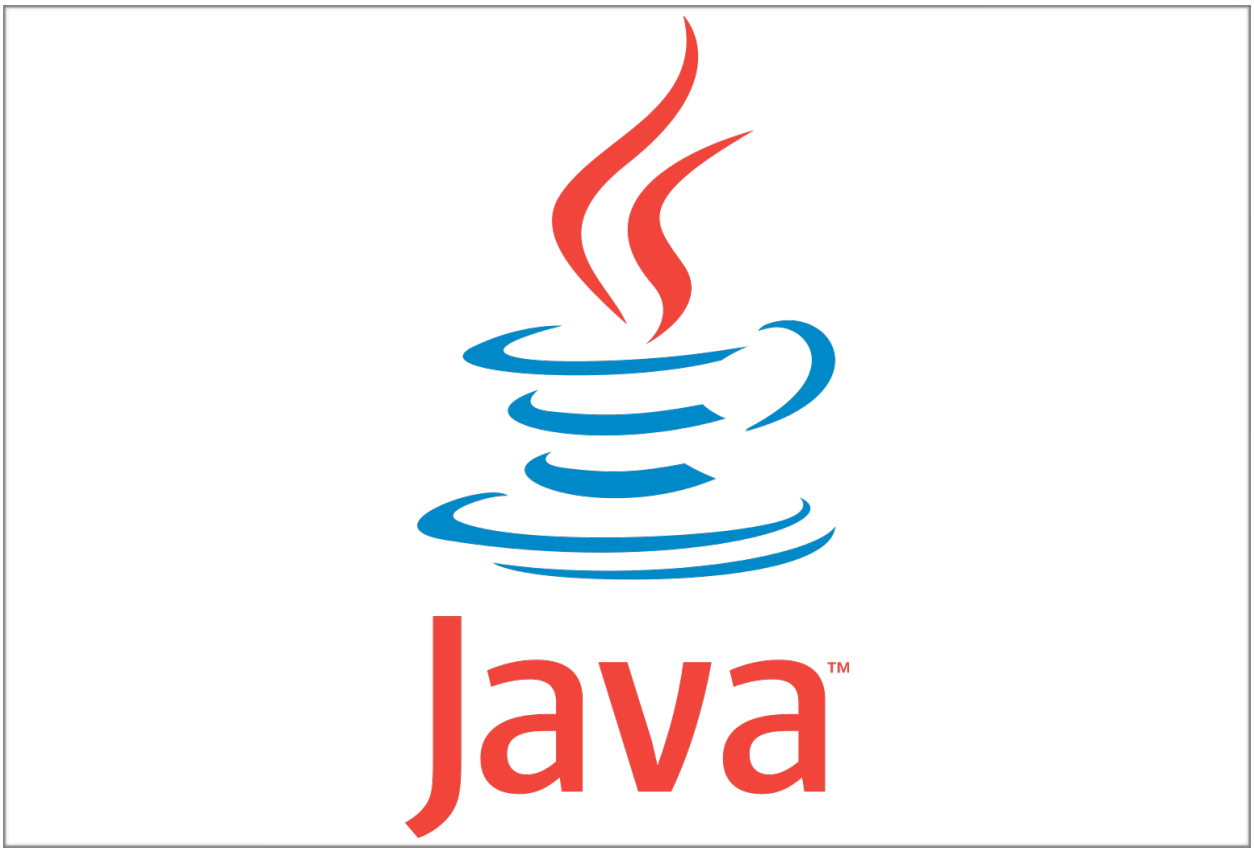
Projet 2

8INF957 : Programmation objet avancée



Professeur : Mcheick, Hamid

Pierre-Antoine Tible

Robin Thoni

Antoine Cormery

Implémentation du projet

Le projet est divisé en quatre parties :

- food
- pigeon
- main
- game

Main

Lance le projet dans une interface Swing. Il est possible de changer le nombre de pigeons par défaut qui est de 10.

Food

Une nourriture est simplement gérée par sa position. En conséquence le constructeur prend en paramètre une position x et une positon y. Des getter setter sont disponibles pour accéder à la positon.

Pigeon

On initialise un pigeon avec un nom et on lui passe son environnement (Game). La classe implémente l'interface Runnable de manière a être lancée dans un thread. Un pigeon a un nombre de mouvements possibles par tour définie par la variable speed, par défaut 10.

Pigeon(nom, Game) : une position aléatoire est donnée au pigeon.

Run() : gère le comportement du pigeon pendant sa vie, celle-ci est gérée avec un while(true). On Synchonized Game au début de la méthode pour protéger les accès concurrents. Ensuite on récupère les nourritures disponibles sur le terrain et on cherche celle la plus proche du pigeon. Si on est à une distance inférieure au déplacement possible du pigeon alors celui-ci mange la nourriture. On supprime celle-ci du terrain de jeu. Un seul pigeon peut manger la nourriture, cette unicité est garantie par le synchronized. Il est possible que le pigeon soit loin de la

nourriture, alors il s'en rapproche mais ne peut pas encore la manger. Quand il n'y a pas de nourriture sur le terrain le pigeon ne se déplace pas, sauf quand celui-ci est effrayé (simulé avec un Random).

Le dernier point de cette méthode est de faire un sleep de 100ms pour éviter de surcharger le processeur avec la boucle.

ComputeDistance(x, y) : prend en paramètre la position d'une nourriture et renvoie la distance de manhattan entre le pigeon et la nourriture. Le but de cette méthode est de se diriger vers la nourriture la plus proche.

ComputeNewPosition(x, y) : prend en paramètre la position d'une nourriture et dirige le pigeon vers celle-ci.

Start() : lancer le thread du pigeon.

Game

La classe Game représente la vie du jeu. Celle-ci extends JPanel de manière à être ajoutée au JFrame de la classe Main et pouvoir gérer les éléments graphiques. La classe implémente également les interfaces MouseListener et ActionListener pour gérer la souris et le rafraichissement. Elle contient en private la liste des pigeons et des nourritures.

Game(nb) : prend en paramètre le nombre de pigeons à créer. Initialise également un timer pour l'action à 50ms.

Paint() : gère le dessin. Les pigeons sont dessinés en rouge et les nourritures en bleu. La liste de nourritures est accédée à l'aider d'un synchonized.

MousePressed() : quand on clique alors on ajoute une nourriture à la position de la souris. La méthode est synchonized pour éviter tout conflit.

RemoveFood(food) : supprime la nourriture de la liste des nourritures. Cette méthode est synchronized et est prévue pour être appelée depuis un thread pigeon.

ActionPerformed() : est l'action à faire toutes les 50ms, dans notre cas un repaint();