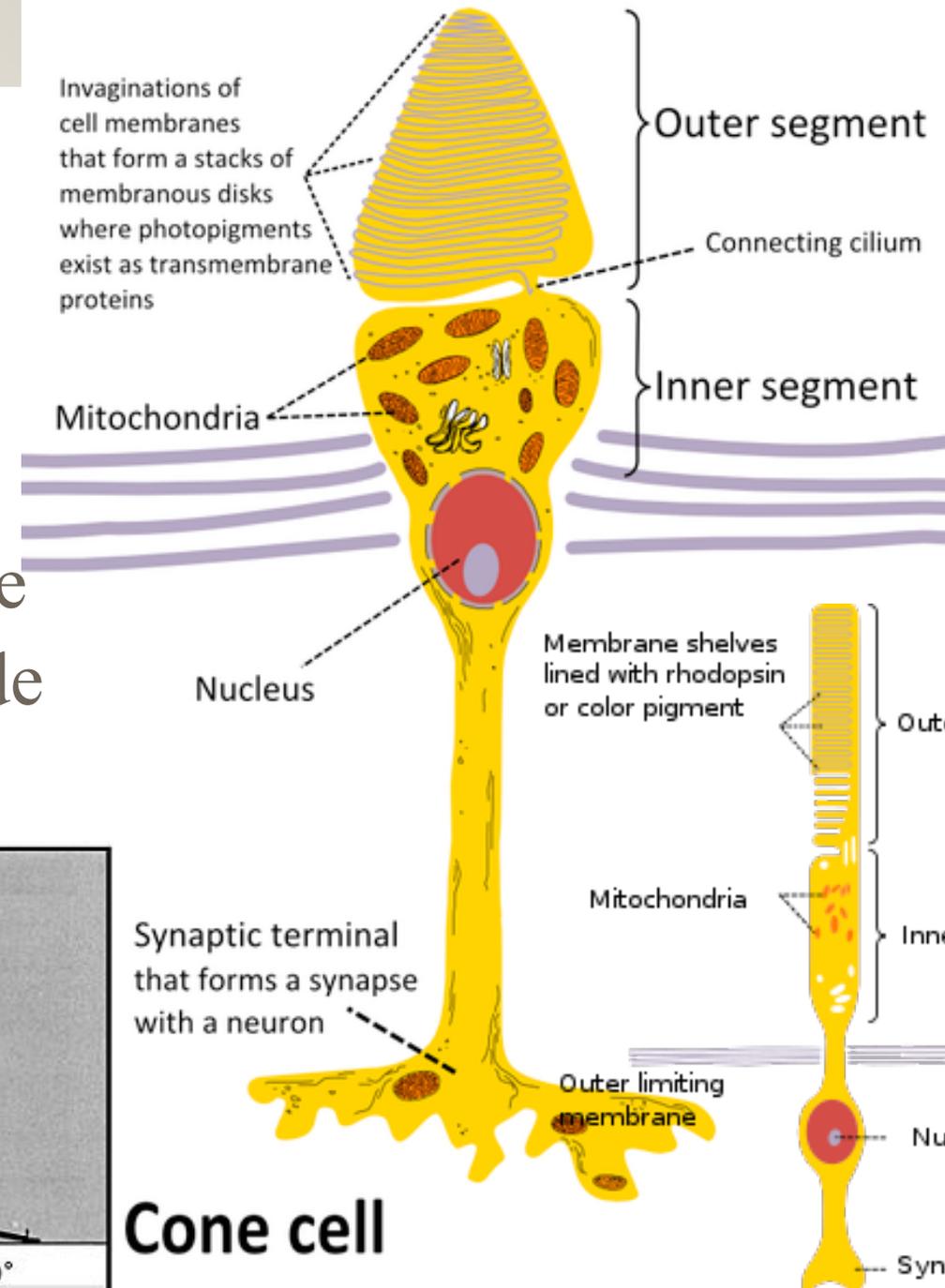
Œil humain

■ Bâtonnets

- 90 M
- Sensibles (vue obs.)

