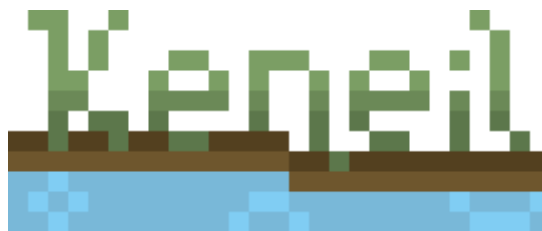


## Projet de simulation d'un tamagotchi

# Spécifications



**Client :** M. Rémy Kessler

**Module :** Génie Logiciel Objet (INF1503)

## Table des matières

I.	Cahier des charges .....	1
II.	Diagramme de cas d'utilisation .....	10
III.	Cas d'utilisation.....	13
IV.	Diagrammes de séquence .....	27
V.	Diagramme de Gantt .....	30

## I. Cahier des charges

# CAHIER DES CHARGES

Version : 5.0

Date : 01/12/2021

<i>Client</i>	<i>Prestataire</i>
Monsieur Rémy KESSLER	ADAM Antoine (Développeur senior) BUCHE Sylvain (Chef de projet) COBAT Guillaume (Analyste des besoins) DEWILDE Yoann (Architecte logiciel) LAGUE Pierre (Gestionnaire qualité/validation)

## INTRODUCTION

---

### Objet du document

Ce document décrit **tous** les services que doivent rendre le produit et ses livrables et toutes les exigences qu'ils doivent satisfaire.

### Portée du document

Ce document est destiné à formaliser le besoin du client, Monsieur Rémy KESSLER dans le cadre du projet Tamagotchi évalué à la fin du premier semestre de 3<sup>e</sup> année de Licence informatique à l'Université Bretagne Sud.

### Terminologie

Terme	Description
Java	Java est un langage de programmation créé par Sun Microsystems en 1995. Java est un langage de programmation compilé, très adapté au développement d'applications.
JVM	Java Virtual Machine (Machine Virtuelle Java) – permet de charger, vérifier et exécuter les programmes java. Elle est connue pour être l'interpréteur Java, car elle permet d'exécuter des programmes java.

Terme	Description
SDK	Software Development Kit (Kit de Développement de Logiciel) – est une collection de logiciels et outils de développements qui peuvent être installés en packages. Ils sont spécifiques à un type de matériel informatique et de système d'exploitation.
Manger	L'avatar se nourrit afin de gagner des points dans sa jauge de nourriture. L'avatar ne peut réaliser cette action que quand il est dans la cuisine.
Dormir	L'avatar dort/se repose et cela lui fait gagner des points d'énergie. Cette action permet d'augmenter ses points d'énergie. L'avatar ne peut réaliser cette action que lorsqu'il est dans sa chambre.
Se laver	L'avatar se lave afin de gagner des points d'hygiène. Il ne peut réaliser cette action que lorsqu'il est dans la salle de bains.
Jouer	L'avatar joue pour gagner des points de divertissement. Il ne peut réaliser cette action que quand il est dans le jardin.
Sigle PEGI 3	Le contenu des jeux classés PEGI 3 est considéré comme adapté à tous les groupes d'âge. Le jeu ne doit pas contenir de sons ou d'images susceptibles d'effrayer les jeunes enfants. Une forme de violence très légère (dans un contexte comique ou enfantin) est acceptable. Aucun langage grossier ne doit être entendu.
Glitch	Événements inattendus déclenchés par une action de l'utilisateur dans le but de tester les limites du logiciel et de produire un plantage ou un comportement anormal. Les glitches ne sont pas prévisibles par l'équipe de développement.
Produit	Terme générique désignant l'objet de la demande du client. Il recouvre aussi bien un système qu'un service, sans préjuger de la part de logiciel et de matériel intervenant dans la réalisation.

## Abréviations

Abréviations	Signification	Libellé
UML	Unified Modelling Language	Principe de modélisation d'une application suivant l'agencement de graphiques représentant le fonctionnement de l'application.
MVC	Modèle – Vue - Contrôleur	MVC est un patron de design de logiciel utilisé pour développer des applications avec interfaces utilisateur.
JAR	Java Archive	Utilisées pour agréger plusieurs fichiers java dans l'optique de créer un programme exécutable.
GIT	Global Information Tracker	Outil utilisé par des développeurs pour gérer des codes sources. Il est utilisé pour gérer les versions d'un logiciel et est très adapté à des travaux en groupe.
VIT	<i>Vitale</i>	<i>Exigences fonctionnelles ou non fonctionnelles indispensables</i>
IMP	<i>Importante</i>	<i>Exigences souhaitées, mais non exigées</i>
MIN	<i>Mineure</i>	<i>Exigences non exigées immédiatement, mais qui devront être prises en compte ultérieurement par le produit (impact sur l'évolutivité)</i>

# LES OBJECTIFS DU PRODUIT

---

## Définition du produit

Le produit constituant le livrable final est une application appelée « Keneil » (« ami » en breton). Cette application est inspirée de la fameuse « Egg Watch » - Tamagotchi créé au Japon par Akihiro Yokoi et Aki Bandai en 1996. Cette application de simulation consiste à gérer l'éducation et le développement d'un avatar à l'aide d'une console miniature. L'application sera exécutable sur ordinateur et aura une interface graphique ainsi qu'un système de persistance des données afin de permettre à l'utilisateur de sauvegarder/reprendre une partie en cours.

## Contexte économique du produit

Ce produit sera positionné sur le marché des jeux de divertissement et des jeux vidéo. Il sera destiné à une clientèle relativement jeune, mais a pour but d'avoir un essor particulier vers de la clientèle plus âgée ayant joué au jeu original. Les zones géographiques de vente se feront d'abord localement, puis après avoir dépassé un seuil, les ventes pourront s'étendre nationalement.

En tenant compte de la nationalité du jeu original, l'ouverture aux autres pays commencera avec une relocalisation des ventes au Japon pour pouvoir toucher un maximum de clientèle.

## Contexte d'exploitation du produit

À ce jour, nous estimons à environ 10 millions le nombre d'enfants de moins de 15 ans en France. Cette application devrait pouvoir toucher un maximum de ces enfants si ce n'est la totalité.

Le code source sera entièrement développé et maintenu par les membres de l'équipe du projet. Une fois prêt, il faudra trouver une entreprise de déploiement d'application sur des plateformes de jeux vidéo (Steam FR, Ubisoft, Epic Games FR, Battle.net, Miniclip, Jeux.fr).

L'application sera proposée comme « Free to play » (gratuite) et après une période - et selon les ventes de l'application - des contenus additionnels seront développés et les clients pourront payer pour y accéder.

À ce jour, la console originale Tamagotchi est encore populaire et est considérée comme console de collection pour les premiers modèles. D'autres jouets similaires sont sortis dans les années 2000 comme « Giga Pets », « Nano Pets ». Plus récemment un jeu sur Nintendo Switch appelé « Pokémon Pikachu » est sorti et a connu un franc succès auprès de la communauté japonaise des jeux type Tamagotchi ainsi que chez les jeunes enfants possédant la console.

Il est impératif de savoir lire pour pouvoir profiter de toutes les fonctionnalités du jeu.

# EXIGENCES SUR LE PRODUIT

---

Dans ce chapitre, les exigences seront décrites sans entrer dans le détail de la mise en œuvre.

