

## Backlog complet

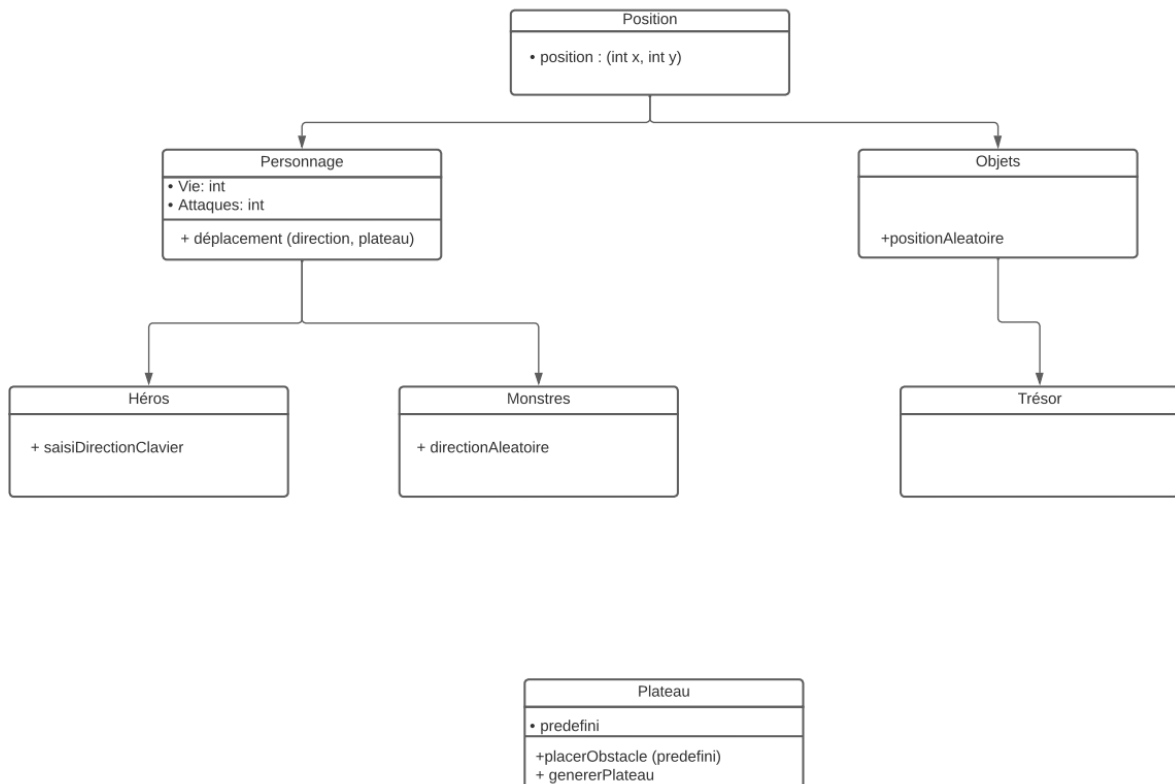
- Un héros est placé sur le plateau de jeu et peu s'y déplacer à l'intérieur.
- Le labyrinthe est généré en fonction du niveau sélectionné. 4) Certains cases du labyrinthe sont spéciales :
  - trésor : si le héros arrive sur la case il a gagné le jeu
  - pièges : quand un personnage arrive sur la case il subit des dégâts
  - magiques : si un personnage arrive sur la case un effet est déclenché
  - passages : un personnage qui arrive sur la case est téléporté à un autre endroit
- Les monstres se déplacent de manière intelligente en essayant d'attraper le héros. Les fantômes sont des monstres qui peuvent traverser les murs.
- Le héros peut attaquer les montres sur la case adjacente.
- Les deux perdent des points de vie.

## SPRINT 1

### Backlog sprint 1

- Générer un labyrinthe avec obstacles par défaut + trésor
- Version texte du déplacement guidé par le terminal qui renvoie la position du héros et celle du monstre. Possibilité pour le perso de pas bouger.
- Créer un monstre qui tue le personnage sans graphisme : déplacement aléatoire + tue le personnage si rencontre

### Diagramme de classe:



### Répartition des tâches

- S41 : Adel (position et personnages)
- S41-42 : Elina (plateau)
- S41-42 : Elie (Héros et monstres)
- S41-42-43 : Thomas (gestion du main, Objet et trésor)

