

Rapport final du mini-projet UML

Groupe 2

MAHDI Chaymae
RHADI Meryam
REYNIER Tony

5 novembre 2022

Ingénierie Logicielle

UE Modélisation Orientée Objet et UML

Responsables

Sylvain Verdy
Richard Dufour

Sommaire

Titre	1
Sommaire	2
1 Introduction	3
2 Description de projet	3
3 Répartition des taches	3
4 L'ergonomie de l'application :	3
5 La modélisation des diagrammes de l'application :	3
5.1 Diagramme de gantt	3
5.1.1 Diagramme de Gantt pour la totalité du mini-projet(temps réel)	4
5.1.2 Diagramme de Gantt initiale(temps estimé)	5
5.2 Diagramme de cas d'utilisation	6
5.2.1 Implémentation	6
5.2.2 Description textuelle	6
5.3 Diagramme de classe	7
5.3.1 Implémentation	7
5.3.2 Description textuelle	7
5.4 Diagramme d'état-transition de joueur	9
5.4.1 Implémentation	9
5.4.2 Description textuelle	9
5.5 Diagramme d'état-transition de système	10
5.5.1 Implémentation	10
5.5.2 Description textuelle	10
5.6 Diagramme de séquence	11
6 Codage et programmation	12
6.1 Page de connexion	12
6.2 Page d'inscription	13
6.3 Page d'accueil	13
6.4 Modification de profil	14
6.5 L'interface de jeu	15
7 Conclusion	16

1 Introduction

Dans le cadre du programme de formation en Licence 3 Informatique dispensé par le Centre d'Enseignement et de Recherche en Informatique à Avignon, nous avons amenées à développer un jeu et la modéliser pour le but de conjuguer notre savoir et notre capacité ainsi de mettre en pratique nos connaissances en matière de programmation avec des nouveaux concepts pour nous : la conception orientée objet et l'UML.

Dans cette perspective, on a développé un jeu, en utilisant des outils informatiques diverses qui nous ont aidés dans un premier temps de faciliter la réalisation et la conception, l'analyse et l'acheminement de notre projet.

2 Description de projet

L'objectif du projet est de mettre en place une application informatique au moyen d'approches issues du génie logiciel, et en particulier au travers d'une modélisation orientée objet.

On a défini un planning de travail, avec une estimation de l'organisation de notre temps. Puis on a réalisé les différents diagrammes UML .

Enfin, on a implémenté les diagrammes de classe permettant de réaliser le logiciel .

3 Répartition des taches

Réalisation des diagrammes :

MAHDI Chaymae, RHADI Meryam et REYNIER Tony.

Programmation des interfaces d'authentification et accueil :

MAHDI Chaymae et RHADI Meryam.

Programmation des interfaces du jeu et des différents fonctionnalités :

MAHDI Chaymae et RHADI Meryam .

4 L'ergonomie de l'application :

- Eclipse
- StarUML
- GanttProject

5 La modélisation des diagrammes de l'application :

5.1 Diagramme de gantt

Le diagramme de Gantt permet à présenter l'évolution de notre travail au cours du ce projet. En effet, nous avons essayé au maximum de respecter le temps estimé au diagramme de gantt de la première séance. et nous avons réussi d'atteindre ce point là.

5.1.1 Diagramme de Gantt pour la totalité du mini-projet(temps réel)

Nom	Date de début	Date de fin
Analyse	26/11/2020	27/11/2020
Prise en main du sujet	26/11/2020	26/11/2020
Analyse des différentes diagrammes de logiciel	27/11/2020	27/11/2020
Estimation de l'application finale	27/11/2020	27/11/2020
Modélisation	28/11/2020	08/01/2021
Réalisation de diagramme de cas d'utilisation	28/11/2020	02/12/2020
Réalisation de diagramme de classe	28/11/2020	18/12/2020
Réalisation de diagramme d'état-transition	28/11/2020	18/12/2020
Modification de diagramme de cas d'utilisation	30/12/2020	08/01/2021
Modification de diagramme de classe	30/12/2020	08/01/2021
Modification de diagramme d'état-transition	30/12/2020	08/01/2021
Conception	08/12/2020	15/01/2021
Structure générale du code	08/12/2020	11/12/2020
Création de la classes personnages et ces types	08/12/2020	18/12/2020
Implémentation du code de l'authentification	18/12/2020	08/01/2021
Implémentation de la page d'accueil	19/12/2020	08/01/2021
Implémentation du jeux	20/12/2020	15/01/2021

Figure 1. Les taches réalisée pour mini-projet version finale

Nom	Date de début	Date de fin
[-] • Analyse	26/11/2020	27/11/2020
• Prise en main du sujet	26/11/2020	26/11/2020
• Analyse des différentes diagramme...	27/11/2020	27/11/2020
• Estimation de l'application finale	27/11/2020	27/11/2020
[-] • Modélisation	28/11/2020	08/01/2021
• Réalisation de diagramme de cas d'...	28/11/2020	02/12/2020
• Réalisation de diagramme de classe	28/11/2020	18/12/2020
• Réalisation de diagramme d'état-tr...	28/11/2020	18/12/2020
• Modification de diagramme de cas...	30/12/2020	08/01/2021
• Modification de diagramme de cla...	30/12/2020	08/01/2021
• Modification de diagramme d'état-...	30/12/2020	08/01/2021
[-] • Conception	08/12/2020	15/01/2021
• Structure générale du code	08/12/2020	11/12/2020
• Création de la classes personnages ...	08/12/2020	18/12/2020
• Implémentation du code de l'authe...	18/12/2020	08/01/2021
• Implémentation de la page d'accueil	19/12/2020	08/01/2021
• Implémentation du jeux	20/12/2020	15/01/2021