## Capacités fonctionnelles

### Description des fonctionnalités

#### Fonctionnalité graphique :

Le système devra mettre à disposition une interface graphique « user-friendly » intuitive permettant à l'utilisateur d'utiliser l'application. Cette interface affichera l'avatar de l'utilisateur ainsi que tout autre élément visuel nécessaire à l'avancement de la partie (boutons, caractéristiques de l'avatar, animations de l'environnement, etc.). En tout le jeu proposera **11** pages dans sa première version :

- Une **page de lancement du jeu**. L'utilisateur pourra choisir de lancer une nouvelle partie, de charger une ancienne partie, de quitter le jeu, de consulter les règles et de consulter les options.
- À partir de la page de lancement, 4 autres pages seront accessibles : la **page des règles** qui récapitule l'utilisation du logiciel, la **page de création de l'avatar** quand l'utilisateur veut lancer une nouvelle

partie, la **page de choix de la sauvegarde** et enfin la **page des options**.

- Après avoir créé son avatar, l'utilisateur se trouvera sur **la page de jeu** où il pourra gérer son avatar et ses caractéristiques. Il pourra accéder à la page des règles, celle des options et celle de **la sauvegarde de partie**.
- Depuis la page de jeu, l'avatar sera amené à changer de pièce/environnement. Il pourra accéder aux pièces suivantes : **le jardin, la cuisine, la salle de bain, la chambre**.

#### **Description des mécaniques en lien avec les caractéristiques de l'avatar :**

L'avatar aura 2 jauges principales : **la santé et le bonheur**, si une de ces 2 jauges arrive à 0, c'est la fin de la partie.

L'avatar aura 4 jauges secondaires : **l'hygiène, l'énergie, la nourriture, le divertissement**.

- **Hygiène** : se laver augmentera la jauge de 15%. Si elle est remplie à plus de 90%, elle commencera à décroître. Si elle est inférieure à 30%, la jauge de **santé** commencera à décroître. Si elle est entre 30% et 90%, la jauge de **santé** augmente.
- **Énergie** : dormir fait augmenter l'énergie de 50% pour 8h de sommeil (temps réel) et de 10% pour 1h30 de sieste. Si l'énergie est supérieure à 90% alors elle commencera à décroître. Si elle est inférieure à 30%, la jauge de **bonheur** va commencer à décroître. Si elle est entre 30% et 90%, la jauge de **bonheur** augmente. Enfin, si l'énergie est inférieure à 50%, le délai entre les actions est plus important.
- **Nourriture** : Si la jauge de nourriture est supérieure à 90% alors la jauge de **divertissement** augmente moins vite. Si elle est inférieure à 30% alors la **santé** décroît. Si la jauge est entre 30% et 90%, la **santé** augmente.
- **Divertissement** : Si la jauge de **divertissement** est supérieure à 90% alors **l'hygiène** décroît plus vite. Si la jauge est inférieure à 30% alors le **bonheur** décroît. Si elle est située entre 30% et 90%, alors le **bonheur** augmente.

Des événements hebdomadaires aléatoires pourront influencer sur l'évolution des caractéristiques de l'avatar (par exemple : l'avatar tombe malade, l'avatar fête son anniversaire, l'avatar se blesse en jouant ...).

#### **Fonctionnalités relatives au développement de l'avatar :**

Une fois l'application lancée, l'utilisateur pourra choisir le type de son avatar ainsi que son nom. Chaque avatar aura des caractéristiques qui lui seront propres.

L'avatar devra posséder plusieurs actions : manger, dormir, se laver, boire, se promener, changer d'environnement. Chacune de ses actions va influencer sur ses caractéristiques : hygiène, propreté, faim, soif (nourriture), condition physique (santé), conditions mentales (bonheur), envie d'aller aux toilettes. Mais ces caractéristiques sont amenées à changer au cours du temps (la faim, la soif, l'inactivité, etc.), beaucoup de facteurs seront la cause de la baisse ou de la hausse des caractéristiques de l'avatar. Afin de bien réguler les caractéristiques de son avatar, l'utilisateur devra lui faire réaliser des actions (expliqué juste au-dessus de ce point).

La santé et le bonheur évolueront en fonction des 4 autres jauges (hygiène, énergie, nourriture et divertissement).

L'utilisateur aura aussi la possibilité de promener son avatar et de le déplacer dans une maison (jardin, salon, cuisine, chambre, toilettes). L'avatar aura accès à certaines pièces en fonction de son emplacement (si l'avatar est dans le jardin, il ne peut accéder qu'au salon ou à la cuisine et pas sa chambre). Chaque lieu proposera des actions différentes pour ajuster les caractéristiques de l'avatar. Enfin, une gestion de la météo sera implémentée afin que le cycle d'activité de l'avatar ne soit pas redondant, et pour proposer une expérience plus intéressante à l'utilisateur.

#### **Fonctionnalités liées à la sauvegarde des parties :**

Si l'utilisateur décide de quitter l'application après s'être occupé de son avatar, il aura la possibilité de sauvegarder sa partie sur son ordinateur. Une fois de retour sur l'application il pourra charger sa partie et pourra reprendre soin de son avatar tel qu'il l'avait laissé. Les caractéristiques auront baissé en fonction du temps que l'utilisateur aura passé loin de son avatar. L'évaluation des caractéristiques sera basée sur la différence de temps de l'horloge interne de l'appareil sur lequel l'application a été installée.

## Interopérabilité

Le logiciel ne devrait pas avoir de lien avec tout autre système d'information ou périphérique externe autre que l'ordinateur sur lequel il est installé.

## Conformité réglementaire

Réglementation liée à l'utilisation du jeu :

Les précautions à prendre dans tous les cas pour l'utilisation d'un jeu vidéo sont les suivantes :

- Évitez de jouer si vous êtes fatigué ou si vous manquez de sommeil.
- Assurez-vous que vous jouez dans une pièce bien éclairée et modérez la luminosité de votre écran.
- Lorsque vous utilisez un jeu vidéo susceptible d'être connecté à un écran, jouez à bonne distance de cet écran et aussi loin que le permet le cordon de raccordement.
- En cours d'utilisation, faites des pauses de dix à quinze minutes toutes les heures.

Le jeu ne sera pas concerné par la réglementation sur l'épilepsie, car il ne pourra en aucun cas mettre en danger une personne photosensible.

Ce jeu sera accessible aux enfants sous la responsabilité des parents, le sigle PEGI 7 apparaîtra sur la page de lancement du jeu sur la plateforme de distribution.

Avant de lancer le jeu, une page d'avertissement apparaîtra pour mettre en garde les utilisateurs des indications précédentes. L'utilisateur devra valider la page en cliquant avec sa souris.

Aucune donnée utilisateur ne sera récoltée lors de l'utilisation du programme.

## Persistance des données

Dans la fonction de sauvegarde, l'instance de l'avatar, l'instance du lieu dans lequel il se trouvait et un timestamp seront sérialisés (via le processus de sérialisation d'objets disponible sur Java) et stockés dans 2 fichiers (un fichier de sauvegarde et un fichier de back-up). Avant la sauvegarde, une vérification de l'espace de stockage disponible sera effectuée et l'utilisateur sera prévenu s'il y a une erreur dans le processus.

## Exigences non fonctionnelles

### Fiabilité

Le taux de défaillance maximum pour ce produit dépendra de la régularité avec laquelle l'utilisateur le met à jour. En fonction de la qualité de la maintenance du jeu, le temps avant la première défaillance du jeu devrait être d'au moins 2 ans. Cependant, cette défaillance dépendrait uniquement du jeu. Il est possible que le système de l'utilisateur ne soit pas en bon état et écourte la durée.

Dans ce cas, le produit aura été mis au point pour effectuer des sauvegardes automatiques pour éviter que le joueur ne perde toute sa progression si son système fait planter le jeu.

Si le jeu est joué sur un ordinateur, il est possible que la modification de la structure des fichiers sources provoque des erreurs au lancement ou pendant la partie. C'est pourquoi il sera notifié au joueur de ne pas modifier les dossiers du jeu. Dans ce cas, l'équipe de production ne sera pas tenue responsable du problème résultant de cette manipulation.

