

# Analyse et Conception

## Objet de Logiciel



Année scolaire 2023/2024

2<sup>e</sup> année ISI G3

Adam Benoit, Pierre Poinas, Steve Lening, Anaïs Douet

# Sommaire :

<b>Sommaire :</b>	<b>1</b>
<b>Cahier des charges</b>	<b>2</b>
Entités	3
Démarrage et fin d'une partie	5
<b>Document d'analyse</b>	<b>6</b>
Cas d'utilisation du jeu	6
Diagramme des cas d'utilisation	6
Rédaction des cas d'utilisation et illustration via un diagramme de séquence	6
Diagramme de classes d'analyse	11
<b>Document de conception</b>	<b>12</b>
Architecture logique du logiciel	12
Modèle en couche du logiciel	12
Diagramme de composant du logiciel	12
Diagrammes de déploiement du logiciel	13
Vue des processus	13
Description de l'incrément	13
Conception détaillée	13
Diagramme de classes logicielles	13
Diagrammes de séquence technique	13
Diagrammes d'états-transitions	14
<b>Manuel utilisateur pour l'incrément choisi</b>	<b>14</b>
1. Objectif du Jeu	14
2. Votre Personnage	14
3. Objets	15
4. Exploration et Combat	15
5. Marchands	15
6. Trésor	15
7. Interface Utilisateur	15
8. Commandes	16
<b>Bilan sur les outils de modélisation utilisés</b>	<b>16</b>
Visual Paradigm Online :	16
LucidChart :	16
Autres modeleurs :	17

# Cahier des charges

## Glossaire :

Objets	Entités	Autres
<ul style="list-style-type: none"><li>- Pièce</li><li>- Potion</li><li>- Arme</li><li>- Clé</li></ul>	<ul style="list-style-type: none"><li>- Joueur</li><li>- Monstre</li><li>- Voleur</li><li>- Marchand</li><li>- Chat</li></ul>	<ul style="list-style-type: none"><li>- Coffre au trésor</li></ul>

## Salles :

Le jeu génère à son chargement un ensemble de salles connectées. Les salles sont contenues dans une matrice de 5x5. La matrice n'est pas forcément entièrement remplie de salles. Toutes les salles sont connectées entre elles (pas de groupe de salles isolé). Les salles et leur disposition sont générées aléatoirement au moment du lancement du jeu. Il y a au moins 8 salles. Le contenu des salles est généré aléatoirement à la volée lorsque le joueur rentre dedans pour la première fois.

Une salle est un quadrillage de 20x30 cases, les entités peuvent se déplacer horizontalement et verticalement sur ce quadrillage.

Il y a plusieurs types de salles qui font varier le contenu et tout contenu est placé aléatoirement sur le quadrillage de la salle :

Les salles **neutres** :

Contient entre 1 et 4 objets choisis aléatoirement (équiprobable) parmi ceux disponibles. Peut contenir un chat avec 30% de chance. Il existe au moins une salle neutre. Le joueur commence forcément son aventure dans une de ces salles.

Les salles **hostiles** :

Contient entre 1 et 2 objets choisis aléatoirement (équiprobable) parmi ceux disponibles. Contient entre 1 et 3 entités hostiles choisies aléatoirement parmi les suivantes : monstre et voleur. Un monstre à 80% de chance d'apparaître et un voleur en a 20%.

Salle du **trésor** :

Contient entre 1 et 3 pièces. Contient en son centre un coffre au trésor. Il y a exactement une salle de ce type.

Salle du **marchand** :

Contient un marchand. Peut contenir un chat avec une probabilité de 50%. Il y a exactement une salle de ce type.

## Entités

### **Joueur :**

Il peut se déplacer, ramasser un objet s'il va sur la case de ce dernier, et le stocker dans son sac à dos. Il peut combattre des entités hostiles avec une arme. Il ne peut pas passer à travers les murs. Le joueur peut également interagir avec le trésor, le chat et le marchand.

Le joueur possède 3 vies au début du jeu et ne peut pas en acquérir plus de 3. S'il perd ses 3 vies, le joueur meurt et la partie s'arrête.

Le joueur possède un sac à dos qui contient 10 places. Les objets ne peuvent pas s'y "stacker" (les objets du même type ne se superposent pas sur une même place). Si le joueur veut ramasser un objet alors que son sac est rempli, il ne se passe rien et l'objet reste au sol.

### **Marchand :**

Le marchand permet d'acheter des objets. Le joueur peut échanger un objet pour obtenir une pièce, échanger deux pièces pour obtenir un objet de n'importe quel type (excepté une clé ou une pièce) ou échanger 3 pièces pour obtenir une clé.

Il est situé dans la salle de type MARCHAND.

### **Chat :**

Il fait partie des éléments du décor. Il peut être caressé par le joueur.

### **Entités de type Hostiles :**

Une entité hostile enclenche le combat avec un joueur s'il se trouve sur la même case que lui.

Elles se déplacent dans les salles de manière à aller vers le joueur en permanence.

### **Monstres :**

Effet de défense : le joueur perd une vie, il disparaît.

### **Voleurs :**

Effet de défense : le joueur perd un objet aléatoirement de son sac à dos, il disparaît.

## Objets

### **Armes :**

Elles peuvent être ramassées par le joueur. Une arme sert à combattre une unique fois un méchant. Après cette utilisation, elle disparaît.

### **Clés :**

Elles peuvent être ramassées par le joueur. Le joueur stocke la clé dans son sac à dos.

### **Pièces :**

Elles peuvent être ramassées par le joueur. Le joueur stocke la pièce dans son sac à dos. Elles permettent de faire des échanges avec le marchand.

### **Potion de vie :**

Elles peuvent être ramassées par le joueur. Lorsque le joueur la récupère, si le joueur n'a plus que 2 ou une vie, il gagne une vie. Si le joueur a toutes ses vies, il garde l'item dans son sac à dos et pourra s'en servir ultérieurement.

## **Trésor**

### **Trésor :**

Si le joueur essaye d'ouvrir le trésor avec moins de 3 clés, un piège se déclenche et le joueur perd une vie.

Si le joueur essaye de l'ouvrir avec 3 clés, il remporte la partie et le jeu se termine.

Il est situé dans la salle de type TRESOR.

## **Combat**

Si le joueur possède une arme, le monstre meurt et un objet aléatoire apparaît dans la salle. L'arme de défense disparaît.

Si le joueur ne possède pas d'arme, le monstre applique son effet de défense. Après ça, il disparaît, mais aucun objet n'apparaît dans la salle.

