

Table des matières

Exercice 1: Bouger une tortue.....	1
Exercice 2: Bouger plusieurs tortues.....	3
Exercice 3: Tortues en couleur.....	5
Exercice 4: [pour avancés].....	5

Exercice 1: Bouger une tortue

Développez une application permettant de dessiner et de déplacer une tortue à l'écran.

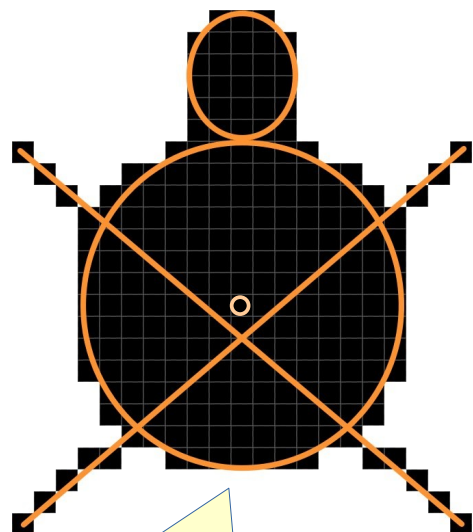
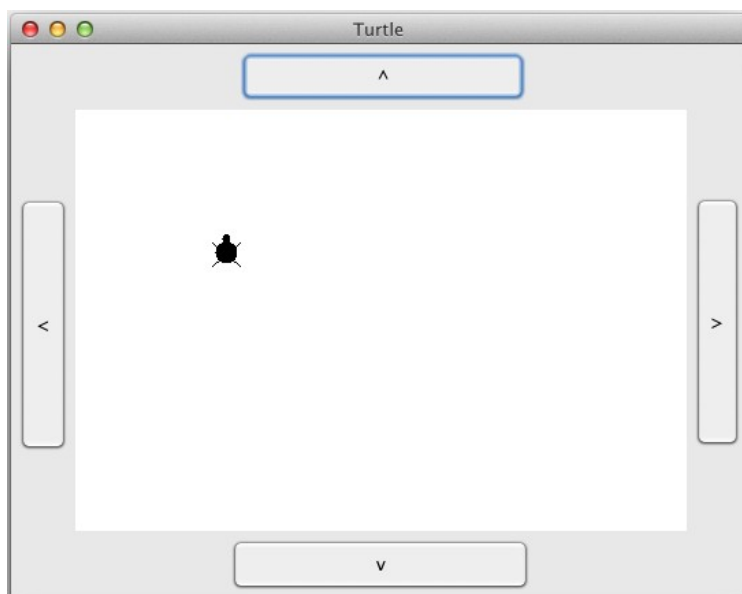
- Le déplacement se fait à l'aide de quatre boutons situés à l'extérieur de la surface de dessin.
- La vitesse de déplacement de la tortue est de 10 pixel par clic.
- La position initiale de la tortue est (100,100).
- Le dessin de la tortue se fait approximativement comme sur le dessin ci-dessous.
- La tortue 'regarde' toujours vers le haut, même si elle se déplace dans une autre direction. La tortue peut se balader aussi en-dehors de la surface visible.

Développer le programme en respectant le modèle MVC et en employant 3 classes.

Dans cette première version, les opérations de dessin sont effectuées dans **paintComponent** de **DrawPanel**.

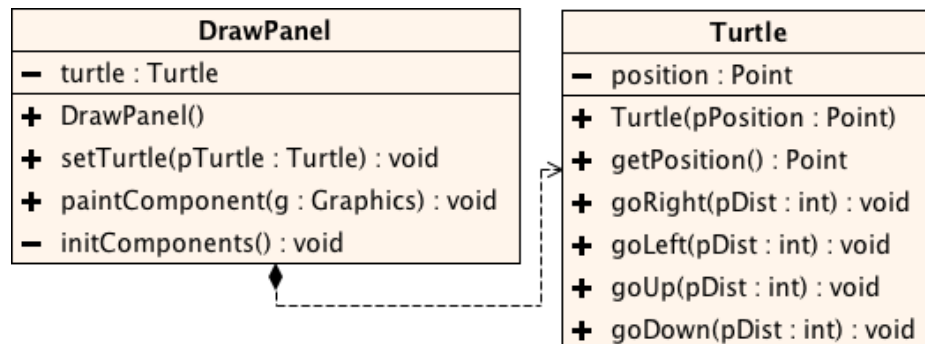
Principe de base :

Une nouvelle tortue est créée dans **MainFrame**. Elle est passée tout de suite à **DrawPanel** (à l'aide de **setTurtle**). Ceci est primordial, puisque **MainFrame** et **DrawPanel** doivent travailler avec LA MÊME tortue.



Largeur: 21 pixels
Hauteur totale : 24 pixels
Diamètre carapace: 15 pixels

Pour vous aider dans la conception du programme, voici le schéma UML de la classe **Turtle** représentant la tortue et de la classe **DrawPanel** pour représenter la tortue graphiquement sur un panneau. L'attribut **position** correspond au centre de la carapace de la tortue (représenté par le petit cercle dans le schéma).