Figure 2. Diagramme de gantt généré pour mini-projet version finale

5.1.2 Diagramme de Gantt initiale(temps estimé)

Tâches		
Nom	Date de début	Date de fin
ANALYSE	26/11/2020	27/11/2020
prise en main de sujet	26/11/2020	26/11/2020
analyse des différents diagrammes de logiciel	27/11/2020	27/11/2020
estimation du l'application final	27/11/2020	27/11/2020
MODELISATION	30/11/2020	04/01/2021
implémentation de diagramme de classe	30/11/2020	04/12/2020
classe personne	30/11/2020	01/12/2020
classe ville	01/12/2020	02/12/2020
classe batiment	02/12/2020	03/12/2020
classe circulation	03/12/2020	04/12/2020
implémentation de Diagramme de cas d'utilisation	07/12/2020	11/12/2020
classe personne	07/12/2020	08/12/2020
classe ville	07/12/2020	08/12/2020
classe batiment	08/12/2020	10/12/2020
classe circulation	09/12/2020	11/12/2020
implémentation de Diagramme d'états-transitions	17/12/2020	24/12/2020
classe personne	17/12/2020	18/12/2020
classe ville	18/12/2020	18/12/2020
classe batiment	21/12/2020	21/12/2020
classe circulation	22/12/2020	24/12/2020
implémentation de Diagramme de séquence ou communication.	28/12/2020	04/01/2021
classe personne	28/12/2020	28/12/2020
classe ville	29/12/2020	31/12/2020
classe batiment	31/12/2020	01/01/2021
classe circulation	01/01/2021	04/01/2021
CONCEPTION	08/12/2020	01/01/2021
implémentation de code de l'application	08/12/2020	31/12/2020
implémentation des interfaces	17/12/2020	01/01/2021