## **Interaction avec l'utilisateur**

### **Gestion des commandes :**

Utilisation des touches fléchées ou des touches ZQSD (AZERTY) pour déplacer le personnage case par case sur le quadrillage.

L'action de ramasser un objet ou d'attaquer un ennemi se réalise automatiquement lorsque le joueur se déplace sur la case contenant un objet ou ennemi.

L'action de caresser un chat peut être réalisée si le joueur se trouve sur la même case que le chat et appuie sur la touche "Entrée".

Pour échanger un objet avec le Marchand, le joueur doit se trouver sur une des 4 cases adjacentes (gauche, haut, droite, bas). Il doit appuyer sur la touche "O" de son clavier pour échanger un objet ou la touche "P" pour échanger des pièces. S'il appuie sur la touche "O", il devra ensuite appuyer sur la touche correspondant à l'index de l'objet à échanger dans son inventaire (touches 0 à 9). Le compteur de pièces du joueur s'incrémentera alors. S'il appuie sur la touche "P", il devra ensuite appuyer sur la touche correspondante au nombre de pièces qu'il souhaite échanger avec le Marchand (touche 2 ou 3). L'objet que le Marchand lui fournira apparaîtra alors dans son sac à dos.

### **Système de navigation entre les salles**

Détection des collisions avec les murs afin d'empêcher le personnage de passer à travers (cf. Entités. Joueur). De ce fait, toute action de déplacement à travers un mur ne sera pas réalisée si une collision est détectée. Le joueur restera sur sa case, dans le même état que si aucune action n'avait été faite de la part de l'utilisateur.

Le joueur peut se déplacer sur une case correspondant à un “trou dans les murs”. S’il effectue ce déplacement, il change de pièce. Il se rendra ainsi dans la salle adjacente (cf. Salles). Les cases “trou dans le mur” se trouvent par nombre de 1 sur les murs partagés par deux salles adjacentes. Ces cases sont adjacentes et, de préférence, situées au milieu des murs.

## **Interface utilisateur graphique**

### **Démarrage et fin d’une partie**

Le jeu démarrera sur un menu contextuel expliquant succinctement les règles du jeu et les commandes permettant de se déplacer et réaliser des actions.

Il ne sera pas possible de charger une partie précédente.

À tout moment dans la partie (hors menu contextuel), le joueur aura la possibilité d’appuyer sur la touche “Échap” afin de quitter le jeu (sans possibilité de sauvegarder sa progression). Un menu contextuel s’affichera demandant confirmation au joueur de son action. Ce menu contextuel précisera à l’utilisateur sur quelle touche appuyer pour valider ou invalider son action de quitter la partie.

### **Représentation des entités**

Les salles seront représentées par un plan 2D dans un style “pixel art”.

Utilisation de graphismes “pixel art” pour créer des personnages, des monstres et des environnements cohérents avec le style rétro du jeu.

Intégration de sprites expressifs pour les personnages et les monstres lors de leur mouvement dans les salles, de leurs attaques et de leurs interactions avec les objets.

### **Informations affichées l’écran**

[À ajouter ?] Placement des éléments d’interface de manière stratégique pour éviter les superpositions et assurer une visibilité maximale. Tous les éléments seront affichés sur les côtés gauches, droit et sur le haut de l’interface. L’affichage des informations, en termes de surface couverte, ne doit pas dépasser plus de 1/3 de l’écran.

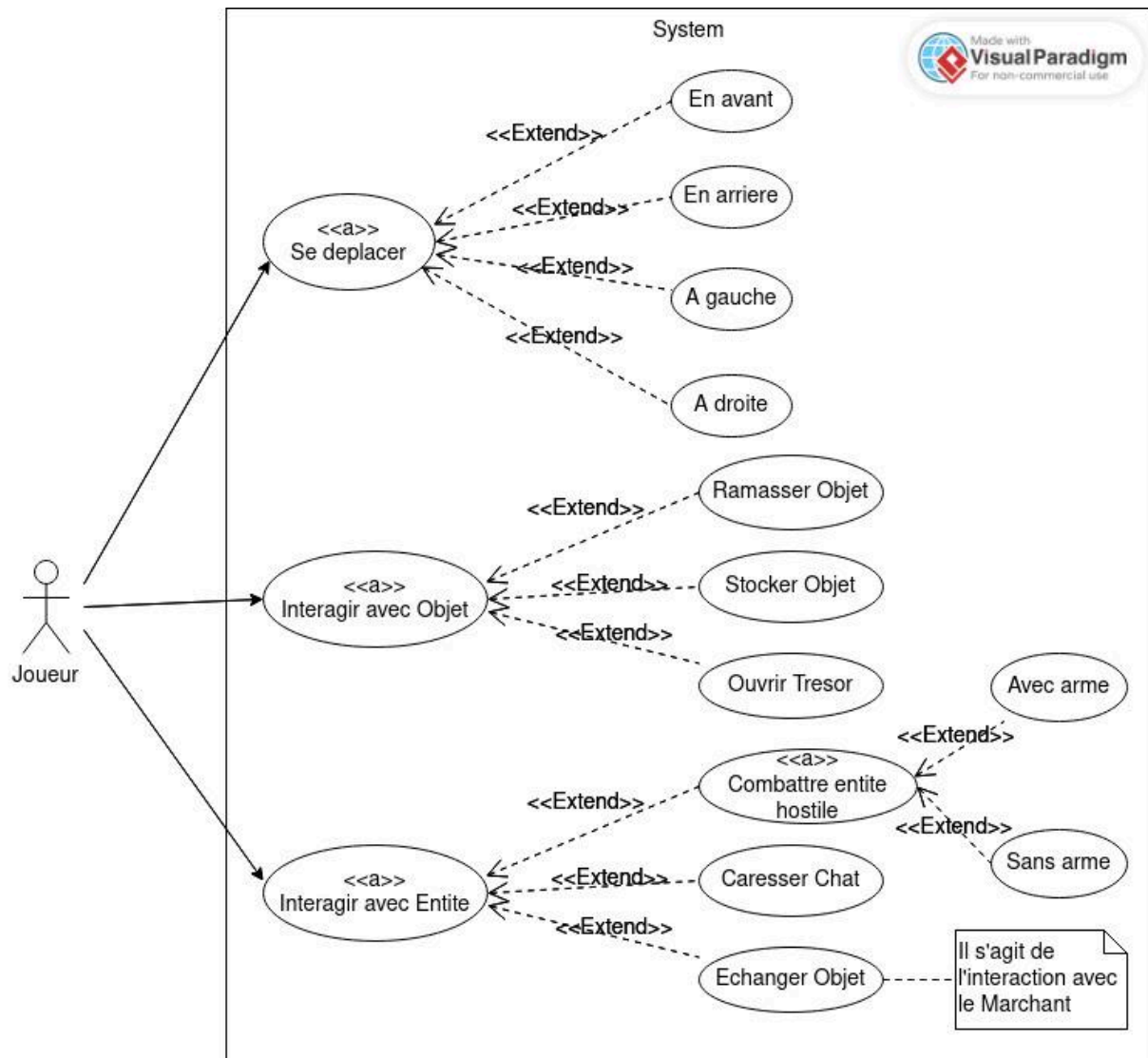
Les informations affichées correspondront au contenu du sac à dos (catégorie d’objets et nombre d’objets par catégorie) ainsi que la barre de vie du joueur. Ces informations seront mises à jour automatiquement à chacun de leur changement.