Exercice 1A : Conception alternative :

Dans la première version, la tortue savait gérer sa position, mais elle ne savait pas se dessiner. Toutes les opérations de dessin étaient effectuées dans **paintComponent** de **DrawPanel**.

Dans cette nouvelle version, la tortue sait se dessiner elle-même sur un canevas. **DrawPanel** va alors 'demander' à la tortue de se dessiner sur son canevas.

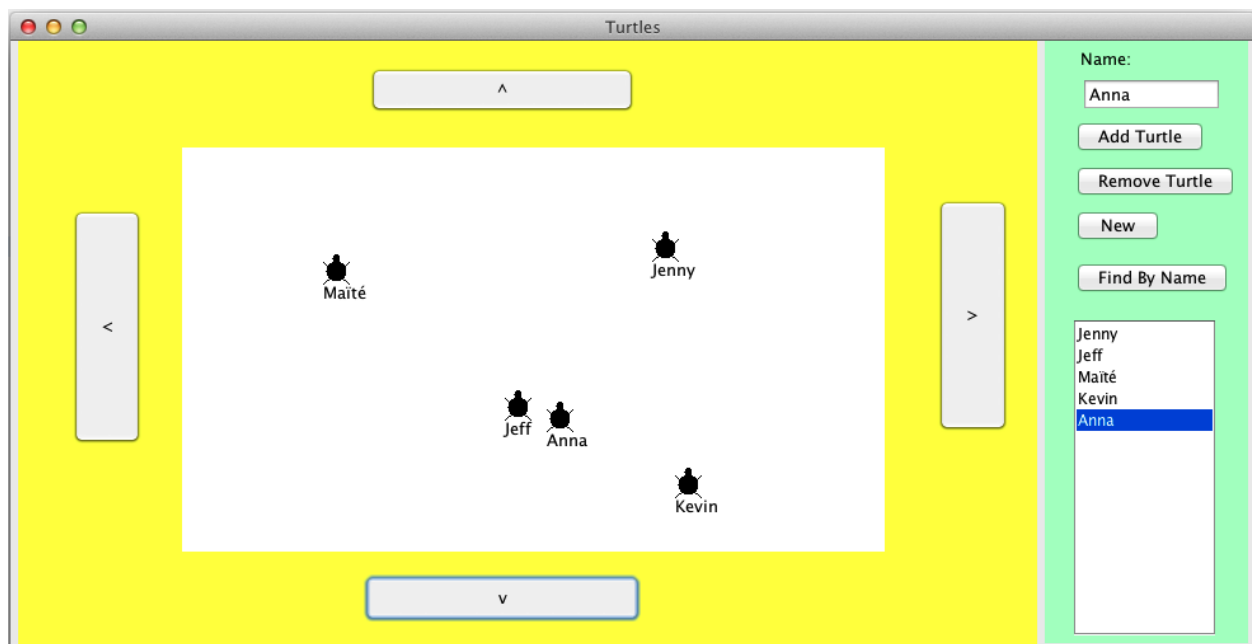
Ajoutez à la classe **Turtle** la méthode **draw(Graphics g)** qui dessine la tortue sur un canevas de dessin et adaptez la méthode **paintComponent** de la classe **DrawPanel**.

Comparez les deux conceptions !

