Rapport Microrécif

Daniel Roulin & Joshua Hurliman

1 Hiérarchie de classe des entités

Pour ce projet, nous avons employées la hiérarchie de classes pour les entités de `Lifeform` et un peu de polymorphisme. Nous avons décidé de créer une super classe `Lifeform` qui est parente de trois sous classes: `Algue`, `Corail` et `Scavenger`. Pour ce qui est des attributs, la superclasse `Lifeform` contient un attribut `age` tandis que le reste des attributs sont gérés par les classes dérivées où un autre module. Dans la classe `Algue` se trouve un attribut `cercle` qui est de type `Cercle` définit dans le module shape qui représente ladite forme dans le monde de la simulation. `Corail` contient les attributs `id`, `statut`, `sens_rot`, `st_dev` et `nb_seg`, qui sont les attributs d'un `Corail` demandées dans la donnée, mais aussi `base` qui est un carré et un vecteur de segments, deux autres formes de shape. Enfin, dans `Scavenger` nous trouvons de nouveau un attribut cercle ainsi que les éléments caractérisant les scavengers: `etat` et `id_cible`.

Concernant les méthodes, toutes les classes possèdent un/des constructeur(s) ainsi que des méthodes s'occupant de vérifier les valeurs données par les fichiers. `Lifeform` possède: `test_age` et `test_pos` tandis que le reste des méthodes de vérification sont mises en œuvre dans les sous-classes. Pour ce qui est du reste des méthodes, `Lifeform` possède `update_age` et `erreur_lecture` qui servent respectivement à mettre à jour l'âge des algues (pour l'instant) et de gérer les problèmes liés à la lecture de fichier pour enclencher une ré-initialisation. Les trois sous-classes possèdent une méthode écriture qui renvoie le string à écrire lors d'une sauvegarde de fichier. `Algue` n'a pas d'autre méthode. Quant aux autres, `Corail` a `add_seg` qui gère l'inclusion des segments de `Corail`, tandis que `Corail` et `Scavenger` ont des getteur et setteurs pour certaines de leurs valeurs.

Pour ce qui est du polymorphisme, il y a seulement la fonction `dessin` qui est purement virtuelle et qui est override dans toutes les sous-classes.

2 Structuration des données des autres entités du Modèle

`Simulation` contient d'abord des attributs utiles lors de la lecture de ficher, tel que des compteurs d'entités et des booléen indiquant la présence d'erreurs ou la fin du fichier. Elle contient aussi les variables d'état `naissance_algue` et `nb_sim`, dont la valeur est affichée dans l'interface utilisateur, ainsi que le `random_engine`. Finalement, des vectors `algues`, `corails` et `scavengers` contiennent les différentes entités de la simulation. Nous avons choisi le type vector car le nombre d'élément de chaque tableau n'est pas fixe et est inconnu lors de la compilation.

3 Description des types du module shape

Le module shape définit trois classes : `Carre`, `Cercle` et `Segment`. Elles représentent des formes géométriques dans l'espace de la simulation. Les trois types possèdent un attribut `pos` de type `S2d` qui représente leur position. De plus, `Carre` possède un attribut `side`, `Cercle` un attribut `rayon` et `Segment` des attributs `angle` et `length`, qui caractérisent la forme exacte de chaque objet. Concernant les méthodes, des constructeurs permettent d'initialiser tous les attributs décrit précédemment. En outre, des getteurs et des setters sont définit pour les valeurs utilisées dans d'autre module. Enfin, une méthode `dessin` permet d'afficher chacune des formes, en faisant appel aux fonctions correspondantes du module graphic. Concernant la

classe `Segment`, elle possède aussi des fonctions permettant d'obtenir son extrémité, de faire des tests de collision entre deux segments et d'obtenir l'écart angulaire avec un point ou un autre segment. Finalement, sa méthode `ecriture` renvoie la représentation textuelle du segment. Nous n'avons pas fait de hiérarchie de classe, car les seuls points communs à chaque forme sont leur position et leur méthode dessin. Comme chaque forme possède un point, une super classe `Point` ne conviendrait pas. De plus, elle ne possèderais qu'une seule méthode virtuelle pure dessin, ce que nous trouvions peu logique.