Figure 3. Les taches générale de projet

Diagramme de Gantt

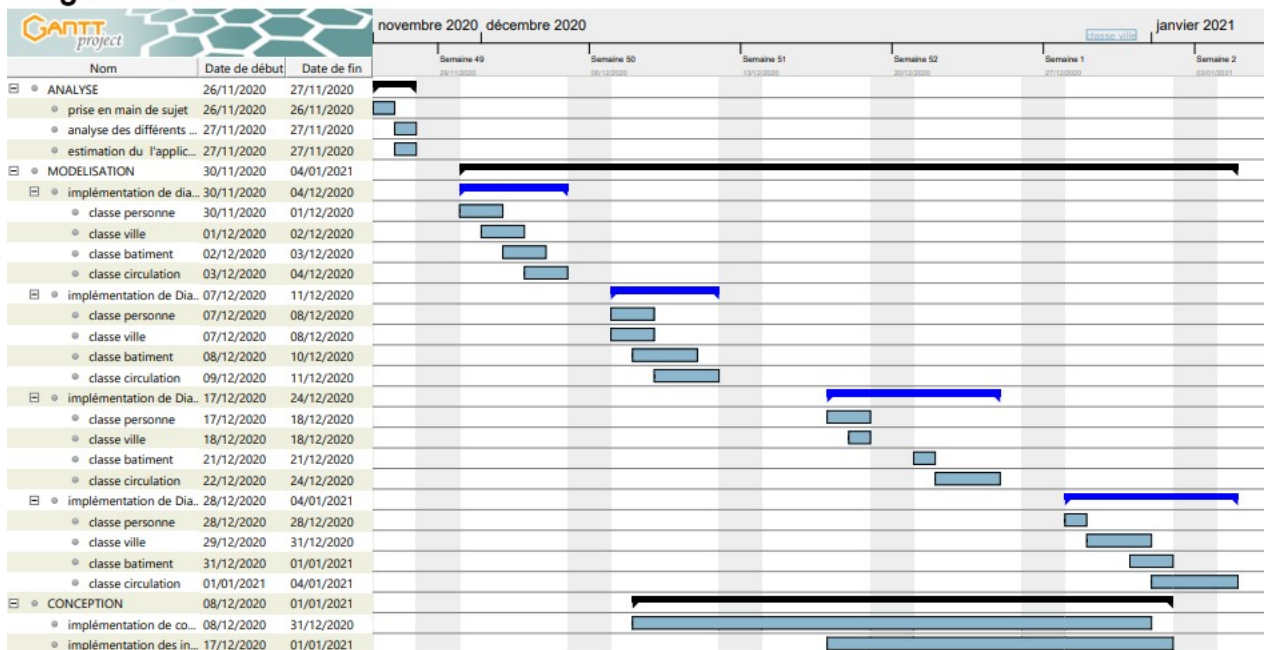


Figure 4. Diagramme de Gantt générale généré

5.2 Diagramme de cas d'utilisation

5.2.1 Implémentation

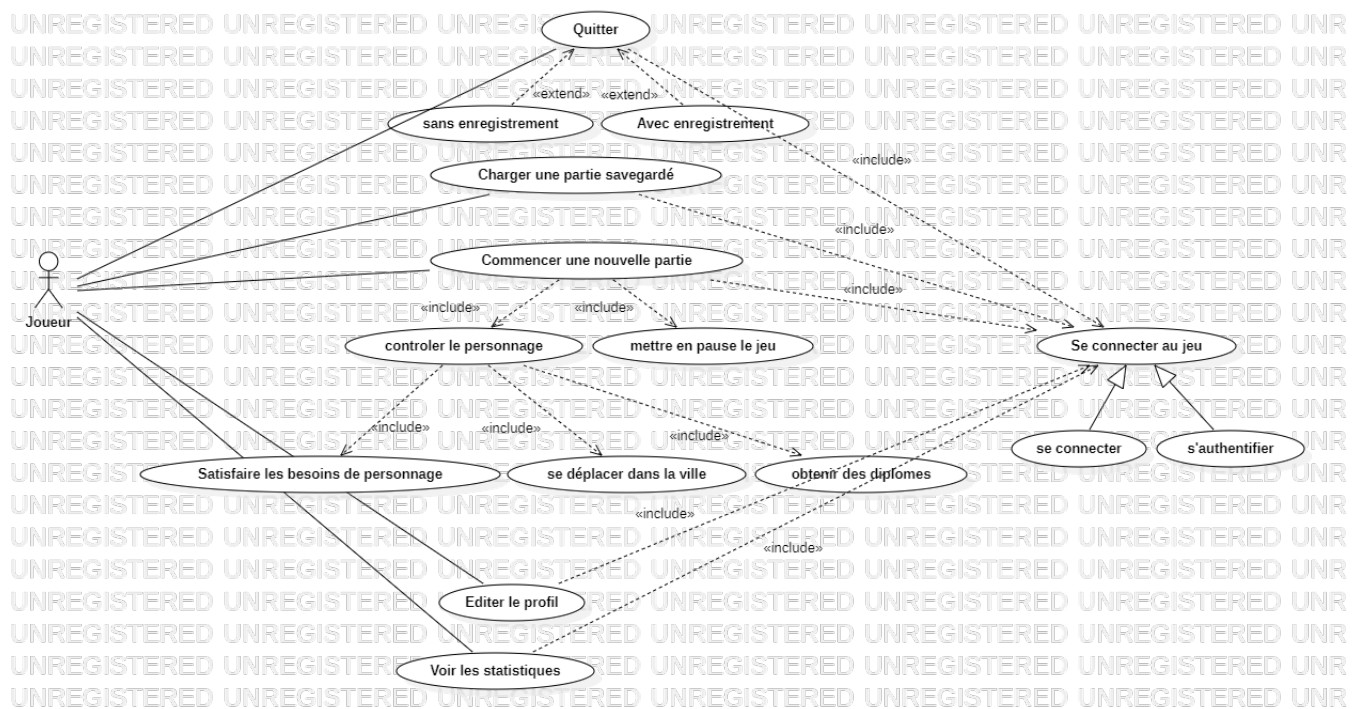


Figure 5. Diagramme de cas d'utilisation

5.2.2 Description textuelle

Le diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur (Joueur dans notre situation) extérieur le voit. Nous commençons donc par un joueur qui représente l'acteur principal de notre jeu.

Dans le jeu, l'acteur commence par son inscription qui peut être une simple connexion pour accéder aux fonctionnalités du jeu ou passer par l'authentification ou il va saisir son nom d'utilisateur, son mot de passe, et le type de joueur (pressé, standard, hippie) dans le cas où le joueur est nouveau.

Quand le joueur sera connecté, il accède directement à la page principale où il va soit commencer une nouvelle partie, charger une partie sauvegardée, consulter les statistiques (Barre de vie...), modifier son profil ou quitter le jeu avec enregistrement de la partie ou sans elle.

Une fois l'acteur commence à jouer, il a le droit de se déplacer dans toute la ville vers les différents endroits programmés dans le jeu, satisfaire ses besoins primaires (boire, soigner...) et obtenir des diplômes.

