

Projet Tamagochie

Benhachani Mohamed (chef du Projet), Kustul Kudret,
Goussoumalet Tanguy.

Vendredi 16 Decembre

Table des matières

I	Tamagochie	3
1	Introduction	4
1.1	Qu'est-ce qu'un Tamagochie	4
1.2	Les fonctionnalités de notre Tamagochie	4
1.3	Code	4
2	Architecture de notre Tamagochie	5
2.1	But de notre Architecture	5
2.2	Modélisation de notre Architecture	6
2.3	Diagramme	6
2.3.1	User-case	6
2.3.2	Classe	7
2.3.3	Etat-Transition	8
2.3.4	Interaction	9
2.3.5	Diagramme de Séquence Créer un Tamagoshi	11
2.3.6	Diagramme de Séquence Charger un Tamagoshi	12
3	Répartition des Tâches	13
3.1	Quelles sont les tâches	13
3.2	Leurs répartitions	14
3.2.1	BENHACHANI Mohamed	14
3.2.2	GOUSSOUMALET Tanguy	15
3.2.3	KUSTUL Kudret	15
4	Planning	17
5	Premier compte rendu	18
5.1	Compte rendu global sur l'avancement du projet	18
5.2	Compte rendu individuel de l'avancement du projet	18
5.2.1	BENHACHANI Mohamed	18
5.2.2	GOUSSOUMALET Tanguy :	19
5.2.3	KUSTUL Kudret	19

6	Compte rendu final	21
6.1	Compte rendu global	21
6.2	Compte rendu individuel	21
6.2.1	BENHACHANI Mohamed	21
6.2.2	GOUSSOUMALET Tanguy	21
6.2.3	KUSTUL Kudret	22
7	Conclusion	23

Première partie

Tamagochie

Chapitre 1

Introduction

1.1 Qu'est-ce qu'un Tamagochie

Le Tamagochie est un jeu dans lequel l'utilisateur doit s'occuper d'un animal pour a terme le voir évoluer.
Si l'utilisateur s'occupe mal du Tamagochie alors il meurt.

1.2 Les fonctionnalité de notre Tamagochie

Le Tamagochie peut boire de l'eau,dormir et se nourrir.Deplus au bout d'un Temps determine le Tamagochie evolue.
L'utilisateur peut aussi grace au fichier de configuration modifier les variables globale dans lesquels évolue le Tamagochie.

1.3 Code

Notre Application sera coder en JAVA, avec comme gestionnaire de version github : [https ://github.com/Benhachani-Mohamed/Tamagochie](https://github.com/Benhachani-Mohamed/Tamagochie) .

Chapitre 2

Architecture de notre Tamagochie

2.1 But de notre Architecture

Pour notre projet nous avons souhaité établir une architecture très genérique, et extrêmement modulable, c'est pourquoi nous avons eu comme ambition premier, d'établir une architecture qui puisse permettre la création d'un panel infini de Tamagotchies avec une diversité sans limite, tout en facilitant la création est la gestion de ces Tamagotchies.

C'est pourquoi nous avons divisé notre application en quatre composantes essentielles.

La première est la composante Menu qui gère la fenêtre et constitue la première interaction avec l'utilisateur, l'utilisateur pourra grâce à celle-ci charger ou créer un nouveau Tamagochie.

La deuxième composante c'est la composante Tamagochie, qui constitue toute la personnalité du Tamagochie, elle constitue le cœur du Tamagochie.

La troisième composante est la composante Moteur, elle sera utilisée par le Tamagochie pour stocker et sauvegarder des données sur le Tamagochie et ainsi assurer la pérennité de certaines données sur le Tamagochie et cela malgré les coupures de l'application.

Enfin la composante Graphique va gérer toute l'interaction graphique entre le Tamagochie et la fenêtre, c'est elle qui gère l'affichage du Tamagochie.

Pour des raisons d'ordre organisationnel nous avons décidé qu'il valait mieux concentrer nos efforts sur un seul Tamagochie.

