|  |
| --- |
| centre de formation technique |
| Animal virtuel |
| Manuel technique |
|  |
| **Jessica Sulzbach** |
| **24/05/2017** |

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

C:\Users\SULZBACHJ_INFO\Desktop\AnimalVirtuel\Images\Born\Born8.png

Table des matières

[1. Introduction 4](#_Toc483385743)

[2. Étude d’opportunité 5](#_Toc483385744)

[2.1 Tamagotchi 5](#_Toc483385745)

[2.1.1 Points forts 5](#_Toc483385746)

[2.1.2 Points faibles 5](#_Toc483385747)

[2.2 My Boo 6](#_Toc483385748)

[2.2.1 Points forts 6](#_Toc483385749)

[2.2.2 Points faibles 6](#_Toc483385750)

[2.3 Conclusion 6](#_Toc483385751)

[3. Analyse fonctionnelle 7](#_Toc483385752)

[3.1 Fonctionnalités 7](#_Toc483385753)

[3.2 Maquettes 8](#_Toc483385754)

[3.2.1 Interface principale 8](#_Toc483385755)

[3.2.2 Interface magasin 10](#_Toc483385756)

[3.2.3 Interface inventaire 11](#_Toc483385757)

[4. Analyse organique 12](#_Toc483385758)

[4.1 Diagramme de classes 12](#_Toc483385759)

[4.1.1 Fenêtre principale 12](#_Toc483385760)

[4.1.2 Fenêtre magasin 13](#_Toc483385761)

[4.1.3 Fenêtre matériaux 13](#_Toc483385762)

[4.1.4 Fenêtre aliments 14](#_Toc483385763)

[4.1.5 Class animal 14](#_Toc483385764)

[4.1.6 Class magasin 15](#_Toc483385765)

[4.1.7 Class récupération de données 15](#_Toc483385766)

[4.1.8 Class inventaire 16](#_Toc483385767)

[4.2 Organigramme 16](#_Toc483385768)

[4.2.1 Fonctionnement général 16](#_Toc483385769)

[4.2.2 Animation depuis la forme principale 17](#_Toc483385770)

[4.2.3 Animation depuis la forme aliments ou matériaux 17](#_Toc483385771)

[4.3 Points techniques 17](#_Toc483385772)

[4.3.1 Méthode Rewrite 17](#_Toc483385773)

[4.3.2 Méthode FileReader 18](#_Toc483385774)

[4.3.3 Méthode AnimalRecover 18](#_Toc483385775)

[5. Protocole de test 19](#_Toc483385776)

[5.1 Plan de tests 19](#_Toc483385777)

[5.1.1 Vue principale 19](#_Toc483385778)

[5.1.2 Vue magasin 19](#_Toc483385779)

[5.1.3 Vue aliments ou matériels 20](#_Toc483385780)

[5.2 Rapport de tests 20](#_Toc483385781)

[5.2.1 Vue principale 20](#_Toc483385782)

[5.2.2 Vue magasin 20](#_Toc483385783)

[5.2.3 Vue aliments ou matériels 21](#_Toc483385784)

[6. Améliorations possibles 22](#_Toc483385785)

[6.1 Audio 22](#_Toc483385786)

[6.2 Fonctionnalités 22](#_Toc483385787)

[7. Conclusion 23](#_Toc483385788)

[8. Table des illustrations 24](#_Toc483385789)

