|  |
| --- |
| Animal virtuel |
| Manuel technique |
| Cette documentation technique est destinée au collège d'experts en charge de l'évaluation du projet. |

Jessica Sulzbach

09/05/2017

Table des matières

[1 Introduction 2](#_Toc482278926)

[2 Étude d’opportunité 3](#_Toc482278927)

[2.1 Tamagotchi 3](#_Toc482278928)

[2.1.1 Points forts 3](#_Toc482278929)

[2.1.2 Points faibles 3](#_Toc482278930)

[2.2 My Boo 4](#_Toc482278931)

[2.2.1 Points forts 4](#_Toc482278932)

[2.2.2 Points faibles 4](#_Toc482278933)

[2.3 Conclusion 4](#_Toc482278934)

[3 Analyse fonctionnelle 5](#_Toc482278935)

[3.1 Fonctionnalités 5](#_Toc482278936)

[3.2 Maquettes 6](#_Toc482278937)

[3.2.1 Interface principale 6](#_Toc482278938)

[3.2.2 Interface magasin 8](#_Toc482278939)

[3.2.3 Interface inventaire 9](#_Toc482278940)

[4 Analyse organique 10](#_Toc482278941)

[4.1 Diagramme de classes 10](#_Toc482278942)

[4.2 Points techniques 10](#_Toc482278943)

[5 Protocole de test 10](#_Toc482278944)

[5.1 Plan de tests 10](#_Toc482278945)

[5.2 Rapport de tests 10](#_Toc482278946)

[6 Améliorations possibles 10](#_Toc482278947)

[7 Conclusion 10](#_Toc482278948)

[8 Table des illustrations 10](#_Toc482278949)

# Introduction

Je suis actuellement en train de réaliser un certificat fédéral de compétence au Centre de Formation Professionnelle Technique de Genève.  Dans le cadre de cette formation, je dois réaliser un travail pratique individuel (TPI) pour l’obtention de mon diplôme.

Il s'agit ici de la documentation technique, cette dernière regroupe les analyses fonctionnelles et organiques, ainsi que le plan et le rapport de test. En outre, le planning du projet, les principales procédures et fonctions, ainsi que le diagramme de classes sont contenu dans ce manuel technique.

L’objectif de ce travail est d’élaborer un projet informatique afin de montrer les compétences acquises pendant les quatre années d’étude au CFPT. Donc, pendant ces prochaines semaines, je dois réaliser un programme en C#. J’ai décidé de programmer un animal virtuel.

Un animal de compagnie virtuel est un type de compagnon artificiel élevé pour l’amusement de son maître. Le but du jeu est de tenir l’animal en vie et de le faire grandir. Pour faire ceci, il faut le nourrir, le nettoyer et le sortir. En fait, il faut le chouchouter comme un vrai animal de compagnie pour éviter qu’il ne s’ennuie ou qu’il ne meure.

# Étude d’opportunité

L’idée de programmer une application d’animal virtuel peut paraitre inutile, il existe déjà de nombreuses applications et sites internet. Voici quelques exemples de l’existant.

## Tamagotchi

Un Tamagotchi est un  [animal de compagnie virtuel](https://fr.wikipedia.org/wiki/Animal_de_compagnie_virtuel) [japonais](https://fr.wikipedia.org/wiki/Japon).

Le jeu consiste à simuler l'éducation d'un animal à l'aide d'une petite console miniature (voir la figure numéro un), de la taille d'une montre, dotée d'un programme informatique.

La petite console est dotée de trois boutons. C’est grâce à ces boutons que l’utilisateur peut interagir avec le petit compagnon virtuel.



Figure 1 : Exemple console Tamagotchi

### Points forts

Grace à seulement trois boutons l’utilisateur peut tout faire. Le format ovulaire est sympathique et diffèrent.

### Points faibles

La petite console à un look chaotique avec des couleurs qui ne vont pas bien ensemble. L’affichage est en noir et blanc. Ceci peut fatiguer les yeux de l’utilisateur.

Il n’y pas de label. Il faut découvrir les fonctionnalités en tâtonnant, ce qui n’est pas très pratique et facile.

## My Boo

Application mobile qui consiste à prendre soin de la balle bleue. Il y a plusieurs fenêtres et c’est tactile.



Figure 2 : Exemple application My Boo

### Points forts

Avec la fonctionnalité Vêtir, l’utilisateur peut habiller son animal. Ceci laisse la possibilité à l’utilisateur de l’adapter plus à lui.

### Points faibles

Application fonctionne seulement si l’utilisateur est connecté à internet. L’animal dans cette application est une balle bleue, ceci est dérangeant car une balle n’a pas de vie.

## Conclusion

Grace à ces exemples, je peux vous dire que mon application a des couleurs douces et accordé. Il y a une interface intuitive, avec un animal existant dans notre magnifique planète.

L’application fonctionne sans internet et sauvegarde automatiquement les données.

# Analyse fonctionnelle

Cette partie de la documentation technique fait la lumière sur la partie visible du point de vue de l’utilisateur final de l’application. Je vais donc vous donner ici les consignes, c'est-à-dire vous expliquer comment le jeu fonctionne précisément.

