

# Sprint review - Sprint 0

## Escape

### Rappel backlog sprint 0 :

- CLI : Victor
- Déplacement du joueur : Théo
- Collisions : Claire
- Première version du moteur graphique : Antoine
- Initialisation du jeu à partir d'un fichier (Mur et Monstre) : Dan

Nous avons pu tout implémenter. Il est ainsi possible de déplacer le héros en CLI. La console affiche l'emplacement du héros suite à ce déplacement.

La première version du moteur graphique fait par Antoine ne propose pas encore l'affichage du Monde, cela sera fait au prochain sprint. Pour l'instant, c'est juste un affichage de "test". En revanche, les fonctionnalités qui nous seront nécessaires sont toutes présentes : changement de vue, gestion des événements clavier, boucle de rendu, mode plein écran, affichage d'image (et mise en cache des images).

### Problèmes rencontrés :

→ Un peu brouillon au début, on aurait dû réfléchir un peu plus aux squelettes des différentes classes directement afin de ne pas être embêté sur des fonctions dont on aurait eu besoin dans plusieurs branches différentes (getter/setter notamment).

Pour le prochain sprint, privilégier des interfaces rendant accessible à tous les méthodes nécessaires ou alors écrire les corps des fonctions (vides) avant de mettre en place les branches liées aux fonctionnalités.

→ Longtemps réfléchi sur comment on devait gérer les collisions dues aux déplacements : vérifie-t-on immédiatement lorsque que le héros bouge, ou vérifie-t-on après que le héros se soit déplacé dans une fonction qui gère toutes les autres collisions et à ce moment-là re-déplacer le héros devant l'élément avec lequel il est en collision.

On a opté pour la 1<sup>re</sup> solution, quitte à changer si on s'aperçoit plus tard, quand on aura d'autres collisions, que la 2<sup>e</sup> solution est plus simple afin de notamment tout centraliser.

→ La mise en place de Maven n'a pas été une mince affaire. En effet, au départ, nous avions une erreur assez peu explicite dès que nous tentions de créer un .jar. Ensuite, la génération de celui-ci produisait une archive qui ne pouvait être lancée. Il a fallu parcourir de multiples forums pour trouver comment créer une archive incluant les dépendances pour que tout fonctionne. Enfin, Maven refusait d'utiliser une version récente de certaines dépendances, produisant des effets inattendus (notamment pour les tests dans le workflow GitHub : il ne voyait pas les fichiers de test), nous obligeant à forcer les dernières versions compatibles dans le pom.xml.

Diagramme de classes général :

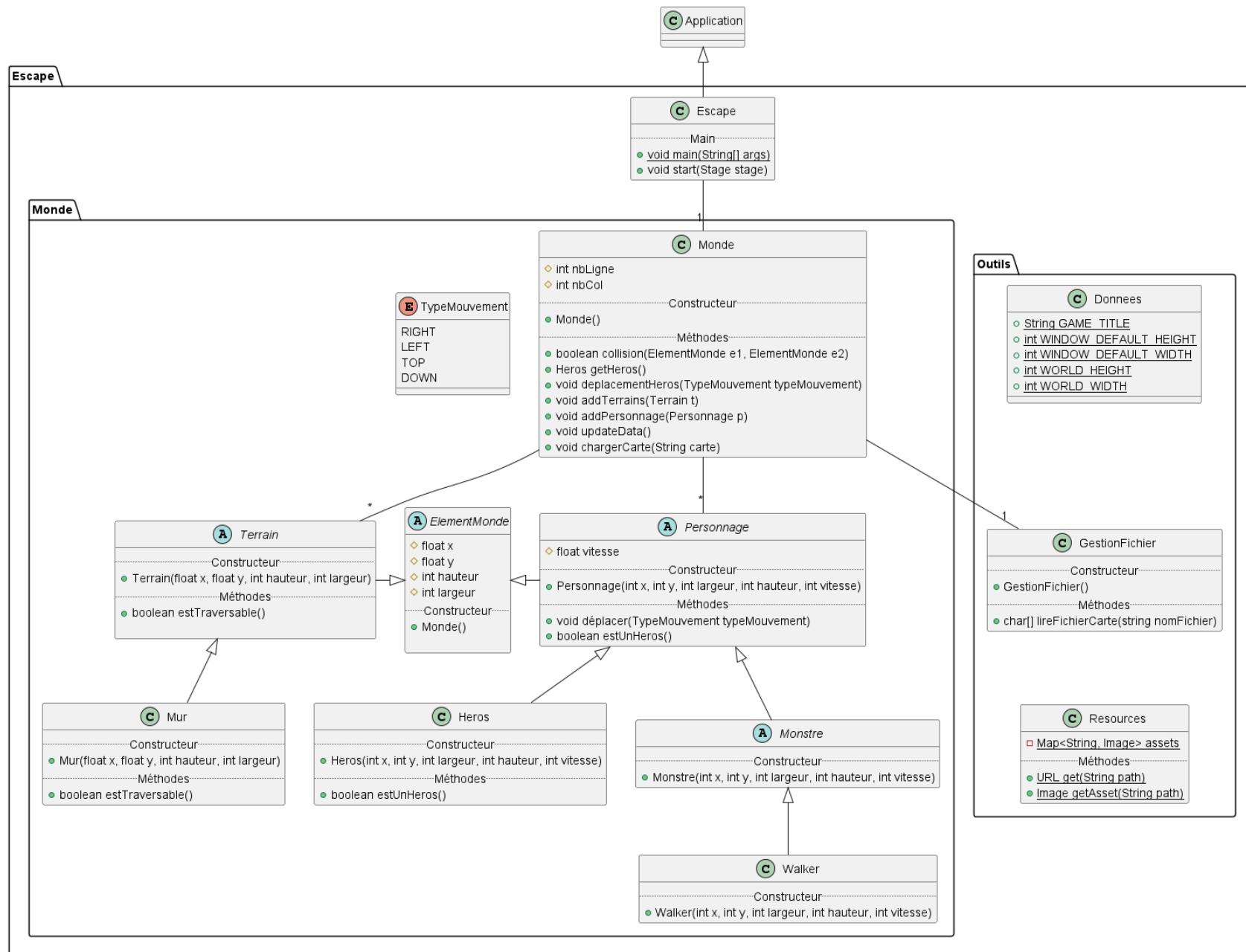


Diagramme de classes de l'interface graphique :

