

*UE – Programmation orientée objet et  
interfaces homme-machine 1  
UE – Anglais*

*L3 informatique  
Année 2023-2024*



# Project POO-IHM1

A text-based adventure game

Enseignants :  
*PELTIER Samuel  
DEFAYE Alex*

Etudiants - Groupe 12 :  
*VESSELLA Emilie  
DARTIGUES Mathieu  
MAITREHENRY Nathanaël*

## **Table des matières**

<b><i>Table des matières</i></b> .....	<b>2</b>
<b><i>1. Introduction</i></b> .....	<b>3</b>
<b><i>2. Documentation utilisateur</i></b> .....	<b>4</b>
A. Installation .....	4
B. Guide .....	4
<b><i>3. Documentation développeur</i></b> .....	<b>5</b>
A. Diagramme de classes (UML) .....	5
B. Diagramme d'états .....	5
A. Diagramme de séquence .....	5
<b><i>4. Organisation</i></b> .....	<b>7</b>
<b><i>ANNEXE</i></b> .....	<b>8</b>
Carte de base du jeu .....	8

# 1. Introduction

Dans ce projet, notre but été de développer un jeu d'aventure textuel interactif en Java. Introduit par Colossal Cave Adventure en 1975, ce type de jeu consiste à ce que l'utilisateur contrôle les actions d'un personnage à travers des ordres écrits en ligne de commande.

Dans notre jeu, le personnage principal a pour but de s'introduire dans un château afin de trouver le trésor qui s'y cache. Pour s'y faire, il devra visiter les différentes pièces du château et inspecter leur contenu. Il pourra également s'équiper de différents items qui pourront l'aider notamment à combattre les différentes créatures qui sont réparties dans le château.

Notre groupe est composé de Nathanaël Maitrehenry, Mathieu Dartigues et Emilie Vessella.

## 2. Documentation utilisateur

### A. Installation

Prérequis :

Avoir Java (<https://www.oracle.com/fr/java/>)

Avoir un JDK (<https://www.oracle.com/fr/java/technologies/downloads/>)

- Étape 1 :

Notre projet est disponible sur GitHub, vous devez télécharger une copie du projet :

<https://github.com/nageur2/projet-Poo>

Si vous avez git, vous pouvez simplement écrire la commande suivante dans un terminal :

git clone <https://github.com/nageur2/projet-Poo.git>

Sinon en haut à droite, il y a un onglet “Code” puis “Download zip”.

- Étape 2 :

Après s’être rendu dans le dossier, vous pouvez simplement exécuter le fichier “Maitrehenry\_Dartigues\_Vessella.jar” à l’aide de la commande java.