## Fonctionnalités

Interface principale affiche différents niveaux liés à la créature, le but c’est de les garder remplis :

* **Santé**

Pour l’augmenter :

* + Le nourrir
  + Hygiène au max
  + Joie au max
* **Hygiène**

Pour l’augmenter :

* + Le doucher
  + Le brosser
* **Energie**

Pour l’augmenter :

* + Le faire dormir
  + Lui faire faire une sieste
* **Joie**

Pour l’augmenter :

* + Le promener
  + Le caresser
  + Lui donner des bonbons

L’interface principale aura aussi trois boutons, chaque bouton permet d’accéder à d’autres fonctionnalités que j’explique plus loin (chapitre 3.2.1) :

* Animal
  + Caresser
  + Dormir
  + Sieste
* Inventaire
  + Alimentations
    - Riz
    - Sushis
    - Bonbons
  + Matériel
    - Shampooing
    - Brosse
    - Nœud papillon/Ruban
* Activités
  + Magasin
  + Promenade

## Maquettes

### Interface principale

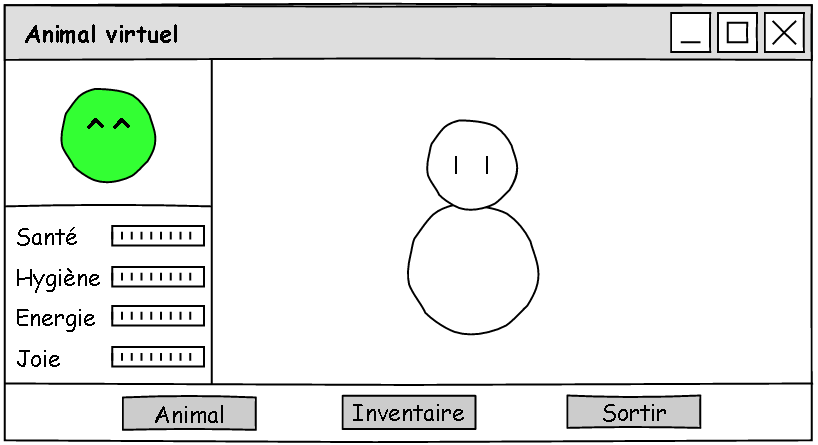


Figure : Interface principale - Animal heureux

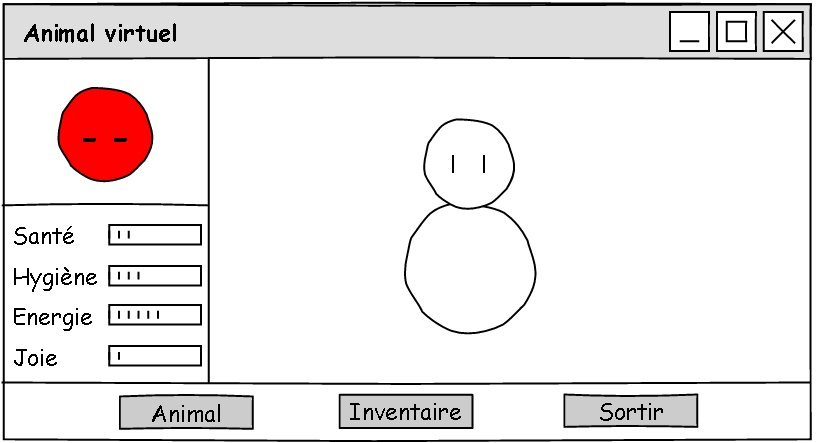


Figure : Interface principale - Animal malheureux

Voici l’interface principale. Ceci est la première fenêtre que l’utilisateur voit quand il lance l’application, sauf quand il y a un œuf à éclore. Donc si l’utilisateur n’a toujours pas d’animal ou son animal est décédé, il verra un œuf éclore pour donner naissance à son nouvel animal et ensuite il est dirigé ici.

Le carré en haut à gauche (avec un cercle vert et des yeux de content) change selon la valeur des quatre niveaux (Santé, Hygiène, Energie et Joie) qui affiche ce que l’animal ressent.

Il y a trois images différentes pour celui-ci :

1. Vert : Niveaux plus que la moitié : Heureux
2. Jaune : Niveaux à moitié remplis : Entre les deux
3. Rouge  : Niveaux moins que la moitié : Malheureux

Les niveaux sont situés en bas à gauche de la fenêtre. Ils sont constamment entrain de diminuer et donc l’utilisateur doit les augmenter grâce à des tâches (voir les fonctionnalités pour plus).

L’interface principale a aussi trois boutons qui sont situés horizontalement en bas de la fenêtre.

**Le bouton Animal**

Ce bouton peut lancer des actions qui ne nécessitent pas de matériel et c’est des actions avec l’animal. L’utilisateur peut choisir entre ces trois possibilités suivantes : Caresser, Dormir et Sieste.

Quand l’utilisateur clique sur Caresser une animation s’affiche et augmente le niveau de Joie d’un demi.

Si l’utilisateur clique sur Dormir, une autre animation se lance et le niveau d’Energie se recharge au maximum.

