

## Activité 5 Changements de repères et de coordonnées

Avant de faire un dessin : `rotate(radians(a))` ; : tourne de a degrés (sens horaire),  
`translate(x,y)` ; `scale(r)` ; : zoome si  $r > 1$ , dézoome si  $0 < r < 1$ .

La fonction "push" permet d'effectuer une "sauvegarde" de l'état du système de coordonnées au moment où cette fonction est exécutée. La fonction "pop" restaure cette sauvegarde. Ci-contre, après `pop()` ;, ce qui est relatif à l'état du système dans le code 2 est annulé.

```
Code1  
push();  
code2  
pop();
```

## Activité 6

**On sort de fonction `setup()`** { point de départ } : `function draw()` { code3 }

La fonction "draw" est appelée (c'est-à-dire que code3 est exécuté) à chaque image, c'est-à-dire un certain nombre de fois par seconde.

Dans `setup()` on peut taper `frameRate(10)` qui donne 10 images par seconde = 10 FPS (Frames per second), sinon par défaut c'est 60 FPS (si possible...).

Pour pouvoir être utilisées dans `draw`, les variables doivent être déclarées en dehors de `setup` :

```
var x ; var r ; // on déclare x et r.  
function setup(){ point de départ ...  
x = 0 ; r = 1 ; // on initialise x et r... }  
function draw(){ code utilisant r et v }
```

Par ailleurs `frameCount` est le nombre de frames depuis le lancement du programme. Avec `frameRate(...)` et une division on se fabrique facilement une horloge.

Parenthèse : mon auteur préféré sur la toile : <https://youtu.be/0jjeOYMjmDU>

**Activité 7 Clics de souris** Pour pouvoir utiliser les fonctions qui suivent, il faut que la fonction "draw" soit présente dans le programme (même si elle est vide).

```
mousePressed(){ code...} mouseReleased(){...} mouseMoved(){...} mouseDragged(){...}
```

**p5js propose 4 fonctions qui devront être complétées par le programmeur :** (ces 4 fonctions ne prennent aucun paramètre et ne retournent aucune valeur.)

- le code se trouvant dans la fonction "**mousePressed**" est exécuté une fois quand l'utilisateur appuie sur un des boutons de la souris
- le code se trouvant dans la fonction "**mouseReleased**" est exécuté une fois quand un bouton de la souris qui avait été préalablement pressé est relâché.
- le code se trouvant dans la fonction "**mouseMoved**" est exécuté à chaque image tant que la souris se déplace
- le code se trouvant dans la fonction "**mouseDragged**" est exécuté à chaque image tant que la souris se déplace et que le bouton de la souris est enfoncé.

**`mouseX` et `mouseY` sont des variables donnant les coordonnées du dernier clic.**

Si "mouseX" et "mouseY" vous donnent la position de la souris à l'instant t, "pmouseX" et "pmouseY" vous permettent d'avoir les coordonnées de la souris à l'instant t-dt, avec dt le temps qui s'est écoulé entre 2 images. Exemple : `function mouseDragged() { line(pmouseX,pmouseY,mouseX,mouseY); }` //permet de dessiner !

### Activité 8 Actions avec des touches

Dans cette activité, on travaille sur le clavier. Deux fonctions avec p5js : Le code se trouvant dans la fonction "**keyPressed**" est exécuté une fois quand l'utilisateur enfonce une touche. Le code se trouvant dans la fonction "**keyReleased**" est exécuté une fois quand une touche du clavier est relâchée.

Il est possible de détecter la touche qui a été frappée grâce à la variable "**keyCode**". Cette variable "keyCode" est égale aux codes des touches JavaScript que vous trouverez [ici](#). On a notamment 37 : left arrow, 38 : up arrow, 39 : right arrow et 40 : down arrow. Exemple : `function keyPressed(){if (keyCode==65){code à exécuter quand la touche "a" est enfoncée} }`

Pour déplacer un objet de position (x,y) de façon continue avec les 4 flèches, on peut aussi taper (par exemple) **dans fonction draw()**: `if (keyIsDown(LEFT_ARROW)) x=x-5; if (keyIsDown(RIGHT_ARROW)) x=x+5; if (keyIsDown(UP_ARROW)) y=y-5; if (keyIsDown(DOWN_ARROW)) y=y+5;`

### Activité 9 Insertion de textes

`text(« Hello World »,50,100)` ; peut être inséré dans n'importe quelle fonction. Il y a 3 paramètres : la chaîne de caractères, la coordonnée x du coin "haut-gauche" du texte, la coordonnée y du coin "haut-gauche" du texte . On peut remplacer « gauche » (comportement par défaut) par centre ou droit avec `textAlign(CENTER)` ; ou `textAlign(RIGHT)` ; "**textSize**" prend pour unique paramètre la taille de la police de caractère. On peut écrire `textSize (50)` ; avant ou après `text( , , ) ;` .

**Activité 10 Fonctions sur les Strings** (= méthodes sur les chaînes de caractères), bof...

**Activité 11 Création d'objets** Exemple : l'objet Balle, il a des attributs indiquant sa position, sa vitesse, sa couleur etc et des méthodes qui définissent son comportement : déplacement et affichage.

Après avoir déclaré la var `tabBalle[]`, on peut créer un tableau de (100) balles avec

```
for (i=0;i<100;i=i+1){ tabBalle.push(new Balle()); // code à améliorer, voir ci-dessous
} // C'est mieux avec : tabBalle[i]=new Balle(); car les balles sont identifiées...
```

Par ailleurs on peut créer une surface de dessin (canvas) prenant toute la fenêtre du navigateur...