

# Projet JavaFX

## DOCUMENTATION

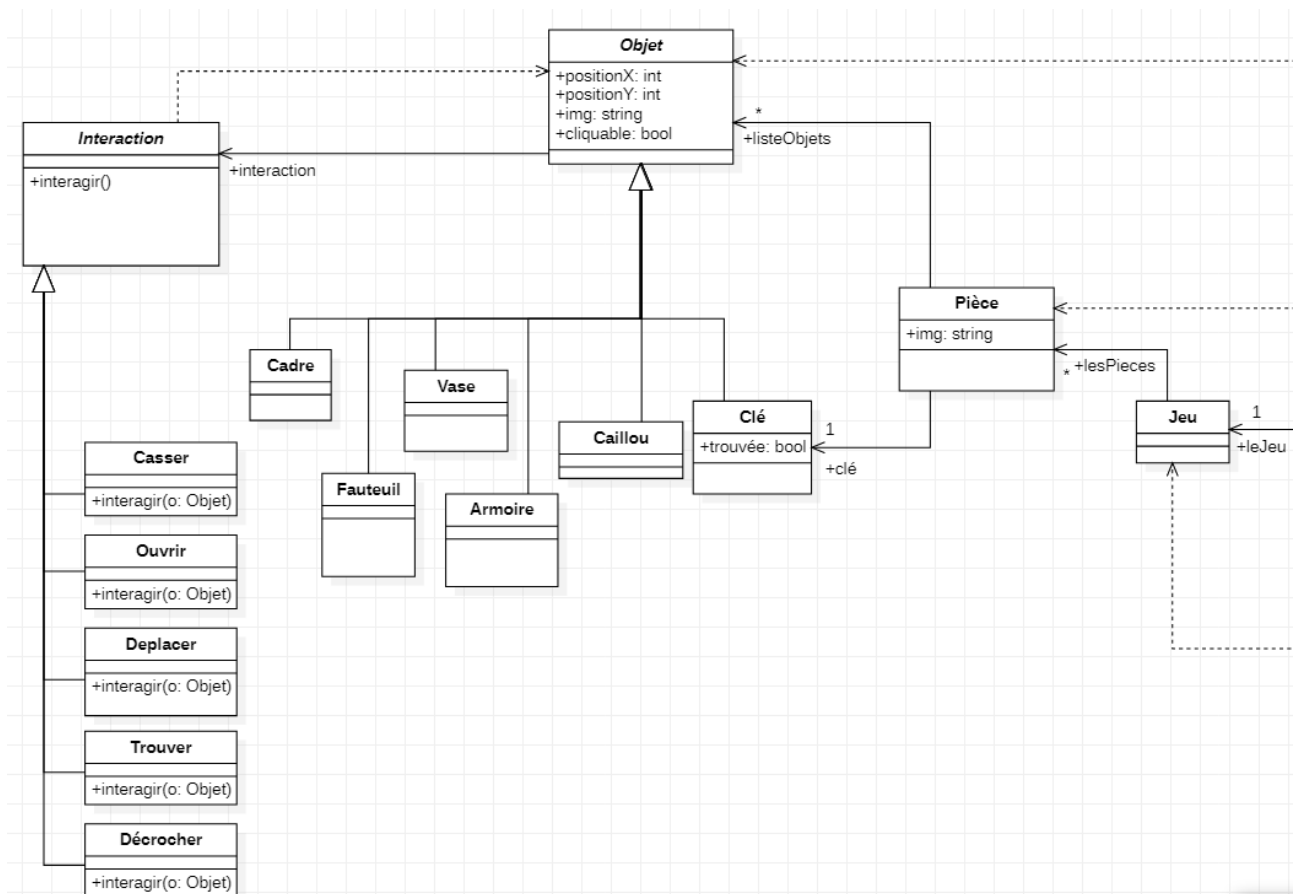
### Contexte

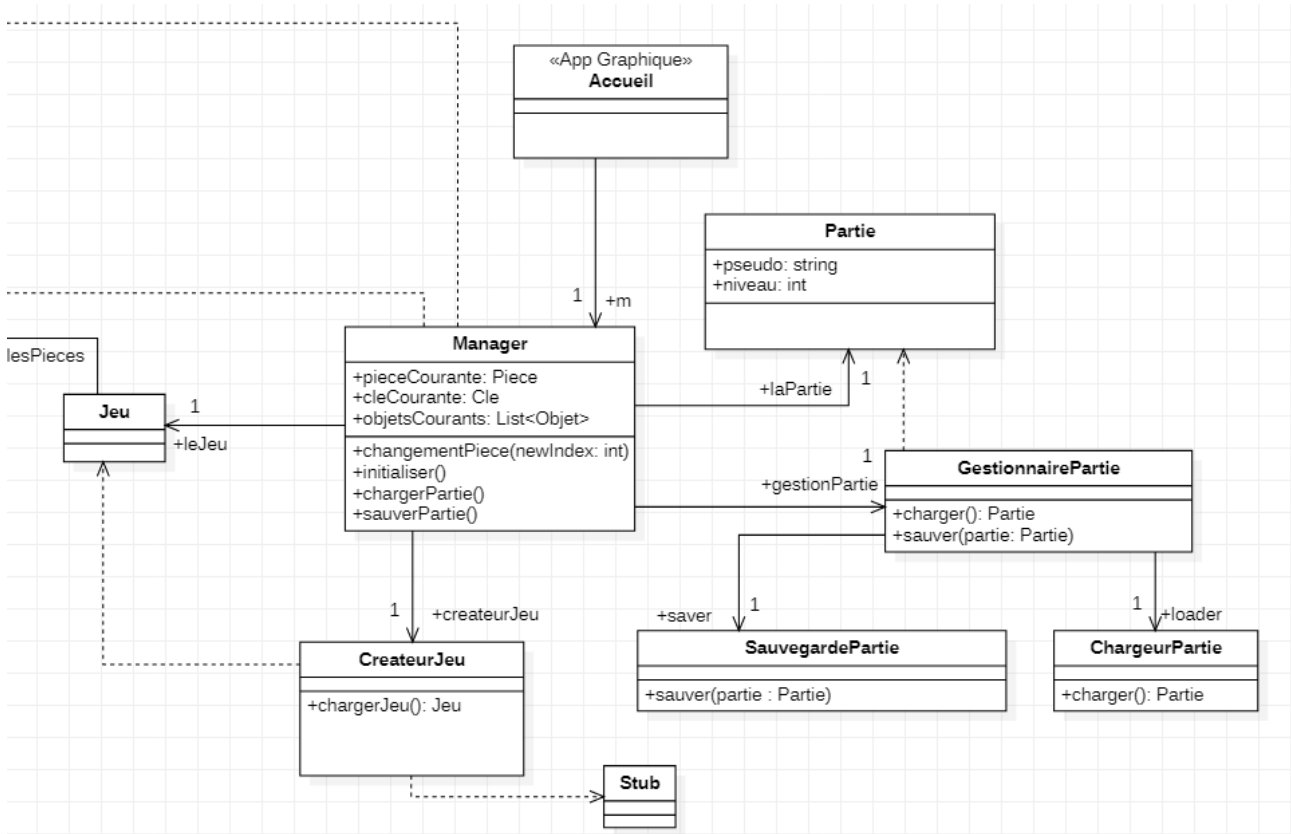
Vous vous réveillez difficilement dans un environnement que vous ne reconnaissez pas. Vos paupières sont encore lourdes, votre crâne vous fait souffrir et votre mémoire n'est clairement plus ce qu'elle était. Comment êtes-vous arrivé là ? Aucune idée. La dernière chose dont vous vous souvenez, c'est de vous être endormi tranquillement dans votre lit la nuit dernière. Ou était-ce celle d'avant ?

Vous reprenez peu à peu vos esprits et parvenez à vous lever. Vous êtes dans une pièce que vous n'avez jamais vue. Devant vous se dresse une grande porte rouge. Vous vous en approchez et tentez de l'ouvrir mais elle semble verrouillée. Vous regardez aux alentours et apercevez un caillou un peu bancal. En le soulevant, vous découvrez une clé cachée dessous. Vous vous en emparez et ouvrez la porte. Mais derrière cette porte, c'est une autre pièce que vous découvrez. Le cauchemar continue. Parviendrez-vous à sortir de ce manoir infernal ?