# 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

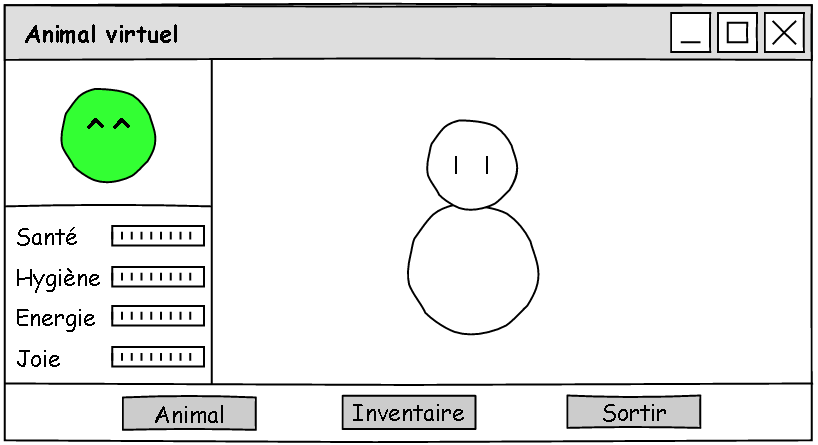


Figure : Interface principale - Animal heureux

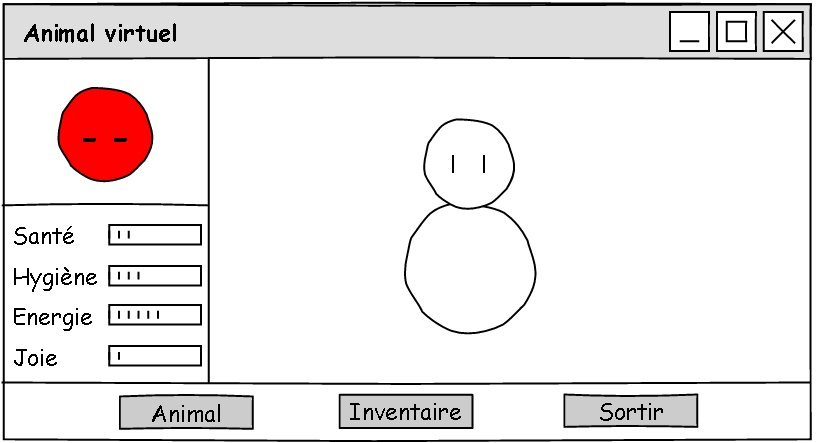


Figure : Interface principale - Animal malheureux

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 : 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

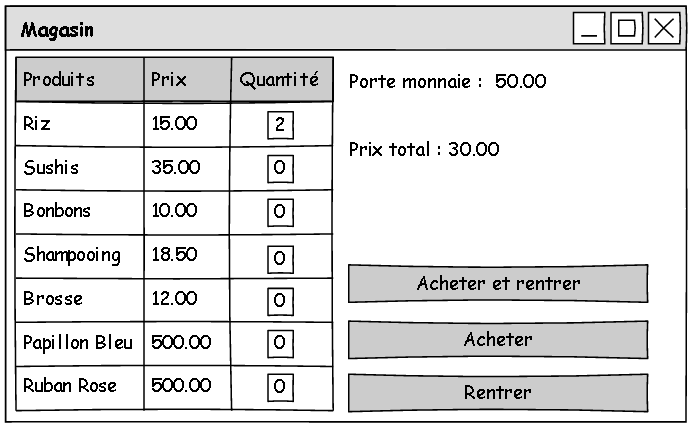


Figure : Interface magasin

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

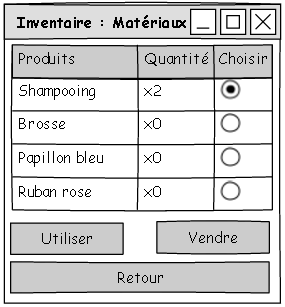
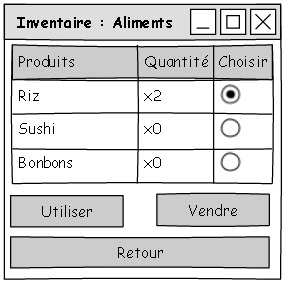


Figure : Inventaire - Aliments

Figure : 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 – Lance une animation
   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

### Fenêtre principale

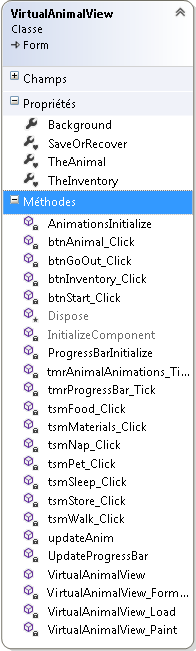
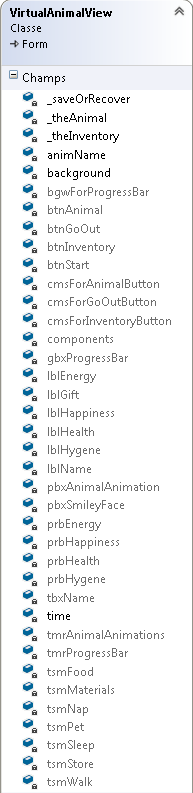