- S44 : retard+bonus
- S45-46 : préparation sprint 2

## Rétrospective Sprint 1

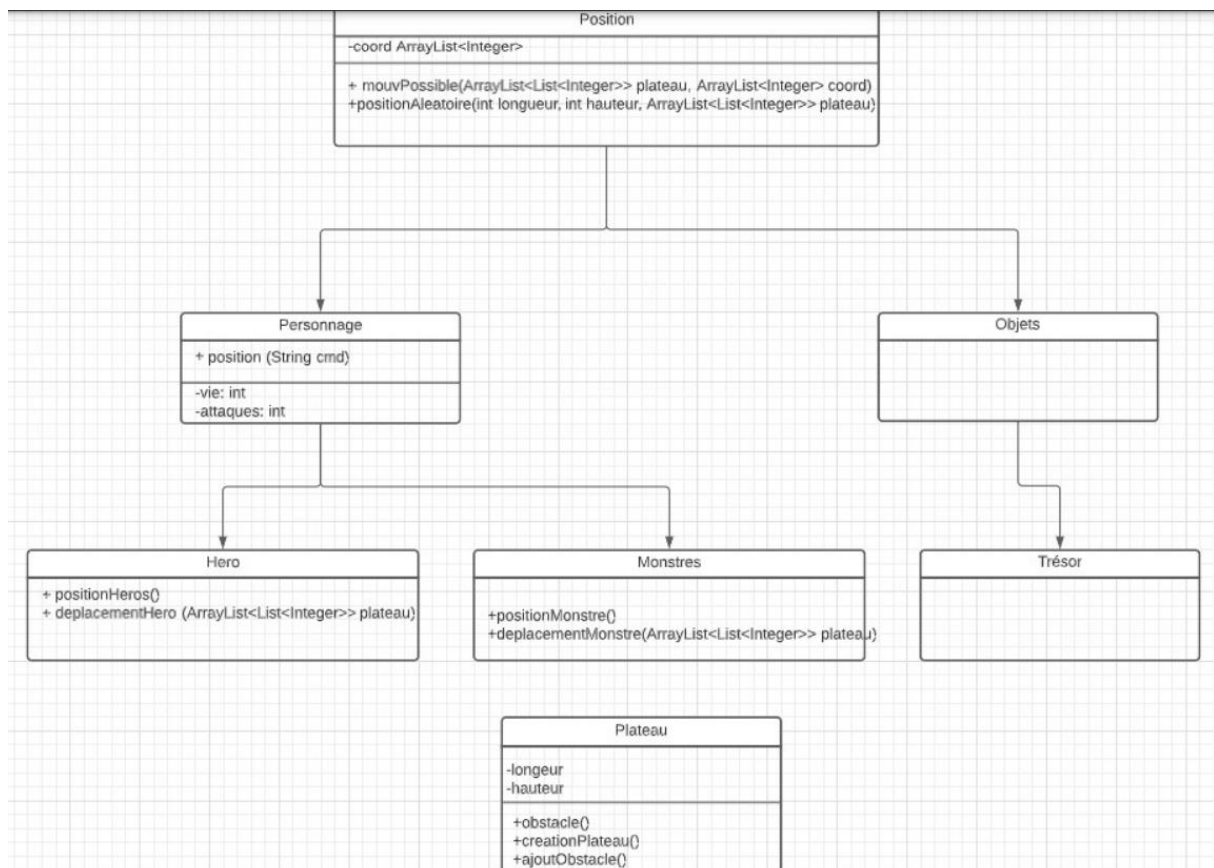
### Observations :

- Labyrinthe et obstacle par défaut ok + trésor aléatoire
- déplacement ok. Si autre touche que D G H B le perso bouge pas et le monstre se déplace quand même
- monstre ok.
- main renvoie position du monstre, trésors, héros.
- diagramme de classe mis à jour
- Trésors et objets pour le moment vide (à compléter plus tard)

### Commentaires prof :

- flèches dans l'autre sens dans les diagrammes
- enlever l'initialisation aléatoire des positions des personnages
- utiliser 2 objets x et y à la place d'un ArrayList coord