2.2 Modelisation de notre Architecture

Après avoir conclu ensemble d'une Architecture viable, tant techniquement que théoriquement, nous avons souhaité pour des raisons d'efficacité et d'efficacité répartir, la rédaction des diagrammes, comme spécifié dans le chapitre 3 Répartition des Tâches.

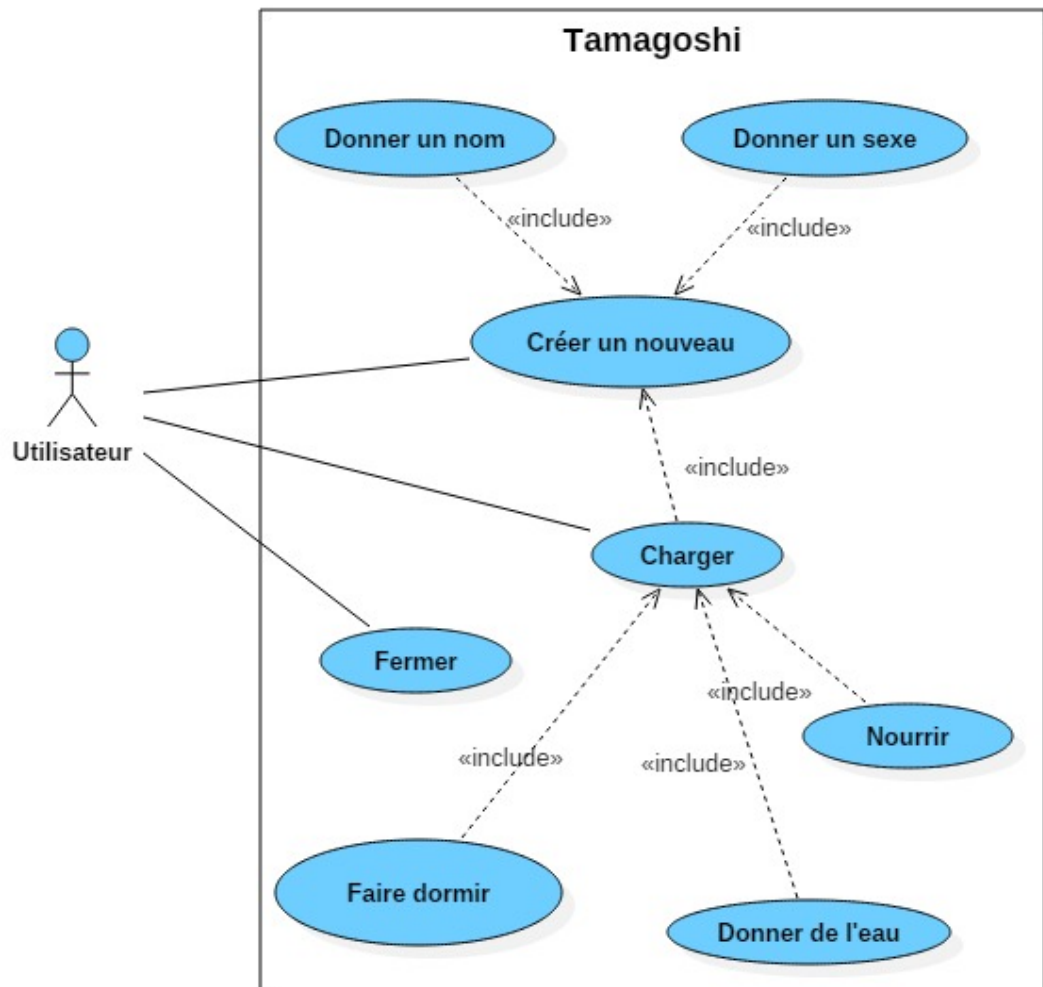
2.3 Diagramme

2.3.1 User-case

Un diagramme de cas d'utilisation est le premier diagramme le plus utilisé dans la modélisation UML qui décrit le comportement du système, à mettre en place, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système.

pour notre cas nous avons un acteur qui est l'utilisateur qui interagisse avec le système. et huit cas d'utilisation comme suit :

- Créer un tamagoshi : revient à lui donner un nom et un sexe.
- Charger un tamagoshi : le système charge tamagoshi et après l'utilisateur pourra agir sur tamagoshi en cliquant sur le bouton pour lui donner de l'eau, de nourriture et de faire l'endormir.
- Fermer : permet à l'utilisateur de quitter le système.