Figure : Diagramme de class fenêtre principale

### Fenêtre magasin

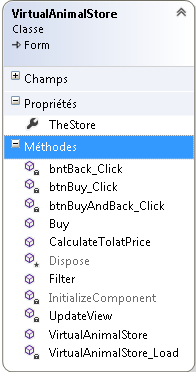
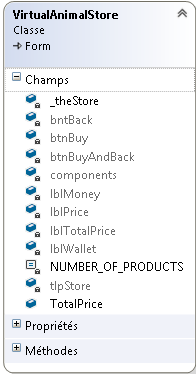


Figure : Diagramme de class fenêtre Magasin

### Fenêtre matériaux

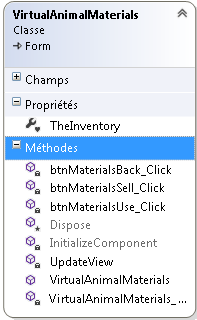
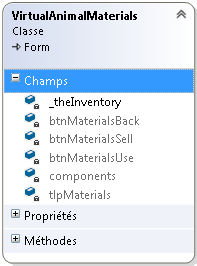


Figure : Diagramme de class fenêtre Matériaux

### Fenêtre aliments

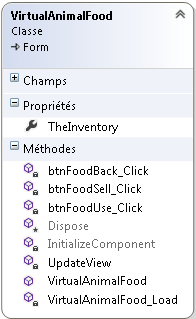
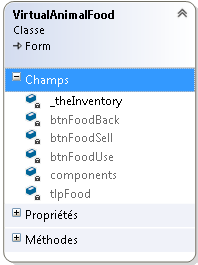


Figure : Diagramme de class fenêtre Aliments

### Class animal

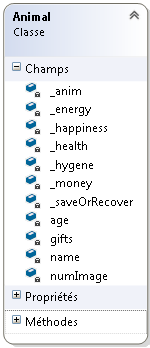


Figure : Diagramme de class Animal

### Class magasin

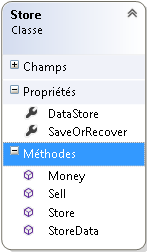
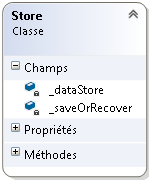


Figure : Diagramme de classe Magasin

### Class récupération de données

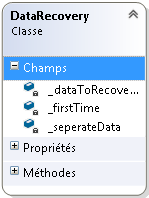
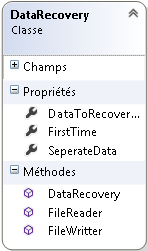
 

Figure : Diagramme de classe Récupération de données

### Class inventaire

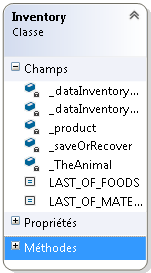
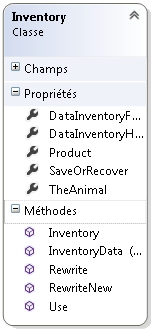
 