Exemple : java -jar Maitrehenry\_Dartigues\_Vessella.jar

### B. Guide

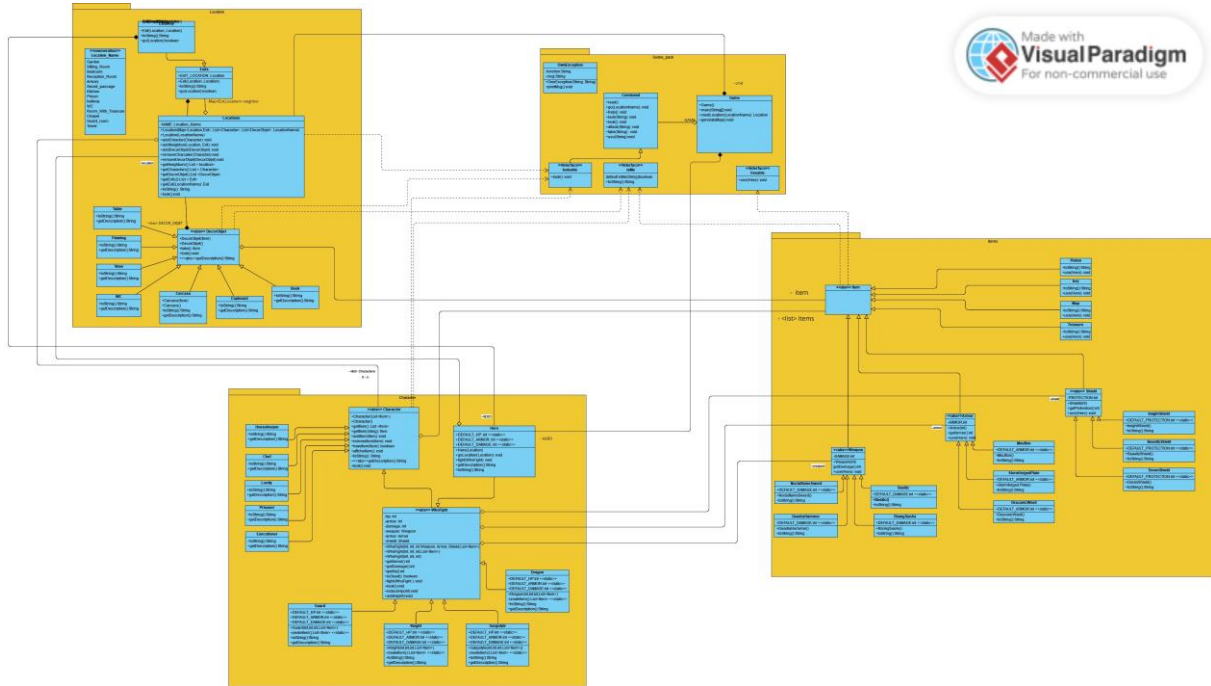
Voici la liste des différentes commandes disponibles ainsi que leurs explications :

- HELP : donne la liste des commandes.
- QUIT : quitter le jeu.
- LOOK [Location] : permet d’explorer une location.
- ATTACK [Personnage] : attaque le personnage visé.
- TAKE [objet] : ramasse l’objet s’il existe dans la location actuelle.
- USE [Item] : utilise l’Item visé s’il est dans l’inventaire

### 3. Documentation développeur

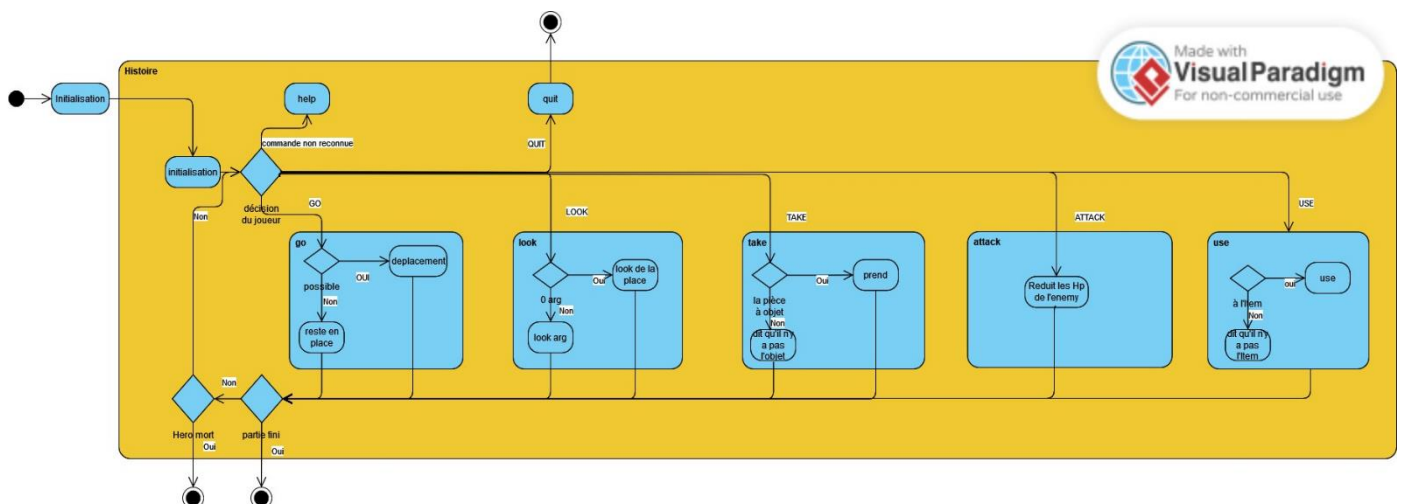
### A. Diagramme de classes (UML)