Dans le cas où le joueur aurait découvert un « glitch », il est impossible de prédire le comportement du jeu. Il sera toutefois difficile d'en trouver, car le panel d'actions est assez limité et le jeu sera géré par des exceptions qui mettront fin aux tentatives de découverte de glitches du joueur.

Dans le cas où le jeu planterait, l'utilisateur doit être en mesure de relancer le jeu pour reprendre sa partie. Un rapport d'erreur sera disponible dans le répertoire du jeu.



## Sécurité

Le logiciel étant un logiciel jouable hors-ligne, aucune information sur les connexions de l'ordinateur ne seront enregistrées, ni sur les profils utilisateur du joueur. Seulement les informations concernant la sauvegarde d'une partie seront récoltées.

Un système de redondance est prévu pour la fonctionnalité de sauvegarde. Si un fichier est compromis, le back-up sera utilisé. Une sauvegarde automatique sera effectuée toutes les 5 minutes par défaut. Le joueur pourra aussi sauvegarder sa partie manuellement avant de quitter le jeu.

## Facilité d'utilisation

Le logiciel a pour but de toucher une clientèle jeune. L'interface graphique sera donc très intuitive :

- Un manuel d'utilisation très détaillé fourni avec le jeu
- Textes clairs, boutons visibles
- Interface épurée et compréhensible
- Aide accessible en jeu via un bouton dans le menu

Un utilisateur lambda sera en mesure de jouer au jeu sans aide en 20 minutes. Les fonctionnalités de base sont très simples.

Cependant, certaines stratégies liées au développement de l'avatar prendront plus de temps à être assimilées (quelques jours pour pouvoir évaluer les phases de sommeil de l'avatar et les pertes en santé par exemple).

## Rendement

Les délais de réalisation des différentes actions dans le jeu ne seront pas observables à l'œil nu (ordre de la milliseconde). Dans le cas où l'action engendre une animation, le délai sera intrinsèquement lié à la durée de l'animation.

Le jeu ne sera jouable que par un seul utilisateur.

La taille d'un jeu équivalent dans son archive .jar est d'environ 80 Ko. Nous estimons que la version finale du produit aura une taille d'environ 90 Ko.

## Maintenabilité

Les mises à jour seront transmises à la plateforme de distribution du jeu. Puis l'architecture des fichiers sera mise à jour en écrasant l'ancienne (tout en préservant les fichiers de sauvegarde).

Le système sera flexible, donc une modification n'entraînera que peu de changements et ne modifiera pas la structure du code. Cela augmentera donc la facilité de correction et la facilité à faire évoluer le logiciel.

De plus le patron de conception utilisé (MVC) assure une maintenance efficace de par son caractère organisationnel du code.

Après une mise à jour, des notes de **patch** seront disponibles pour informer les utilisateurs des changements.

## Portabilité

Le logiciel sera conçu pour pouvoir être exécuté sur les systèmes d'exploitation : UNIX, Windows et OS X.

Cependant il sera impératif d'avoir une JVM de version correspondante à celle utilisée dans le jeu.

# Exigences concernant le développement du produit

## Objectifs de délais

Dans le cadre de ce projet, trois rendus sont programmés :

1. Un rendu le 27 octobre 2021 contenant :
  - a. Cahier charges,
  - b. Diagramme de cas d'utilisation,
  - c. Cas d'utilisation (min. 3),
  - d. Cas de test (min. 3 x 5 scénarii = min. 15 scénarii tests)
  - e. Diagrammes de séquence
  
2. Un deuxième rendu le 1er décembre 2021 contenant :
  - a. Cahier charges (version modifiée),
  - b. Diagramme de cas d'utilisation (version modifiée),
  - c. Cas d'utilisation (min. 3) (version modifiée),
  - d. Cas de test (min. 3 x 5 scénarii = min. 15 scénarii tests) (version modifiée),
  - e. Diagramme UML de classes,
  - f. Diagrammes de séquence (min. 3) (pour les scénarii des cas d'utilisation décrits à l'étape précédente),
  - g. Schéma conceptuel des données persistantes.
  
3. Un troisième rendu le 4 janvier 2022 :
  - a. Tableau de suivi de l'avancement du projet, idéalement avec un diagramme Gantt, etc.,
  - b. Diagramme de cas d'utilisation,
  - c. Cas d'utilisation (min. 3),
  - d. Cas de test (min. 3 x 5 scénarii = min. 15 scénarii tests),
  - e. Diagramme UML de classes,
  - f. Diagrammes de séquence (min. 3) (pour les scénarii des cas d'utilisation décrits à l'étape précédente),
  - g. Schéma conceptuel des données persistantes.
  - h. Code source, tests unitaires et un fichier readme.txt (avec le nom de l'équipe et les prénoms et noms des membres de l'équipe).
  - i. Archive jar exécutable avec le jeu

## Objectifs de coûts

L'équipe de développement ne sera pas rémunérée pour ce projet. On estime le nombre d'heures par semaine d'environ 3-4 heures consacrées au projet.

## Exigences de réalisation

Le projet sera réalisé en utilisant deux IDE (Environnement de développement intégré), Visual Studio Code et IntelliJ Idea, en fonction des préférences des membres de l'équipe.

Le patron de programmation utilisé sera le patron MVC cela facilitera l'implémentation de l'interface utilisateur et la liaison entre les fichiers.

Le logiciel sera programmé en Java.