### Diagramme de classe:

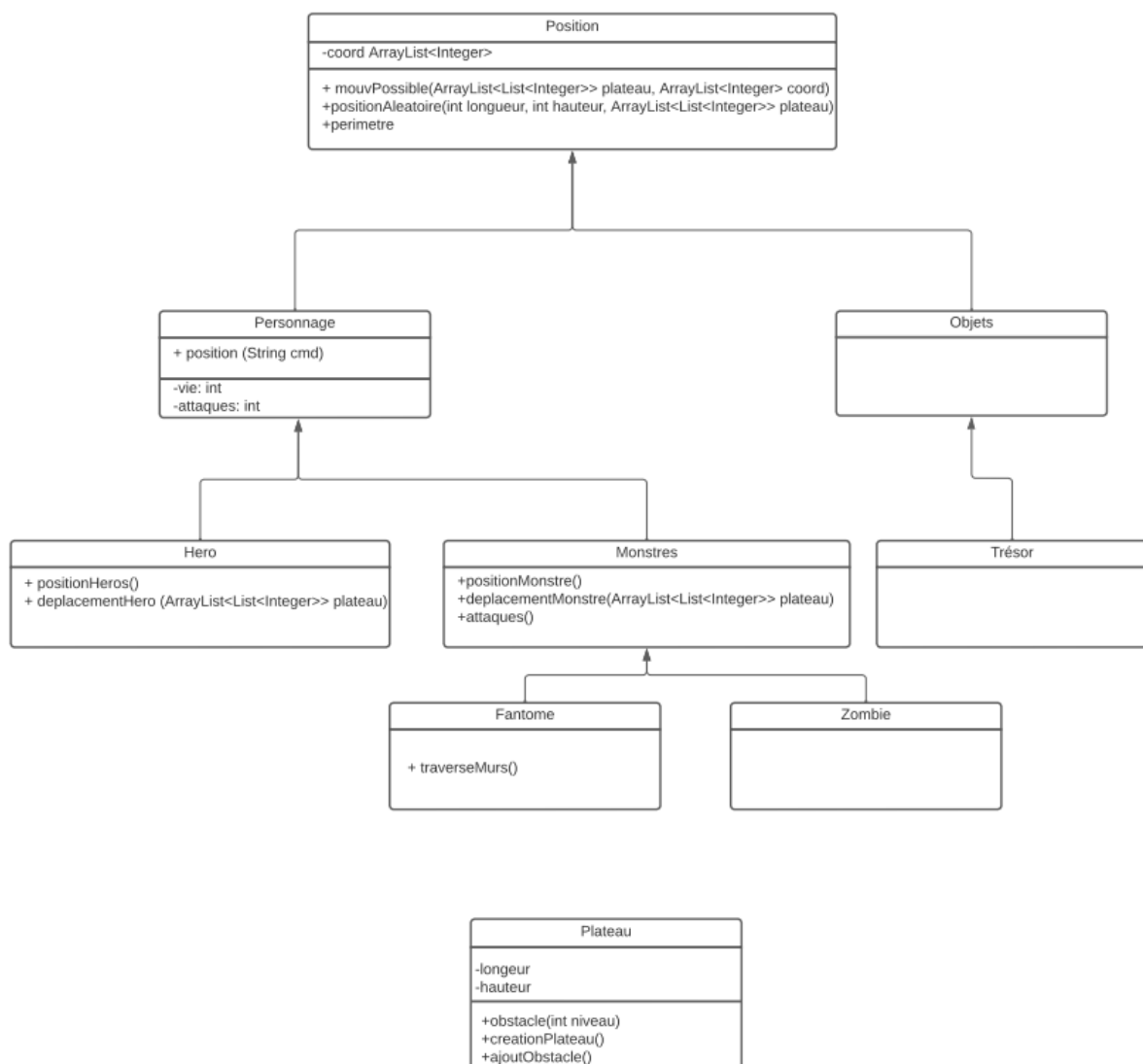


## SPRINT 2

### Backlog sprint 2:

- plateau: obstacle aléatoire + en fonction du niveau (taille fixe, nb d'obstacle qui change)
- faire le graphisme
- faire différents monstres en fonction du niveau (qui attaque et un qui traverse les murs et attaque)
- initialiser la position des monstres de façon aléatoire en respectant un périmètre autour du héros
- prédéfinir la position du héros
- ajouter aussi un périmètre autour du trésor à l'initialisation