Affichage de toute la salle en permanence à l’écran.

# Document d'analyse

## Cas d'utilisation du jeu

### Diagramme des cas d'utilisation



### Rédaction des cas d'utilisation et illustration via un diagramme de séquence

#### - Se déplacer :

##### PRECONDITIONS

- Le jeu a démarré
- Le joueur est vivant (il a un nombre de vies  $\geq 1$ )
- Le joueur se trouve sur une case dans une salle de n'importe quel type
- Le joueur n'a pas encore ouvert le trésor

##### DESCRIPTION

1. L'utilisateur se déplace en appuyant sur une touche de direction du clavier parmi "up", "down", "left" ou "right".

1.1. Si la case sur laquelle il souhaite aller n'est pas disponible, c'est-à-dire qu'il est face à un mur par exemple (il y a un mur dans la direction souhaitée de son déplacement), alors il ne se passe rien et le joueur reste au même endroit.

1.2. Si la case sur laquelle il souhaite aller est disponible, alors le joueur se déplace en un seul saut sur cette case.

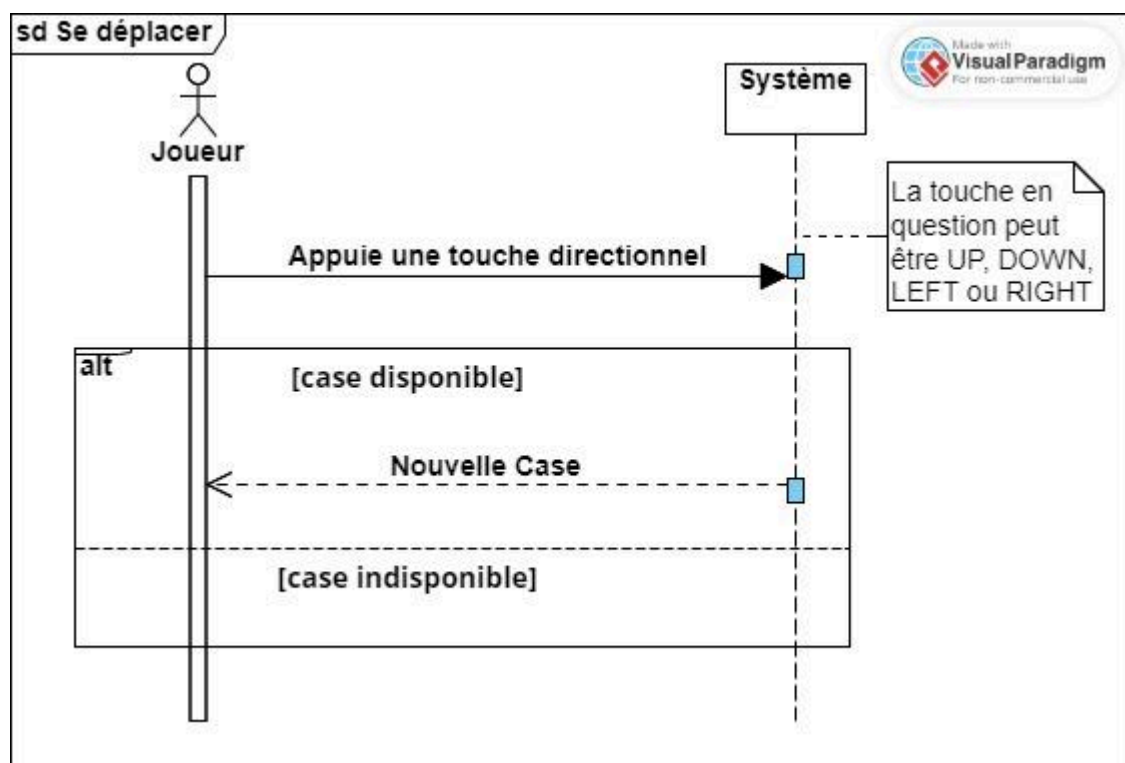
#### POSTCONDITIONS

- Le joueur se trouve soit sur la même case qu'au départ, soit sur une case différente selon que la case visée par son déplacement était ou pas accessible.

#### EXCEPTIONS

- Si le joueur est déjà mort, il ne se passe rien puisque la partie est terminée
- Si le joueur a déjà ouvert le trésor, alors il ne se passe rien non plus, parce qu'il a terminé la partie.

#### DIAGRAMME DE SÉQUENCE



- **Interagir avec un objet :**

#### PRECONDITIONS

- Le jeu a démarré
- Le joueur est vivant
- La case où se trouve le joueur contient un objet

#### DESCRIPTION



1. Si le sac du joueur contient moins de 10 objets, alors, le joueur ramasse l'objet qui se trouve sur la même case que lui.
2. Si le sac du joueur contient déjà 10 objets, alors, il ne se passe rien, et l'objet reste sur le sol.