# SYNTHÈSE DES EXIGENCES

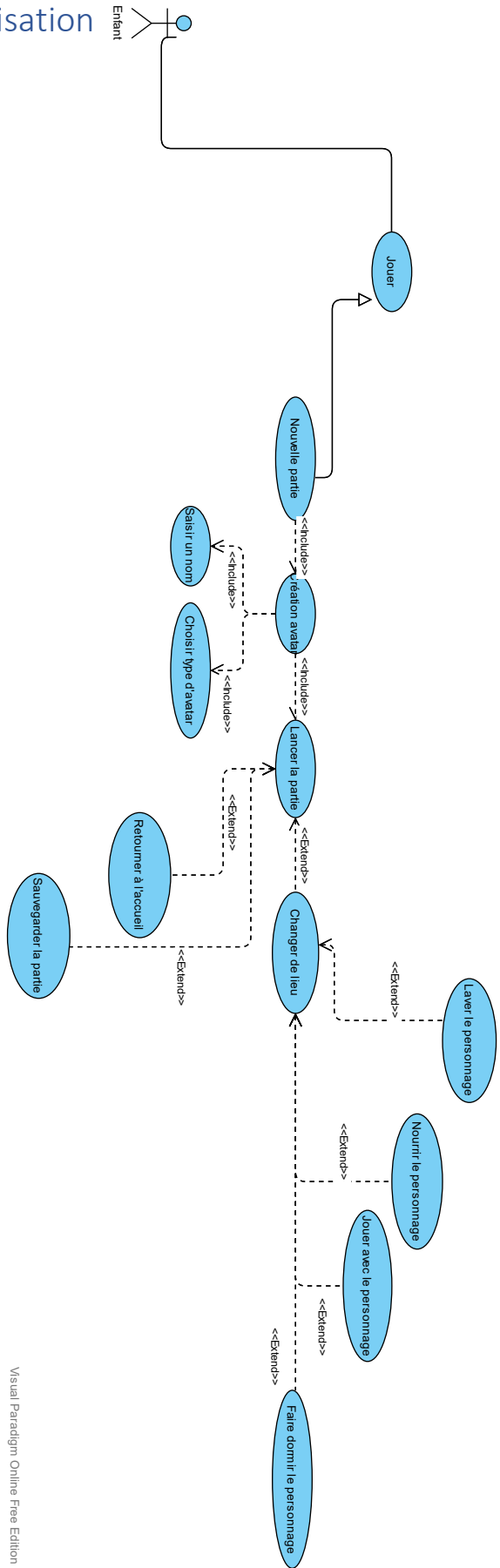
---

## Hierarchisation des exigences fonctionnelles

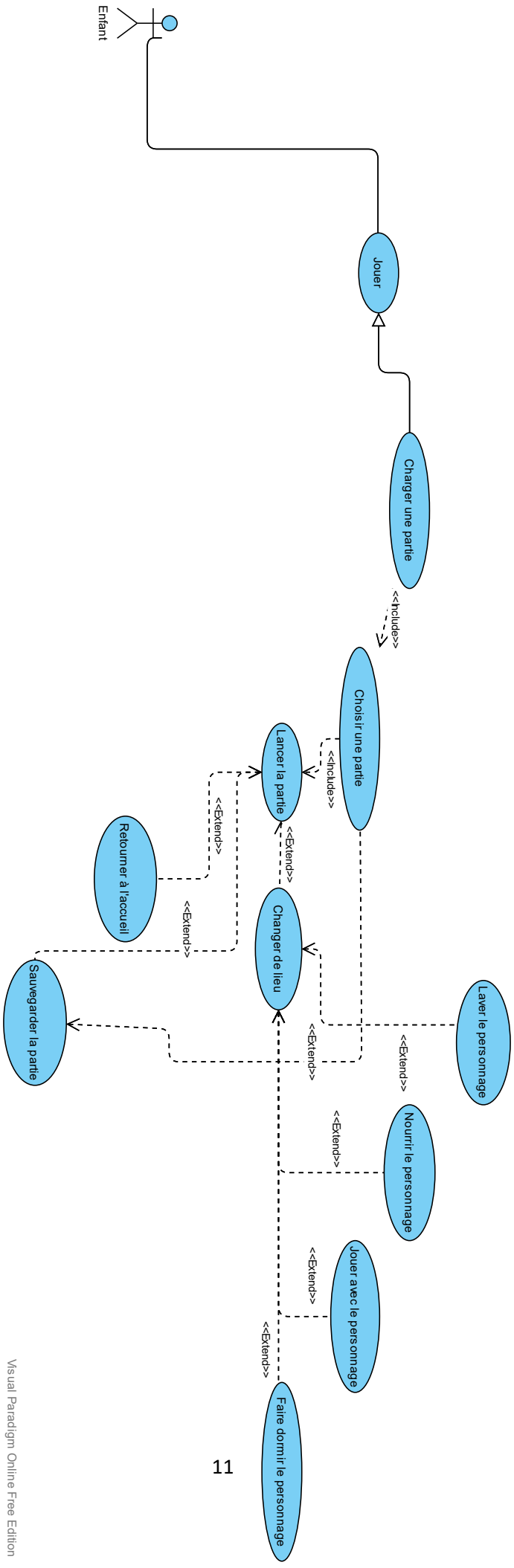
Nom fonction	Importance
Fonction de navigation dans l'application	VIT
Fonctions de contrôle des caractéristiques de l'avatar	VIT
Fonction d'affichage de l'avatar sur l'interface utilisateur en jeu	VIT
Fonctions de l'environnement dans lequel se trouve l'avatar	VIT
Fonctions d'animations de l'avatar sur l'interface utilisateur en jeu	IMP
Fonction de sauvegarde	VIT
Implémentation des pages annexes (options, règles, page dédiée à la sauvegarde)	IMP
Fonction du système de météo	MIN
Fonction de mise à jour des caractéristiques de l'avatar quand on reprend une partie	IMP
Fonction d'affichage des standards de sécurité au début du jeu	MIN
Implémentation de la fonction de sérialisation pour effectuer la sauvegarde du jeu	IMP

## Hierarchisation des exigences non fonctionnelles

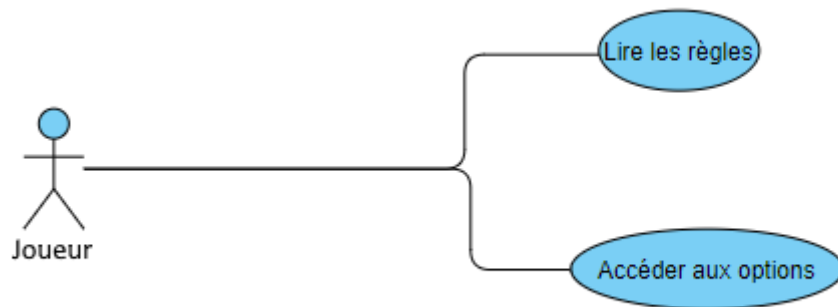
Exigence	Importance
L'application doit garder son intégrité tout au long de l'installation/utilisation	VIT
La maintenance de l'application doit se faire méthodiquement et rapidement en suivant le patron MVC	IMP
Les fichiers de sauvegardes doivent garder leur intégrité lorsqu'ils sont chargés et stockés	VIT
Accès au manuel d'utilisateur depuis l'application et aux options en jeu	IMP
L'application peut fonctionner sur différents systèmes d'exploitation (UNIX, Os X ..)	MIN
Facilité d'utilisation, interface graphique esthétique et compréhensible	VIT



## II. Diagramme de cas d'utilisation



Cas d'utilisation : l'utilisateur accède aux options et lit les règles



### III. Cas d'utilisation

Éléments de cas d'utilisation	Description spécifique au projet
Cas d'utilisation	Lancement d'une nouvelle partie
Acteurs	Usager
Événements déclencheurs	Appui sur le bouton "Nouvelle partie" depuis la page d'accueil
Parties prenantes et leurs intérêts	Usager - lancer une partie depuis le départ avec création d'un nouvel avatar
Niveau	Stratégique
Portée	<p>Données usager pour la création d'une nouvelle partie :</p> <ul style="list-style-type: none"><li>• Nom de joueur - chaîne de caractères contenant uniquement des lettres majuscules, minuscules et chiffres (pas de caractères spéciaux) et d'une longueur maximale de 16 caractères.</li><li>• Nom d'avatar - chaîne de caractères contenant uniquement des lettres majuscules, minuscules et chiffres (pas de caractères spéciaux) et d'une longueur maximale de 16 caractères.</li><li>• Type d'avatar - carrousel proposant différents types d'avatars possibles à incarner (chien, chat, oiseau, poulpe, robot)</li></ul>
Précondition	Le jeu doit être sur l'ordinateur. L'ordinateur doit être allumé. Le système doit être stable.
Postcondition	Un nouveau joueur est créé avec son avatar et une partie est lancée
Scénario nominal	<ol style="list-style-type: none"><li>1. L'utilisateur remplit le champ "Nom de joueur" (cf. Portée)</li><li>2. L'utilisateur remplit le champ "Nom d'avatar" (cf. Portée)</li><li>3. L'utilisateur choisit le type d'avatar qu'il souhaite avec le carrousel.</li><li>4. L'utilisateur valide son choix en cliquant sur le bouton valider.</li><li>5. Chaque champ est vérifié par le validateur afin de s'assurer que les saisies de l'utilisateur sont correctes.</li><li>6. Le joueur et son avatar sont créés, une nouvelle partie est lancée</li></ol>
Extensions	<ol style="list-style-type: none"><li>1. Si le validateur ne valide pas un des champs de saisie, un message d'erreur apparaît en indiquant qu'une des saisies est incorrecte</li></ol>