### Diagramme de classe:



### Répartition des tâches:

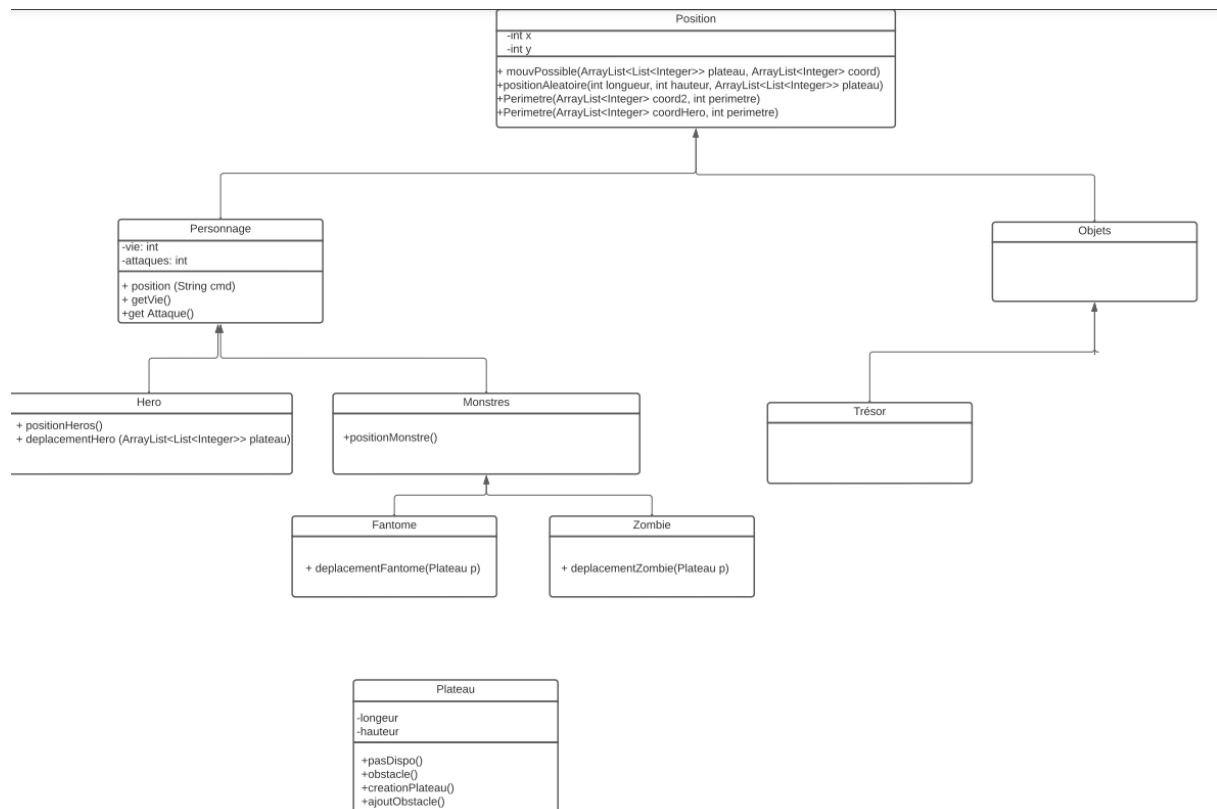
- plateau: Elina
- graphisme: Elie
- périmètre: Adel
- monstre: Thomas

## Rétrospective Sprint 2

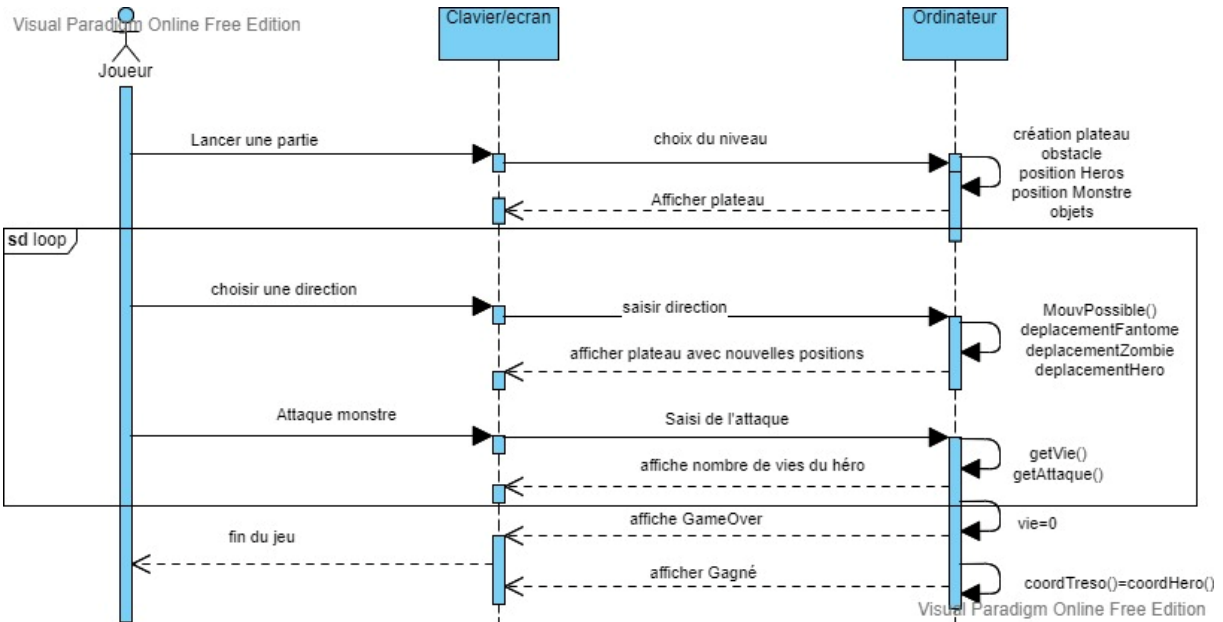
### Observations :