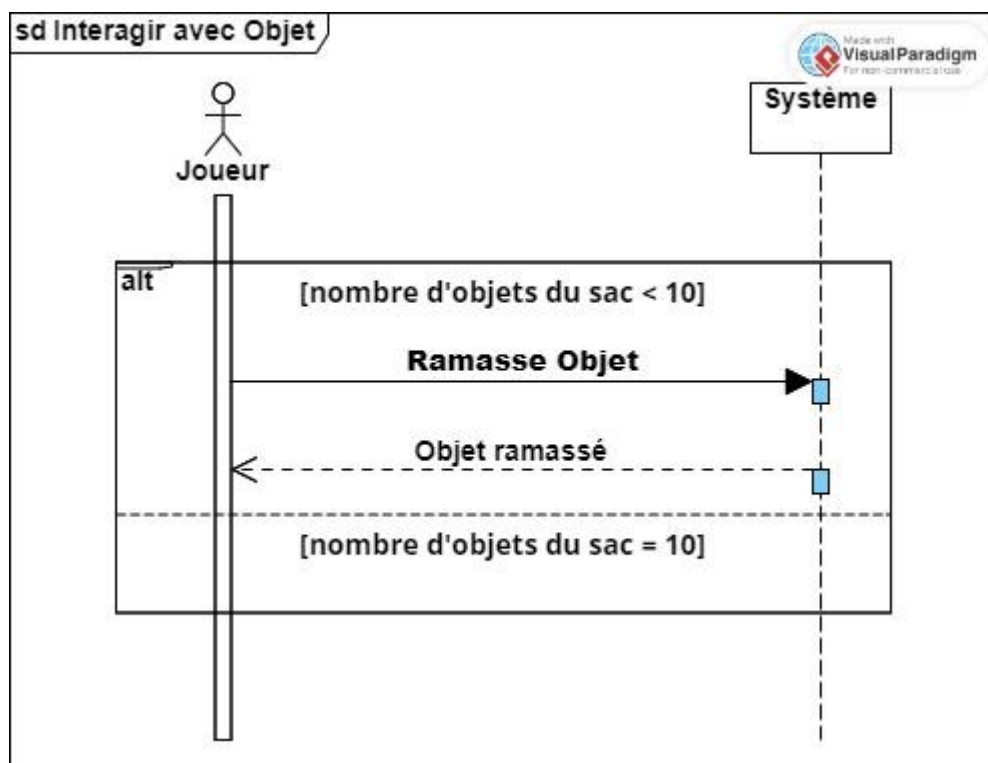
#### POSTCONDITIONS

- Le nombre d'objets présents dans le sac du joueur a augmenté ou stagné selon qu'il avait moins de 10 objets ou pas avant d'arriver sur la case courante.

#### EXCEPTIONS

- Si la case où se trouve le joueur ne contient aucun objet, alors rien ne se passe.

#### DIAGRAMME DE SÉQUENCE



- **Interagir avec une entité :**

#### PRECONDITIONS

- Le jeu a démarré
- Le joueur est vivant
- La case où se trouve le joueur contient une entité

#### DESCRIPTION

1. Si la case contient une entité hostile et que le joueur possède une arme
  - 1.1. S'il s'agit d'un Voleur, alors le joueur combat le voleur avec son arme, le voleur ainsi que l'arme utilisée disparaissent au terme du combat.
  - 1.2. S'il s'agit d'un Monstre, alors le joueur combat également le monstre avec son arme, le monstre ainsi que l'arme du joueur disparaissent et un objet aléatoire apparaît aléatoirement dans la salle (c'est-à-dire à une position aléatoire dans la salle).

2. Si la case contient une entité hostile et que le joueur ne possède pas d'arme
  - 2.1. S'il s'agit d'un voleur, alors le voleur ainsi qu'un objet aléatoire se trouvant dans le sac à dos du joueur disparaissent.
  - 2.2. S'il s'agit d'un monstre, alors le monstre disparaît et le joueur perd une vie.
3. Si la case contient une entité non hostile
  - 3.1. S'il s'agit du Marchand, alors le joueur peut échanger un/des objet(s) avec celui-ci en échange d'autres objets.
  - 3.2. S'il s'agit d'un chat, alors le joueur peut tout simplement le caresser.

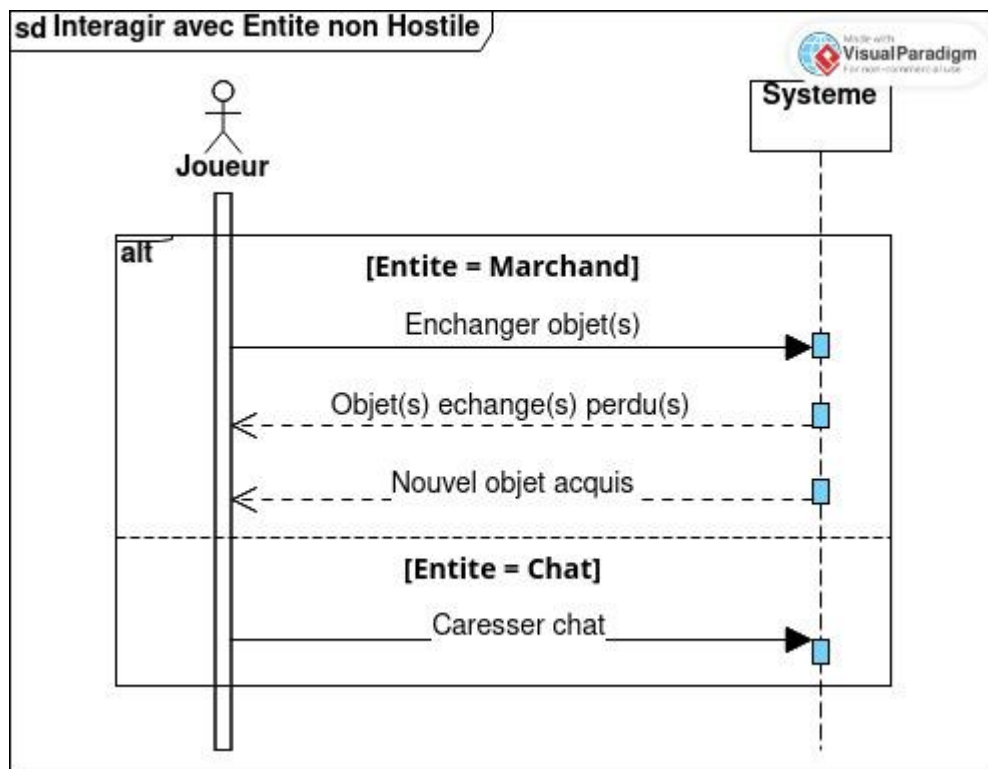
#### POSTCONDITIONS

- Si le joueur a interagit avec un chat, alors rien n'a changé.
- S'il a interagit avec le Marchand, alors, potentiellement le contenu de son sac à dos est différent selon que le joueur ait décidé d'échanger ou pas des objets.
- S'il a interagit avec un voleur, il a un objet de moins dans son sac à dos, soit une arme en moins soit un autre objet.
- S'il a interagit avec un monstre, il a potentiellement une vie en moins selon qu'il ait affronté le monstre avec ou sans arme.