■ Cônes

- 4.5 M
- Moins sensible à la lumière
- Sensible changement rapide
- 3 types (R, V, et B)



Cone cell

- Pourquoi ne pas parler de graphique à haut niveau
 - Peindre la Joconde sur une forme de théière vue de dessus...
 - Mieux comprendre = mieux utiliser
 - Jeux et Applications exigeantes sont construites sur les niveaux bas
 - Sens historique
 - Il reste des choses nouvelles et prometteuses à faire
- Instructions et Données de base
 - Instruction non décomposable
 - Données sous la forme la plus proche de sa forme matérielle

Palette de couleurs



256 couleurs
de la palette VGA

Clairs / foncés

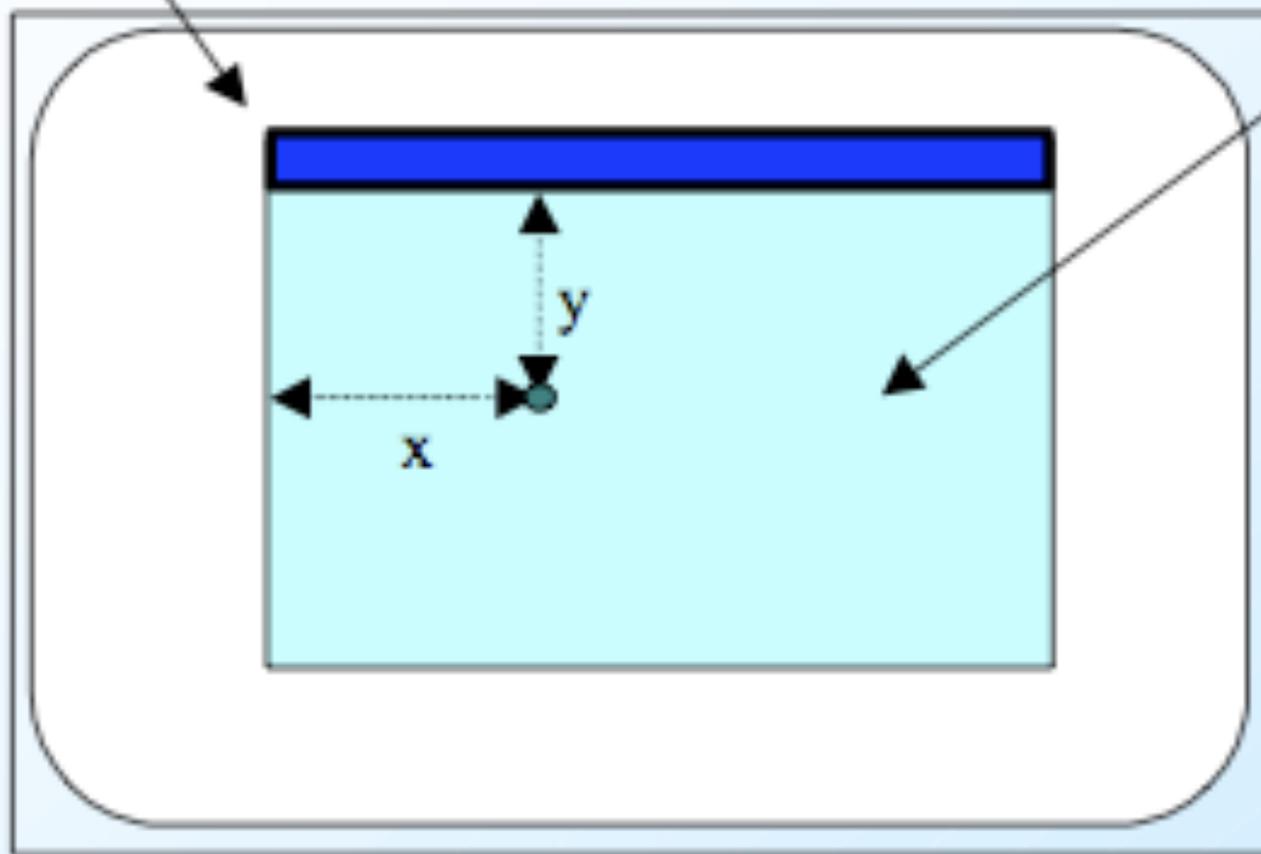
Tons

Couleurs vives et
pastels

Si dessin à l'écran ou dans une fenêtre

Display Device

Window for Graphics



Normal meaning for *SetPixel(x,y,green)*

■ Tailles standards

■ Mode portrait (hauteur > largeur) 