5.3 Diagramme de classe

5.3.1 Implémentation

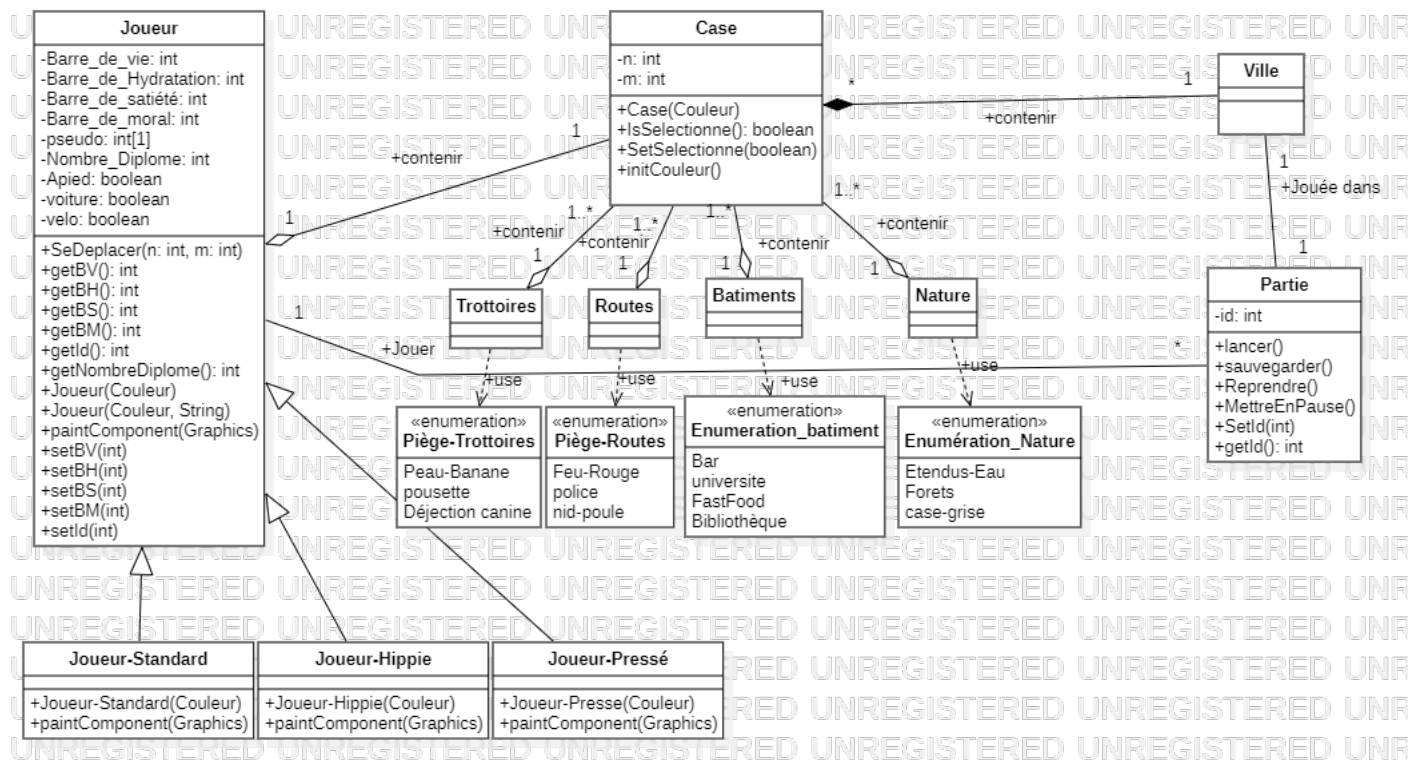


Figure 6. Diagramme de classe

5.3.2 Description textuelle

Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application.

En effet, L'objectif principal de ce jeu étant de simuler la vie d'un personnage au sein d'une ville .

classe Joueur

On va donc créer une première classe « Joueur », qui est de type abstrait pour empêcher de créer une instance depuis cette classe, et il a possédé de plusieurs attributs et méthodes. Pour les attributs de cette classe, nous avons mis :

- Barre de vie
- Barre d'hydratation
- Barre de satiété
- Barre de moral
- Le pseudo pour identifier chaque joueur
- le nombre de diplôme

Ce personnage a la possibilité de circuler librement dans la ville alors qu'il peut se déplacer :

- A pied
- En voiture
- A vélo

C'est pour ça nous avons pensé d'ajouter les différents possibilités des déplacements dans la classe personne comme des attributs de type booléen .

On a ajouté aussi des méthodes telles que `Sedéplacer()`, les constructeurs...

Et depuis cette classe "Joueur", on a créé des classes filles qui héritent de la classe personne, et on a spécifié un constructeur pour chaque sous classe dont on a initialisé les valeurs des attributs de la classe selon l'énoncé. qui spécifient exactement le type de personne qui peut être un :

- Joueur standard
- Joueur hippie
- Joueur pressé

classe Ville

La ville représente l'endroit principal de déplacement de notre personnage, elle se compose de plusieurs cases(n,m) ce qui fait nous avons mis une relation de composition entre les deux classes ces endroits sont soit :

- Bâtiments
- Routes
- Trottoirs
- Forêts
- Cases grisées
- Étendue d'eau

classe Case

la classe case contient plusieurs endroits soit de type «bâtiment», «routes», «trottoirs», «forêts», «cases grisées» ou «étendues d'eau » ce qui fait nous avons fait une relation d'agrégation entre la classe case et ses différents sous classes . Des différents bâtiments jalonnent la ville qui sont :

- Maison
- Université
- Bibliothèque
- Bar
- Fast-food

Pour les types des bâtiments dans la séance précédente, nous avons eu l'idée d'une relation d'héritage entre la classe bâtiment et ses sous-types mais lors de l'implémentation de code, nous avons remarqué que ces sous-types ne doivent pas être considérés comme des classes sachant qu'ils n'ont pas d'attributs ou de méthodes spécifiques.

Nous avons alors choisi que tout ce qui est type de bâtiment devra être pensé comme énumération de la classe bâtiment.

nous avons pensé à créer une nouvelle classe nature qui va utiliser une énumération de Forêts, cases grisées, Étendue d'eau. nous avons ajouté la classe trottoir qui utilise une énumération contient les pièges qui sont peau de banane, Poussette, Déjection canine ainsi que la classe de routes qui utilise également une énumération contient les pièges qui sont Feu rouge, Police, Nid de poule

classe Partie

nous avons pensé d'ajouter une autre classe partie selon notre code qui a une relation entre la classe ville et la classe joueur ,en effet le joueur peut jouer plusieurs partie dans le jeu et la partie jouée dans une seule ville .