#### EXCEPTIONS

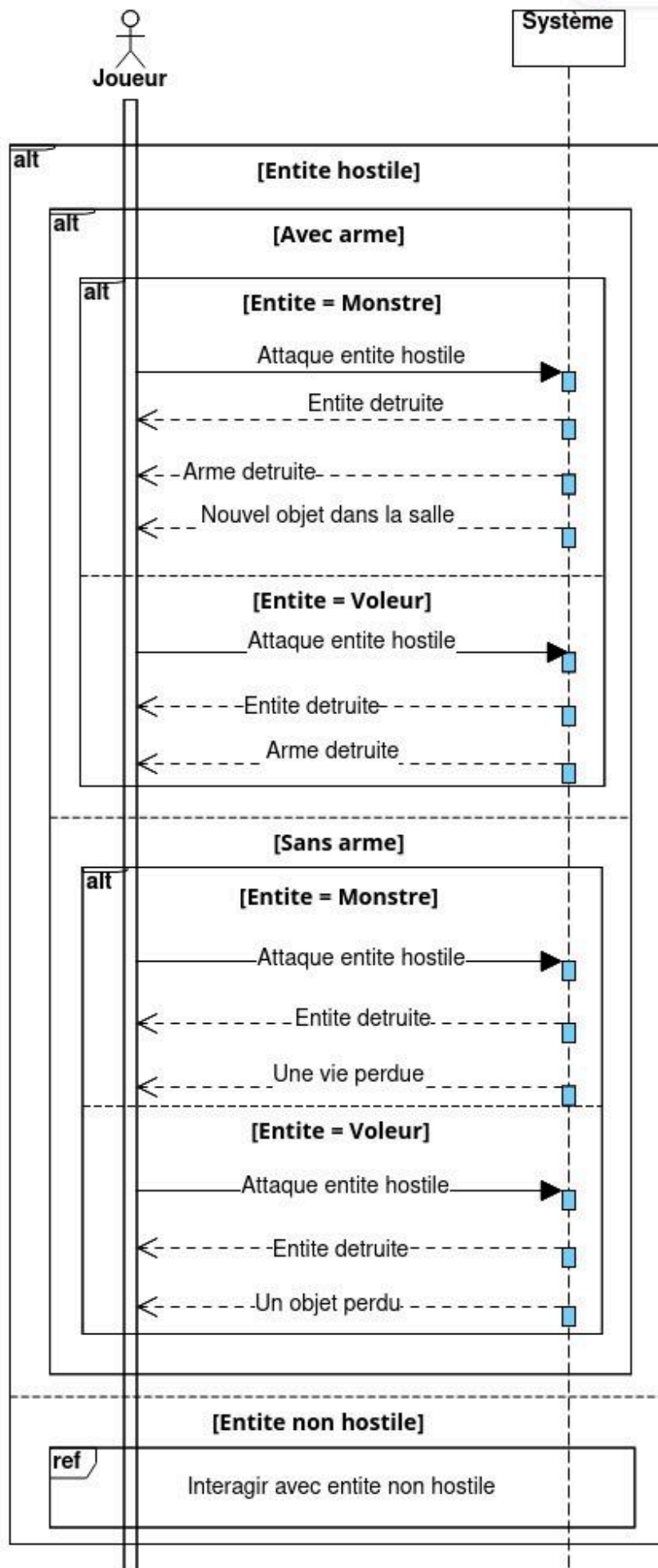
- Si la case où se trouve le joueur ne contient aucune entité, alors rien ne se passe.

#### DIAGRAMME DE SÉQUENCE



Ci-dessous le diagramme de séquence système représentant une interaction avec une entité de n'importe quel type. Ce diagramme référence celui de l'interaction avec une entité non hostile ci-dessus.

# sd Interagir avec Entite



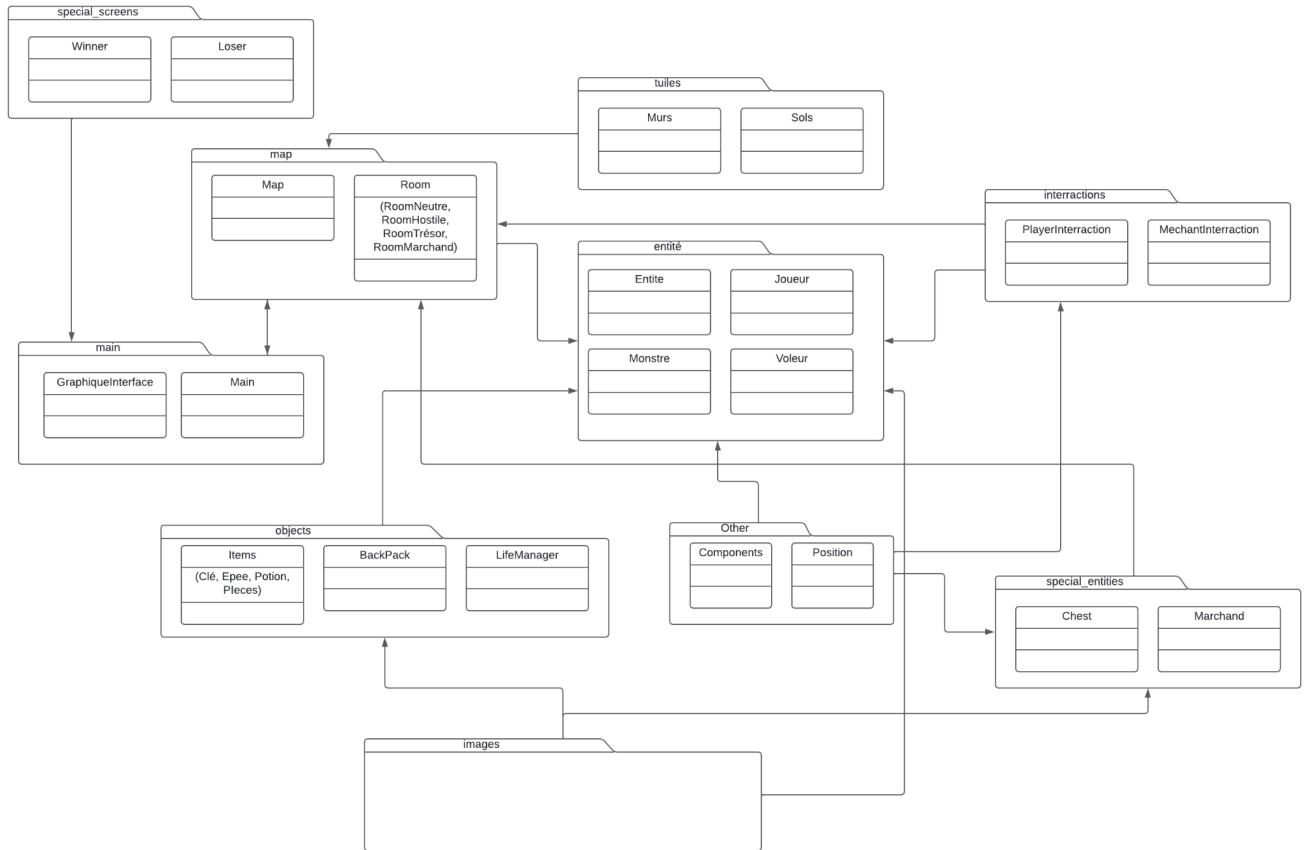
Made with  
**VisualParadigm**  
For non-commercial use



# Document de conception

## Architecture logique du logiciel

### Modèle en couche du logiciel



## Diagramme de composant du logiciel



## Diagrammes de déploiement du logiciel

Le diagramme de déploiement n'est volontairement pas réalisé ici. En effet, notre jeu ne s'exécute que sur une machine et ne sollicite que les ressources de celle-ci. De plus, nous n'utilisons pas de base de données, mais réalisons tout le stockage d'information dans la mémoire vive (jeu relativement peu gourmand).