	<p>2. Si le validateur détecte un champ vide, un message d'erreur apparaît en indiquant qu'un des champs n'est pas rempli</p> <p>3. Si le nom de joueur et le nom d'avatar sont identiques, un message d'erreur apparaît en indiquant qu'il faut changer l'un des deux champs</p>
Contraintes	

Éléments de cas d'utilisation	Description spécifique au projet
Cas d'utilisation	Chargement d'une partie déjà sauvegardée
Acteurs	Usager, base de données
Événements déclencheurs	Appui sur la sauvegarde concernée dans le Menu "Charger une partie"
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>• Usager : reprendre une partie là où il l'a arrêtée</li> <li>• Base de données : donne accès à toutes les sauvegardes, pour les lister et les charger.</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>• Données de sauvegarde <ul style="list-style-type: none"> <li>• Pseudonyme du joueur</li> <li>• Nom de l'avatar</li> <li>• Type de l'avatar</li> <li>• Temps total de jeu</li> <li>• Statistiques de l'avatar</li> <li>• Date et heure de la fin de la session de jeu</li> </ul> </li> <li>• Données du modèle <ul style="list-style-type: none"> <li>• Nom de l'avatar</li> <li>• Type de l'avatar</li> <li>• Temps total de jeu</li> <li>• Statistiques de l'avatar</li> <li>• Dates et heures du début de la nouvelle session de jeu</li> </ul> </li> </ul>
Précondition	Le joueur a sauvegardé au moins une partie
Postcondition	La partie reprend au même endroit qu'avant



Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur se rend, depuis l'accueil, dans le menu "Charger une partie"</li> <li>2. L'interface affiche toutes les sauvegardes, avec un bouton "Charger"</li> <li>3. L'utilisateur clique sur une sauvegarde pour la sélectionner, puis sur le bouton "Charger" pour la lancer</li> <li>4. Le jeu récupère toutes les informations de la sauvegarde en base de données</li> <li>5. Le jeu lance la partie, avec toutes les informations contenues dans la sauvegarde.</li> </ol>
Extensions	<ul style="list-style-type: none"> <li>• S'il n'y a aucune sauvegarde, le bouton "Charger une partie" de l'accueil est grisé : l'utilisateur n'a pas accès à la base de données.</li> </ul>
Contraintes	

Éléments de cas d'utilisation	Description spécifique au projet
Cas d'utilisation	Sauvegarder et quitter une partie.
Acteurs	Usager, Base de données
Événements déclencheurs	Appuie sur le bouton "Sauvegarder", puis appuie sur le bouton "Quitter".
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>• Usager : Quitter le jeu en sauvegardant la partie en cours.</li> <li>• Base de données : enregistrer les données de la partie</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>• Données de sauvegarde <ul style="list-style-type: none"> <li>• Pseudonyme du joueur</li> <li>• Nom de l'avatar</li> <li>• Type de l'avatar</li> <li>• Temps total de jeu</li> <li>• Statistiques de l'avatar</li> <li>• Date et heure de la fin de la session de jeu</li> </ul> </li> <li>• Données du modèle <ul style="list-style-type: none"> <li>• Nom de l'avatar</li> <li>• Type de l'avatar</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Temps total de jeu</li> <li>• Statistiques de l'avatar</li> <li>• Dates et heures du début de la nouvelle session de jeu</li> </ul>
Précondition	Le joueur a lancé une nouvelle partie, ou chargé une partie déjà existante.
Postcondition	La partie est sauvegardée, les données sont enregistrées dans la base de données, le jeu retourne à l'écran d'accueil.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le rouage.</li> <li>2. L'utilisateur clique sur le bouton "Sauvegarder".</li> <li>3. L'utilisateur clique sur le bouton "Quitter".</li> </ol>
Extensions	Si l'utilisateur clique sur le bouton "Quitter" sans cliquer sur le bouton "Sauvegarder", un message d'avertissement apparaît et prévient l'utilisateur qu'aucune sauvegarde n'a été faite. S'il continue sans sauvegarder, le jeu retournera au menu d'accueil sans être sauvegardé.
Contraintes	

Éléments de cas d'utilisation	Description spécifique au projet
Cas d'utilisation	Réaliser une action avec l'avatar.
Acteurs	Usager
Événements déclencheurs	Appuie sur l'icône d'action pour interagir avec l'avatar.
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>• Usager : Effectue une action pour influencer les statistiques de l'avatar</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>• Nom : Une chaîne de caractères qui définit l'appellation de l'avatar.</li> <li>• Type : un type d'avatar compris dans l'énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT.</li> <li>• Santé : entier compris entre 1 et 10</li> <li>• Bonheur : entier compris entre 1 et 10</li> <li>• Nourriture : entier compris entre 1 et 10</li> <li>• Hygiène : entier compris entre 1 et 10</li> </ul>

Précondition	Une partie doit être en cours
Postcondition	Les statistiques de l'avatar sont mises à jour
Scénario nominal	L'utilisateur clique sur l'action à effectuer
Extensions	Les actions ne sont pas les mêmes selon le lieu dans lequel se situe l'avatar
Contraintes	

Éléments de test de cas d'utilisation	Description spécifique au projet
Cas de test	Test du lancement d'une nouvelle partie
Acteurs	Usager
Événements déclencheurs	Appui sur le bouton "Nouvelle partie" depuis la page d'accueil
Parties prenantes et leurs intérêts	Usager - lancer une partie depuis le départ avec création d'un nouvel avatar
Niveau	Stratégique
Portée	Données usager pour la création d'une nouvelle partie : <ul style="list-style-type: none"> <li>Nom de joueur - chaîne de caractères contenant uniquement des lettres majuscules, minuscules et chiffres (pas de caractères spéciaux) et d'une longueur maximale de 16 caractères.</li> <li>Nom d'avatar - chaîne de caractères contenant uniquement des lettres majuscules, minuscules et chiffres (pas de caractères spéciaux) et d'une longueur maximale de 16 caractères.</li> <li>Type d'avatar - carrousel proposant différents types d'avatars possibles à incarner (chien, chat, oiseau, poulpe, robot)</li> </ul>
Précondition	Aucune
Postcondition	Un nouveau joueur est créé avec son avatar et une partie est lancée
Scénarii de test	<ol style="list-style-type: none"> <li>Toutes les données d'entrée vides (test limite/extrême) : <ol style="list-style-type: none"> <li>Nom_Joueur, Nom_Avatar = " " (chaîne vide)</li> <li>L'utilisateur soumet les données pour créer une nouvelle partie</li> <li>Réponse attendue : un message d'erreur indiquant que les champs sont vides</li> </ol> </li> <li>Données d'entrée incomplète : <ol style="list-style-type: none"> <li>Le champ Nom_Joueur ou Nom_Avatar = " " (chaîne vide) et l'autre est correctement rempli</li> <li>L'utilisateur soumet les données pour créer une nouvelle partie</li> <li>Réponse attendue : un message d'erreur indiquant que le champ est incorrect.</li> </ol> </li> <li>Données d'entrée incorrecte 1</li> </ol>

	<ol style="list-style-type: none"> <li>1. Nom_Joueur ou Nom_Avatar contient des caractères non autorisés</li> <li>2. L'utilisateur soumet les données pour créer une nouvelle partie</li> <li>3. Réponse attendue : un message d'erreur indiquant que le champ contient des caractères non autorisés (en précisant que seuls des majuscules, minuscules et nombres sont autorisés)</li> <li>4. Données d'entrée incorrecte 2 <ol style="list-style-type: none"> <li>1. Nom_Joueur ou Nom_Avatar contient une chaîne de caractères trop longue</li> <li>2. L'utilisateur soumet les données pour créer une nouvelle partie</li> <li>3. Réponse attendue : un message d'erreur indiquant que le champ est trop long (en précisant qu'il ne peut contenir que 16 caractères maximum)</li> </ol> </li> <li>5. Données d'entrée existant déjà dans la base de données : <ol style="list-style-type: none"> <li>1. Nom_Joueur et Nom_Avatar sont correctement renseignés, mais existent déjà tous les deux dans la BDD</li> <li>2. L'utilisateur soumet les données pour créer une nouvelle partie</li> <li>3. Réponse attendue : un message d'erreur indiquant que le personnage existe déjà et qu'il faut changer le Nom_Joueur ou le Nom_Avatar</li> </ol> </li> </ol>
Extensions	
Contraintes	