- plateau ok
- graphique ok
- différents monstres sont créés en fonctions du niveau
- pop monstre ok
- périmètres respectés
- changement des coordonnées passer d'une liste à x et y

### Diagramme de classe:



### Diagramme de séquence :



## SPRINT 3

### **Backlog:**

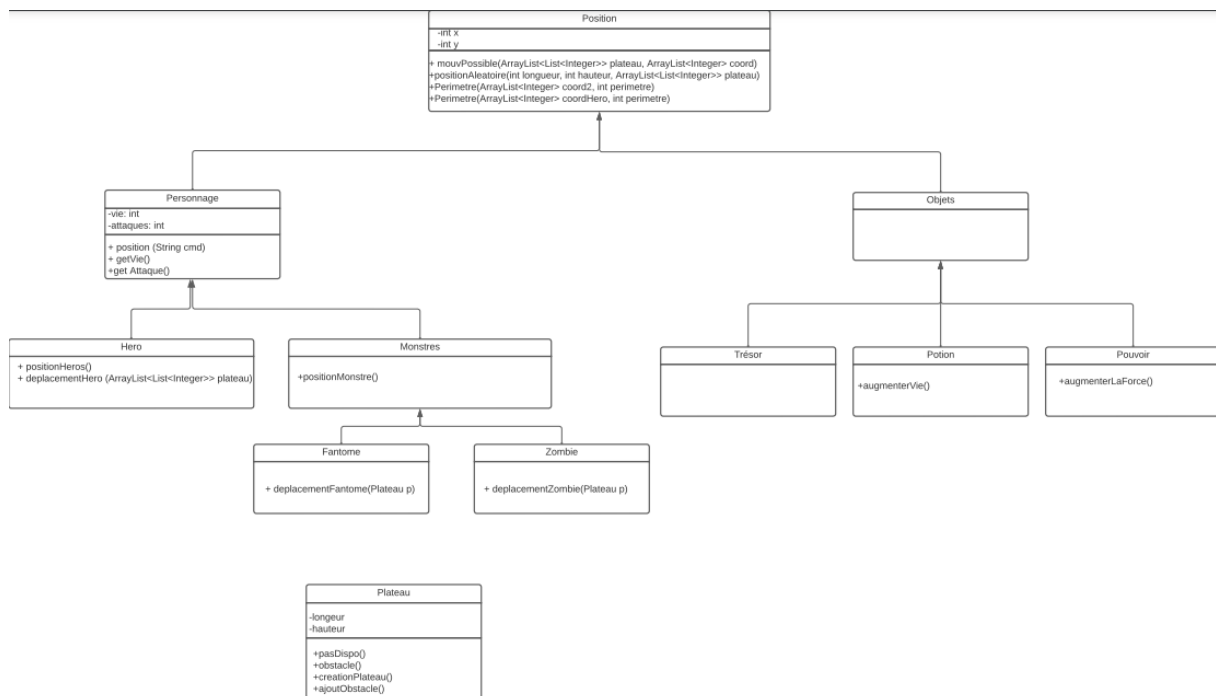
- si un monstre meure, un nouveau monstre apparait
- trouver comment insérer des images à la place des couleurs
- trouver les images pour améliorer le graphisme
- créer des cases avec téléportation, un pouvoir supplémentaire (le héros peut attaquer le monstre), vie supplémentaire pour le héros