5.4 Diagramme d'état-transition de joueur

5.4.1 Implémentation

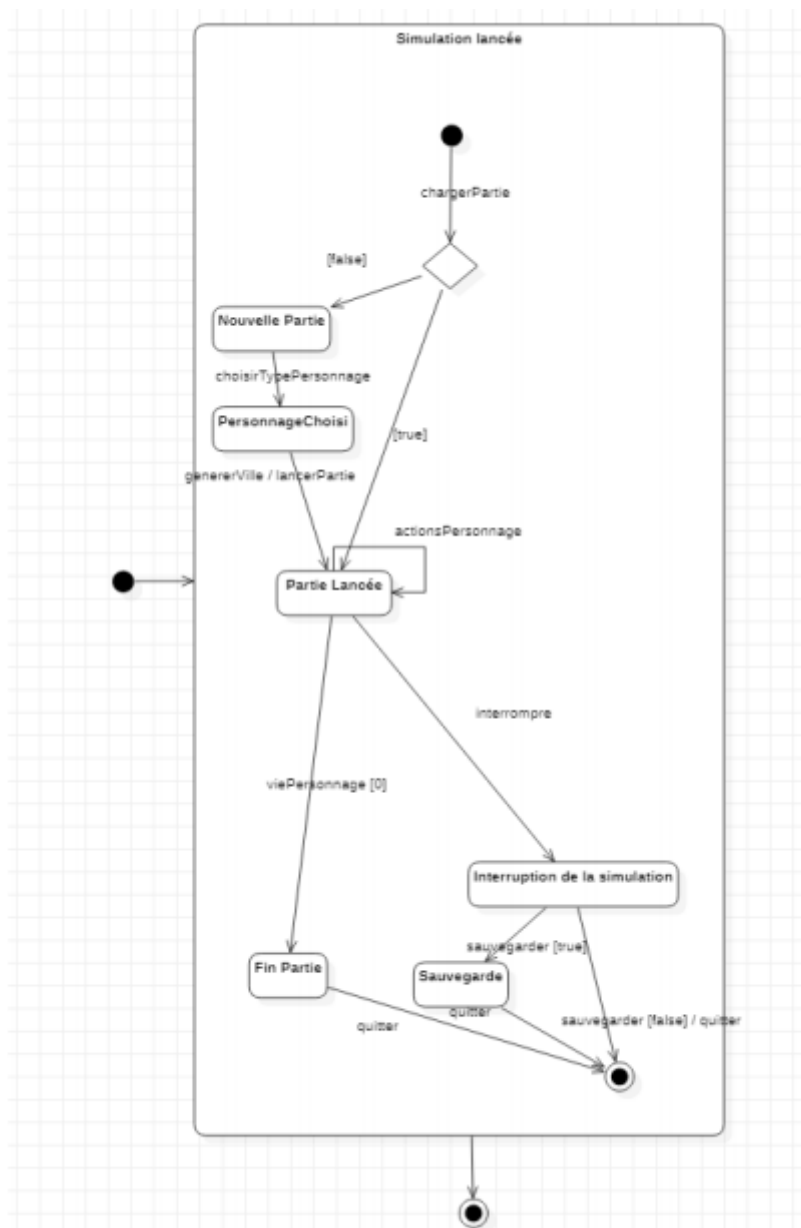


Figure 7. Diagramme d'état-transition

5.4.2 Description textuelle

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis.

En effet, tout commence lorsque l'on décide de lancer une simulation, qui en est soit une nouvelle, soit la reprise d'une sauvegarde d'une ancienne simulation encore en cours. Nous nous retrouvons alors dans la partie en train d'être jouée où le personnage peut faire ses actions librement jusqu'à ce que certains événements signifiant la fin (allant vers l'état final) surviennent.

Ces événements sont l'interruption (avec ou sans sauvegarde préalable pour reprendre plus tard) de la simulation ou la fin de vie du personnage, signifiant le "Game Over".