Éléments de test de cas d'utilisation	Description spécifique au projet
Cas de test	Test de chargement d'une partie déjà sauvegardée
Acteurs	Usager, base de données
Événements déclencheurs	Appui sur la sauvegarde concernée dans le Menu "Charger une partie"
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>• Usager : reprendre une partie là où il l'a arrêtée</li> <li>• Base de données : donne accès au jeu à toutes les sauvegardes, pour les lister et les lancer</li> </ul>

Niveau	Stratégie
Portée	<ul style="list-style-type: none"> <li>Données de sauvegarde <ul style="list-style-type: none"> <li>Pseudonyme du joueur</li> <li>Nom de l'avatar</li> <li>Type de l'avatar</li> <li>Temps total de jeu</li> <li>Statistiques de l'avatar</li> <li>Date et heure de la fin de la session de jeu</li> </ul> </li> <li>Données du modèle <ul style="list-style-type: none"> <li>Nom de l'avatar</li> <li>Type de l'avatar</li> <li>Temps total de jeu</li> <li>Statistiques de l'avatar</li> <li>Dates et heures du début de la nouvelle session de jeu</li> </ul> </li> </ul>
Précondition	Le joueur a sauvegardé au moins une partie
Postcondition	La partie reprend au même endroit qu'avant
Scénarii de test	<ol style="list-style-type: none"> <li>Cas nominal : une partie a été sauvegardée <ol style="list-style-type: none"> <li>Dans le menu "Charger une partie", les informations suivantes sont affichées pour chaque sauvegarde : "Sauvegarde n°N", noms du joueur et de son avatar</li> <li>Aucune sauvegarde n'est sélectionnée par défaut → le bouton "Charger" est grisé</li> <li>Le joueur sélectionne une sauvegarde en cliquant dessus <ul style="list-style-type: none"> <li>→ mise en avant de la sauvegarde (eg. Surli-gner)</li> <li>→ le bouton "Charger" devient accessible.</li> </ul> </li> <li>Le joueur lance la partie → les informations sont récupérées : noms du joueur et de l'avatar, statistiques, temps total de jeu, date et heure, lieu</li> </ol> </li> <li>Aucune sauvegarde n'est disponible <ol style="list-style-type: none"> <li>Le bouton "Charger une partie" du menu d'accueil est grisé → on n'a pas accès au menu des sauvegardes</li> </ol> </li> <li>Sauvegarde corrompue <ol style="list-style-type: none"> <li>Lors de la sauvegarde, 2 copies sont faites, au cas où l'une des deux soit corrompue</li> <li>Si seulement l'une est corrompue → le jeu charge l'autre</li> </ol> </li> </ol>

	3. Si les 2 sont corrompues → le jeu affiche l'erreur "Erreur : sauvegarde <nom de la sauvegarde> corrompue"
Extensions	
Contraintes	

Éléments de test de cas d'utilisation	Description spécifique au projet
Cas de test	Test de changement de lieu avec son avatar
Acteurs	Usager
Événements déclencheurs	Appuie sur la flèche de gauche ou sur la flèche de droite.
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>Usager : Changer d'environnement de jeu.</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>Environnement : instance d'une classe Environnement qui définit l'environnement dans lequel se trouve l'avatar.</li> </ul>
Précondition	Le joueur a lancé une nouvelle partie, ou chargé une partie déjà existante.
Postcondition	L'avatar du joueur se situe dans un nouvel environnement.
Scénarii de test	<p>1. Cliquer sur la flèche de gauche. Réponse attendue : Le Tamagotchi se situe dans un nouvel environnement.</p> <p>2. Cliquer sur la flèche de droite. Réponse attendue : Le Tamagotchi se situe dans un nouvel environnement.</p> <p>3. Cliquer sur la flèche de gauche, puis sur la flèche de droite. Réponse attendue : Le Tamagotchi a changé d'environnement, et est retourné dans son environnement de départ.</p>

	<p>4. Cliquer sur la flèche de gauche jusqu'à ce qu'elle disparaisse.</p> <p>Réponse attendue : Au bout d'un certain nombre d'environnements, le Tamagotchi est obligé de faire demi-tour. La flèche de Gauche doit donc disparaître, car il ne peut plus avancer.</p> <p>5. Cliquer sur la flèche de droite jusqu'à ce qu'elle disparaisse.</p> <p>Réponse attendue : Au bout d'un certain nombre d'environnements, le Tamagotchi est obligé de faire demi-tour. La flèche de gauche doit disparaître, car il ne peut plus avancer.</p>
Extensions	
Contraintes	

Éléments de test de cas d'utilisation	Description spécifique au projet
Cas de test	Test de réalisation d'une action avec l'avatar.
Acteurs	Usager
Événements déclencheurs	Appuie sur l'icône d'action pour interagir avec l'avatar
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>Usager : Effectue une action pour influencer les statistiques de l'avatar</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>Nom : Une chaîne de caractères qui définit l'appellation de l'avatar.</li> <li>Type : un type d'avatar compris dans l'énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT.</li> <li>Santé : entier compris entre 1 et 10</li> <li>Bonheur : entier compris entre 1 et 10</li> <li>Nourriture : entier compris entre 1 et 10</li> <li>Hygiène : entier compris entre 1 et 10</li> </ul>
Précondition	<p>Une partie doit être en cours.</p> <p>L'avatar doit être créé.</p> <p>Les boutons d'interactions doivent être disponibles.</p>



Postcondition	Les statistiques de l'avatar sont mises à jour.
Scénarii de test	<ol style="list-style-type: none"> <li>1. Si l'environnement sélectionné est la salle à manger : le bouton d'interaction situé en bas de l'écran doit afficher le texte correspondant à l'action "Manger". L'utilisateur doit cliquer sur le bouton. L'animation correspondant à l'action "Manger" doit se déclencher. Réponse attendue : les statistiques suivantes doivent être mises à jour : Nourriture, <ol style="list-style-type: none"> <li>1. Si Nourriture &gt; 50% alors Bonheur augmente</li> </ol> </li> <li>2. Si l'environnement sélectionné est le salon : le bouton d'interaction situé en bas de l'écran doit afficher le texte correspondant à l'action "Jouer". L'utilisateur doit cliquer sur le bouton. L'animation de "Jouer" se déclenche. Réponse attendue : les statistiques suivantes doivent être mises à jour : Divertissement, <ol style="list-style-type: none"> <li>1. Si Divertissement &gt; 50% alors Bonheur augmente</li> </ol> </li> <li>3. Si l'environnement sélectionné est la chambre: le bouton d'interaction situé en bas de l'écran doit afficher le texte correspondant à l'action "Dormir". L'utilisateur doit cliquer sur le bouton pour que l'animation se déclenche. Réponse attendue : les statistiques suivantes doivent être mises à jour : Énergie, <ol style="list-style-type: none"> <li>1. Si Énergie &gt; 50% alors Santé augmente</li> </ol> </li> <li>4. Si l'environnement sélectionné est la salle de bain : le bouton d'interaction situé en bas de l'écran doit afficher le texte correspondant à l'action "Se laver". L'utilisateur doit sélectionner le bouton pour que l'animation se déclenche. Réponse attendue : les statistiques suivantes doivent être mises à jour : Hygiène, <ol style="list-style-type: none"> <li>1. Si Hygiène &gt; 50% alors Santé augmente</li> </ol> </li> </ol>
Extensions	
Contraintes	