Notre projet est un jeu d'énigmes qui consiste à interagir avec des objets d'une scène afin de débloquent la scène suivante. Le joueur peut cliquer sur des objets afin d'interagir avec eux dans le but de trouver une clé qui lui permettra d'ouvrir la porte pour atteindre la scène suivante. Le joueur gagne et termine le jeu lorsqu'il parvient à résoudre la dernière scène.

### Diagramme de classes





Le jeu est composé de plusieurs pièces. Chaque pièce contient des objets (une clé et des meubles) et chaque objet a une interaction définie. Les classes abstraites *Objet* et *Interaction* représentent des points d'extension puisqu'il est aisé d'ajouter de nouveaux objets ou de nouvelles interactions sans avoir à modifier du code existant.

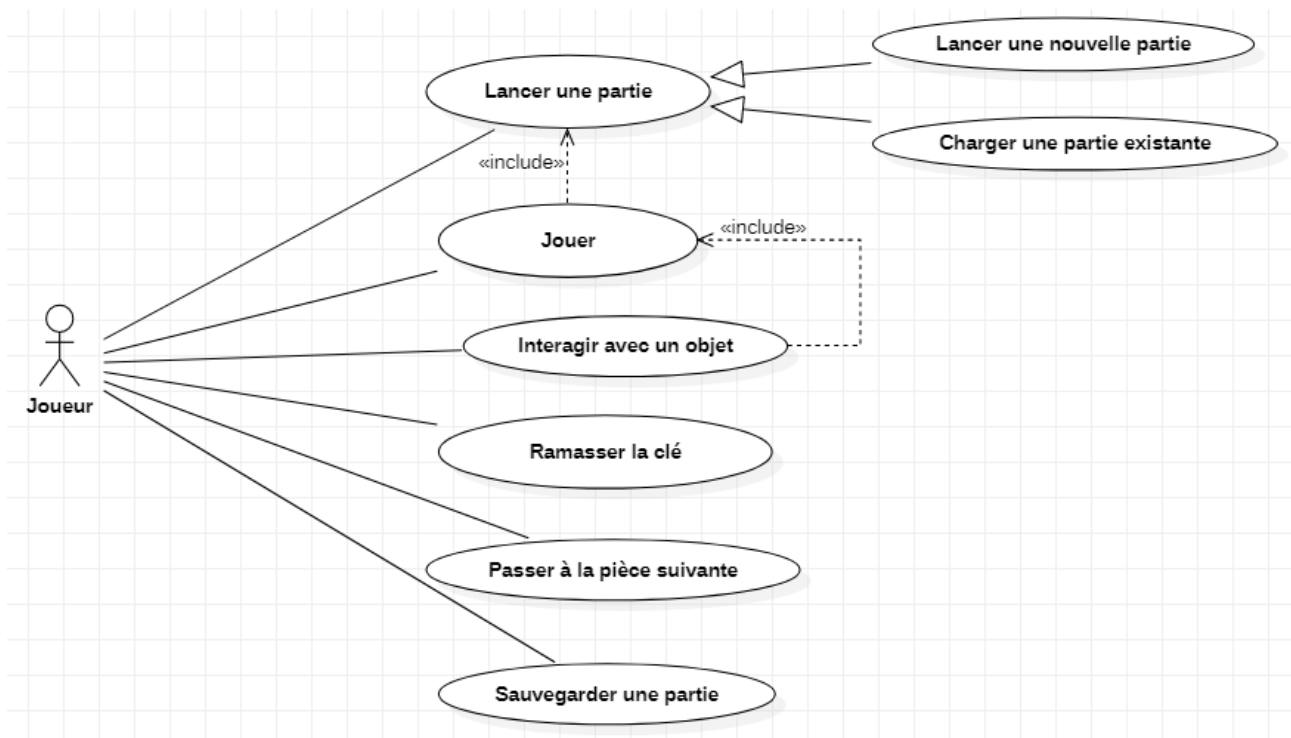
La dépendance entre *Objet* et *Interaction* permet de choisir à l'exécution le comportement d'un objet. On pourrait donc décider de déplacer les vases ou de les casser en changeant simplement l'instance d'interaction qui est attribut des vases plutôt que de devoir modifier une méthode.