5.5 Diagramme d'état-transition de système

5.5.1 Implémentation

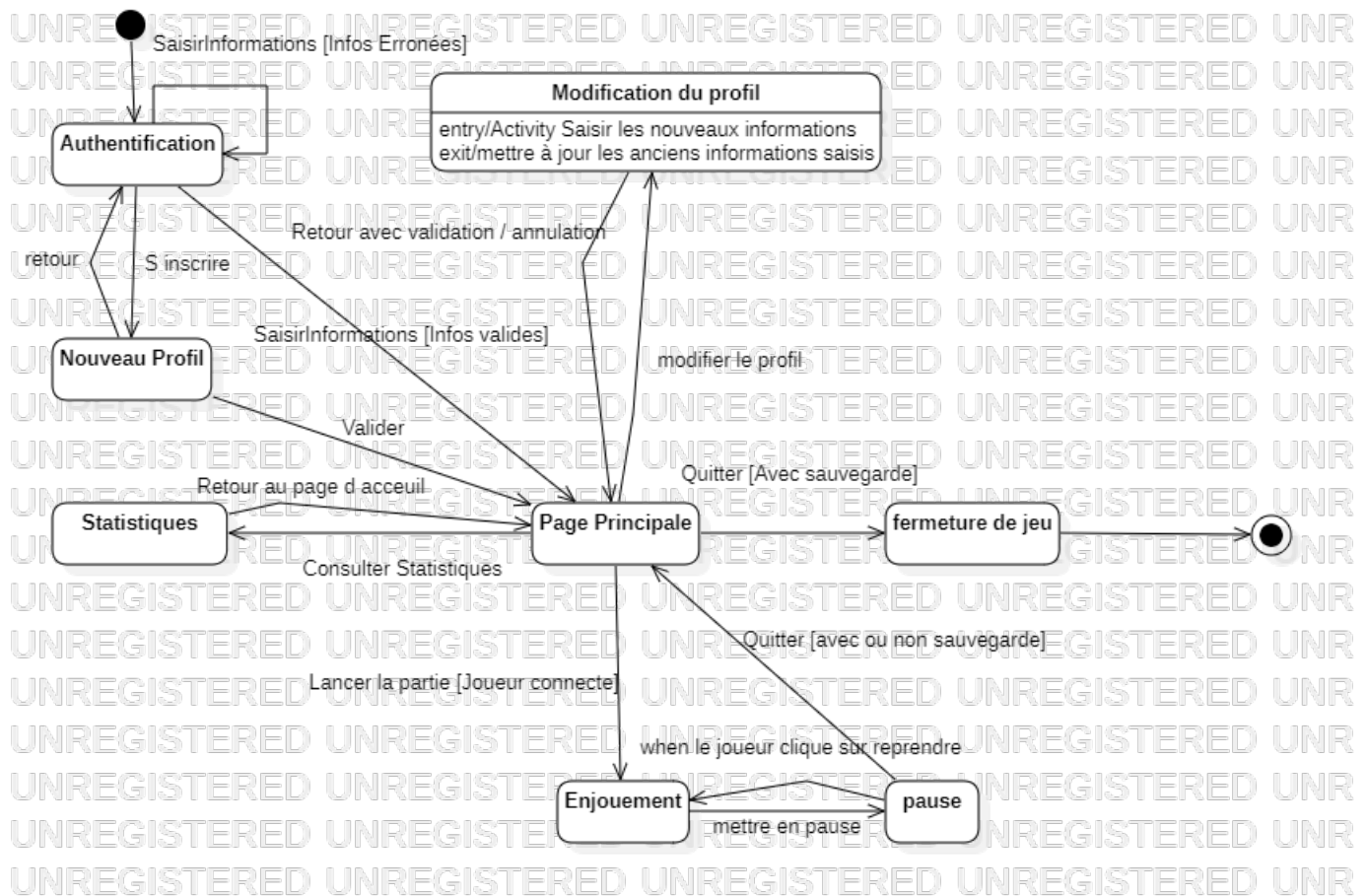


Figure 8. Diagramme d'état-transition

5.5.2 Description textuelle

Ce diagramme d'état transition a pour objectif d'implémenter la stratégie générale de l'utilisation de jeu ce qui fait nous avons en premier temps la connexion au jeu si le joueur a déjà inscrit sinon il faut faire son inscription avant d'accéder aux fonctionnalités de jeu qui sont trouvés dans une page principale où il peut jouer, modifier son profil, consulter les statistiques, mettre en pause le jeu, charger une partie sauvegardée ou quitter le jeu dont on arrive à notre état final.

5.6 Diagramme de séquence

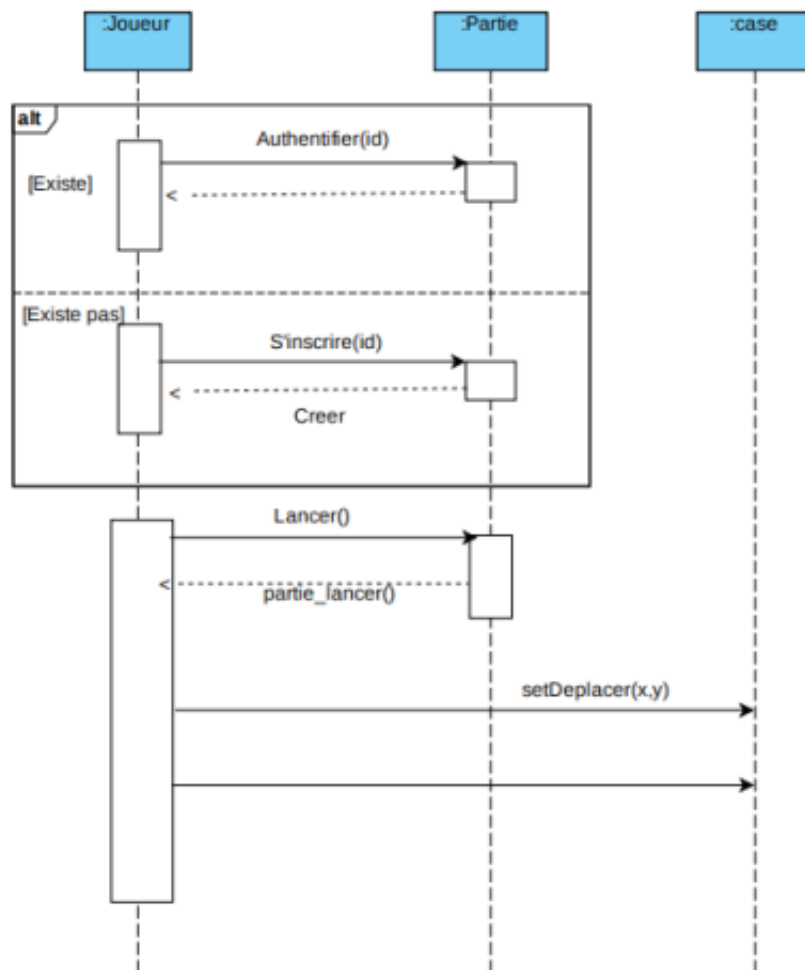


Figure 9. Diagramme de séquence

Le diagramme de séquence est une représentation graphique de la chronologie des échanges de messages avec le système ou au sein du système :

- « Vie » de chaque entité représentée verticalement.
- Échanges de messages représentés horizontalement.