Figure : Diagramme de classe Inventaire

## Organigramme

### Fonctionnement général

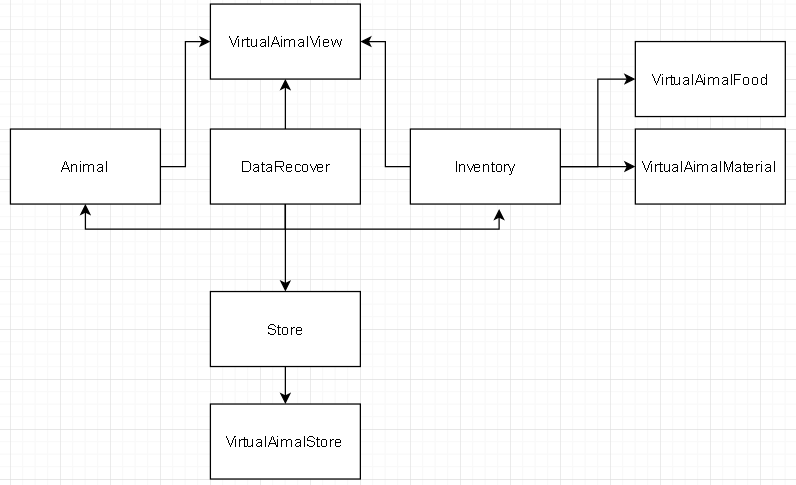


Figure : Organigramme générale

### Animation depuis la forme principale

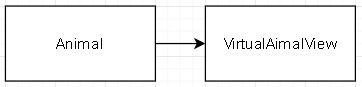


Figure : Organigramme animation depuis vue principale

### Animation depuis la forme aliments ou matériaux

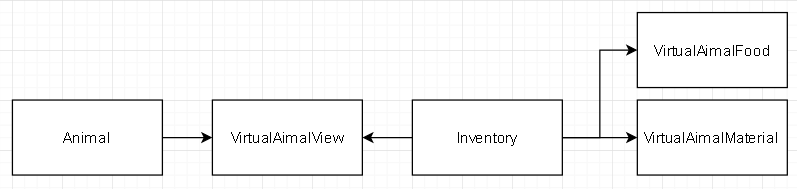


Figure : Organigramme animation depuis vue aliments ou matériaux

## Points techniques

### Méthode Rewrite

Cette méthode se trouve dans la classe inventaire (Inventory).

Celui-ci commence par rendre le booléen FirstTime vrai. Il fait ceci, car au moment d’écrire la première ligne il faut spécifier qu’il n’est pas sensé l’ajouter à la fin, mais écraser les données d’avant.

Les données comme le nom du produit et son prix sont stocké dans une liste nommé SeperateData. Voici un exemple de ce que la liste contient :

|  |  |
| --- | --- |
| Index | Contenu |
| 0 | Riz |
| 1 | 15.00 |
| 2 | 0 |
| 3 | Sushi |
| 4 | 35.00 |
| 5 | 0 |

La premier boucle for permet de retrouver les produits grâce au index et le deuxième permet de retrouver le prix de chaque produit.

Le foreach permet de parcourir chaque élément d’une liste. Ici la boucle permet de parcourir les éléments du dictionnaire de données et récupéré la valeur qui représente la nouvelle quantité.

Ensuite grâce au string. Format cette méthode regroupe les trois données (nom produit, prix produit et quantité) et ensuite le réécrit dans le fichier texte en appelant FileWriter.

### Méthode FileReader

Cette méthode se trouve dans la classe récupération de données (DataRecovery).

FileReader prend un string en paramètre. Ce string contient le nom du fichier texte à être lu.

Stream Reader est lancer, cette class permet la lecture des fichiers. Ensuite, le chemin d’accès au fichier est instancié. La liste de données est remise à zéro.

Finalement selon le nom du fichier et tant que la ligne de lecture n’est pas vide, les données sont ajouté à la liste et le fichier fermé.

### Méthode AnimalRecover

Cette méthode se trouve dans la classe animale (Animal) et commence par appeler la méthode FileReader pour lire le fichier Save\_Animal.txt.

Ceci récupère les données du fichier et instancie les niveaux de vie de l’animal et l’argent. Ces données sont enregistrées dans un ordre précis :

1. Santé
2. Hygiène
3. Energie
4. Joie
5. Argent

Ensuite, les mêmes étapes sont refaites sur le fichier Animal\_Age\_Name.txt, et récupérer le nom de l’animal et son âge, dans cette ordre.