### **Répartition des tâches:**

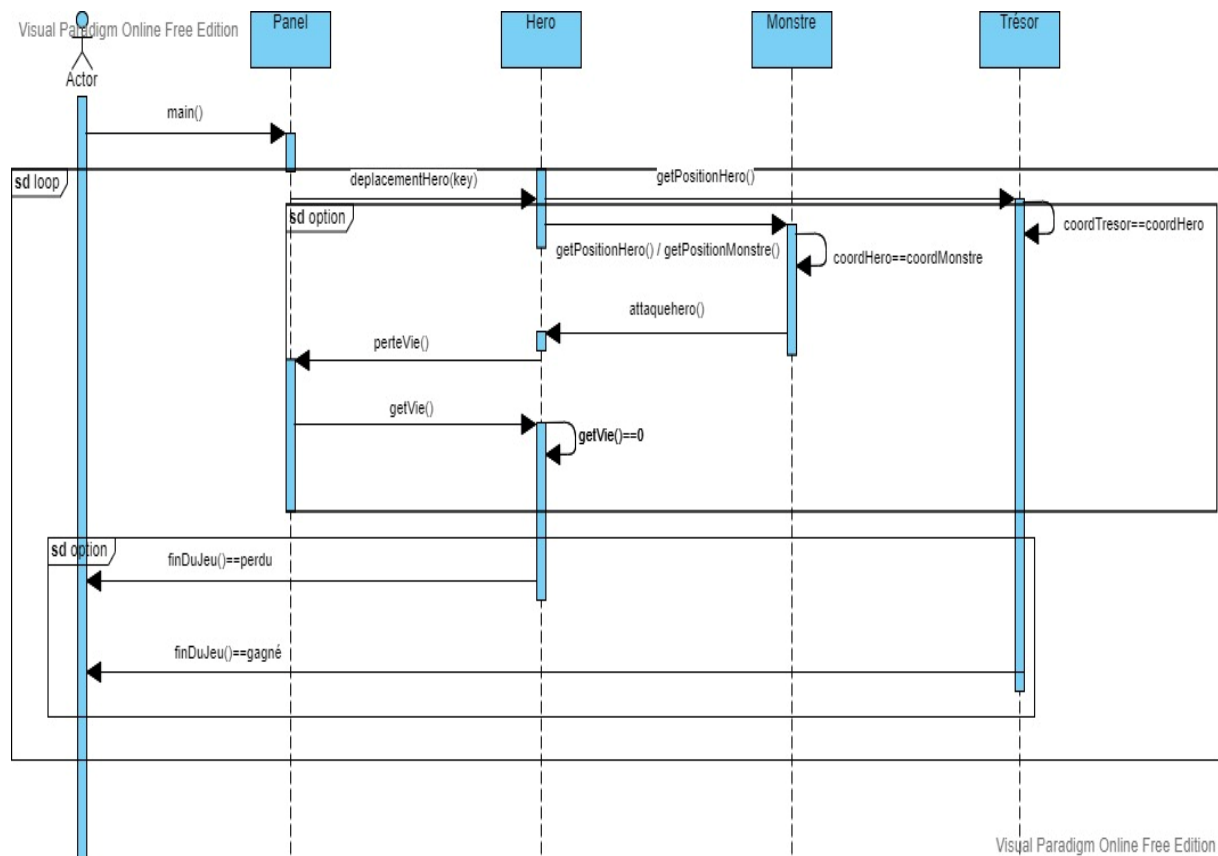
- nouveau monstre: Thomas
- trouver comment insérer des images: Elie
- trouver les graphismes: Elina
- téléportation, attaque supp, vie supp: Adel

RQ : pas beaucoup de temps + sprint 4 encore plus court attention !!!

