**Document de Conception**

**ACL 2019 – JeuTropBien**

LANGEVIN Titouan (TitooLang)  
PASDELOUP Rémi   
HUAN Quentin   
RICHARD Raphaël (despokin)

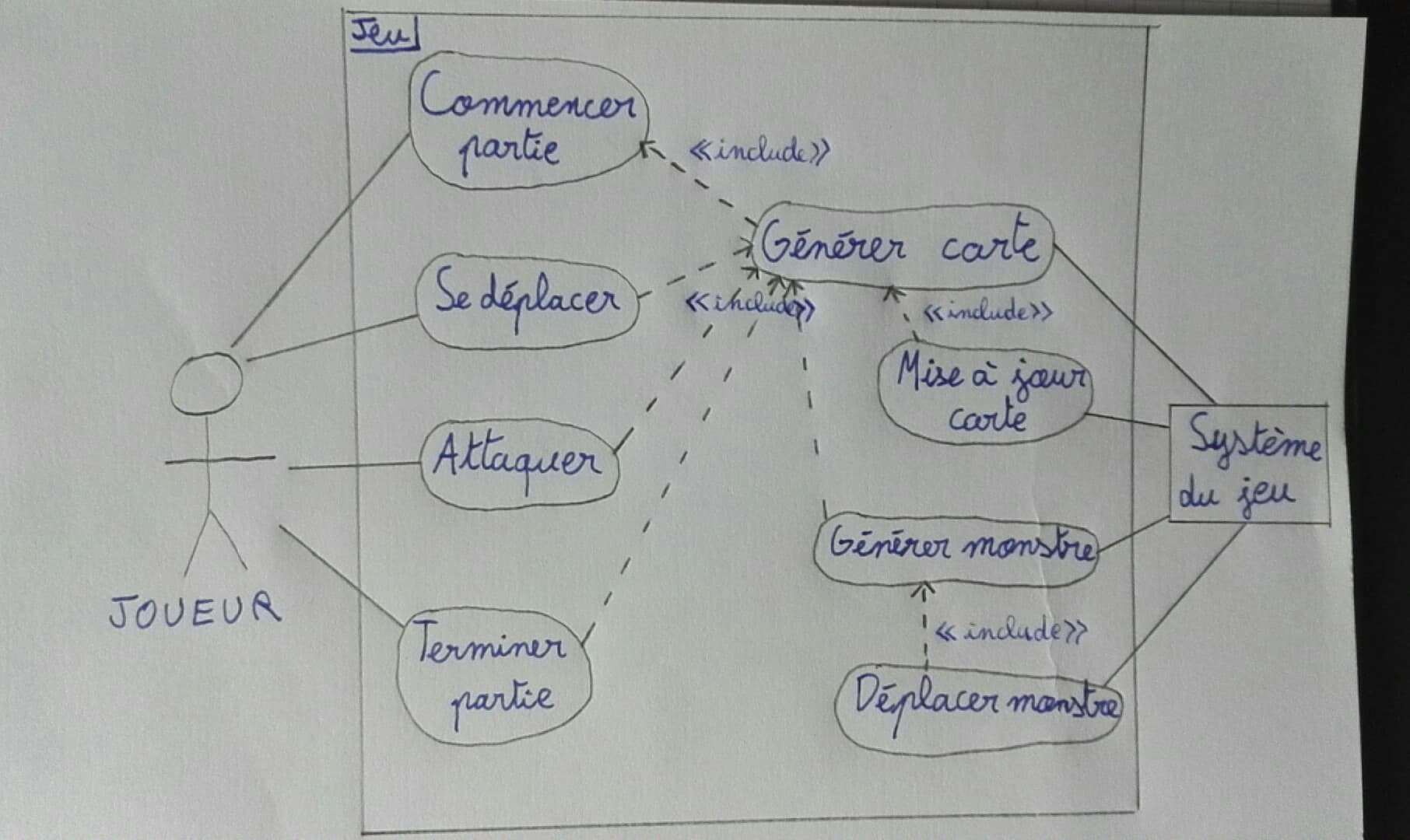
**Liste des backlogs :**

Essentiel :

* Accéder au menu
* Mettre à jour l’affichage graphique
* Démarrer une partie
* Se déplacer
* Attaquer
* Collision avec les murs
* Générer le terrain par case avec un fichier de graphismes
* Collision avec un monstre
* Déplacer les monstres
* Créer le héros
* Générer un monstre
* Écran de victoire
* Écran de défaite
* Retourner au menu