Celui-ci est présent dans les fichiers du jeu (de manière plus lisible).



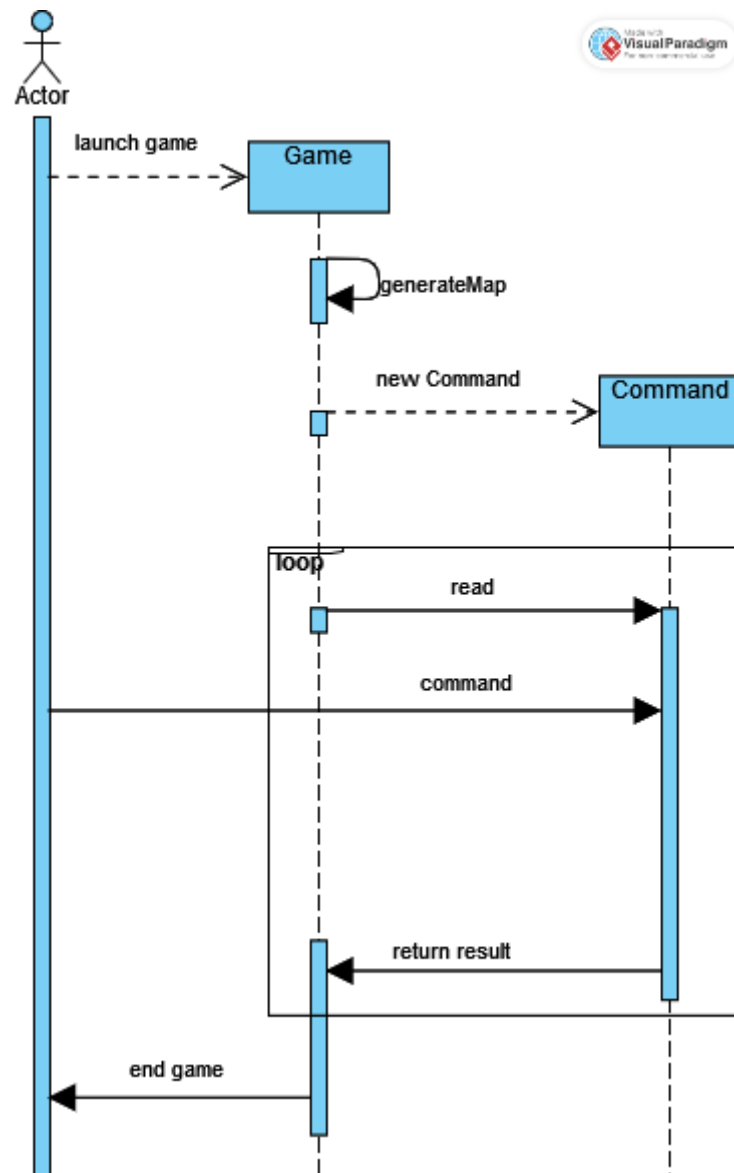
### B. Diagramme d'états

Celui-ci est présent dans les fichiers du jeu (de manière plus lisible).



### A. Diagramme de séquence

Celui-ci est présent dans les fichiers du jeu (de manière plus lisible).



## 4. Organisation

Dans un premier temps, nous nous sommes concertés pour savoir quel scénario et ambiance notre jeu allait avoir.