6 Codage et programmation

6.1 Page de connexion

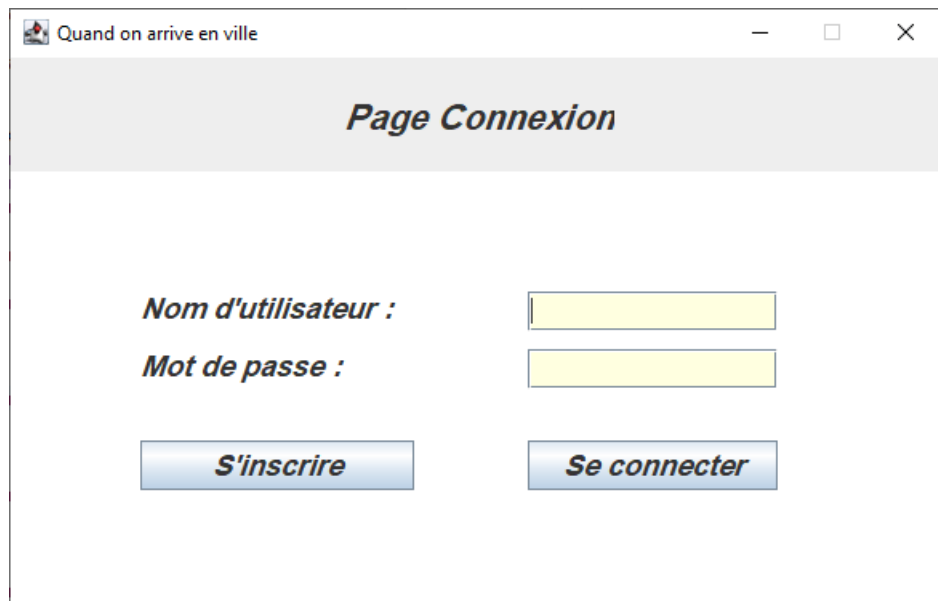


Figure 10. L'interface de connexion

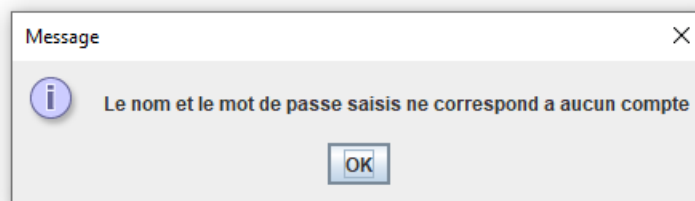
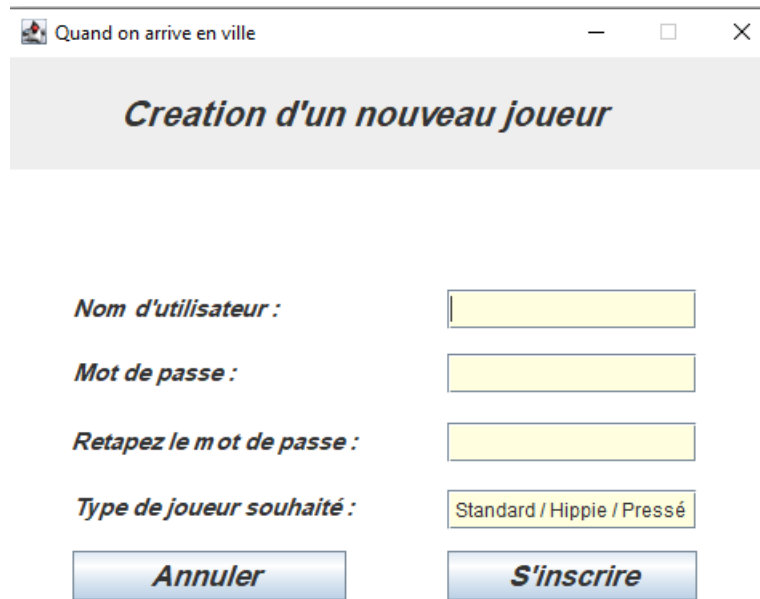


Figure 11. L'interface de connexion échoué

Cette interface permet au joueur de se connecter pour commencer à jouer, en entrant son nom d'utilisateur et mot de passe, sinon si il n'a pas encore de compte (nouveau) il a le droit de s'inscrire en entrant son nom d'utilisation, son mot de passe et le type de joueur souhaité.

6.2 Page d'inscription

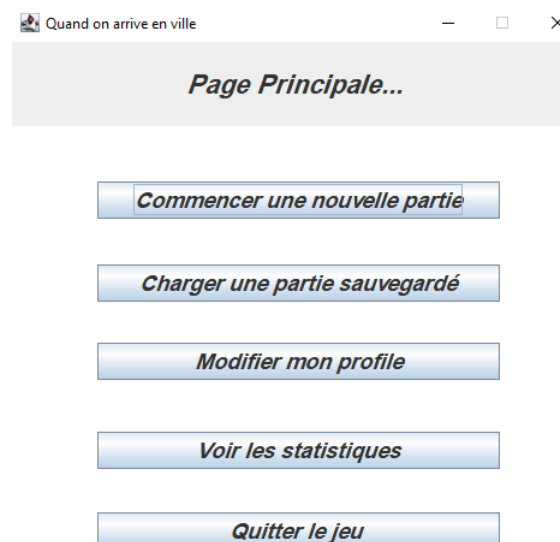


The screenshot shows a window titled "Quand on arrive en ville" with a subtitle "Creation d'un nouveau joueur". It contains four input fields: "Nom d'utilisateur :", "Mot de passe :", "Retapez le mot de passe :", and "Type de joueur souhaité :". The "Type de joueur souhaité" field has a dropdown menu with options "Standard / Hippie / Pressé". At the bottom, there are two buttons: "Annuler" and "S'inscrire".

Figure 12. L'interface d'inscription

Cette interface permet à créer un nouveau joueur, en remplissant les champs requis, et on choisissant le type de joueur, ses informations d'inscription sont stocké dans un fichier **.txt** pour qu'on puisse vérifier après lors de la connexion.

6.3 Page d'accueil



The screenshot shows a window titled "Quand on arrive en ville" with a subtitle "Page Principale...". It contains five buttons: "Commencer une nouvelle partie", "Charger une partie sauvegardé", "Modifier mon profile", "Voir les statistiques", and "Quitter le jeu".

Figure 13. L'interface d'accueil

L'interface de la page d'accueil du jeu, permet soit de commencer une nouvelle partie, charger une partie sauvegardé, éditer son profile on changeant le mot de passe ou le type de joueurs, voir les statistiques qui présente les caractéristiques de la vie du joueur et sortir de jeu.

6.4 Modification de profil



The screenshot shows a window titled "Quand on arrive en ville" with standard window controls (minimize, maximize, close). The main content area has a light gray background with the title **Modification de Profile** in a bold, italicized font. Below the title, there are three labels with corresponding input fields:

- Nouveau nom d'utilisateur :** followed by a yellow rectangular text input field.
- Nouveau mot de passe :** followed by a yellow rectangular text input field.
- Nouveau Type de joueur:** followed by a yellow rectangular dropdown menu showing the options "Standard / Hippie / Pressé".