Il n'est donc pas pertinent de réaliser un tel diagramme dans notre cas de figure.

## Vue des processus

Notre logiciel fonctionne via deux threads distincts. Le premier, exécuté au lancement du logiciel, s'attelle à créer et maintenir la fenêtre `JFrame` (package `"main"`, classe `"GraphiqueInterface"`).

## Description de l'incrément

Pour simplifier la programmation d'une première version de notre jeu, nous avons choisi certaines fonctionnalités à implémenter plutôt que d'autres.

Dans un premier temps, pour ce qui est de la génération des salles, nous avons décidé de la garder, comme prévu, dans le cahier des charges. En effet, la génération de l'environnement de jeu est un aspect primordial du jeu et doit donc être complètement implémenté. L'ensemble des différents types de salles vont donc être implémentés aussi à l'exception des probabilités d'apparition des différents éléments. En effet, tout élément apparaît de façon équiprobable dans cet incrément. Or leur contenu va varier dans cet incrément.

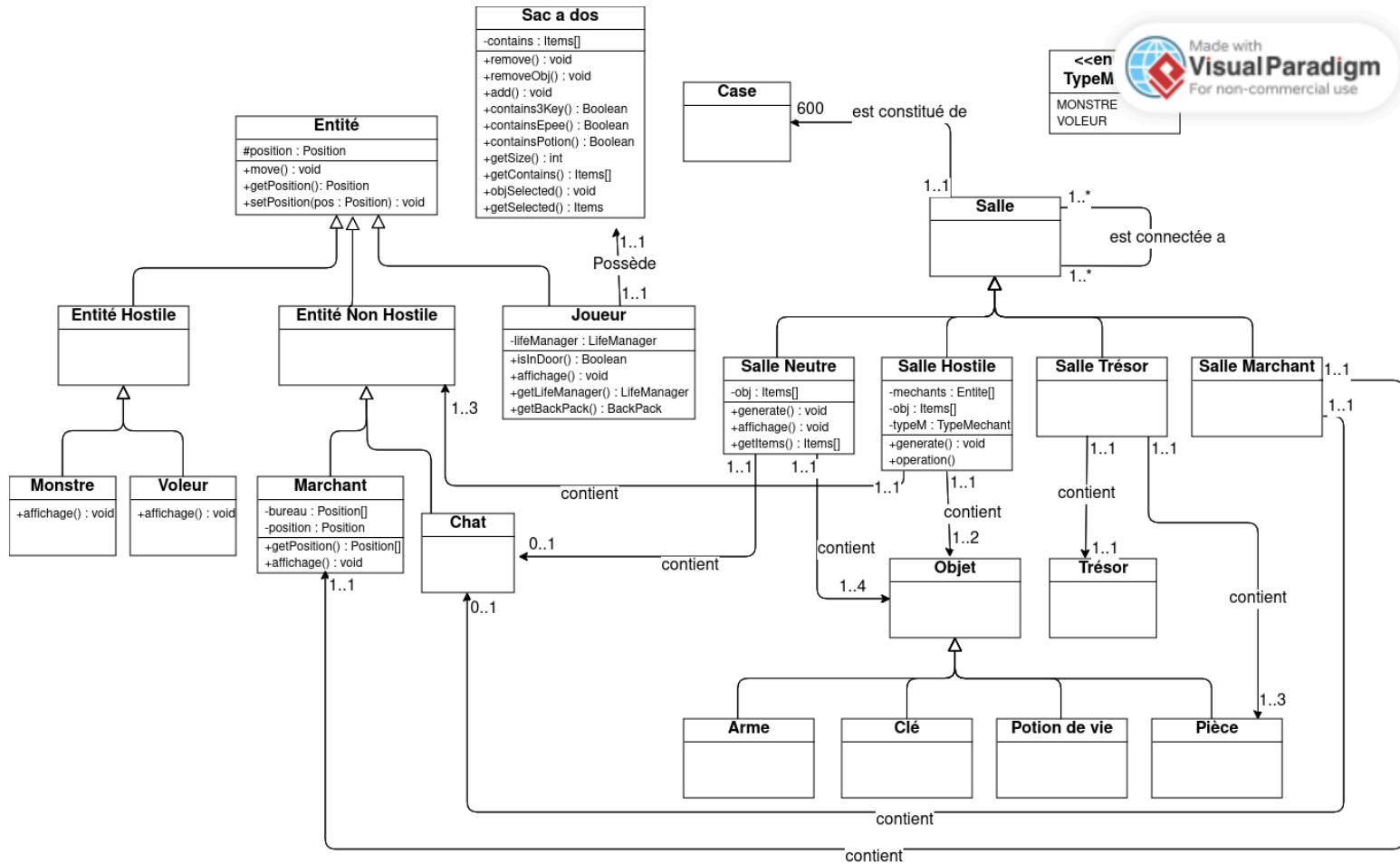
Dans un premier temps, nous n'allons pas implémenter le code permettant de générer et interagir avec l'entité passive chat.

Ensuite, pour l'utilisation du clavier par le joueur, nous avons choisi de finalement utiliser que les touches flèche, la touche espace et la touche entrée pour simplifier la compréhension du jeu par l'utilisateur (cf. Manuel Utilisateur).

Finalement, les fonctionnalités de menus d'ouverture et de fermeture du jeu ne seront pas non plus implémentées. En effet, elles ne servent qu'à la qualité de vie du joueur sans être très intéressantes par rapport à la conception du reste du jeu.

Pour le reste, c'est-à-dire, la salle du trésor, la salle du marchand et ses interactions, les combats, les objets et l'interface graphique, tout sera implémenté.

