

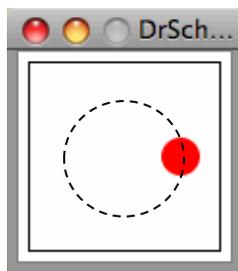
# Programmation Fonctionnelle I, Printemps 2017 – TD4

<http://deptinfo.unice.fr/~roy>

Dans toutes les animations qui suivent, il est essentiel de bien choisir le *Monde*, c'est-à-dire les variables indépendantes (souvent une seule, parfois plusieurs sous la forme d'une structure ou d'une liste) qui gouvernent le phénomène !

**Exercice 4.1** On se propose de programmer une **animation** dans une scène carrée de côté 100. Une balle rouge de rayon 8 va parcourir à vitesse constante un cercle centré au centre de la scène et de rayon 30.

- On note  $t$  l'angle polaire d'un point  $M(x,y)$  mobile sur le cercle. Exprimez les coordonnées  $x$  et  $y$  en fonction de  $t$ .
- A tout moment, l'état du système [ou l'image à dessiner] est déterminé par la valeur de  $t$ . On prend donc la variable  $t$  comme *Monde mathématique*. Programmez les fonctions (suivant  $t$ ) et (dessiner  $t$ ).
- Lancez l'animation en invoquant les fonctions `big-bang`, `on-tick` et `on-draw`.
- Placez le tout dans une fonction (`anim-balle`) comme indiqué dans le cours page 10.
- Rajoutez ce qu'il faut à cette fonction pour que la balle stoppe après 3 tours complets.



*N.B. Que faudrait-il changer pour que la trajectoire soit une ellipse au lieu d'un cercle ?*

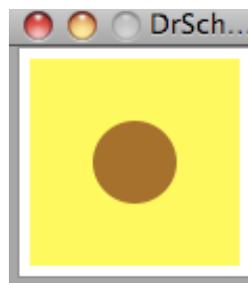
**Exercice 4.2** Modifiez l'animation précédente pour que :

- la balle change de sens de parcours lorsque l'utilisateur presse la barre espace.
- l'animation stoppe lorsqu'il clique sur la souris

On se posera la question fondamentale : quelles sont les variables qui déterminent l'état du système à un moment donné ?

**Exercice 4.3** On dessine au centre d'un canvas 100 x 100 un disque de rayon 20, de couleur verte au départ de l'animation. Cette dernière consiste seulement à écouter le clavier [faut-il une horloge ?]. Sur pression de la touche « flèche en haut » la couleur s'approche très légèrement du rouge, et sur pression de « flèche en bas » elle s'approche du vert. La couleur du disque varie donc continuellement entre le vert et le rouge, avec 256 couleurs intermédiaires.

*N.B. Afin d'utiliser des couleurs autre que les couleurs de base "red", "green", etc. passées sous forme de chaînes de caractères, il faut utiliser la primitive (make-color r g b) vue dans le cours 3 page 4.*



# Programmation Fonctionnelle I, Printemps 2017 – TP4

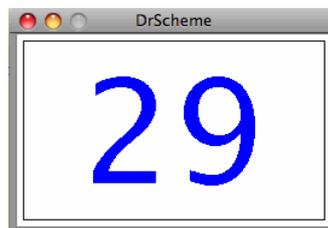
<http://deptinfo.unice.fr/~roy>

Dans toutes les animations qui suivent, il est essentiel de bien choisir le Monde, c'est-à-dire les variables indépendantes (souvent une seule, parfois plusieurs sous la forme d'une structure ou d'une liste) qui gouvernent le phénomène !

**Exercice 4.1** Il s'agit de programmer une **animation** illustrant un *compteur numérique de secondes* dans une fenêtre de dimensions 300 x 200. La simulation consistera à faire afficher en très gros caractères bleus le nombre de secondes écoulées depuis le début de la simulation... On pourra prendre une police de taille 150 par exemple.

a) Quelle sera la variable mathématique qui caractérisera l'état du Monde à un moment donné ? Placez cette information sous forme de commentaire juste avant la définition de la variable INIT. Procédez ainsi dans chaque animation !

b) Programmez l'animation dans une fonction (horloge-numerique T). L'animation stoppe lorsque le compteur dépasse T secondes.

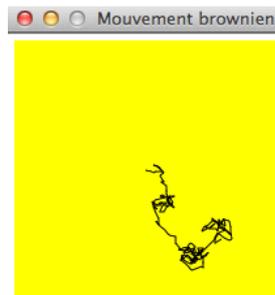


29 secondes après le lancement de la simulation...

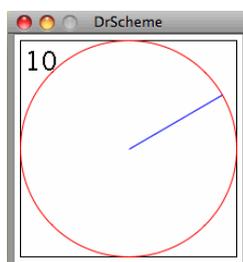
**Exercice 4.2** Programmez l'animation (brownien N) d'une particule rouge de rayon 2 qui se promène au hasard dans un canvas jaune 200 x 200. Elle fait à chaque top d'horloge un petit déplacement dans une direction arbitraire au hasard [on fera varier un peu son x et son y]. On tracera au fur et à mesure la trajectoire parcourue. *Indication : animation à 3 paramètres !* Testez avec  $N = 200$  et une horloge qui bat 4 fois par seconde.

Pour ajouter une ligne dans la scène d'une animation, mieux vaut utiliser `scene+line` que `add-line`. En effet, `scene+line` fait un "cropping" automatique avec la scène, alors que `add-line` peut agrandir la "bounding-box" !

Complément : voir l'exercice 5.6.4 du livre de cours PCPS...



**Exercice 4.3** Transformez votre compteur numérique en une **horloge analogique**, avec une seule aiguille bougeant à chaque seconde, et un affichage en mode texte des secondes courantes en haut à gauche. La solution complète prend une quinzaine de lignes. Il vous faudra bien entendu de la trigonométrie pour gérer l'angle de l'aiguille...



N.B. Pour votre entraînement personnel et une préparation optimale aux cc-flash et partiels, faites TOUS les exercices du livre de cours. Vous en voulez plus ? Programmez le célèbre EYES (<http://arc.id.au/XEyes.html>), vous trouverez la photo de Dali sur <http://deptinfo.unice.fr/~roy/Images/Dali.png>