

## TP4 Visualisation d'informations : Interaction

P. Cubaud <cubaud @ cnam.fr>

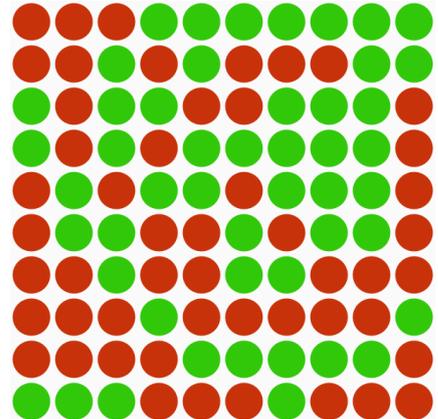
L'objectif de ce TP est de poursuivre l'étude de l'environnement Processing avec les fonctions d'animation et d'interaction.

### Exercice 1. Processing en mode animé

Lire la documentation de Processing concernant *setup()*, *draw()* et *frameRate()*. Reprendre le code de l'exercice 1 du TP mais en mettant tout le code du dessin du pavage à l'intérieur de la fonction *draw()* et juste l'ordre *size()* dans *setup()*

Que se passe-t'il ? Etudier l'impact de *frameRate()* sur l'animation.

Modifier ensuite le code pour que soit affiché une matrice de cercles, dont la couleur de remplissage est rouge avec une probabilité 0.4 ou verte sinon, comme sur la figure ci-contre.



### Exercice 2. Fonctions d'interaction

Lire la documentation de *keyPressed()* et de *saveFrame()*. On veut que l'utilisateur puisse enregistrer à la demande le contenu de la fenêtre dans un fichier graphique (touche 's').

On veut aussi que le dessin soit régénéré par de nouveaux tirages au sort (touche espace). Pour bloquer l'animation dans *draw()*, on peut utiliser les fonctions *noLoop()* et *loop()* : voir leur documentation.

### Exercice 3. Programmation objet

Réorganiser le code de l'exercice 1 en introduisant une classe *Truc* qui se charge du tracé du cercle, conserve sa position et son état binaire (pour être dessiné en vert ou rouge).

Ajouter ensuite des méthodes à la classe *Truc* pour que les actions utilisateur de l'exercice 2 fonctionnent aussi, mais sans utilisation de *loop()* et *noLoop()*.

### Exercice 4. Sélection

Lire la documentation concernant *mouseX*, *mouseY*. S'en servir pour que s'affiche au centre de l'écran l'identifiant du *Truc* désigné par la souris. L'identifiant sera (par exemple) constitué du numéro de ligne et du numéro de colonne du *Truc*.

Pour savoir quel *Truc* est concerné par la souris, on se ramène à la condition qu'un point (x,y) soit à l'intérieur d'un cercle de centre (cx,cy) et de rayon R. Quelle est-elle ? En faire une méthode supplémentaire de la classe *Truc*, appelée *rollover()*.

Remarque : pour afficher du texte, on utilise la fonction *text()*.

## Exercice 5. Interaction

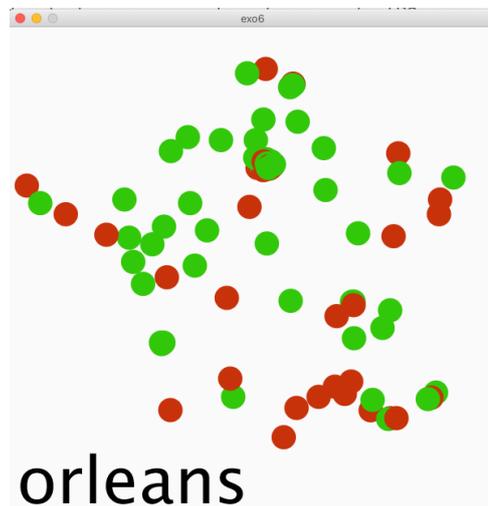
Avec la fonction *MousePressed()*, ajouter au code précédent la prise en compte du clic souris sur un Truc. L'action du clic sera de forcer l'état du Truc à celui qui correspond à la couleur verte.

## BONUS Exercice 6. De vraies données

Modifier le code précédent pour utiliser des données plus utiles. Au lieu de placer dans la fenêtre les Trucs sous la forme d'une matrice, utiliser des positions issues d'une carte.

Pour cela, on utilisera le fichier *communes100.csv* sur la page Moodle du cours. Celui-ci a été produit à partir du fichier utilisé au TP2 de visualisation par extraction des 100 communes les plus peuplées.

On constate qu'il suffit de peu de modifications, essentiellement au sein du code de la classe *Truc*.



Remarque :

Comme certaines villes sont très proches les unes des autres, il est compliqué de cliquer dans le cercle correspondant. Il faudrait ajouter une interface spécifique.