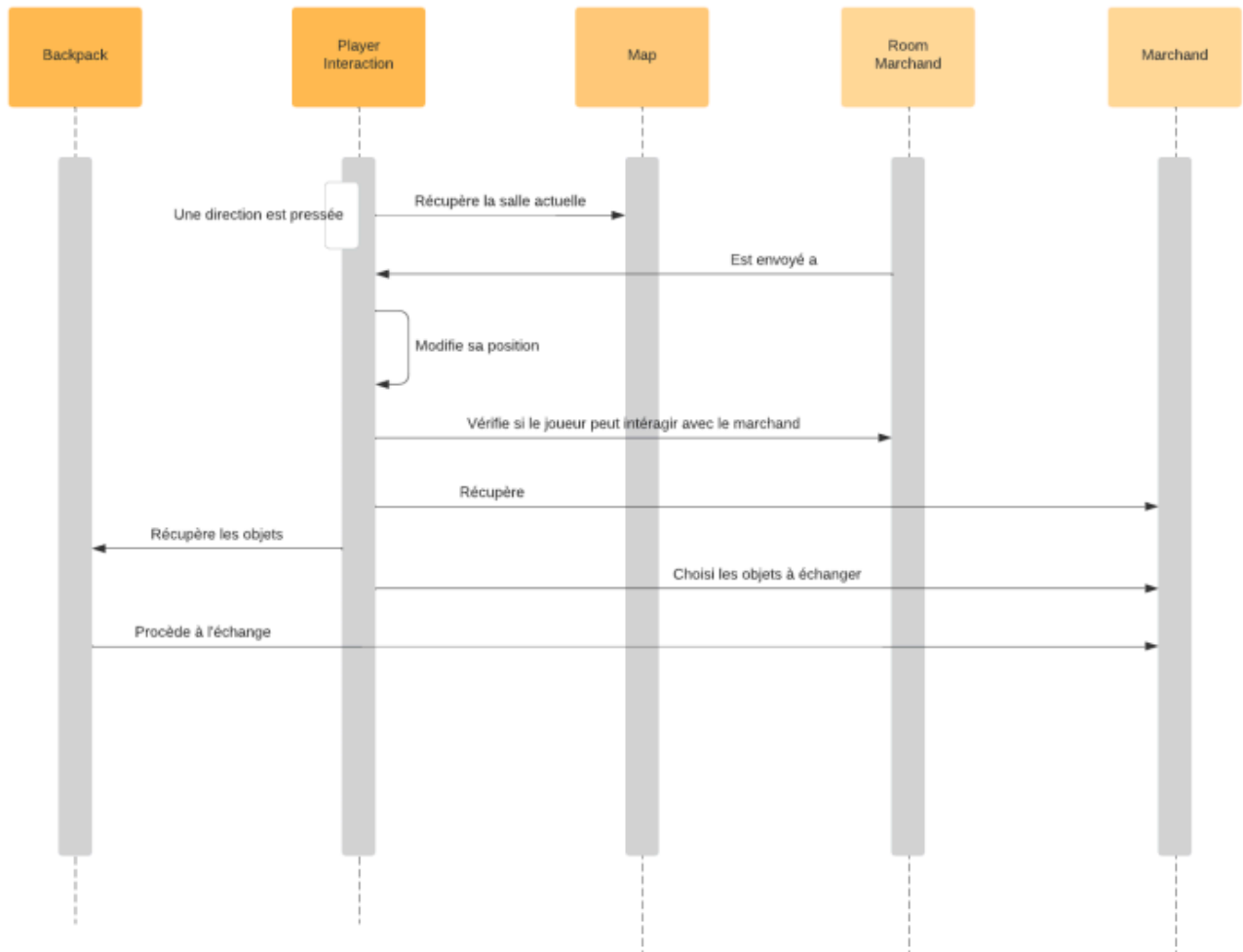
## Diagramme de classes logicielles





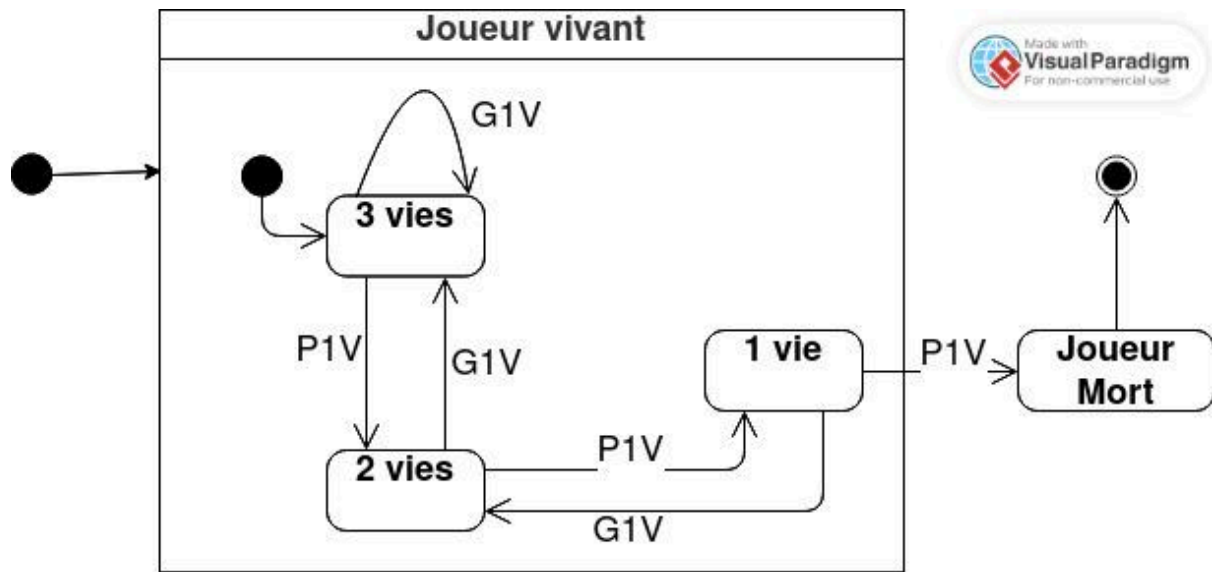
## Diagrammes de séquence technique

Diagramme de séquence d'un joueur qui interagit avec le marchand



## Diagrammes d'états-transitions

- Etat du joueur :



### Legende :

P1V = Perd une vie (en combattant un monstre sans arme par exemple)  
G1V = Gagne une vie (en utilisant une potion de vie)

# Manuel utilisateur pour l'incrément choisi

Bienvenue dans "ZELDO" ! Préparez-vous à plonger dans un monde rempli de défis, de trésors et de rencontres étranges. Ce manuel est conçu pour vous aider à naviguer à travers ce jeu fascinant. Suivez ces instructions pour une expérience optimale.

