

H.O.N.K.

Honk On Nasty Kids

Marius FORILLIERE, Ewen DENIAU,
Jean LE CRAS

26 octobre 2023

L3 - MIS
Informatique

Génie Logiciel Objet
Version 0.1

Professeur
Rémy Kessler



Sommaire

Titre	1
Sommaire	2
0 Introduction	3
0.1 Produit demandé	3
0.2 Terminologie	3
0.2.1 Termes généraux	3
0.2.2 Différentes Actions et Statistiques	3
0.3 Abréviations	3
1 Objectifs du produit	4
1.1 Définition du produit	4
1.2 Contexte économique	4
2 Fonctionnalités	4
2.1 Animaux	4
2.1.1 Les choix d'animaux	4
2.1.2 Comment s'en occuper	4
2.1.3 Vie et Bonheur	5
2.1.4 Niveaux / Expérience	6
2.2 Matrice d'incidence et équilibrages	6
2.2.1 Incidence des actions et du temps sur les statistiques	6
2.2.2 Statistiques par défaut	6
2.2.3 Intervalle de stabilité	7
2.3 Menus et Sauvegarde	7
2.3.1 Écran Titre	7
2.3.2 Menu de Pause	7
2.3.3 Fonctionnement de la Sauvegarde	8
2.4 Maison	8
3 Exigences non fonctionnelles	9
3.1 Fiabilité	9
3.2 Sécurité	9
3.3 Facilité d'utilisation	9
3.4 Rendement	9
3.5 Maintenabilité	9
3.6 Portabilité	9
4 Hiérarchisation des Fonctionnalités	9
4.1 Hiérarchisation des exigences fonctionnelles	9
4.2 Hiérarchisation des exigences non fonctionnelles	10
5 Crédits	10
5.1 Document	10

0 Introduction

0.1 Produit demandé

HONK est un projet de clone de Tamagotchi. Ce projet a été apporté par le client M. Kessler pour son fils de 12 ans. Le but sera de s'occuper de plusieurs animaux virtuels grâce à des actions ayant une influence sur les statistiques de l'animal choisi.

0.2 Terminologie

0.2.1 Termes généraux

- **Animaux** : Personnages principaux desquels le joueur devra s'occuper. Aussi appelés "*Tamagotchis*".
- **Statistique** : représente l'état d'une variable de l'animal, telle que sa satiété, son repos ou autre.
- **Intervalle de stabilité** : Chaque statistique aura un palier minimum et un palier maximum. Cela correspond à des seuils critiques dans lesquels des comportements spécifiques pourront être appliqués.
- **Actions** : activités de durée variable possibles pour le Tamagotchi, qui influencent ses statistiques.
- **Maison** : le jeu se déroule en grande partie dans une maison divisée en plusieurs salles.
- **Pièces** : salles de la Maison, il sera possible de naviguer de l'une à l'autre, différenciables par les actions possibles dans chacune d'entre elles.
- **GUI** : Ensemble d'éléments visuels extra-diégétiques, montrant l'état des statistiques et les actions possibles via des jauges horizontales et des boutons.
- **Écran de jeu** : écran où l'on joue.
- **Écran Titre** : écran de lancement du jeu, où est présenté le titre du jeu, ainsi que quelques menus.
- **Temps** : dans le jeu, un compteur de temps sera mis en place pour modifier automatiquement les statistiques à des intervalles réguliers.
- **Journée** : dans le jeu, représente un grand intervalle de temps au bout duquel certaines statistiques peuvent être modifiées.
- **Début de jeu** : commence au moment où le joueur choisit un animal.
- **Fin de jeu / Game Over** : moment où l'animal meurt.
- **Partie de jeu** : temps passé à s'occuper d'un animal, du début de jeu au Game Over.
- **Niveaux / Expérience** : à chaque fin de journée, si les conditions sont remplies, l'animal pourra gagner en expérience. Avec suffisamment d'expérience, l'animal pourra monter de niveau, et à un certain niveau avec un Tamagotchi, il sera possible d'en débloquer un autre.

0.2.2 Différentes Actions et Statistiques

- **Satiété** : quand cette statistique est basse, l'animal a faim.
- **Repos** : relatif à la quantité de sommeil de l'animal. Si elle est basse, l'animal est fatigué.
- **Énergie** : endurance restante de l'animal, si elle est basse l'animal est exténué.
- **Poids** : statistique trinaire, variant entre "sous-poids", "normal", et "surpoids" en fonction de la satiété et du temps.