# Protocole de test

## Plan de tests

### Vue principale

|  |  |  |
| --- | --- | --- |
| **N° de test** | **Description test** | **Résultat attendu** |
| 1 | Lorsque le bouton Activités/Magasin est cliqué. | La fenêtre magasin s’ouvre. |
| 2 | Lorsque le bouton Activités/Promenade est cliqué. | L’animation se lance. Niveau de joie augmente. Niveaux d’hygiène et d’énergie diminue. |
| 3 | Lorsque le bouton Inventaire/Aliments est cliqué. | La fenêtre aliments s’ouvre. |
| 4 | Lorsque le bouton Inventaire/Matériaux est cliqué. | La fenêtre matériaux s’ouvre. |
| 5 | Lorsque le bouton Animale/Caresser est cliqué. | L’animation se lance. Niveau de joie augmente. |
| 6 | Lorsque le bouton Animale/Dormir est cliqué. | L’animation se lance. Niveau d’énergie augmente au max. |
| 7 | Lorsque le bouton Animale/Sieste est cliqué. | L’animation se lance. Niveau d’énergie augmente. |
| 8 | Lorsqu’un des niveaux vitaux est égal à zéro. | Animation mourir se lance. Message indiquant à l’utilisateur que le jeu a fini et recommence. |
| 9 | Recommence ou première lancée. | Affiche animation naissance, TextBox et un bouton Commencer. Remets tout à zéro. |

### Vue magasin

|  |  |  |
| --- | --- | --- |
| **N° de test** | **Description test** | **Résultat attendu** |
| 1 | Rentrer des lettres dans le TextBox. | Le TextBox reste vide, car les lettres ne sont pas acceptées. |
| 2 | Rentrer des caractères spéciaux dans le TextBox. | Le TextBox reste vide, car les caractères spéciaux ne sont pas acceptés. |
| 3 | Rentrer des chiffres a virgule dans le TextBox. | Le TextBox affiche seulement les chiffres et pas la virgule (ni un point). |
| 4 | Rentrer des chiffres entiers inférieurs à 1 dans le TextBox. | Le TextBox affiche seulement les chiffres et pas le moins. |
| 5 | Rentrer un chiffre entier dans le TextBox. | Affiche le prix total correct. |
| 6 | Lorsque le bouton Acheter et rentrer est cliqué.  Assez d’argent. | La fenêtre se ferme. Quantité ajouté. |
| 7 | Lorsque le bouton Acheter et rentrer est cliqué.  N’a pas assez d’argent. | Message d’erreur s’affiche. La fenêtre se ferme. |
| 8 | Lorsque le bouton Acheter est cliqué.  Assez d’argent. | TextBox se vident et l’argent se remet à jour. |
| 9 | Lorsque le bouton Acheter est cliqué.  N’a pas assez d’argent. | Message d’erreur s’affiche. TextBox se vide. |
| 10 | Lorsque le bouton Rentrer est cliqué. | Ferme la fenêtre. |

### Vue aliments ou matériels

|  |  |  |
| --- | --- | --- |
| **N° de test** | **Description test** | **Résultat attendu** |
| 1 | Sélectionner produit, cliquer sur Utiliser.  Quantité produit n’est pas égal à zéro. | Ferme fenêtre, diminue quantité produit de 1. Lance animation. |
| 2 | Sélectionner produit, cliquer sur Utiliser.  Quantité produit est égal à zéro. | Message d’erreur. |
| 3 | Aucun produit sélectionné, cliquer sur Utiliser. | Fenêtre se ferme. |
| 4 | Sélectionner produit, cliquer sur Vendre.  Quantité produit n’est pas égal à zéro. | Quantité produit égale à zéro. |
| 5 | Sélectionner produit, cliquer sur Vendre.  Quantité produit est égal à zéro. | Rien. |
| 6 | Aucun produit sélectionné, cliquer sur Vendre. | Rien. |

## Rapport de tests

### Vue principale

|  |  |  |  |
| --- | --- | --- | --- |
| **N° de test** | **Date test** | **Résultat obtenu** | **OK** |
| 1 | 23.05.2017 | La fenêtre magasin s’ouvre. | ok |
| 2 | 23.05.2017 | L’animation se lance. Niveau de joie augmente. Niveaux d’hygiène et d’énergie diminue. | ok |
| 3 | 23.05.2017 | La fenêtre aliments s’ouvre. | ok |
| 4 | 23.05.2017 | La fenêtre matériaux s’ouvre. | ok |
| 5 | 23.05.2017 | L’animation se lance. Niveau de joie augmente. | ok |
| 6 | 23.05.2017 | L’animation se lance. Niveau d’énergie augmente au max. | ok |
| 7 | 23.05.2017 | L’animation se lance. Niveau d’énergie augmente. | ok |
| 8 | 23.05.2017 | Animation mourir se lance. Message indiquant à l’utilisateur que le jeu a fini et recommence. | ok |
| 9 | 23.05.2017 | Affiche animation naissance, TextBox et un bouton Commencer. Remets tout à zéro. | ok |

