

## Rapport rendu 2

### Hierarchie des classe (citer tout ou juste ceux de lifeform ?)

Dans notre superclasse *Lifeform*, nous avons défini trois méthodes en public : *get\_coord* (retourne les coordonnées x, y d'une entité), *get\_age* (retourne l'âge) et *older* (modifie l'âge par incrémentation d'une unité). Ces trois méthodes sont toutes utilisées par nos classes dérivées : *Algue*, *Corail* et *Scavenger*.

Pour la classe *Algue* :

- Il n'y a pas besoin de méthode supplémentaire, celles héritées de la classe *Lifeform* suffisent.

Pour la classe *Corail* :

- Huits méthodes ont été définies dans public : *get\_seg\_vector*, *get\_nbr\_segments*, *get\_id*, *get\_statut\_cor*, *get\_dir\_rot*, *get\_statut\_dev*, *get\_is\_alive*, *add\_seg\_vector*.
- Dix attributs ont été définis dans private : *seg\_vector*, *id*, *statut\_cor*, *dir\_rot*, *statut\_dev*, *nbr\_segments*, *is\_alive*, *is\_extend*, *color*, *cor\_life*

Pour la classe *Scavenger* :

- Quatre méthodes ont été définies dans public : *init\_corail\_id\_cible*, *get\_rayon*, *get\_statut\_sca*, *get\_corail\_id\_cible*.
- Trois attributs ont été définis dans private : *rayon*, *statut\_sca*, *corail\_id\_cible*.

### Structuration des données des autres entités du modèle

Dans notre classe *Simulation* nous avons pour les trois différents types d'entités, un attribut de type unsigned int contenant son nombre et un vecteur contenant chacun des instances d'une classe d'entité : *Algue*, *Corail* et *Scavenger*.

De plus, pour assurer la bonne exécution de la simulation, nous avons également un attribut *type* de type int et un *TYPE\_lecture* de type enum.

Les données lues lors de la lecture de fichier sont affectées à des variables à portée limitée afin d'être vérifiées et ensuite elles sont stockées dans une instance d'*Algue*, *Corail* ou *Scavenger* qui sont elles-mêmes stockées dans les vecteurs correspondants.

### Brève description des types mis en œuvre dans **shape**

Dans **shape** nous avons une class *Segments* et des méthodes et fonctions. Certaines fonctions vont appeler des fonctions de **graphic** afin de dessiner les entités.

## Répartitions des responsabilités

Eileen (372924) : widgets(open/save), Makefile, lecture/écriture des données, méthode seed()  
(plutôt simulation)

Orjowane (379270) : widgets(start/step), méthode on\_draw, timer, absence de distortion  
(plutôt interface graphique)