Figure 5 : Exemple bouton

Pour finir, si l’utilisateur clique sur Sieste il y a aussi une animation qui se lance, mais le niveau d’Energie augmente de deux.

**Le bouton Inventaire**

Le bouton Inventaire redirige l’utilisateur sur la fenêtre des Aliments ou Matériaux (voir plus bas).

**Le bouton Activités**

Ce bouton englobe toutes les actions qui nécessitent de prendre l’air. Dans le jeu l’utilisateur peut aller faire une promenade ou aller faire des courses au supermarché (voir Interface magasin).

La promenade dure dix secondes. Pendant la promenade l’animal peut trouver des cadeaux (entre zéro et cinq), dans les cadeaux il y a une somme d’argent (entre cinq et cinquante). Quand l’animal a trouvé un cadeau l’utilisateur entend un bruit de caisse et l’argent est ajouté directement dans le porte-monnaie.

La promenade augmente le niveau de joie de trois et diminue le niveau d’énergie et hygiène aussi de trois.

Pour conclure sur l’interface principale, il y a un dernier carré, plus grand que les autres, qui prend le reste de la fenêtre. Ce carré représente où l’utilisateur voit les animations correspondant à l’action qu’il lance.

### Interface magasin

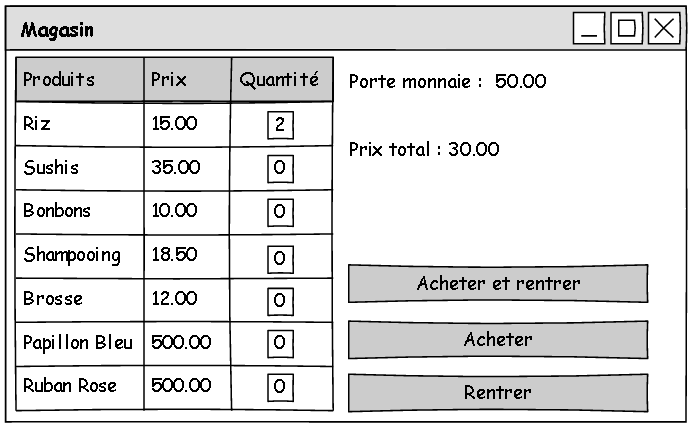


Figure : Interface magasin

L’interface magasin, est implémentée par un tableau qui affiche les produits qui sont en vente, le prix de ce produit et une zone de saisie pour que l’utilisateur puisse rentrer la quantité qu’il aimerait acheter. Ce tableau se situe à gauche.

À droite il y a l’argent de l’utilisateur, le prix total des produits et trois boutons.

1. L’utilisateur rentre la quantité à acheter et ensuite clique sur le premier bouton «Acheter et rentrer». Ce bouton ajoute ce que l’utilisateur a acheté à l’inventaire et retourne à l’interface principale.
2. L’utilisateur choisit ce qu’il voudrait acheter et ensuite clique sur le deuxième bouton «Acheter». Ce bouton ajoute ce que l’utilisateur a acheté à l’inventaire et reste sur l’interface magasin.
3. L’utilisateur change d’avis et clique sur le bouton « Rentrer ». Ce bouton retourne à l’interface principale.

### Interface inventaire

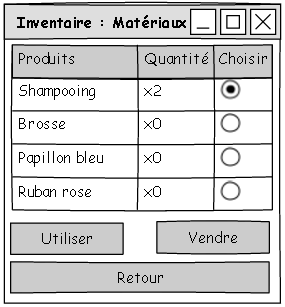
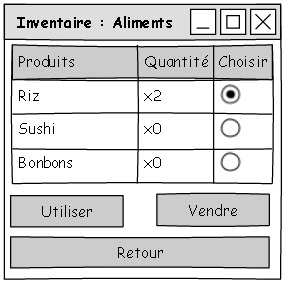


Figure 7 : Inventaire - Aliments

Figure 8 : Inventaire - Matériaux

L’inventaire a deux fenêtres, une pour les matériels et un autre pour les aliments. Les deux fenêtres se ressemble beaucoup et fonctionne de la même manière.

L’inventaire a un tableau avec trois colonnes :

1. Produits  : Qui contiennent le nom du produit
2. Quantité : La quantité que vous avez de ce produit
3. Choisir  : Sélectionner celui que vous voulez utiliser ou vendre

Ensuite, il y a trois boutons :

1. Utiliser : Diminue la quantité de un, ferme l’inventaire
   1. Riz, Sushi, Bonbons – Lance une animation
   2. Shampooing, brosse – L’utilisateur doit cliquer sur l’animal trois fois pour compléter l’action
   3. Papillon bleu, Ruban rose – L’animal portera désormais un accessoire
2. Vendre  : Vend le produit choisi au prix d’achat
3. Retour : Retourne à l’interface principale

# Analyse organique

L’analyse organique décrit le fonctionnement interne de l’application. Il s'agit d'analyser ce qui se passe en interne, du point de vue du développeur.

## Diagramme de classes

## Points techniques

# Protocole de test

## Plan de tests

## Rapport de tests

# Améliorations possibles

# Conclusion

# Table des illustrations