■ Mode Paysage (hauteur < largeur) 

■ 640*480 VGA (Video Graphics Array)

■ 948 Ko (24=3*8 bit par pixel)

■ 1,2 Mo (32 bits par pixel)

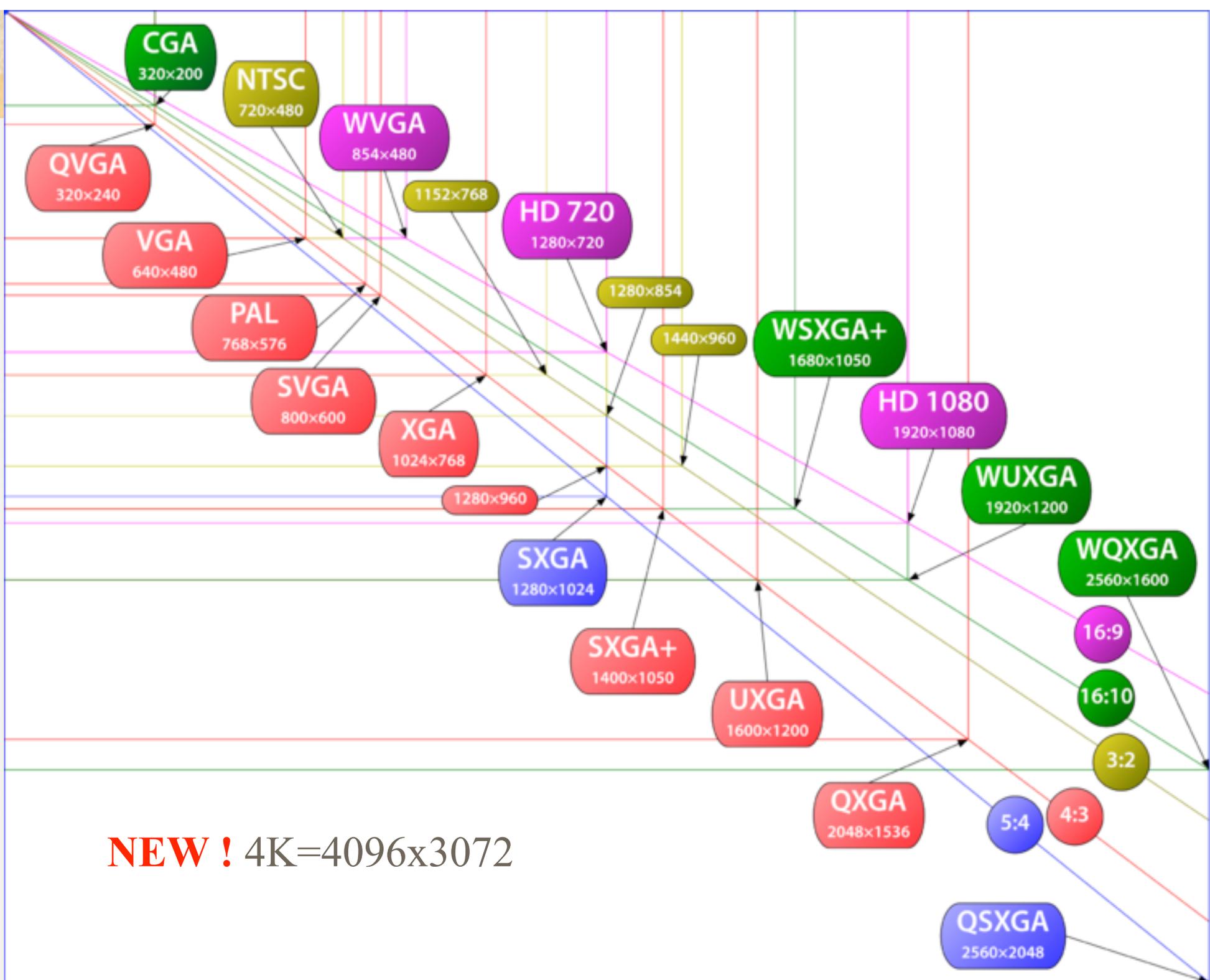
