

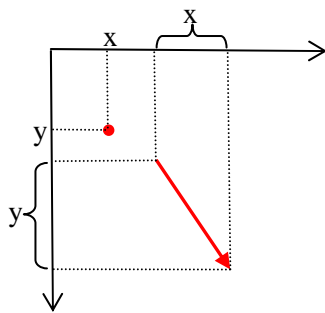
# Programmation 1

## Examen

Les *boids* sont des oiseaux virtuels qui volent dans un espace à 2 dimensions en obéissant à 3 règles :

- Eviter de se rapprocher trop de ses voisins les plus proches : règle de *séparation*.
- Chercher à ne pas trop s'éloigner de ses voisins le plus proches : règle de *cohésion*.
- Suivre le cap moyen de ses voisins : règle d'*alignement*.

Les *boids* sont représentés à un moment donné par une position et une vitesse. Pour représenter une position ou une vitesse on utilisera la classe *Vecteur* :



```
public class Vecteur{
    public double x, y;
    public Vecteur(double x, double y)
    public Vecteur(Vecteur v)
    public void setXY(double x, double y)

    public double longueur()
    public void normaliser()

    public void plus(Vecteur v)
    public void moins(Vecteur v)
    public void fois(double nbr)
}
```

### **Question 1.** (2 points)

Définir les constructeurs et la méthode *setXY* de la classe *Vecteur*.

### **Question 2.** (2 points)

Définir la méthode *longueur* et la méthode *normaliser* de la classe *Vecteur*.

La longueur d'un vecteur est la racine carrée de  $x^2+y^2$ . Normaliser un vecteur c'est diviser chacune de ses composantes par sa longueur, si elle est différente de 0. ( on obtient la racine carrée d'une valeur réelle par la méthode *double Math.sqrt( double v )* )

### **Question 3.** (3 points)

Définir les méthodes *plus*, *moins* et *fois* :

- *plus* modifie les composantes *x* et *y* en leur ajoutant les composantes *x* et *y* du paramètre.
- *moins* modifie les composantes *x* et *y* en leur soustrayant les composantes *x* et *y* du paramètre.
- *fois* modifie les composantes *x* et *y* en les multipliant par *nbr*.

La classe *BoidAbstrait* est définie de la façon ci-contre :

```
abstract public class BoidAbstrait {
    static double distanceSeparation = 20;
    static double distanceCohesion = 100;
    static double distanceAlignement = 100;
    Vecteur position;
    Vecteur vitesse;
    Univers lUnivers ;
    public BoidAbstrait(Univers u)
    protected Vecteur forceCohesion(BoidAbstrait b) { . . . }
    protected Vecteur forceAlignement(BoidAbstrait b) { . . . }
    protected Vecteur forceSeparation(BoidAbstrait b) { . . . }
    protected boolean proche(BoidAbstrait b, double dist)
}
```

**Question 4.** (2 points)

Définir la méthode *proche* qui retourne *true* si le *boïd* *b* est dans le cercle centré sur la position du *boïd*, et de rayon *dist*, et *false* sinon.

La Classe *Boid* est définie de la façon suivante :

```
public class Boid extends BoidAbstrait{
    public Boid()

    protected Vecteur separation()
    protected Vecteur cohesion()
    protected Vecteur alignement()

    private void appliquer(Vecteur f)
    public void deplace()
}
```

**Question 5.** (5 points)

Définir les méthodes *separation*, *cohesion* et *alignement* qui retournent respectivement les forces de séparation, de cohésion, et d'alignement à appliquer sur un *boïd*.

- La force de séparation est la somme des forces de séparation pour tous les *boïds* éloignés de moins de *distanceSeparation*.
- La force de cohésion est la somme des forces de cohésion pour tous les *boïds* éloignés de moins de *distanceCohesion*, divisée par le nombre de *boïds* qui participent à cette force.
- La force d'alignement est la somme des forces d'alignement pour tous les *boïds* éloignés de moins de *distanceAlignement*, divisée par le nombre de *boïds* qui participent à cette force.

L'attribut *lUnivers* est une référence à un objet décrivant l'univers dans lequel évoluent les *boïds*. C'est essentiellement un tableau public de *boïds* : *Boid[] lesBoïds* dont chaque élément est une référence à un *boïd* de l'univers.

**Question 6.** (2 points)

Définir la méthode *deplace* qui déplace un *boïd*. Le déplacement d'un *boïd* se fait en calculant la somme des forces qui s'exercent sur lui et en lui appliquant cette force. ( on ne demande pas de définir la méthode *appliquer* ).

**Question 7.** (4 points)

Un *BoidSolitaire* est un *boïd* pour lequel la force de cohésion est le dixième de la force de cohésion d'un *boïd* normal. Définir la classe *BoidSolitaire*.

Un *BoidMilitaire* est un *boïd* pour lequel la force d'alignement est dix fois la force d'alignement d'un *boïd* normal, et la distance de séparation deux fois plus petite. Définir la classe *BoidMilitaire*.