### **Diagramme de classe :**



### Diagramme de séquence :

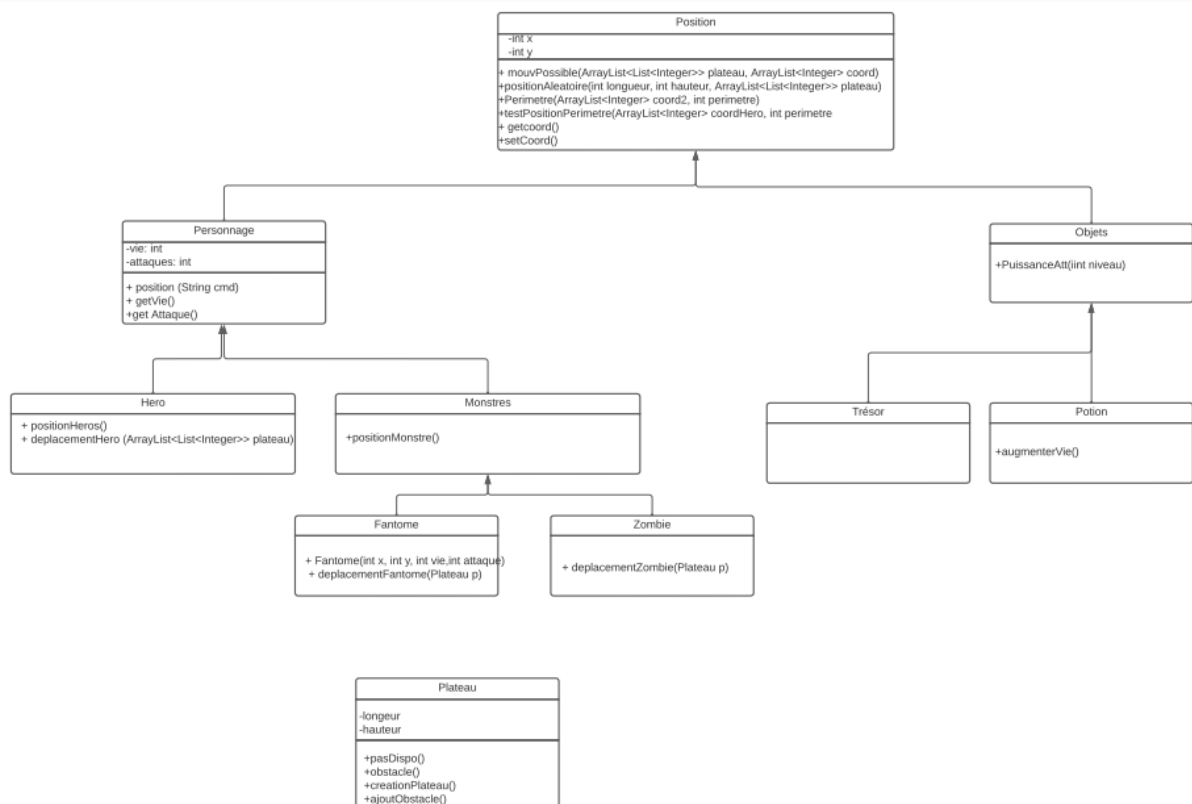


## Rétrospective Sprint 3

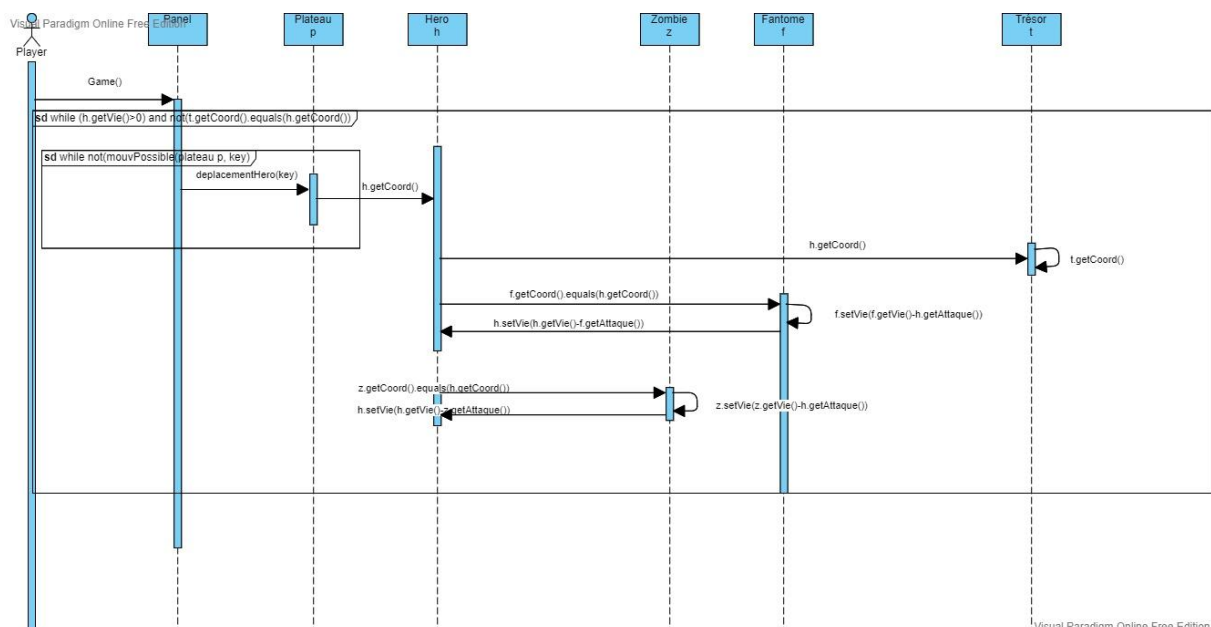
### Observations :