■ à 72dpi = 22,58cm * 16,93cm

■ Rapport 4/3 entre largeur et hauteur

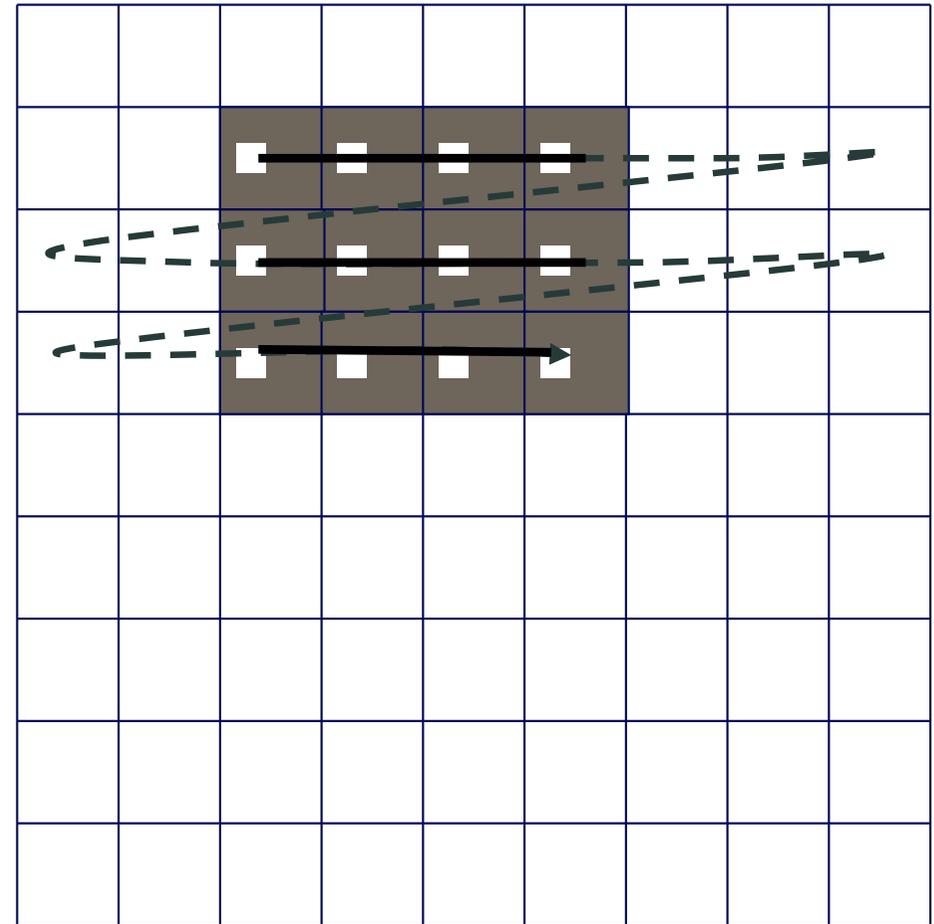
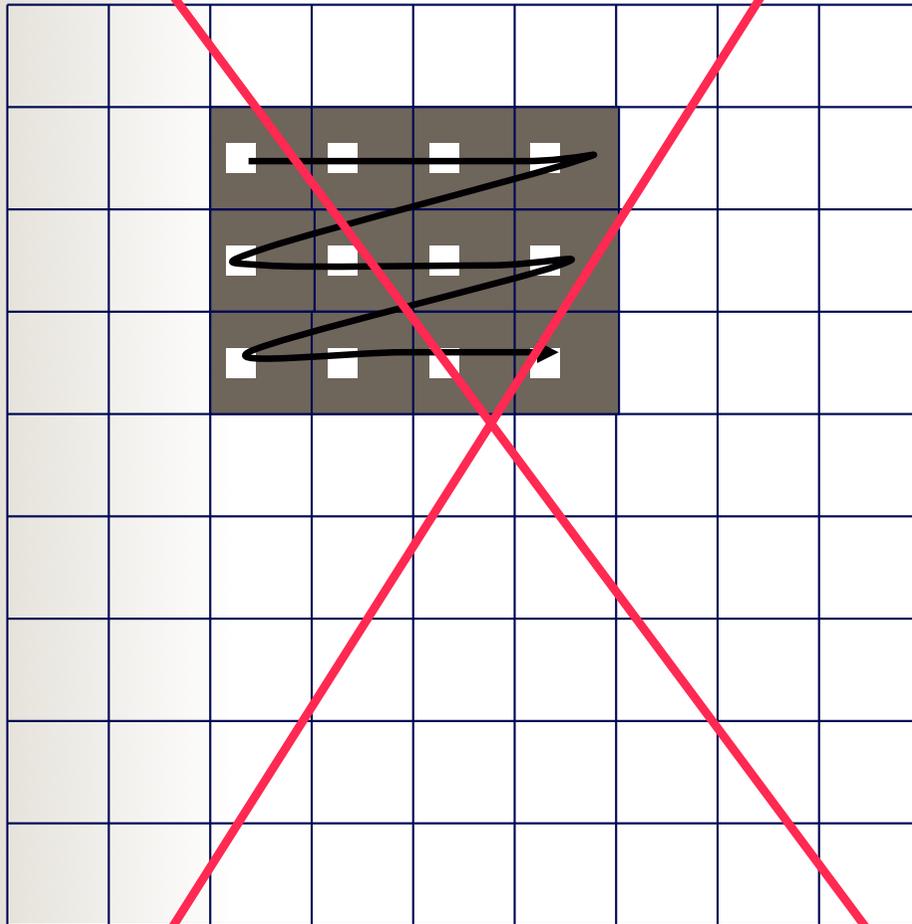
■ 5/4 ou 4/3 ou 3/2 ou 16/10 ou 16/9