Éléments de test de cas d'utilisation	Description spécifique au projet
Cas de test	Test de sauvegarde manuelle d'une partie.
Acteurs	Usager
Événements déclencheurs	L'utilisateur ouvre le menu et appuie sur le bouton de sauvegarde.
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>Usager : Effectue une action pour sauvegarder sa partie</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>Nom : Une chaîne de caractères qui définit l'appellation de l'avatar.</li> <li>Type : un type d'avatar compris dans l'énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT.</li> <li>Santé : entier compris entre 1 et 10</li> <li>Bonheur : entier compris entre 1 et 10</li> <li>Nourriture : entier compris entre 1 et 10</li> <li>Hygiène : entier compris entre 1 et 10</li> <li>Lieu : valeur comprise dans l'énumération : JARDIN, SALLE DE BAIN, SALON, CHAMBRE.</li> <li>Date : la date système (Sysdate) à laquelle la partie a été sauvegardée.</li> </ul>
Précondition	<p>Une partie doit être en cours.</p> <p>L'avatar doit être créé.</p> <p>Les boutons d'interactions doivent être disponibles.</p> <p>La fonction de sauvegarde doit être fonctionnelle.</p> <p>Le système de persistance des données doit être implémenté</p>
Postcondition	La partie est sauvegardée dans un fichier situé dans l'arborescence de l'application.
Scénarii de test	<p>On considère que le fonctionnement de la sauvegarde sera identique pour n'importe quel environnement du jeu.</p> <ol style="list-style-type: none"> <li>La partie est en cours et l'utilisateur décide de sauvegarder sa partie. S'il est en jeu et que son avatar est en action. Il peut ouvrir le menu et cliquer sur le bouton de sauvegarde. Il entre le nom de la sauvegarde. Il confirme la sauvegarde.</li> </ol>

	<p>Réponse attendue : il y a un temps d'attente et la partie est sauvegardée dans un fichier situé dans un dossier de sauvegarde dans l'arborescence du jeu.</p> <p>2. Si une erreur survient (manque d'espace disque ou autre), le jeu affiche le message "Sauvegarde impossible : une erreur est survenue".</p>
Extensions	
Contraintes	

Éléments de test de cas d'utilisation	Description spécifique au projet
Cas de test	Test de sauvegarde automatique d'une partie.
Acteurs	Usager et Système
Événements déclencheurs	5 minutes se sont écoulées depuis le chargement de la partie ou depuis la dernière sauvegarde automatique.
Parties prenantes et leurs intérêts	<ul style="list-style-type: none"> <li>Usager : jouer au jeu sans interruption et sans se soucier de sauvegarder la partie.</li> <li>Système : sauvegarder la partie</li> </ul>
Niveau	Stratégique
Portée	<ul style="list-style-type: none"> <li>Nom : Une chaîne de caractères qui définit l'appellation de l'avatar.</li> <li>Type : un type d'avatar compris dans l'énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT.</li> <li>Santé : entier compris entre 1 et 10</li> <li>Bonheur : entier compris entre 1 et 10</li> <li>Nourriture : entier compris entre 1 et 10</li> <li>Hygiène : entier compris entre 1 et 10</li> <li>Lieu : valeur comprise dans l'énumération : JARDIN, SALLE DE BAIN, SALON, CHAMBRE.</li> <li>Date : la date système (Sysdate) à laquelle la partie a été sauvegardée.</li> </ul>
Précondition	<p>Une partie doit être en cours.</p> <p>L'avatar doit être créé.</p>

	<p>Les boutons d'interactions doivent être disponibles.</p> <p>La fonction de sauvegarde doit être fonctionnelle.</p> <p>Le système de persistance des données doit être implémenté</p>
Postcondition	La partie est sauvegardée dans un fichier situé dans l'arborescence de l'application.
Scénarii de test	<p>On considère que le fonctionnement de la sauvegarde sera identique pour n'importe quel environnement du jeu.</p> <p>La partie est en cours et 5 minutes se sont écoulées depuis le chargement de la partie ou depuis la dernière sauvegarde automatique : la fonction de sauvegarde automatique se déclenche sans action de l'utilisateur.</p> <p>Réponse attendue : un icône de sauvegarde automatique apparaît dans un coin de l'écran et la partie est sauvegardée dans un fichier situé dans un dossier de sauvegarde dans l'arborescence du jeu.</p> <p>S'il n'y a plus d'espace disque disponible pendant la sauvegarde automatique, l'icône de sauvegarde est remplacée par une icône d'erreur.</p>
Extensions	
Contraintes	

## IV. Diagrammes de séquence

Diagramme représentant le lancement du programme et la création d'une nouvelle partie.

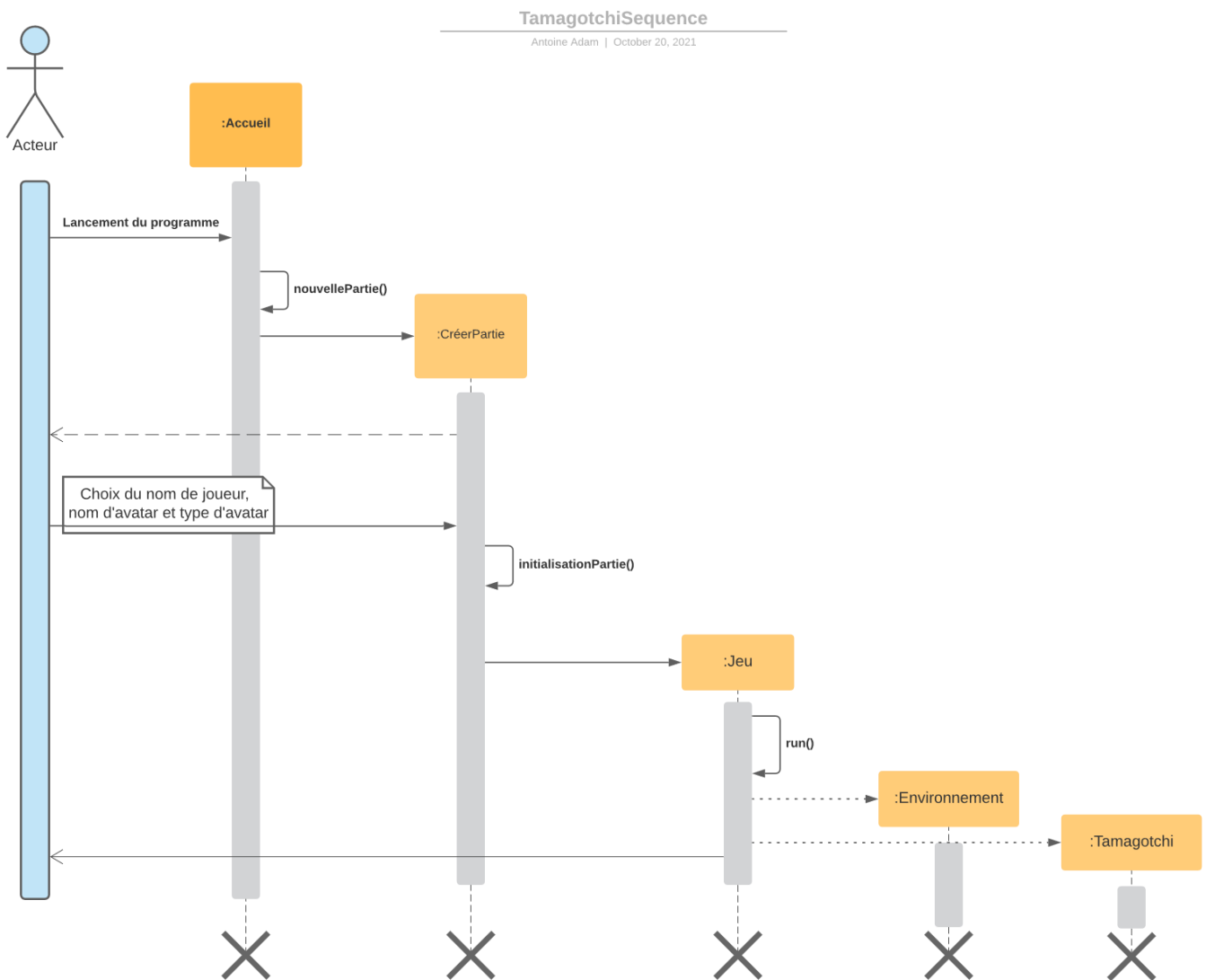


Diagramme représentant les interactions de l'utilisateur avec l'environnement afin de réaliser différentes actions avec son avatar.