La classe *Jeu* joue le rôle de porte d'entrée sur le modèle. Ainsi, le manager n'a besoin de connaître qu'une seule instance de *Jeu* et non pas chaque objet de chaque pièce, qu'il peut récupérer via le jeu qu'il connaît.

Le manager est le seul objet contenu dans l'application graphique. Il se gère de retourner les données nécessaires et de déléguer les intentions métier. Par exemple, il appelle le gestionnaire de partie pour charger une partie, sans le faire lui-même. Ce gestionnaire se charge donc d'appeler soit le chargeur de partie, soit la sauvegarde de la partie afin de gérer la sérialisation de la partie.

Le manager a également un attribut instance de *CreateurJeu* qui permet de charger une instance de *Jeu* à partir d'un fichier binaire (ou d'un *Stub* si le fichier cause une erreur).

## Diagramme de cas d'utilisation



### Cas n°1 : Lancer une partie

Ce cas est divisé en deux cas (cas n°1.1 et cas n°1.2) décrits ci-dessous :

#### Cas n°1.1 : Lancer une nouvelle partie

- Acteur : Joueur
- Condition d'utilisation : aucune
- Description : Un joueur peut lancer une nouvelle partie en lançant le jeu.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur clique sur « nouvelle partie ».
  - Le système affiche la vue de nouvelle partie.
  - Le joueur entre son pseudo.
  - Le joueur clique sur « jouer ».
  - Le système lance une nouvelle partie.

#### Cas n°1.2 : Charger une partie existante

- Acteur : Joueur
- Condition d'utilisation : une sauvegarde doit exister
- Description : Un joueur peut charger une partie commencée précédemment et sauvegardée.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur clique sur « charger une partie ».
  - Le système lance le jeu là où le joueur avait sauvegardé sa partie.

Cas n°2 : Jouer

- Acteur : Joueur
- Condition d'utilisation : le joueur doit avoir lancé une partie
- Description : Un joueur peut jouer au jeu.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur joue.

Cas n°3 : Interagir avec un objet

- Acteur : Joueur
- Condition d'utilisation : le joueur doit être en train de jouer et ne pas avoir interagit avec l'objet auparavant.
- Description : Un joueur peut interagir avec les objets d'une scène.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur clique sur un objet.
  - Le système vérifie que l'objet n'a pas déjà été cliqué.
  - L'interaction a lieu.
- Scénario d'échec :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur clique sur un objet.
  - Le système détecte que l'objet a déjà été cliqué.
  - Rien ne se passe.

Cas n°4 : Ramasser la clé

- Acteur : Joueur
- Condition d'utilisation : le joueur doit être en train de jouer et ne pas encore avoir ramassé la clé.
- Description : Un joueur peut ramasser la clé pour déverrouiller la porte. Cas particulier du cas n°3.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur clique sur la clé.
  - Le système vérifie que la clé n'a pas déjà été cliquée.
  - La clé est déplacée dans le coin inférieur droit de l'écran et la porte est déverrouillée.
- Scénario d'échec :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur clique sur la clé.

- Le système détecte que la clé a déjà été cliquée.
- Rien ne se passe.

Cas n°5 : Passer à la pièce suivante

- Acteur : Joueur
- Condition d'utilisation : le joueur doit être en train de jouer et avoir déjà ramassé la clé de la scène dans laquelle il se trouve.
- Description : Un joueur peut passer à la scène suivante en cliquant sur la porte.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur ramasse la clé.
  - Le joueur clique sur la porte.
  - Le système vérifie que la clé a déjà été trouvée.
  - Le système affiche la scène suivante.
- Scénario d'échec :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur clique sur la porte.
  - Le système détecte que la clé n'a pas été trouvée.
  - Rien ne se passe.

Cas n°6 : Sauvegarder la partie

- Acteur : Joueur
- Condition d'utilisation : le joueur doit être en train de jouer.
- Description : Un joueur peut sauvegarder sa partie en cours afin d'y revenir plus tard.
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur lance une partie.
  - Le joueur joue.
  - Le joueur clique sur « sauvegarder et quitter ».
  - Le système sauvegarde la partie (pseudo et niveau en cours).
  - Le système ferme le jeu.
  - Le joueur dispose d'une sauvegarde pour terminer le jeu plus tard.