Optionnel :

* Suivre le personnage avec la caméra
* Jouer des sons et de la musique
* Récupérer le coffre aux trésors
* Faire des cases pièges
* Créer différents types de monstres
* Gérer les paramètres du jeu
* Ajouter des items
* Ajouter des sorts
* Améliorer l’IA

**Diagramme de cas d’utilisation :**

**Scénarios :**

* 1. **deplacer**

Cette méthode permet de déplacer un personnage à partir de sa vitesse, sa direction et de la carte. Elle est appelée en continue durant le jeu pour chaque Personnage.   
On suppose que la vitesse V est faible.

Date de création : 02/11/2019

Version : V.0.2

Acteur : Système

Responsable : Langevin Titouan

Précondition : l’objet Personnage est créé

Postcondition : les coordonnées du Personnage sont modifiées ou non

Description par étape :

Scénario nominal :

1. On appelle contactCarte() pour le Personnage
2. On test contactCarte()
3. Si la méthode renvoie *false*, on récupère les coordXY (X et Y), la vitesse (V) et la direction (DX et DY) du Personnage
4. On définit les nouveaux coordonnées du Personnage (X,Y) = (X+V\*DX, Y+V\*DY)
5. On appelle la fonction contactPers() si le Personnage est un Monstre.

Scénario alternatif :

A1) contactCarte() renvoie *true.*Ce test a lieu à l’étape 2, on passe alors directement à l’étape 5 du scénario nominal.

Scénario exceptionnel : aucun

* 1. **Générer la carte (méthode generer)**

Créé un tableaux bidimensionnel d'entier représentant la carte à partir d'un fichier .csv

date de création: 02/11/19

version: v0.2

acteur: système

responsable: Quentin Huan

pré condition:

Le fichier .csv nommé "carte.csv" est placé à la racine du projet. Il s'agit d'un fichier tableur, de N cellules de large et M cellules de haut. Chaque cellule contient un entier positif.

Le tableau nommé "donnees", membre de la classe Carte, est initialisé: il s'agit d'un tableau de N\*M entiers tous nul.

post condition:

Le tableau nommé "donnees", est remplis grâce au fichier carte.csv:

description par étape:

Scénario nominal :

1) on charge le fichier carte.csv

2) Pour[0<=i<N et 0<=j<M], à chaque élément d'indice (i\*N + j) du tableau, on assigne la valeur de la cellule d'indice (i,j) du fichier carte.csv

3) on renvoie le tableau d’entier ainsi généré

Scénario Alternatif :

A1) Si le fichier est vide ou qu’il n’existe pas, alors on renvoie un tableau de taille N\*M dont tout les éléments sont nuls.

A2) Si le fichier ne respecte pas le format demandé en précondition, alors on renvoie un tableau de taille N\*M dont tout les éléments sont nuls.

Scénario exceptionnel  : aucun

* 1. **contactPers**

Cette méthode a pour but de détecter s’il y a un contact entre 2 personnages.

Date de création : 02/11/19

Version : V0.2

Acteurs : Le système

Responsable : Pasdeloup Rémi

Pré condition : 2 personnages qui existent

Post condition : résultat de la méthode : boolean

Description par étape :

Scénario nominal :

1. La méthode va en premier lieu récupérer les coordonnées des 2 personnages

2. Elle calcule la distance D entre ces derniers.

3. Ensuite, la méthode va tester si la somme de seuilContact de chaque personnage (S1+S2) est inférieure ou non à D.

4. Si cette somme est inférieure à D, alors la méthode va retourner True. Cela signifie qu’il y a bien contact entre les 2 personnages.

Scénario alternatif :

A1) Test (S1+S2<D) renvoie False.

Si le test de l’étape 3. renvoie False, alors l’étape 4. devient :

4.La méthode retourne False. Il n’y a pas contact entre les 2 personnages.

Scénario exceptionnel : aucun

**4) attaquer**

Cette méthode permet de résoudre l’attaque du héros.

Date de création : 02/11/2019

Version : V.0.2

Acteur : joueur

Responsable : Raphaël RICHARD

Précondition : Le joueur presse la touche d’attaque.

Postcondition : La liste des monstres est modifiée (des monstres peuvent avoir été éliminés ou avoir perdu des points de vie).