- si un monstre meure, un nouveau monstre apparait ok
- trouver comment insérer des images à la place des couleurs ok
- trouver les images pour améliorer le graphisme ok
- créer des cases avec téléportation, un pouvoir supplémentaire (le héros peut attaquer le monstre), vie supplémentaire pour le héros. OK

### Diagramme de classe:



### Diagramme de séquence :





## SPRINT 4

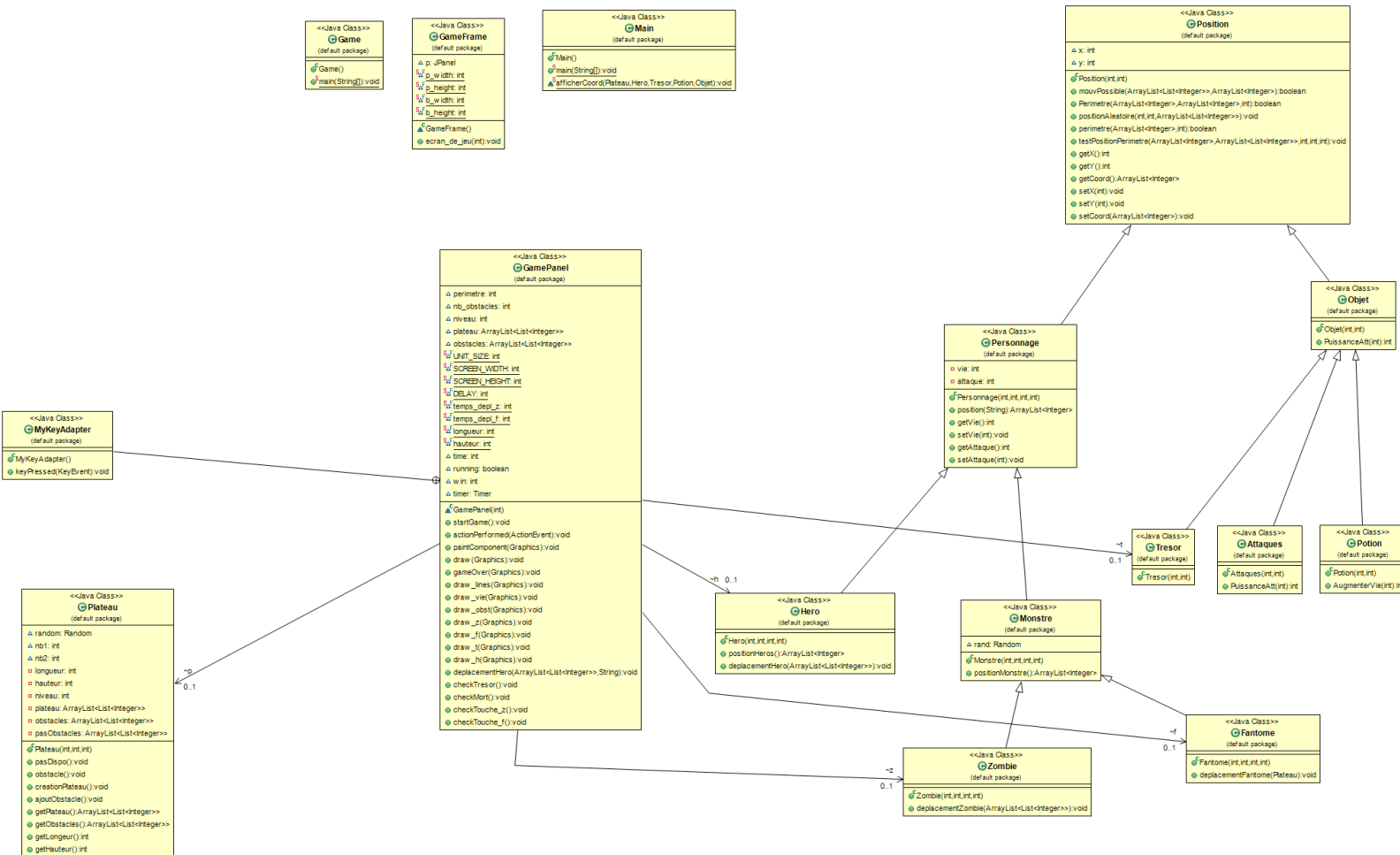
### Backlog:

- graphisme des nouvelles fonctions et intégration des images
- JUNIT
- Soutenance + Diagrammes

### Répartition des tâches:

- Elina : soutenance + diagramme
- Thomas et Adel : Tests JUNIT
- Elie : Graphisme

### Diagramme de classe :



## Diagramme de séquence :