0.3 Abréviations

- **GUI** : *Graphical User Interface* – Interface Graphique Utilisateur
- **XP** : points d'expériences

1 Objectifs du produit

1.1 Définition du produit

Le produit est un jeu de Tamagotchi avec un style graphique de type *pixel art*, où il sera possible de choisir entre plusieurs animaux et d'en débloquent d'autres. Le but sera de s'occuper d'un animal à l'aide de diverses actions réalisables dans des pièces dédiées de la maison. Il sera aussi possible de sauvegarder les données d'un animal et de les charger plus tard.

1.2 Contexte économique

Les Tamagotchis sont des jouets inventés par la société japonaise Bandai en 1996. La première version était un petit gadget avec un écran et trois boutons permettant de nourrir, laver et soigner le compagnon virtuel. Sans cette attention, le Tamagotchi meurt.

Ce jouet destiné aux jeunes japonaises est sorti pour la première fois le 23 novembre 1996 au Japon, et en mai 1997 dans le reste du monde. En France, ils seront introduits par Sébastien Breteau.

Si nous pouvons nous permettre de placer nos efforts dans un tel projet, c'est grâce aux efforts de l'équipe de marketing. Le produit est dédié à un public d'adolescentes et de jeunes adultes en Europe de l'ouest. Ils espèrent attiser une euphorie de Tamagotchis comme pour toutes les éditions précédentes, cela se fera par un biais nostalgique pour les acheteurs les plus âgés, et un biais de mode pour les plus jeunes. Par un effet de réseau et de mode, de nouveaux types d'acheteurs pourront s'ajouter au public cible de base. Ainsi, le public cible sera étendu artificiellement.

2 Fonctionnalités

2.1 Animaux

2.1.1 Les choix d'animaux

Le joueur pourra choisir parmi 4 animaux au début du jeu :

- Chat
- Chien
- Lapin
- Robot

Puis, en faisant monter en niveau les animaux, le joueur pourra débloquent respectivement :

- Lynx
- Renard
- Raton
- UwUcopter

2.1.2 Comment s'en occuper

Pour s'occuper de son animal, on aura à notre disposition un set d'actions ayant une incidence, globalement positive, sur ses statistiques. Dans la maison, toutes les actions seront réparties dans des pièces spécifiques reliées entre elles.

Statistiques

L'animal a deux types de statistiques : les statistiques de santé, indispensables à sa survie, et les statistiques globales.

Statistiques de santé : Les statistiques de santé sont au nombre de deux. Si l'une d'entre elle est nulle, le Tamagotchi meurt, c'est alors un *Game Over*.

Nous avons :

- La vie qui correspond à la santé physique du Tamagotchi.
- Le bonheur qui représente sa santé mentale.

Ces deux statistiques sont représentées par une barre de 5 coeurs.

Statistiques Globales : Il y en a 5, elles représentent des traits influencés par les actions, et elles influencent les statistiques de santé. Elles seront visibles sur la fenêtre sous la forme de jauge graduées de 0 à 100%.

- L'énergie
- La satiété
- Le repos
- L'hygiène
- Le poids (en trois parties pour représenter les différentes situations de poids : sous-poids, normal et surpoids)

Les valeurs de ces statistiques changeront d'un animal à l'autre (voir Table 2). Au fur et à mesure du temps, ces statistiques vont diminuer petit à petit. Ne pas s'occuper de l'animal conduira à sa mort.

Actions

Plusieurs actions sont possibles afin d'améliorer l'état du compagnon virtuel. En fonction de l'animal, les actions s'effectueront dans des pièces différentes de la maison.