At the bottom of the form, there are two blue buttons with white text: **Annuler** on the left and **Modifier** on the right.

Figure 14. L'interface de modification de profil

6.5 L'interface de jeu

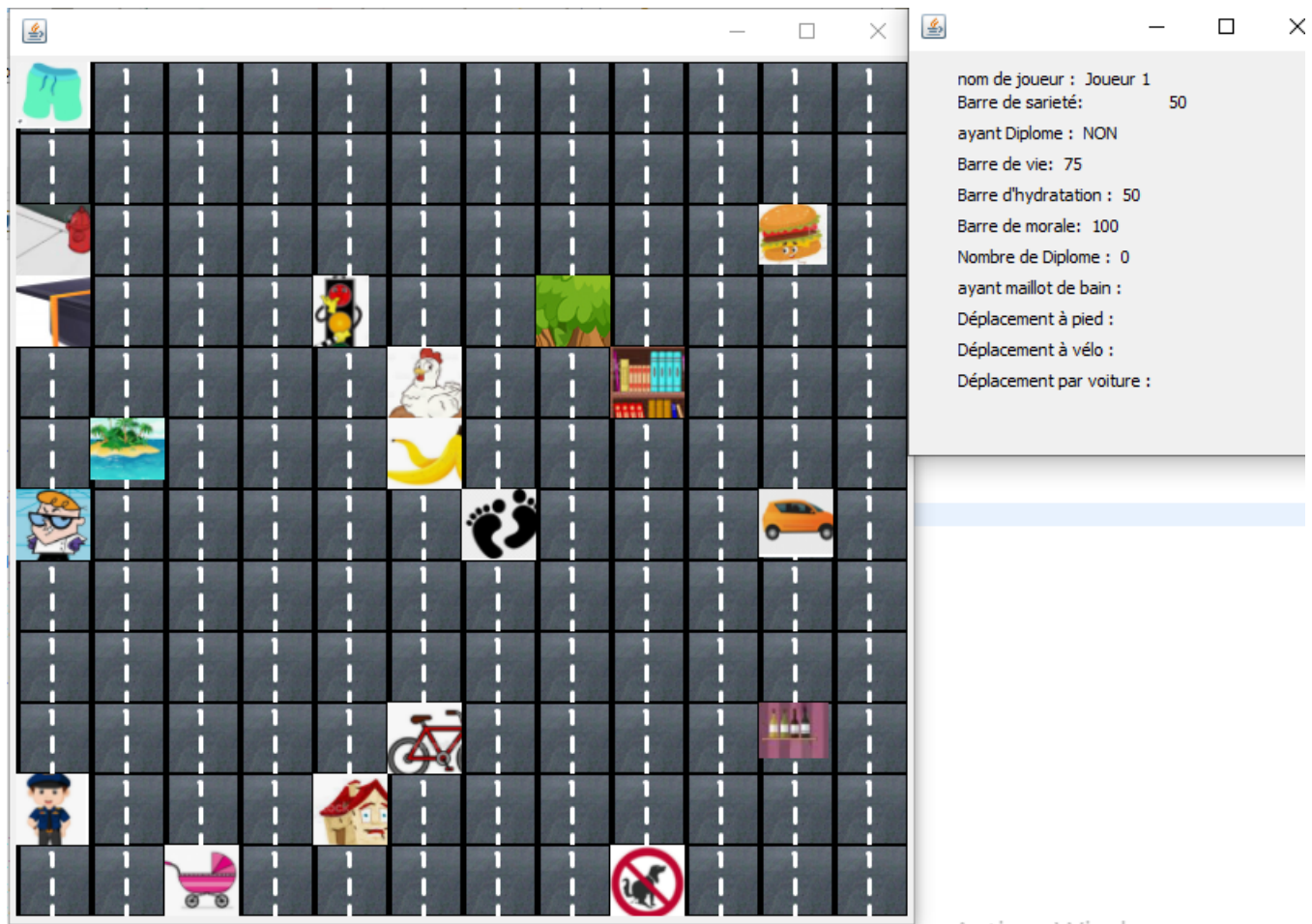


Figure 15. L'interface du jeux

Cette interface sert à présenter un plateau de case qui permet au joueur de se déplacer dans les différentes cases, soit l'université, restaurant, bar, maison, routes,... tout cela est représenté avec des photos dans chaque case, et qui contient au même temps des méthodes pour faire le traitement nécessaire sur chaque case selon son inutilité, et on a aussi une petite fenêtre à côté qui représente le statut du joueur et ses barres.

Sinon quand le joueur a eu fin des ces barres (barre de vie = 0...), une popup qui annonce la finalité du jeu déclenchera.

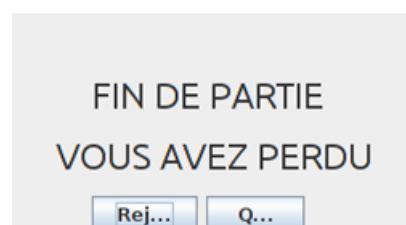


Figure 16. L'interface du jeux

7 Conclusion

Ce projet nous a permis de connaître tous les contours de la modélisation UML et les paramètres qui interviennent dans la programmation Java . Nous déplorons le fait de n'avoir pas pu réaliser une application un peu plus évoluée avec Authentification et création de plusieurs sessions ; cela à cause d'une difficulté d'acquisition du langage de programmation nécessaire et contrainte de temps. En dehors de cela nous n'avons pas rencontré de difficultés majeures ; nous avons pu récolter un grand nombre d'informations en matière de la maitrise de la modélisation UML et s'y adapter au niveau applicatif.