Exercice 2: Bouger plusieurs tortues

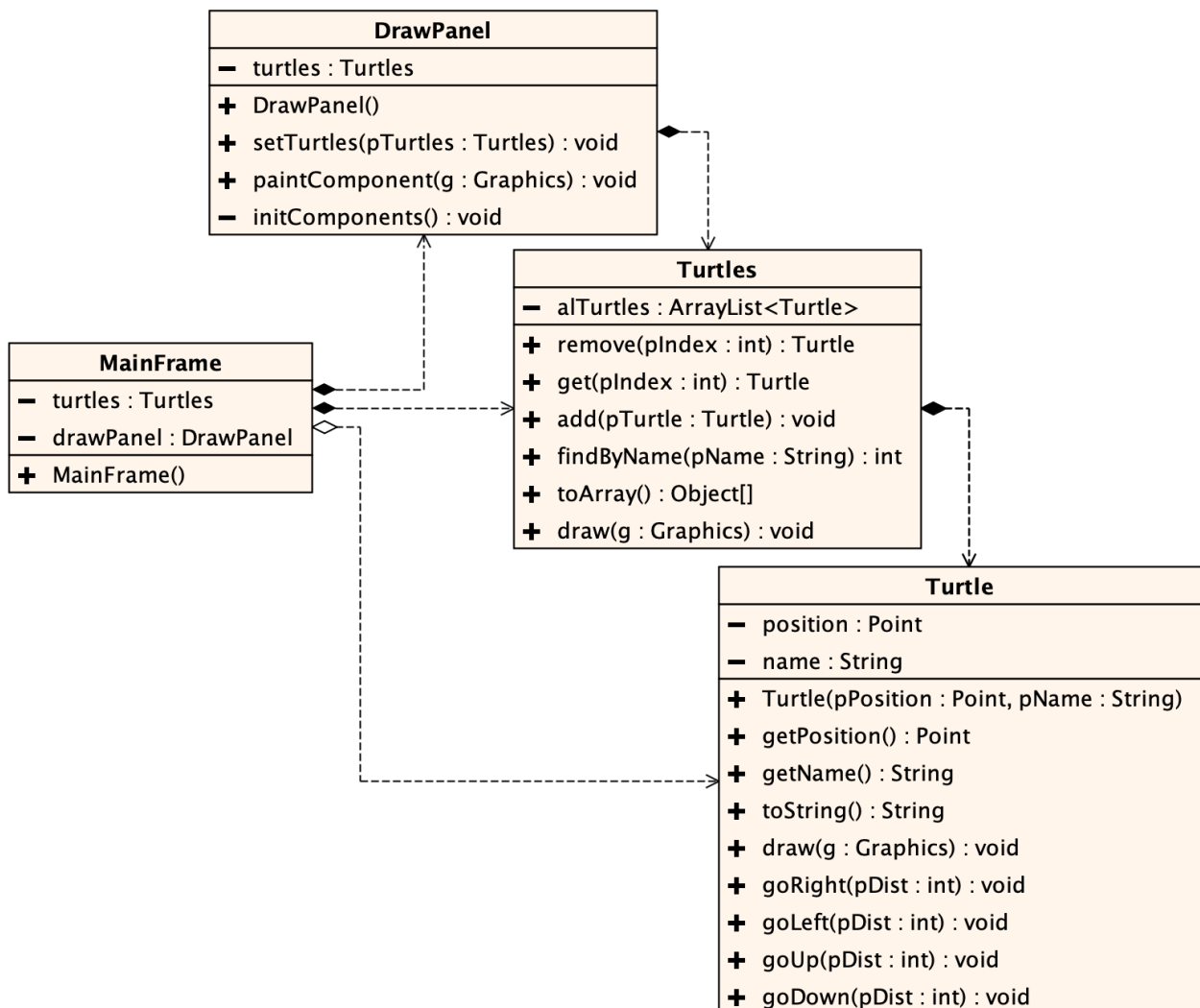
Ce programme consiste à étendre le programme précédent (**version 1A**) de la manière suivante :

- Plusieurs tortues peuvent être affichées et déplacées sur le panneau
- Une tortue a un nom **name** qui est affiché sur le panneau
- Les noms des tortues sont également affichés dans un composant de type **JList**
- La tortue sélectionnée dans la liste peut être déplacée sur le panneau
- Le bouton **Add Turtle** permet d'ajouter une nouvelle tortue qui est ensuite placée à une position aléatoire sur le panneau. Cette tortue porte le nom saisi dans la boîte d'édition **Name**
- Le bouton **Remove Turtle** supprime la tortue sélectionnée dans la liste
- Le bouton **New** crée une nouvelle liste de tortues
- Le bouton **Find By Name** permet de chercher dans la liste la tortue qui porte le nom saisi dans la boîte d'édition **Name**. Si une tortue est trouvée alors elle est marquée dans la liste. Si plusieurs tortues portent le nom recherché alors la première est sélectionnée.
- la classe **Turtles** possède une méthode **draw(Graphics g)** qui dessine toutes les tortues de la liste sur un canevas **g**.



Pour vous aider dans la conception du programme, vous trouvez dans la suite le schéma UML des **CLASSES** :

- la classe **Turtle** représentant une tortue,
- la classe **Turtles** qui gère une liste de tortues,
- la classe **DrawPanel** qui fait dessiner les tortues sur son canevas,
- la classe **MainFrame** qui est représentée de façon simplifiée (sans les composants et les méthodes de réaction). Mais il est bien visible qu'elle contrôle le déroulement et les interactions entre les autres classes.



A la page suivante (à droite), vous trouvez un diagramme représentant les **OBJETS** au cours de l'exécution du programme. Dans cette représentation, on a 5 tortues (*Jenny, Jeff, Maïté, Kevin et Anna*) sur le canevas.

Exercice 3: Tortues en couleur

Ajoutez un attribut **color** à la classe **Turtle**, qui est initialisé lors de la création d'une tortue. Lors du dessin, la tortue est dessinée avec la couleur **color**. Ajoutez à l'interface graphique une possibilité de choisir une couleur.

Conseil : Ajoutez un composant **JColorChooser** à votre fiche et nommez-le **colorChooser**. Ainsi vous avez accès à sa couleur actuelle par : `colorChooser.getColor()`

Exercice 4: [pour avancés]

Ajoutez à la classe **Turtle** un attribut **orientation** pour mémoriser si la tortue regarde vers le haut, le bas, la gauche ou la droite. L'orientation est changée lors du déplacement par les boutons. Lors du dessin, dessinez la tortue de façon à ce qu'elle regarde toujours dans la direction actuelle (c.-à-d. celle du dernier déplacement).