2.3.2 Classe

Le diagramme de classe permet de visualiser les liens inter classes. En effet, dans ce diagramme nous allons illustrer toutes nos classes et établir des liens entre ces dernières. Ce diagramme va permettre au lecteur de mieux comprendre le fonctionnement du logiciel.

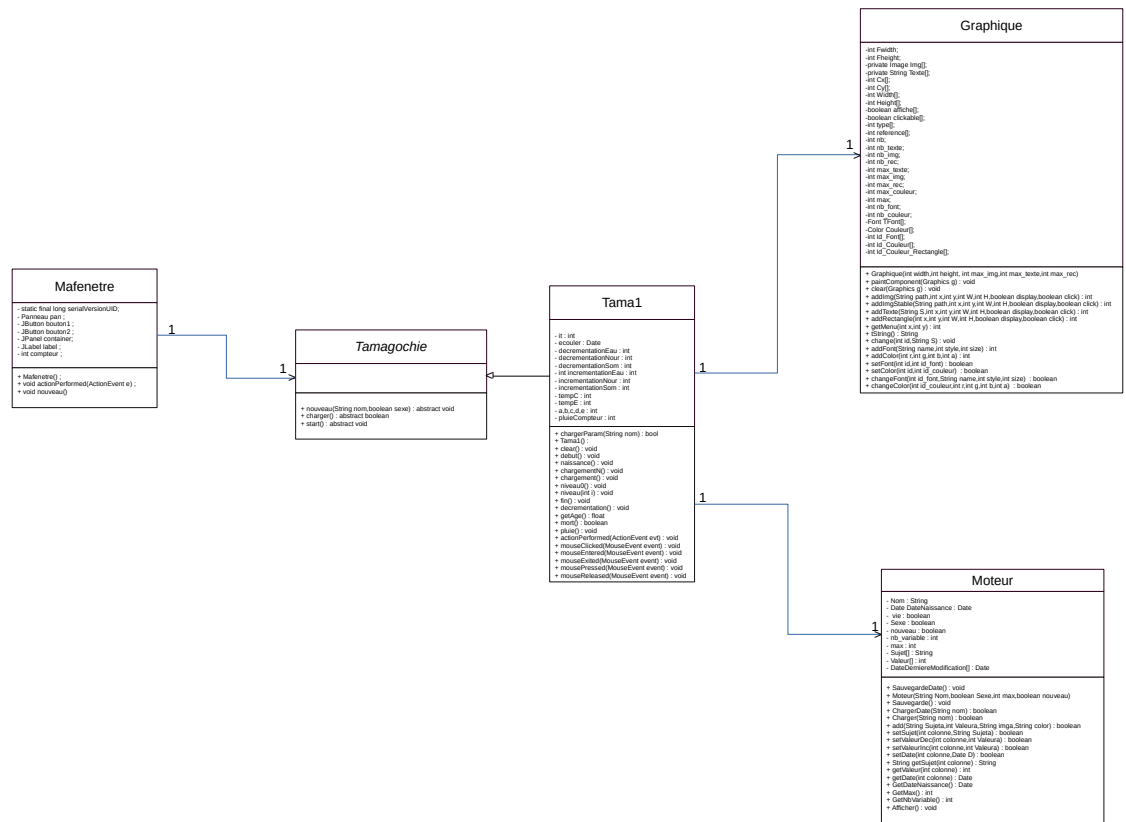
Nous avons 3 classes : Tamagochie , Graphique , Tama , Moteur.