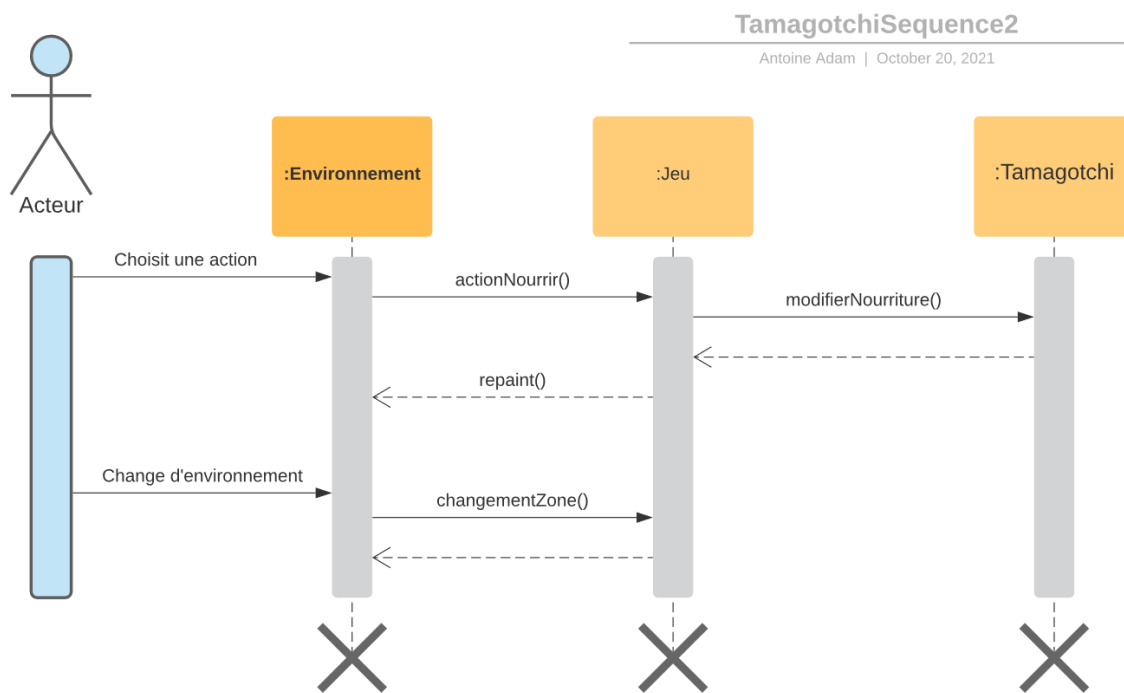
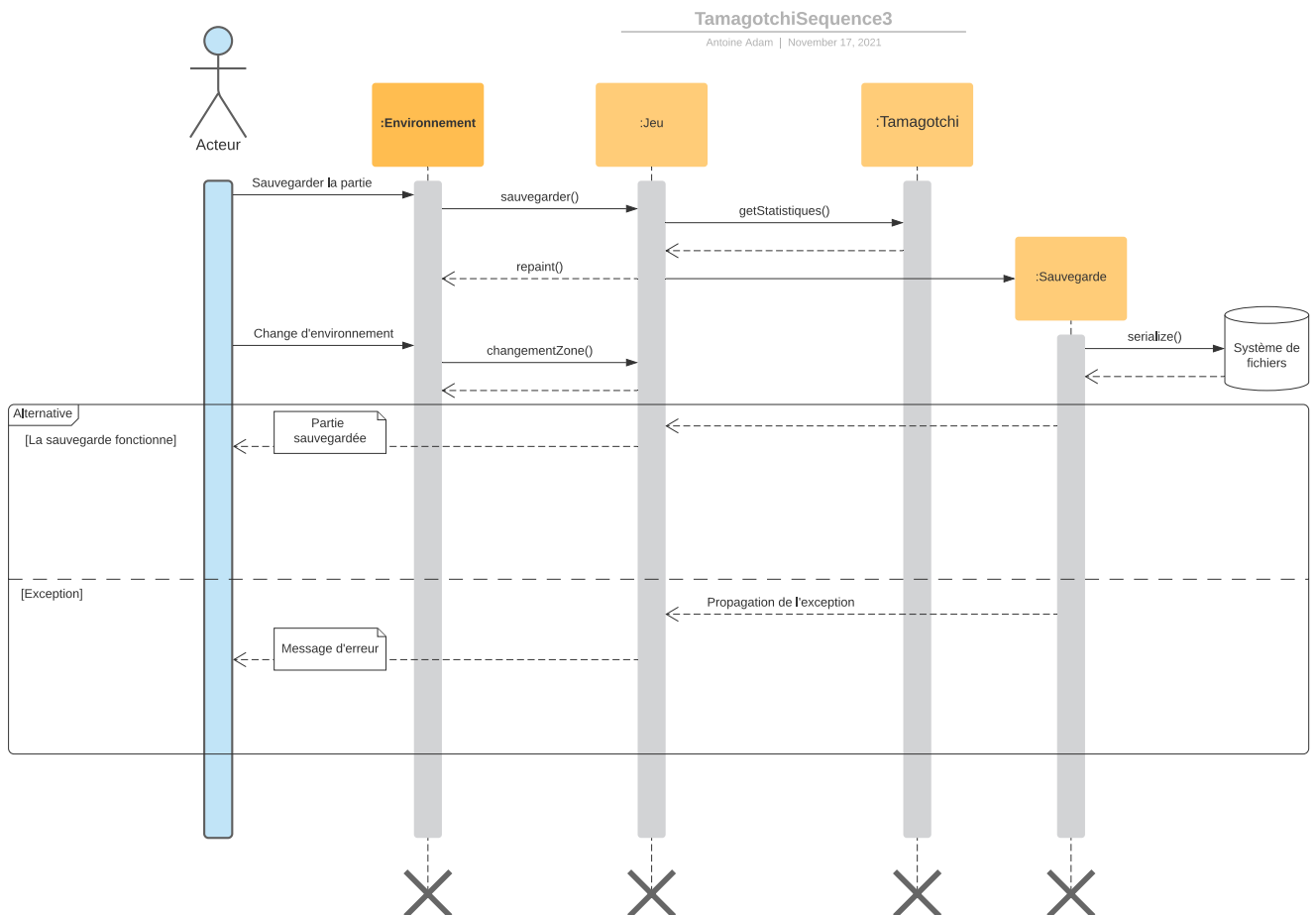


Diagramme représentant l'interaction de l'utilisateur avec l'environnement afin de sauvegarder une partie en cours.



V. Diagramme de Gantt