### Vue magasin

|  |  |  |  |
| --- | --- | --- | --- |
| **N° de test** | **Date test** | **Résultat obtenu** | **OK** |
| 1 | 23.05.2017 | Le TextBox reste vide, car les lettres ne sont pas acceptées. | ok |
| 2 | 23.05.2017 | Le TextBox reste vide, car les caractères spéciaux ne sont pas acceptés. | ok |
| 3 | 23.05.2017 | Le TextBox affiche seulement les chiffres et pas la virgule (ni un point). | ok |
| 4 | 23.05.2017 | Le TextBox affiche seulement les chiffres et pas le moins. | ok |
| 5 | 23.05.2017 | Affiche le prix total correct. | ok |
| 6 | 23.05.2017 | La fenêtre se ferme. Quantité ajouté. | ok |
| 7 | 23.05.2017 | Message d’erreur s’affiche. La fenêtre se ferme. | ok |
| 8 | 23.05.2017 | TextBox se vident et l’argent se remet à jour. | ok |
| 9 | 23.05.2017 | Message d’erreur s’affiche. TextBox se vide. | ok |
| 10 | 23.05.2017 | Ferme la fenêtre. | ok |

### Vue aliments ou matériels

|  |  |  |  |
| --- | --- | --- | --- |
| **N° de test** | **Date test** | **Résultat obtenu** | **OK** |
| 1 | 23.05.2017 | Ferme fenêtre, diminue quantité produit de 1. Lance animation. | ok |
| 2 | 23.05.2017 | Message d’erreur. | ok |
| 3 | 23.05.2017 | Fenêtre se ferme. | ok |
| 4 | 23.05.2017 | Quantité produit égale à zéro. | ok |
| 5 | 23.05.2017 | Rien. | ok |
| 6 | 23.05.2017 | Rien. | ok |

# Améliorations possibles

Je peux citer quelques améliorations qui serait possible d’apporter par la suite au programme.

## Audio

Dans la fenêtre principale, lorsqu’on lancera une animation, il serait intéressant d’ajouter du son. C’est un animal virtuel, mais qui est sensé représenté la réalité. Donc, l’ajout audio rendra les animations plus vif et du coup plus proche de la réalité.

## Fonctionnalités

Un autre point important serait d’ajouter la possibilité d’avoir plusieurs animaux par partie et plusieurs parties, vu que pour le moment le programme gère un animal et une partie.

Au lancement du programme il aurait un menu qui permettrait à l’utilisateur de choisir la partie qu’il aimera jouer et dans les activités ajouter une sortie pet shop, ou il pourrait acheter d’autres animaux.

En parlant des activités, une compétition de saut d’obstacle peut être ajoutée comme une manière de gagner de l’argent selon le score. Ceci apprendra à l’utilisateur que l‘argent rentre selon le travail fournie.

# Conclusion

Il existe un certain nombre de différences entre l’idée de base et le produit terminé. Les différences sont la cause de contraintes de fonctionnement ou de moyens mis en œuvre pour faciliter l’utilisation de l’utilisateur.

Au niveau des produits, la différence est que l’option de changement d’habits (papillon ou ruban) n’a pas pu être réalisé, car je n’avais pas d’images prêtes et plus le temps de les modifier.

Les difficultés rencontrées n’ont pas été nombreuses. D’autres problèmes comme une faute de frappe ou encore l’exécution de code alors qu’il ne remplissait pas les conditions voulues. Mais rien de très important qui a empêché l’avancement du projet

Dans l’ensemble du projet, je suis content du résultat obtenu grâce au fait d’avoir mené ce projet du début jusqu’à la fin.

Au niveau de l’environnement de travail, les conditions ont été excellentes. Le matériel mis à disposition et le fait de pouvoir compter sur l’aide de certains professeurs voir même collègues durant la réalisation du projet en cas de problèmes est des choses que j’ai fortement appréciées.