Autres



- Rempli un rectangle d'une couleur uniforme
 - `fillRect(x, y, w, h, color) / fillRect(x1, y1, x2, y2, color)?`



■ Mémoire linéaire

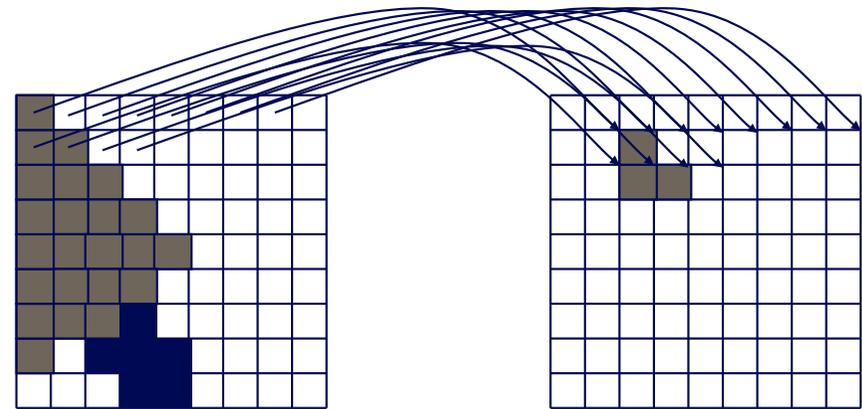
- On ne balaie jamais de haut en bas !
- Mémoire et processeur = 2 entités

■ Performances

- dépendent de la largeur de écran, jamais de la hauteur
- Mise en cache des images et du contenu de l'écran
 - Registres (i.e. 8x64 bits)
 - L1 (i.e 64k)
 - L2 (i.e 1M) \leq 1 petite image mais pas un écran entier !
 - Mémoire vive =RAM (i.e 8G)
 - Disque dur (i.e 1T)

- Copie (Blit surface)

- Copy (adresse1, x1, y1, w1, h1, adresse2, x2, y2)



- La copie efface les pixels de la destination
- La copie ignore les débordements
 - si $(x2+w1 > \text{largeurE})$ copy (... largeurE-x2...)
- La copie ne transforme pas les pixels selon leur encodage
- La copie utilise 2 fois plus de cache !

- RGB (pour RedGreenBlue)
 - La composante en rouge est encodée sur les bits de poids **fort** (l'équivalent des centaines en quelques sorte)
 - La composante en bleu est encodée sur les bits de poids **faible** (l'équivalent des unités en quelques sorte)
 - La composante en verte est au **milieu** (dizaine)
- chaque valeur est comprise entre 0 et 255
- La couleur qui **ajoute** du rouge et du bleu
 - = un violet/rose ! 
- Les couleurs sont additives (plus on en met plus c'est lumineux) contrairement à la peinture (plus on en mélange plus on a un gris foncé)
 - $(r=0, g=0, b=0) \Rightarrow$ noir
 - $(r=255, g=255, b=255) \Rightarrow$ blanc