## 1. Objectif du Jeu

Votre mission est de trouver et d'ouvrir le trésor caché dans l'une des salles. Pour y parvenir, vous devrez explorer les différentes salles, ramasser des objets utiles, combattre des ennemis hostiles. Vous pourrez également interagir avec des entités et personnages non joueurs.

## 2. Votre Personnage

- Votre personnage commence avec 3 vies.
- Vous possédez un sac à dos pouvant contenir jusqu'à 10 objets, chaque objet occupe un emplacement dans votre sac.
- Si votre sac est plein et que vous essayez de ramasser un objet supplémentaire, il restera au sol.

## 3. Objets

- Déplacez-vous sur la case de l'objet pour le ramasser automatiquement.
- Ramassez des armes pour combattre les ennemis.
- Ramassez des clés pour ouvrir le trésor.
- Ramassez des pièces pour échanger avec le marchand.
- Ramassez des potions de vie pour vous éviter de mauvaises aventures.

Lorsque vous récupérez une potion de vie, celle-ci vous redonne une vie si vous en avez perdu une ou deux. Si vous avez toutes vos vies, vous stockez l'item dans votre sac à dos et vous pourrez vous en servir ultérieurement.

Si vous perdez une vie alors qu'une potion se trouve dans votre sac à dos, celle-ci sera automatiquement utilisée (vous gagnerez une vie et elle disparaîtra de votre inventaire).

## 4. Exploration et Combat

- Explorez les salles pour trouver des objets, des pièces et des ennemis.
- Combattez les ennemis en vous déplaçant sur leur case. Les ennemis peuvent également venir sur votre case pour se battre avec vous.
- Si vous possédez une arme, elle sera utilisée automatiquement lors du combat. Vous vaincrez alors l'ennemi qui disparaîtra.

- Si vous ne possédez pas d'arme, le monstre appliquera un effet de défense. Vous perdrez alors une vie, mais le méchant disparaîtra.
- Il existe deux types de méchants, les voleurs et les monstres. L'effet de défense du voleur est de vous voler un objet dans votre sac à dos. Celui des monstres est de vous faire perdre une vie. Prenez garde !

## 5. Marchands

- Vous trouverez le marchand au milieu d'une unique salle sur la carte.
- Vous pouvez échanger un objet pour obtenir une pièce, deux pièces pour obtenir un objet ou trois pièces pour obtenir une clé. Pour ce faire, vous devez vous mettre sur une case à côté de son bureau et aller vers lui. Vous aurez alors la possibilité de sélectionner un ou plusieurs objets. Pour passer d'un objet à un autre, il vous suffit d'appuyer sur la flèche de droite ou de gauche de votre clavier. Pour sélectionner l'objet à échanger, appuyez sur la touche ENTRER.
- Une fois vos échanges terminés, vous pouvez retourner explorer avec la touche ESPACE.

## 6. Trésor

- Vous devez collecter au moins 3 clés pour ouvrir le trésor et gagner la partie.
- La partie s'arrête une fois le trésor a été ouvert.
- Pour tenter d'ouvrir le trésor, il vous suffit d'être sur une case adjacente à ce dernier et d'essayer de vous déplacer dessus.
- Si vous essayez d'ouvrir le trésor avec moins de 3 clés, un piège se déclenche et vous perdez une vie.

## 7. Interface Utilisateur

- Les informations sur votre sac à dos ainsi que votre barre de vie sont affichées à l'écran.
- Toutes les salles sont affichées en permanence à l'écran.

## 8. Commandes

- Utilisez les touches fléchées pour déplacer votre personnage dans les différentes directions sur le quadrillage de la salle.
- Appuyez sur la touche "Close" de la fenêtre graphique pour quitter définitivement le jeu à tout moment de la partie

Maintenant que vous êtes prêt, que l'aventure commence !

# Bilan sur les outils de modélisation utilisés

Lors de notre projet, nous avons pu utiliser différents modeleurs UML. Chacun avec leurs qualités et leurs défauts. En voici un bilan :

## Visual Paradigm Online :

La version offline de ce logiciel existe, mais nous avons décidé d'utiliser la version online parce qu'elle est collaborative et nous permet de travailler de façon simultanée et potentiellement à distance sur un même diagramme. Cet outil permet de créer à peu près tous les diagrammes UML qui existent, nous l'avons utilisé pour créer typiquement le diagramme de classe d'analyse, le diagramme des cas d'utilisation et le diagramme d'état transition. Il est très simple et très intuitif à utiliser, de plus il donne la possibilité d'utiliser les formes de plusieurs types de diagrammes différents dans le même diagramme, ce qui peut être très utile dans certains cas. Une potentielle limite de cet outil est le fait qu'elle ajoute son logo sur l'image finale du diagramme exporté, ce qui peut cacher une partie du diagramme. Pour résoudre ce problème, nous avons adapté la position des éléments afin d'éviter que le logo de l'application ne cache une partie de notre diagramme.

## LucidChart :

Lucidchart est un très bon modeleur UML (mais pas que) en ligne que nous avons utilisé. Il nous a particulièrement été utile pour réaliser le modèle en couches et le diagramme de composant de notre logiciel. Il est intuitif d'utilisation et pratique. Sa seule limitation réside dans le fait que la version gratuite autorise le 60 figures par diagrammes et limite l'utilisateur à 3 diagrammes. Cela ne nous a pas posé de problème pour réaliser les schéma que vous avez pu observer. Cela nous contraint seulement à ne plus pouvoir les développer au-delà d'une certaine taille (nombre d'éléments présents sur le diagramme). Nous n'avons donc pas pu inclure tout le diagramme de classe logicielles dans le modèle de couches, nous avons seulement choisi de citer les classes les plus importantes.

## Autres modeleurs :

Nous avons également essayé d'utiliser *Modelio* comme préconisé sur la page Chamilo du cours. Cependant ce logiciel, lourd à l'installation, est doté d'une interface graphique peu intuitive. Les diagrammes étaient donc difficiles à réaliser. Malgré les tutoriels et autres pages "wiki" disponibles sur Internet pour nous aider, nous n'avons pas retenu ce modeleur pour réaliser nos diagrammes.

Aussi, nous avons essayé d'utiliser Draw.io, bien qu'étant un outil permettant de dessiner des diagrammes UML en ligne et de façon collaborative, la complexité d'utilisation de cet outil, le nombre de formes limité et l'apparence peu ergonomique des diagramme obtenu nous a poussé à opter pour d'autre modeleurs UML.