Il existe aussi une différence entre le planning avant le début du projet et le déroulement du projet.

Planning temporaire:

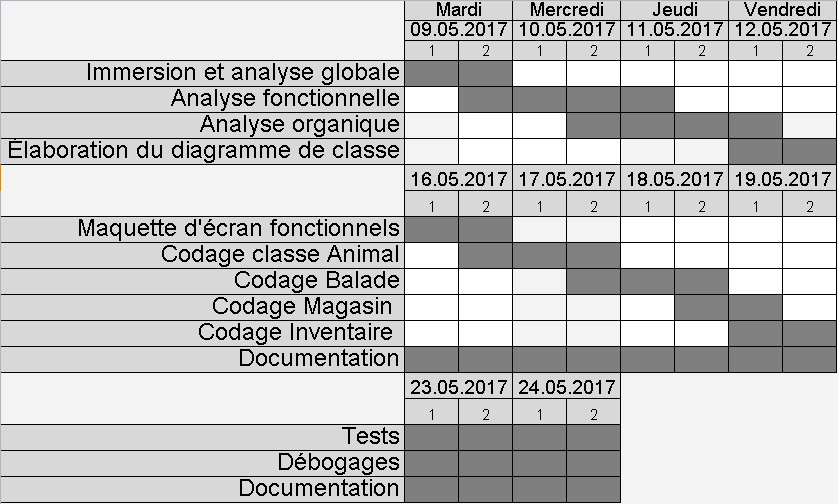


Figure : Planning temporaire

Planning :

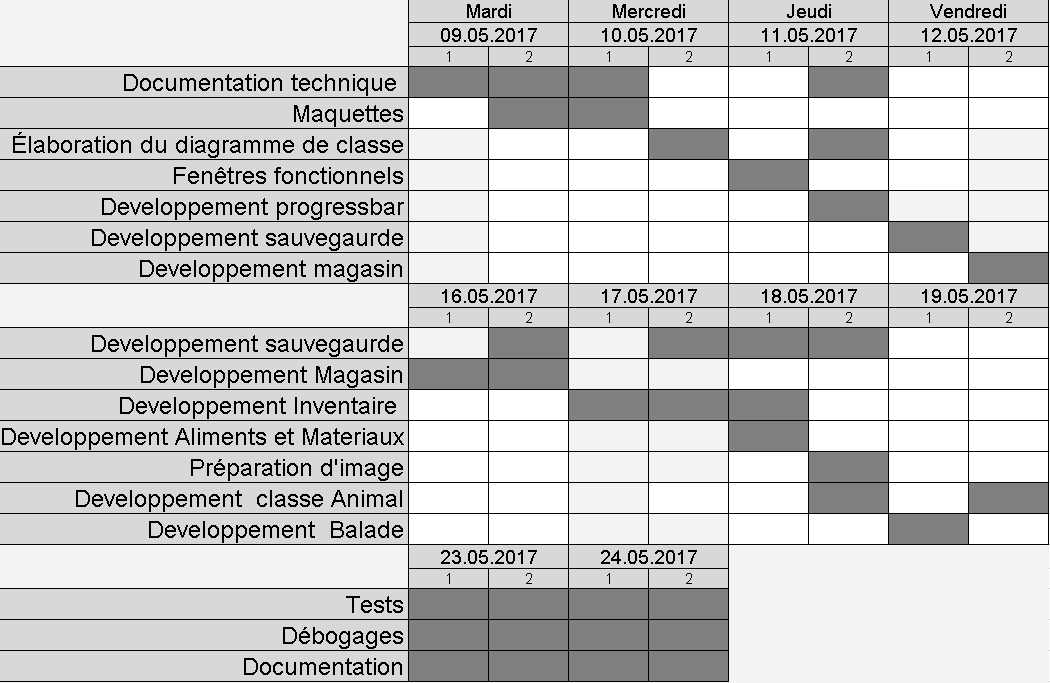


Figure : Planning final

# Table des illustrations

[Figure 1 : Exemple console Tamagotchi 5](#_Toc483315181)

