

## 8INF957 - Programmation objet avancée

### Rapport TP2

#### Objectifs :

Création d'une simulation d'alimentation de pigeons. Il est possible de déposer des graines à l'aide de la souris. Les pigeons sont attirés par les graines les plus récentes. Une fois un délais dépassé, les graines ne sont plus mangeables et reste affichées. Il peut arrivé que les pigeons aient peurs, ils se déplacent alors de manière aléatoire.

#### Lancement de l'application :

Un exécutable est disponible à la racine du fichier Zip, au lancement de l'application un bouton *Launch* permet de faire apparaitre les pigeons et de déposer des graines.

#### Implémentation :

*Pigeon.cs : Cette classe comprend toutes les fonctions liées au pigeon.*

`window_limit()` permet d'empêcher le pigeon de sorti de la fenêtre, notamment dans son état de peur.  
`find_freshest_seed()` prend en compte les croyances du pigeon pour déterminer la graine la plus proche de lui, permet de modifier son attribut désire.

`move_to_food()` en fonction de la position du désire, permet de fixer la direction que va suivre le pigeon.

`move_random()` renvoi une direction aléatoire, simulant la peur du pigeon.

`distance_seed_calculation(Graine seed)` calcul la distance entre la position du pigeon et une graine, pour ensuite déterminer la plus proche.

`distance_target_calculation()` calcul la distance entre le pigeon et son désire.

`next_position(int delta_time, String new_state)` en fonction de l'état dans lequel est le pigeon, de sa position et de son désire, cette fonction renverra une coordonnée correspondant à la prochaine position du pigeon.

`eat()` permet de renvoyer la coordonnée de la graine qu'à manger le pigeon.

*Environnement.cs : Cette classe s'occupe de l'environnement.*

`initialise()` est la méthode permettant de créer les pigeons et les liés à leurs threads.

`randomPigeonMove()` est la fonction qui modifie le comportement du pigeon de façon aléatoire.

`deleteSeed(int seedX, int seedY)` supprime la graine de l'affichage et de la liste de graines.

`CriticalZone(int seedX, int seedY)` implémente le mutex qui permet d'empêcher plusieurs pigeons de manger une même graine.

*Graine.cs : Cette classe comporte les attributs et les méthodes liées aux graines.*

`expiration_date` : entier qui désigne la date d'expiration d'une graine, à chaque itération celui-ci décrémente, à 0 la graine est considérée comme immangeable.

`rotten_img` : booléen permettant de savoir l'image affichée.

`update_seed_expiration()` : méthode qui décrémente l'attribut `expiration_date` et vérifie que la graine est encore mangeable.