Voici les actions possibles, accompagnées des lieux pour les différents animaux et le temps qu'elle prendrait :

- **Manger (1h) :**
 - Chien, Chat, Lapin ⇒ Cuisine
 - Robot ⇒ Garage
- **Dormir (8h) :**
 - Chien, Chat ⇒ Salon
 - Lapin, Robot ⇒ Chambre
- **Jouer (2h) :**
 - Chien, Chat, Lapin ⇒ Hall, Jardin
 - Robot ⇒ Salon, Jardin
- **Promenade (3h) :**
 - Tous ⇒ Jardin
- **Toilette (0.75h) :**
 - Chien, Chat, Lapin ⇒ Salle de bains
 - Robot ⇒ Cuisine

Chaque action va prendre une certaine durée, cette durée sera rajoutée au timer du jeu afin de l'accélérer. De plus, les actions auront une influence plus ou moins forte sur une ou plusieurs statistiques (voir Table 1).

2.1.3 Vie et Bonheur

Avec le temps, la vie augmentera si au moins quatre des statistiques globales ont une valeur supérieure ou égale au palier maximum associé. En revanche, elle baissera si au moins quatre des statistiques ont une valeur inférieure ou égale au palier minimum associé.

Quant au bonheur, il est possible d'en gagner grâce aux actions (jouer et se promener), et d'en perdre avec le temps.

2.1.4 Niveaux / Expérience

À la fin de chaque journée, le Tamagotchi va monter en XP. Ce gain est calculé en multipliant la somme des coeurs en valeur entière par un coefficient $\gamma = 10$. Ce coefficient peut changer à l'avenir par soucis d'équilibrage. Le nombre d'XP nécessaire à une montée de niveau sera calculé par la formule : $lv(xp) = \frac{xp}{16} \ln(xp^2)$

2.2 Matrice d'incidence et équilibrages

Cette section se concentrera sur l'équilibrage du jeu et sur les valeurs par défaut. Il est important de préciser que les valeurs peuvent être amenées à changer pour des raisons diverses et variées. Le jeu ne doit être ni trop facile ni trop difficile, c'est pourquoi une action peut influencer plus ou moins fortement sur les statistiques, tout comme les maximums de ces statistiques peuvent être modifiées d'un Tamagotchi à l'autre.

2.2.1 Incidence des actions et du temps sur les statistiques

La matrice d'incidence (Table 1) expose l'incidence des actions et du temps sur les statistiques.

[Exemple] Manger va plus remplir la jauge de satiété que celle d'énergie. En revanche, cela n'aura aucune influence sur le repos.

Si une statistique n'a pas été modifiée depuis un certain intervalle de temps, celle-ci verra sa valeur baisser (voir ligne "Temps" de la Table 1).

Attention ! Les exceptions étant le poids et le bonheur dont le timer ne sera jamais réinitialisé, peu importe les modifications lors de l'intervalle de temps.

Ces timers seront sensibles aux sauts de temps causés par les actions,

[Exemple] Si le timer est additionnée d'une heure, les timers des statistiques seront eux aussi additionnés d'une heure.

Actions \ Statistiques (% par rapport au max)	Énergie	Satiété	Repos	Hygiène	Poids	Bonheur (en 1/2 coeur)
Manger	+30	+75				
Dormir	+80	-50	+100			
Jouer	-30	-20	-20	-15	-5	+1
Promenade	-60	-45	-50	-40	-10	+3
Toilette				+100		
Temps	-5/h	-15/2h	-10/2h	-5/2h	[-10,-5,+10]/24h	-4/24

Table 1. Matrice d'incidence des actions et du temps

L'influence du temps sur le poids se fait en fonction de la satiété toutes les 24 heures.

- Si la satiété est supérieur à son palier maximum, le Tamagotchi gagne 10% de poids.
- Si elle est inférieure au palier minimum, il perd 10%.
- Sinon, il perd 5%.

2.2.2 Statistiques par défaut

Chaque Tamagotchi aura une valeur maximum différentes par statistiques. Ainsi, le joueur doit s'adapter aux besoins spécifiques de son personnage. Cela concerne seulement l'énergie, la satiété, le repos et l'hygiène (voir la Table 2).

Pour les animaux débloqués plus tard : leurs maximums correspondent à une augmentation de 20% des maximums de base.

Max	Énergie	Satiété	Repos	Hygiène
Chat	105	75	80	140
Chien	125	100	100	75
Lapin	110	110	120	100
Robot	180	20	150	50

Table 2. Maximum des statistiques par Tamagotchi

2.2.3 Intervalle de stabilité

Les statistiques globales ont un intervalle de stabilité (voir Table 3). C'est-à-dire qu'elles ont un palier minimum en dessous duquel des conséquences néfastes seront appliquées, et un palier maximum au dessus duquel des conséquences bénéfiques seront gratifiées.