[Figure 2 : Exemple application My Boo 6](#_Toc483315182)

[Figure 3 : Interface principale - Animal heureux 8](#_Toc483315183)

[Figure 4 : Interface principale - Animal malheureux 8](#_Toc483315184)

[Figure 5 : Exemple bouton 9](file:///C:\Users\SULZBACHJ_INFO\Desktop\AnimalVirtuel\SULZBACH_Doc_Technique.docx#_Toc483315185)

[Figure 6 : Interface magasin 10](#_Toc483315186)

[Figure 7 : Inventaire - Aliments 11](file:///C:\Users\SULZBACHJ_INFO\Desktop\AnimalVirtuel\SULZBACH_Doc_Technique.docx#_Toc483315187)

[Figure 8 : Inventaire - Matériaux 11](file:///C:\Users\SULZBACHJ_INFO\Desktop\AnimalVirtuel\SULZBACH_Doc_Technique.docx#_Toc483315188)

[Figure 9 : Diagramme de class fenêtre principale 12](#_Toc483315189)

[Figure 10 : Diagramme de class fenêtre Magasin 13](#_Toc483315190)

[Figure 11 : Diagramme de class fenêtre Matériaux 13](#_Toc483315191)

[Figure 12 : Diagramme de class fenêtre Aliments 14](#_Toc483315192)

[Figure 13 : Diagramme de class Animal 14](#_Toc483315193)

[Figure 14 : Diagramme de classe Magasin 15](#_Toc483315194)

[Figure 15 : Diagramme de classe Récupération de données 15](#_Toc483315195)

[Figure 16 : Diagramme de classe Inventaire 16](#_Toc483315196)

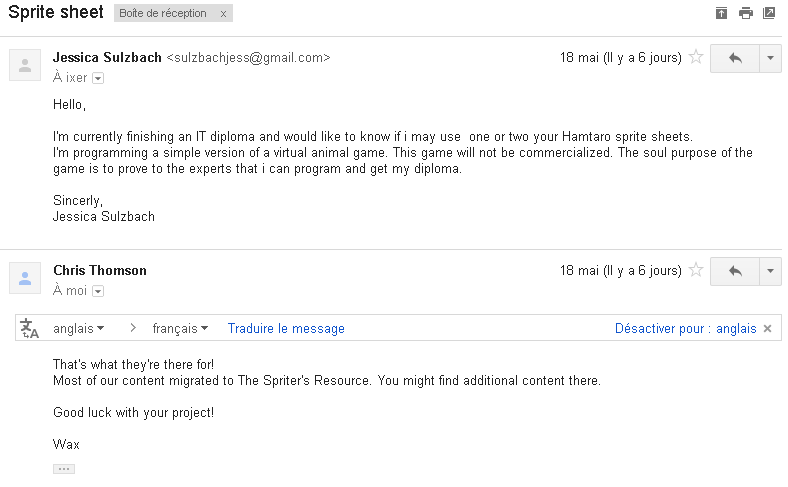
[Figure 17 : Organigramme générale 16](#_Toc483315197)

[Figure 18 : Organigramme animation depuis vue principale 17](#_Toc483315198)

[Figure 19: Organigramme animation depuis vue aliments ou matériaux 17](#_Toc483315199)

# Bibliographie

* Images animation
  + Demande de droit d’usage :



* + Site : http://tsgk.captainn.net/index.php?p=search&q=hamtaro
* Image background douche
  + Filtrage pour le droit d’usage Google : Réutilisation autorisée
  + Site : http://www.freepik.es/vector-gratis/vector-cuarto-de-bano-vacio\_722518.htm
* Image background promenade
  + Filtrage pour le droit d’usage Google : Réutilisation autorisée
  + Site : https://pixabay.com/fr/pierre-boulder-nature-granit-576669/
* Bouton à choix multiple
  + Site : https://stackoverflow.com/questions/10803184/windows-forms-button-with-drop-down-menu
* Dictionnaire de données
  + Site : https://www.dotnetperls.com/dictionary
* Récupéré la valeur d’un Object crée dynamiquement
  + Site : https://stackoverflow.com/questions/28276545/get-value-of-dynamically-created-textbox