

SMP - TP8

Le but de ce TP est d'écrire les bases d'un mini jeu de vie artificielle. **Lisez la totalité de l'énoncé avant de commencer.**

1 RÈGLES DU JEU

Le plateau est un tableau bidimensionnel $\text{maxX} \times \text{maxY}$ de cellules. Les cellules sont peuplées par des animaux. A chaque tour, tous les animaux du plateau effectuent un mouvement qui dépend de leur type et qui sera détaillé ensuite. Si deux animaux se retrouvent sur la même cellule, ils se battent jusqu'à disparition de l'un d'entre eux.

Pour les bagarres, elles se jouent au "pierre-feuille-ciseaux" (shifumi). Les règles du shifumi sont les suivantes :

- La pierre écrase les ciseaux et gagne.
- La feuille enveloppe la pierre et gagne.
- Les ciseaux découpent la feuille et gagnent.

Ces règles s'appliquent mais :

- Un match nul est réglé par l'utilisation du générateur aléatoire.
- S'il y a plus de deux animaux sur la même cellule, deux sont sélectionnés au hasard pour se battre et ce jusqu'à ce qu'il ne reste qu'un seul animal.

Nous avons affaire à différents types d'"animaux" :

- les lions : ils se déplacent de manière aléatoire à chaque tour dans une des 4 directions (1,1), (1,-1), (-1,-1) ou (-1,1) (Fig. 1). Ils attaquent de manière aléatoire avec feuille ou ciseaux.
- les pierres : elles ne se déplacent pas. Elles attaquent uniquement avec des pierres!
- les loups : ils se déplacent (au hasard sur n'importe quelle cellule du plateau) et attaquent avec l'une des trois attaques possibles au hasard à chaque tour.

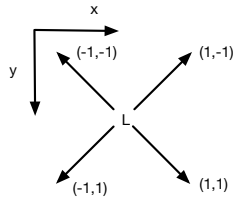


FIGURE 1.1 – Déplacements des Lions.

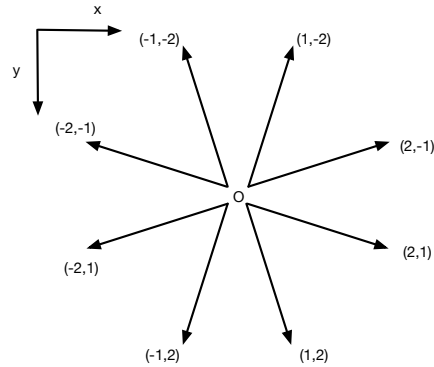


FIGURE 1.2 – Déplacements des Ours.

- les ours : ils se déplacent au hasard à chaque tour selon une des 8 directions suivantes (2,1), (1,2), (-1,2), (-2,1), (-2,-1), (-1,-2), (1,-2), (2,-1) (Fig. 1). Ils attaquent uniquement avec feuille.

Quand un animal arrive au bord de la carte, il se retrouve de l'autre côté. Autrement dit, s'il arrive à la dernière colonne et qu'il va à droite, il se retrouve sur la première (idem pour les lignes). L'initialisation se fait avec 25% du plateau rempli, avec le même nombre d'animaux de chaque type, distribués de façon aléatoire sur la carte.

2 TRAVAIL À FAIRE

Vous devez implémenter les animaux en suivant le diagramme de classe de la figure 2. `deplace` et `setAttaque` sont deux fonctions **virtuelles pures**. `setAttaque` affecte l'attaque avant le combat dans `attaque`. La fonction `resoudreAttaque` de la classe `Attaque` renvoie vrai si l'attaque sur laquelle est appliquée la fonction est "plus forte" que l'attaque passée en paramètre.

Votre fonction principale initialisera un plateau de taille 10×10 qui respecte les contraintes. Tant que l'utilisateur demandera à continuer le jeu (et qu'il reste des animaux vivants sur le plateau) pour chaque tour vous :

- déplacerez l'ensemble des animaux
- résoudrez les conflits
- afficherez sous forme texte le plateau comme dans l'exemple figure 2.

Pour simplifier le problème, vous commencerez uniquement avec des loups et des pierres comme animaux puis vous ajouterez les ours et les lions.