Statistiques	Énergie	Satiété	Repos	Hygiène	Poids
Min-Max (en %)	30-80	50-90	30-75	10-95	33-66

Table 3. Paliers minimum et maximum des statistiques globales.

2.3 Menus et Sauvegarde

Le jeu sera agrémenté de menus dans lesquels les joueurs pourront naviguer. Ces menus permettront de lancer le jeu, de créer des sauvegardes, etc.

2.3.1 Écran Titre

Trois choix seront disponibles sous formes de boutons : "Nouvelle Partie", "Charger une Partie", "Options". Il suffira de cliquer dessus pour ouvrir la page correspondante. Lorsque la souris passe sur l'un des boutons, celui-ci changera de ton de couleur, pour bien indiquer le bouton choisi.

Nouvelle Partie

Ce menu sera composé de trois boutons représentant les différents fichiers de sauvegarde alloués, il sera alors possible d'en choisir un en cliquant dessus, ce qui aura pour effet de soit : créer un nouveau fichier en cliquant sur un bouton marqué "Sauvegarde Vide" ou d'écraser une sauvegarde précédente, suite à une confirmation du joueur.

Charger une Partie

Ce menu sera composé de boutons représentant les fichiers de sauvegarde existants. En cliquant dessus, le jeu nous ramène à l'écran de jeu avec les données de la sauvegarde chargées. Cela concerne le timer et le Tamagotchi avec ses statistiques et les valeurs de la dernière session. Il est éventuellement prévu d'affecter un malus sur les statistiques en fonction du temps d'absence.

2.3.2 Menu de Pause

Ce menu apparaîtra lorsque l'on cliquera sur un bouton en particulier sur l'écran de jeu. il sera composé de 4 boutons, un petit en haut à droite pour revenir au jeu, et trois gros boutons disposés en colonne :

- **Sauvegarder et Continuer** : Met à jour le fichier de sauvegarde, et retourne sur l'écran de jeu.
- **Sauvegarder et Quitter** : Met à jour le fichier de sauvegarde, et revient à l'écran titre.
- **Quitter sans Sauvegarder** : Revient simplement à l'écran titre.

2.3.3 Fonctionnement de la Sauvegarde

La sauvegarde se fera dans un fichier .json où l'on enregistrera l'état de l'animal, toutes ses statistiques et leurs timers, ainsi que la date et l'heure de sauvegarde pour pouvoir simuler le temps passé hors exécution.

Au prochain lancement du jeu, les données seront donc chargées et appliquées à partir du .json, la position dans la maison sera aléatoire, puis les statistiques seront mises à jour en fonction du temps passé hors exécution.

2.4 Maison

Le joueur et ses Tamagotchis évoluent dans une maison. Il sera possible d'aller dans chaque pièce afin d'effectuer une action pour son Tamagotchi. Sachant que chaque Tamagotchi exécute ses actions dans des lieux différents des autres, il est du devoir du joueur de retenir où est-ce qu'ils doivent aller pour telle ou telle action (la liste complète se trouvant section 2.1.2 dans le paragraphe sur les actions).