Description par étape :

Scénario nominal :

Pour chaque monstre :

1)  la méthode “ toucher(Monstre)” est appelée. Elle renvoie le booléen *true* si le monstre est à portée et *false* sinon.   
2) Si “toucher(monstre)” renvoie *true*, les points de vie sont actualisés.

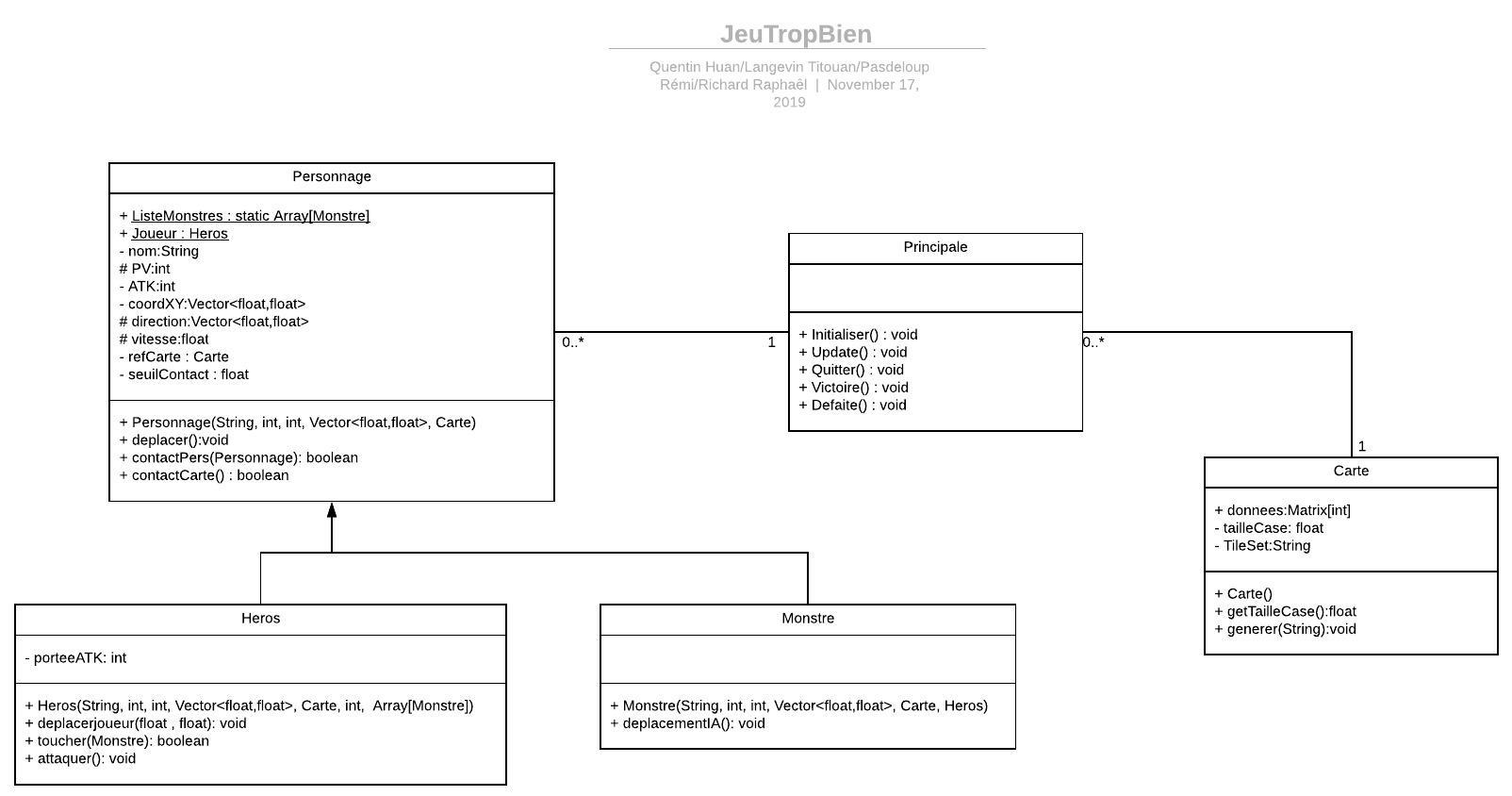
Scénario alternatif :

 A0) pas de monstre   
1) On ne fait rien.

