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

09/05/2017

Table des matières

[Introduction 2](#_Toc482106769)

[Étude d’opportunité 3](#_Toc482106770)

[Tamagotchi 3](#_Toc482106771)

[Points forts 3](#_Toc482106772)

[Points faibles 3](#_Toc482106773)

[My Boo 4](#_Toc482106774)

[Points forts 4](#_Toc482106775)

[Points faibles 4](#_Toc482106776)

[Conclusion 4](#_Toc482106777)

[Analyse fonctionnelle 5](#_Toc482106778)

[Fonctionnalité 5](#_Toc482106779)

[Promenade 6](#_Toc482106780)

[Maquette 6](#_Toc482106781)

[Interface principale 6](#_Toc482106782)

[Interface magasin 7](#_Toc482106783)

[Inventaire 7](#_Toc482106784)

[Analyse organique 8](#_Toc482106785)

[Diagramme de classes 8](#_Toc482106786)

[Points techniques 8](#_Toc482106787)

[Protocole de test 8](#_Toc482106788)

[Plan de tests 8](#_Toc482106789)

[Rapport de tests 8](#_Toc482106790)

[Améliorations possibles 8](#_Toc482106791)

[Conclusion 8](#_Toc482106792)

[Table des illustrations 8](#_Toc482106793)

# 

# 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 d'états 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 peux habiller son animal. Ceci laisse l’utilisateur l’adapter plus à lui.

### Points faibles

Application qui 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 aura des couleurs douces et accordé. Il sera bien étiqueté, 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.

## Fonctionnalité

Interface principale affichera différents niveaux liés à la créature :

* **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
  + Le faire prendre une sieste
* **Joie**

Pour l’augmenter :

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

L’interface principale aura aussi trois boutons, chaque bouton à des actions :

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

### Promenade

La promenade dura 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 entendra un bruit de caisse et l’argent sera ajouter directement dans le porte-monnaie.

## Maquette

### Interface principale

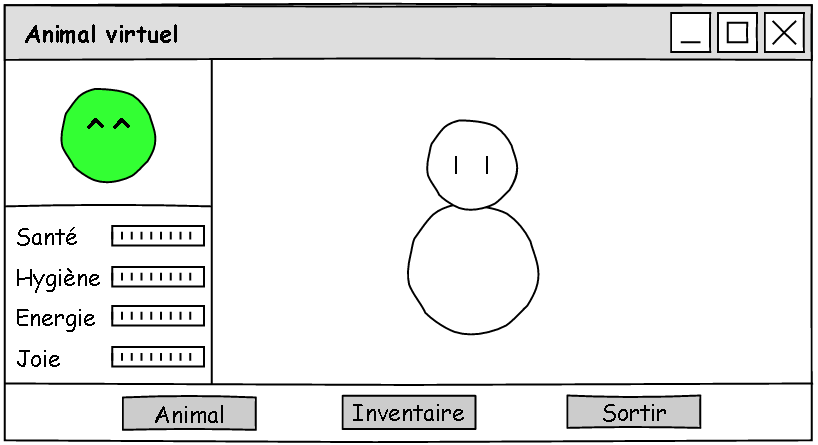


Figure 3 : Interface principale - Animal heureux

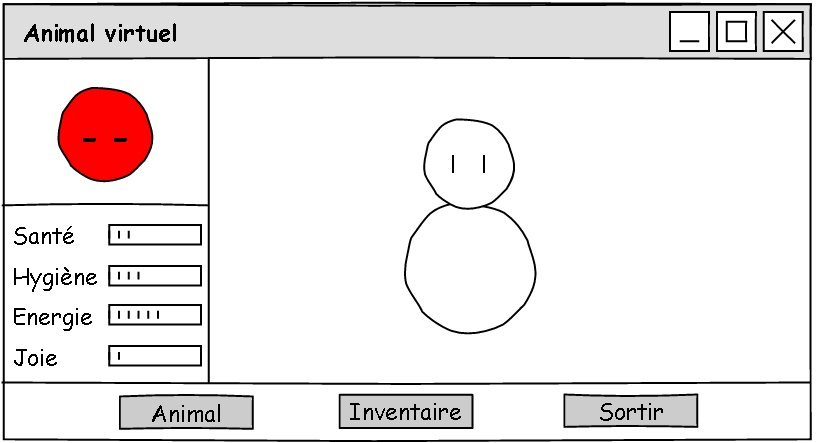


Figure 4 : Interface principale - Animal malheureux

### Interface magasin

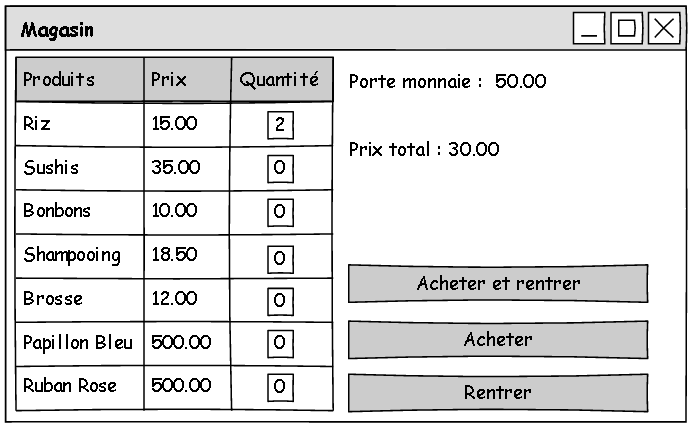


Figure 5 : Interface magasin

### Inventaire

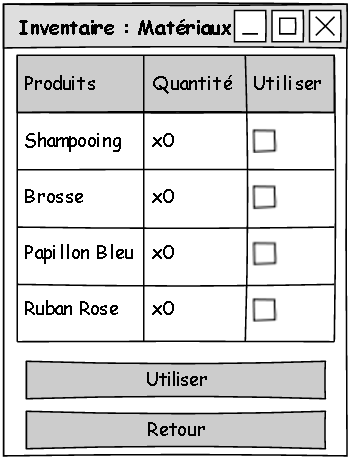
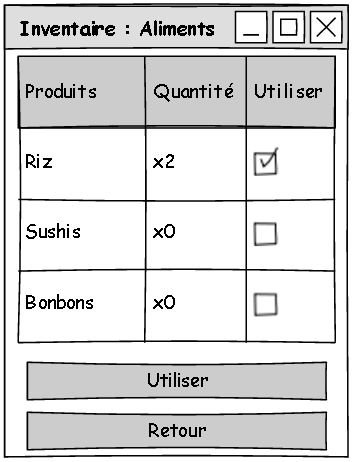
 

Figure 6 : Inventaire - Matériaux

Figure 7 : Inventaire - Aliments

# 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