

CR TP8 : Classes

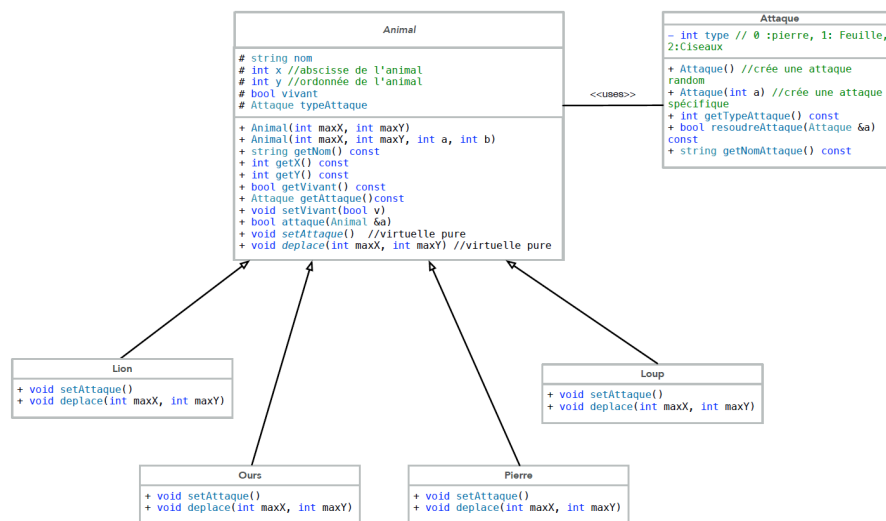
Aurélien GENTE - Laurène BERNIER

Objet

L'objectif de ce TP est de réaliser un mini jeu de vie artificielle dont les règles sont détaillées dans l'énoncé. Il faudra implémenter des animaux (classe mère) d'espèces différentes (classes filles) avec des attaques et des modes de déplacement différents.

Classes

Dans un premier temps, on commence par créer les classes Attaque, Animal, et les classes filles associées à Animal : Ours, Lion, Pierre, Loup en suivant le diagramme suivant :



Premier essai

Dans notre fichier main.cpp, nous allons dans un premier temps créer 4 animaux (1 de chaque type) à des positions précises et ainsi tester la suite du code.

```
fonction collision
fonction collision()
algorithme
debut
    POUR chaque animal i dans le tableau animaux :
        SI animaux[i] est vivant ALORS
            POUR chaque animal j (de la fin vers le début) dans le tableau animaux :
                SI animaux[j] est vivant ET animaux[j] != animaux[i] :
                    SI animaux[i] et animaux[j] sont sur la même case (X, Y) ALORS
                        Afficher le combat entre animaux[i] et animaux[j]
                        Résultat = animaux[i] attaque animaux[j]
                        SI Résultat (animaux[i] a gagné) ALORS
                            Afficher victoire de animaux[i]
                            animaux[j] devient non-vivant
                        SINON
                            Afficher victoire de animaux[j]
                            animaux[i] devient non-vivant
                        SORTIR de la boucle interne
                    FIN SI
                FIN SI
            FIN SI
        FIN POUR
    FIN SI
```

```

    FIN POUR
fin

```

```

fonction
    fonction tableau(entier m, entier n)
        Parametres entier m (nb de lignes)
                     entier n (nb de colonnes)
    algorithme
    debut
        tableau = Allouer un tableau de m pointeurs
        POUR i allant de 0 à m-1
            tableau[i] = Allouer un tableau de n entiers
            POUR chaque ligne i allant de 0 à m-1 :
                POUR chaque colonne j allant de 0 à n-1 :
                    Afficher "|----"
                FIN POUR
            Afficher "|"
            POUR chaque colonne j allant de 0 à n-1 :
                animalTrouve = faux
                POUR chaque animal k dans animaux :
                    SI animal[k].vivant ET animal[k].X == i ET animal[k].Y == j ALORS
                        Afficher "| xx " où xx = premières lettres de animal[k].nom
                        animalTrouve = vrai
                    STOP
                FIN SI
            FIN POUR
            SI animalTrouve == faux ALORS
                Afficher "|  "
            FIN SI
        FIN POUR
        Afficher "|"
    FIN POUR
    Détruire tableau
fin

```

Test

En exécutant notre programme, nous obtenons le résultat suivant :

```

=== Verification des collisions initiales ===
Combat entre W et P !
W a vaincu P !
Combat entre L et O !
L a vaincu O !
|----|----|----|----|----|
|    |    |    |    |    |
|----|----|----|----|----|
| L  | W  |    |    |    |
|----|----|----|----|----|
|    |    |    |    |    |
|----|----|----|----|----|
|    |    |    |    |    |
|----|----|----|----|----|

```

Lorsque deux animaux se trouvent sur la même case, l'attaque à lieu et le vainqueur est affiché

Suite

A l'état actuel, notre programme créer une grille de taille donnée avec 4 animaux d'espèce donnée à des position précises. Les objectifs suivants serait donc :

- Choisir une dimension de grille personnalisée

- Générer un nombre aléatoire d'animaux, de type aléatoire et à une position aléatoire
- Actualiser la grille tour par tour afin d'observer l'évolution du jeu de vie