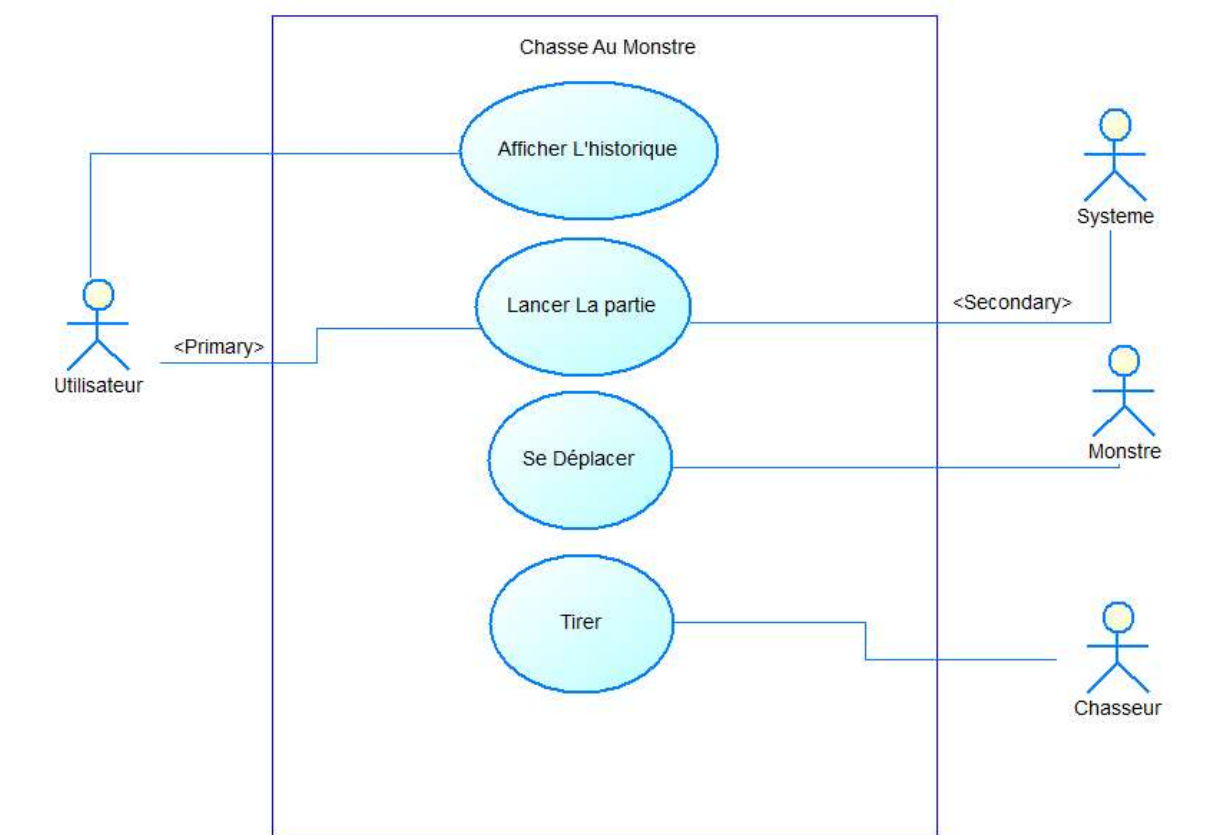


## SAE3.02 : Analyse Chasse au Monstre

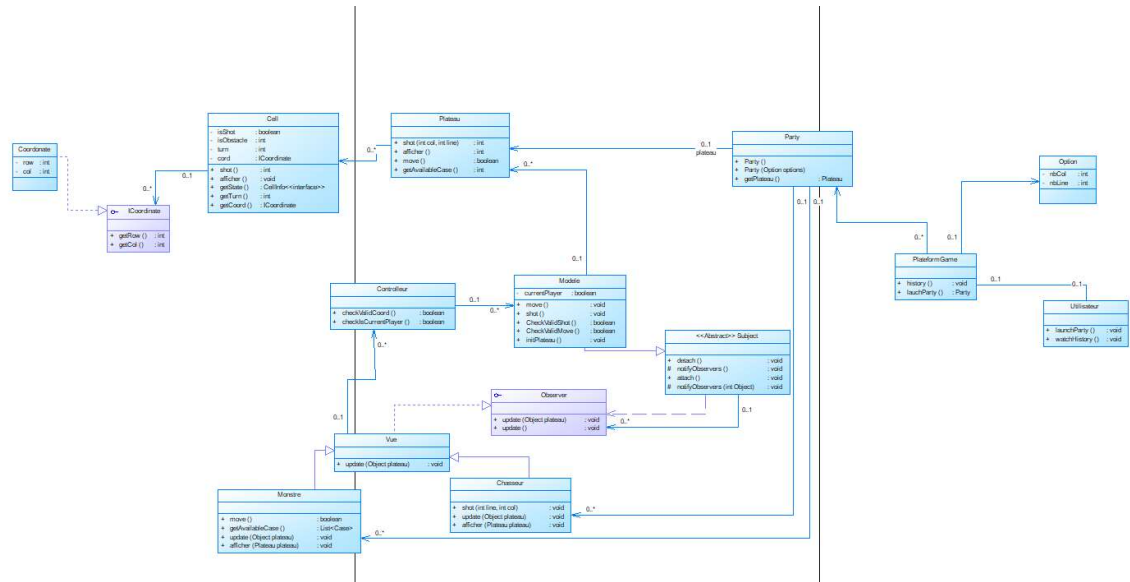
### Groupe K2 :

Ezechiël Leibovici  
Lonny Fauchoit  
Julien Desrumaux  
Xavier Moyon

### Diagramme de Cas D'utilisation :



### Diagramme de Classes :



## Description de l'implémentation pour les fonctionnalités principales (Déroulement du partie classique):

### Lancement d'une partie :

L'utilisateur appelle la méthode launchParty, cette dernière appelle cette même méthode dans PlateformGame qui va alors créer une partie et l'ajouter à sa liste de partie.

Le constructeur de Party va alors créer un plateau en fonction des parametres par défaut.

Le modele va ensuite abonner Chasseur et Monstre via la méthode attach

Ensuite le modele ajoute le plateau via la méthode initPLateau, cette dernière va utiliser la méthode notifyObservers permettant aux vues de déclencher la méthode update permettant de déclencher l'affichage du plateau pour les Vues (Monstre et Chasseur).

### Jouer un tour (fonctionnalités bouger/tirer ):

La partie commence, le joueur qui commence est le monstre, il utilise la méthode move qui va appeler les methodes CheckCurrentPlayer et ChackValidCoord du Controleur qui vont vérifier que le joueur qui tente de modifier le modele est bien le joueur courant et que les coordonnées se trouvent bien dans le plateau.

Si ces deux conditions sont réunis le contrôleur appelle la méthode CheckValidMove qui vérifie que le monstre peut bien se déplacer sur une case, si c'est le cas alors le modele modifie l'état de la case et appelle la méthode move qui va alors appeler la méthode notifyObservers appelant les méthodes update des deux vues permettant de changer l'affichage des vues via les méthodes afficher.

Ensuite le joueur courant change et le c'est au tour du chasseur.

Le monstre utilise la méthode shot qui va appeler les methodes CheckCurrentPlayer et ChackValidCoord du Contrôleur qui vont vérifier que le joueur qui tente de modifier

le modele est bien le joueur courant et que les coordonnées se trouvent bien dans le plateau.

Si ces deux conditions sont réunies le contrôleur appelle la méthode `CheckValidShot` qui vérifie que le chasseur peut bien tirer sur une case, si c'est le cas alors le modele modifie l'état de la case et appelle la méthode `move` qui va alors appeler la méthode `notifyObservers` appelant les méthodes `update` des deux vues permettant de changer l'affichage des vues via les méthodes `afficher`.