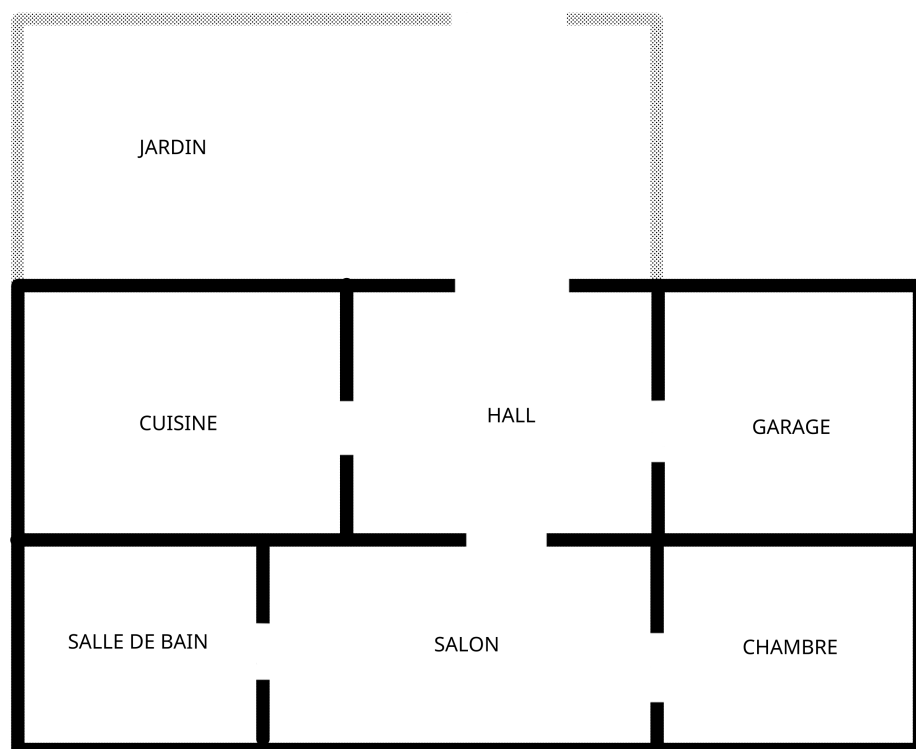


Figure 1. Schéma de la maison où le joueur évolue.

3 Exigences non fonctionnelles

3.1 Fiabilité

Le programme ne devra pas utiliser trop de ressources pour tourner suffisamment bien et que les actions du joueurs répondent rapidement et les animations soient fluides. Étant un jeu pour enfants, nous veillerons à ce que le programme soit robuste afin de prévenir tout risque de *crash*, car cela peut repousser les joueurs.

3.2 Sécurité

Pour éviter tout risque de perte de données, une sauvegarde automatique est prévue tous les 10 minutes. Nous prévoyons également d'encoder le fichier de sauvegarde de sorte à ce que l'utilisateur ne puisse pas facilement changer les données et rendre difficile la triche.

3.3 Facilité d'utilisation

Des gros boutons et une police claire seront appliqués. Des retours visuels pour les actions et les statistiques seront présentes. L'animal s'animerà en fonction de l'action ou de son état actuel (fatigué, heureux, triste, malade, quand il mange ou dort).

3.4 Rendement

Cette application sera destinée à un utilisateur particulier n'ayant pas forcément d'ordinateur puissant. L'avantage du *pixel art* est sa résolution assez basse (avec des graphismes simples, sans modèles 3d ou animations lourdes), demandant ainsi peu de temps de chargement et mobilisant moins de ressources en mémoire. N'étant pas multijoueur, que ce soit en ligne ou en local, nous n'avons pas besoin de gérer les serveurs ni l'interface et les ressources pour cela. Le stockage des données sera aussi très léger car seuls quelques nombres et dates seront enregistrés.

3.5 Maintenabilité

Tous les paramètres concernant l'équilibrage ou les statistiques d'un animal seront rassemblés dans la même classe.

[Par exemple] Tout ce qui concerne le robot, c'est-à-dire ses statistiques / lieux / état, seront dans la classe Robot.

En outre, nous appliquerons la méthode MVC pour organiser nos classes afin de garantir une maintenabilité optimale.

3.6 Portabilité

Notre objectif est de le rendre accessible sur Windows et Linux au minimum. Ce jeu sera entièrement jouable à la souris, nous pouvons envisager une écoute de clavier pour remplacer la souris. Nous envisagerions aussi faire notre propre console pour y jouer, cela se fera sur la base d'un Raspberry π .

4 Hiérarchisation des Fonctionnalités

4.1 Hiérarchisation des exigences fonctionnelles

1. Animaux
2. Statistiques

3. Salles
4. Temps
5. XP des animaux et leur évolution

4.2 Hiérarchisation des exigences non fonctionnelles

1. Sauvegarde
2. Menus et interfaces
3. Menus et interfaces plus poussés (avec les textures)

5 Crédits

5.1 Document

Le template du document provient de Vincent Labatut (vincent.labatut@univ-avignon.fr) et est modifié par Simon Leclercq (leclercq.e2102543@etud.univ-ubs.fr). Il a été ré-ajustée pour ce projet en particulier par Ewen Deniau.

Nous remercions [@that.one.stupid.goat](#) qui nous a produit la page de garde.