Remarque : pour appeler le générateur de nombre aléatoire vous pouvez utiliser la fonction `random` de la bibliothèque `stdlib.h`. Vous prendrez garde de bien initialiser le générateur de nombres aléatoires en utilisant la fonction `srand` (une seule fois au début du main). Vous pouvez par exemple faire appel à `srand (time(NULL))` ; pour l'initialiser en fonction du

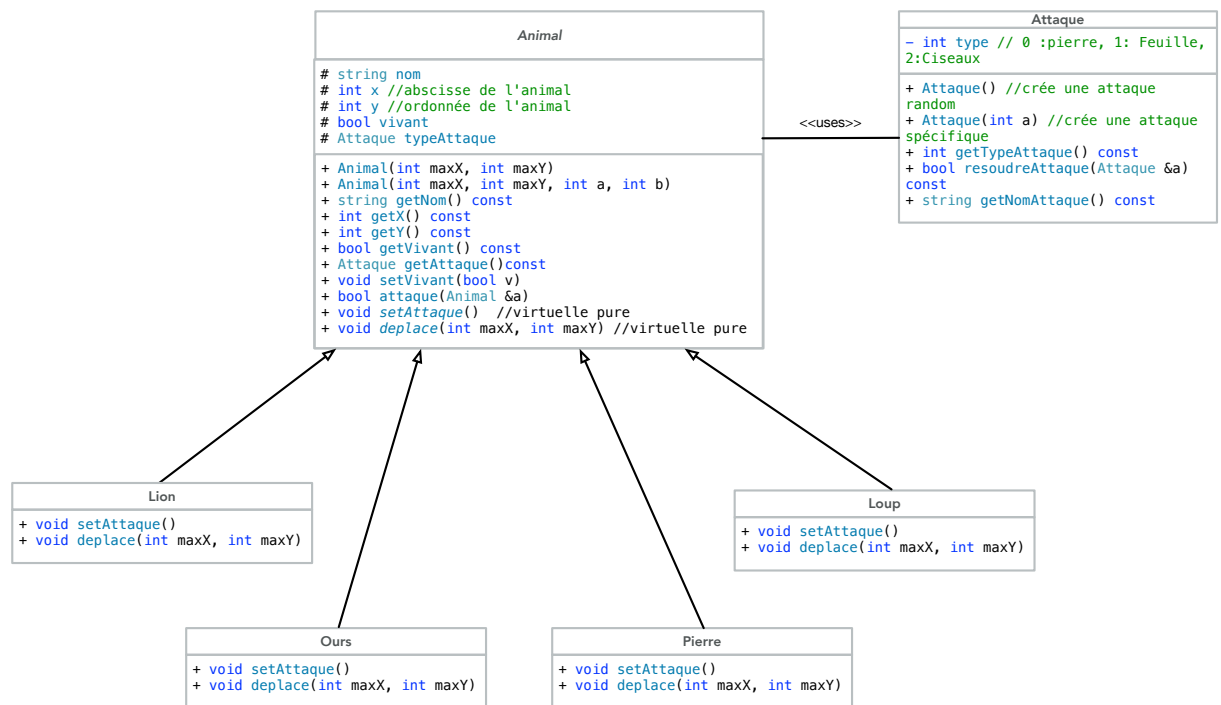


FIGURE 2.1 – Diagramme des classes à implémenter.

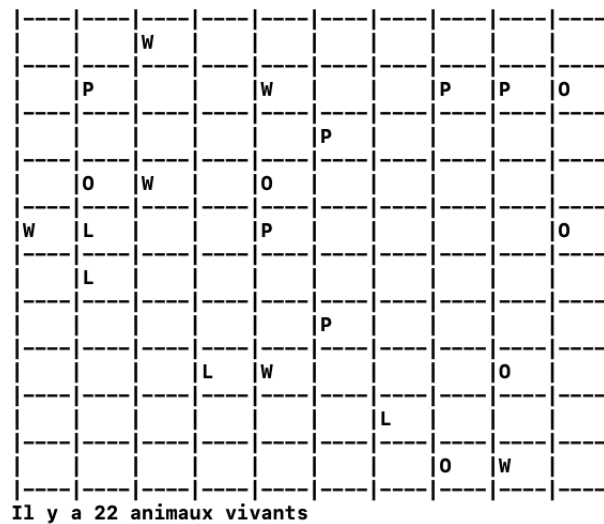


FIGURE 2.2 – Exemple d’affichage du plateau.

temps (en incluant dans ce cas la bibliothèque `time.h`).

Vous déposerez vos comptes rendus sur hippocampus et vos code sur github classroom.