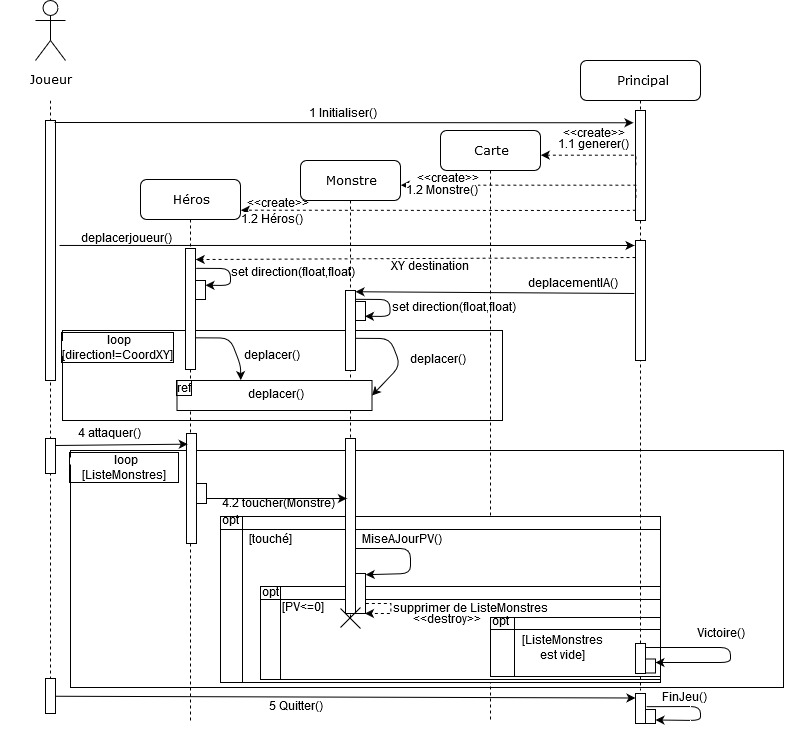
 A1) le monstre testé n’est pas à portée   
Cet enchainement démarre au point 1 du scénario nominal   
2) Il ne se passe rien. (On passe au test du monstre suivant)

  A2) Les points de vie du monstre tombent à zéro (ou négatif)   
Cet enchainement démarre au point 2 du scénario nominal.     
3) Si les points de vie d’un monstre tombent à zéro, le monstre est éliminé de la partie.    
4) Si c’était le dernier monstre en vie, on appelle la méthode “victoire()” qui met fin à la partie.

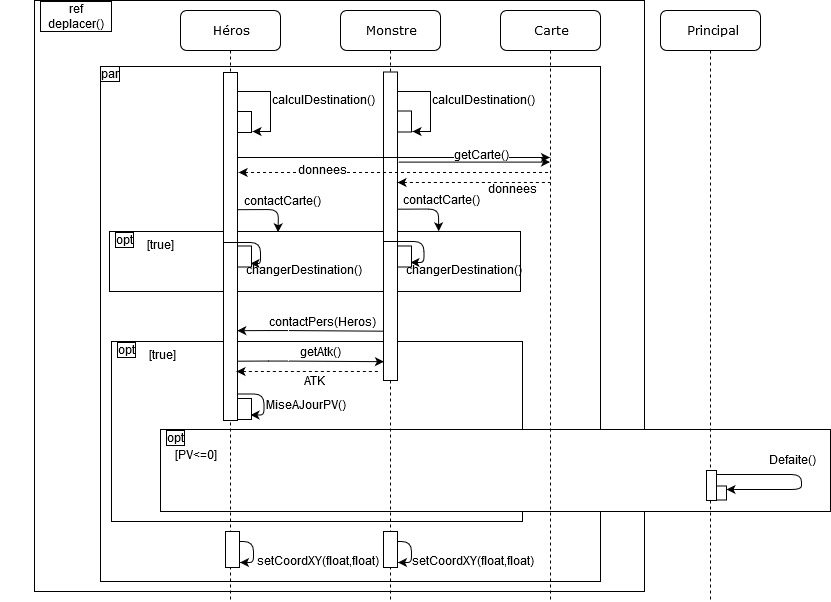
               Scénario exceptionnel : aucun

**Diagramme de classe :**

**Diagramme de séquence :**

Diagramme global :

setPV()

Diagramme ref deplacer() :

**Liste des Sprints :**

Sprint 1 :

* Afficher une carte simple avec 4 murs
* Démarrer une partie
* Déplacer le héros
* Collision avec les murs