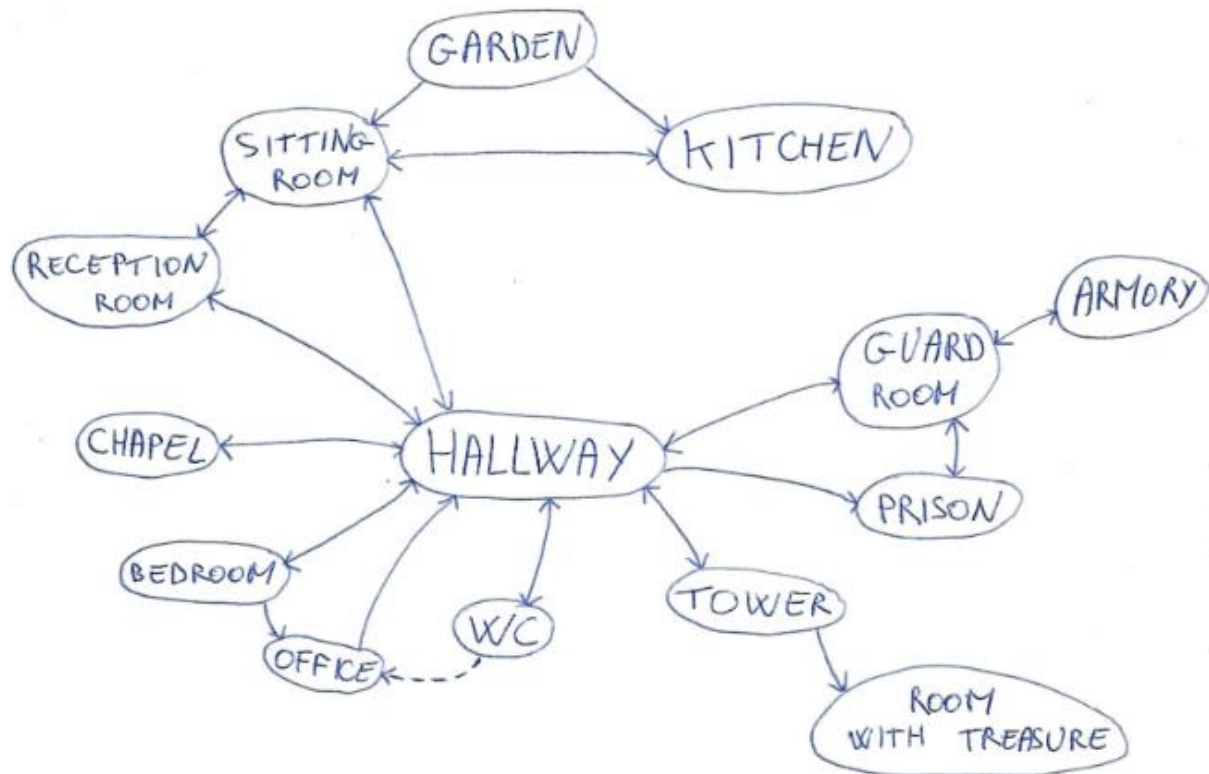
Ensuite, nous avons réalisé l'UML ensemble sur feuille avant de le transposer sur ordinateur avec Visual Paradigm.

En second lieu, nous nous sommes réparti les classes. Nathanaël a fait la structure de base du projet avec les fichiers correspondants, et puis nous avons pu remplir les méthodes de chaque classe.

Emilie s'est principalement occupé de la carte du jeu ainsi que des classes en rapport avec les Locations. Mathieu a réalisé les tests ainsi que les Items. Nathanaël s'est occupé des packages Game et Character. Au niveau du rapport, nous nous sommes partagé le travail.

# ANNEXE

Carte de base du jeu



Legend:

(ROOM)

→ exit in one way  
 ↔ exit in two directions  
 ---→ secret passage