Tamagochie : Tamagochie est une classe abstraite

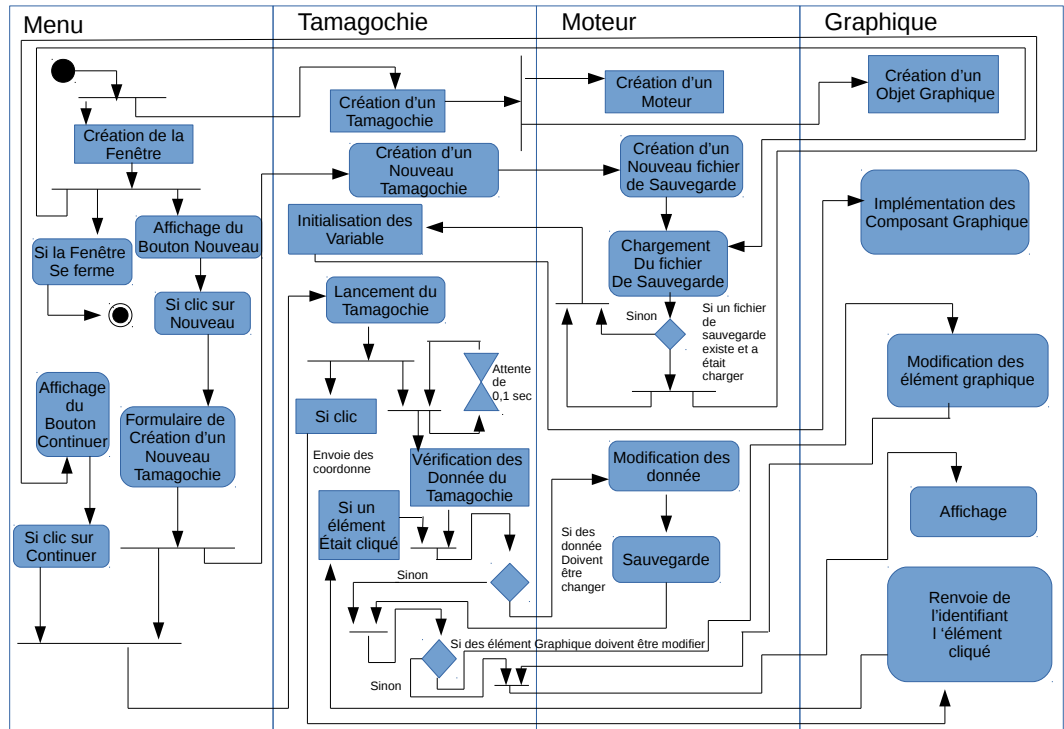
Graphique : Cette classe permet d'afficher des texte, des image , des rectangle coloré , des gifs.

Tama1 : Cette classe à pour attribut un Grapique et un moteur , en effet cette

Explication du diagramme : Un tamal est un moteur et un graphique le moteur gère la vie le nom le sexe du tamagoschie et le graphique va permettre l'affichage des images suivant l'âge du tamagoschie et la météo par exemple. Tamal est elle une classe abstraite de tamagoschie cette classe abstraite peut nous permettre de créer plusieurs tamagoschie par exemple mais nous avons limité le nombre de tamagoschie à 1 pour des raisons d'ergonomie.



Ce diagramme est diviser en quatre composante la composante Menu,Tamagochie,Moteur,Graphique.



2.3.4 Interaction

- Diagramme de Séquence

Il permet de décrire les cas d'utilisation de façon à mettre en évidence les interactions entre les instances des classes (objets) du logiciel. Il montre la séquence dans le temps des échanges de message entre les objets participant à un scénario.

Dimension verticale : le Temps

- l'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme

- le temps s'écoule de haut en bas Dimension Horizontale : les Objets et (les acteurs)

- l'ordre de disposition des objets sur l'axe horizontale est sans importance.

dans notre cas nous avons modéliser deux diagramme de séquences à savoir diagramme de séquence de cas d'utilisation créer un tamagoshi et diagramme de séquence de cas d'utilisation charger un tamagashi.

- Diagramme de séquence de cas d'utilisation Créer un tamagoshi :

l'utilisateur clic sur nouveau le système affiche un formulaire de création là l'utilisateur saisie le nom et définit le sexe de tamagoshi, il aura une phase de test

par le système pour vérifier si les informations saisie par l'utilisateur sont correct.

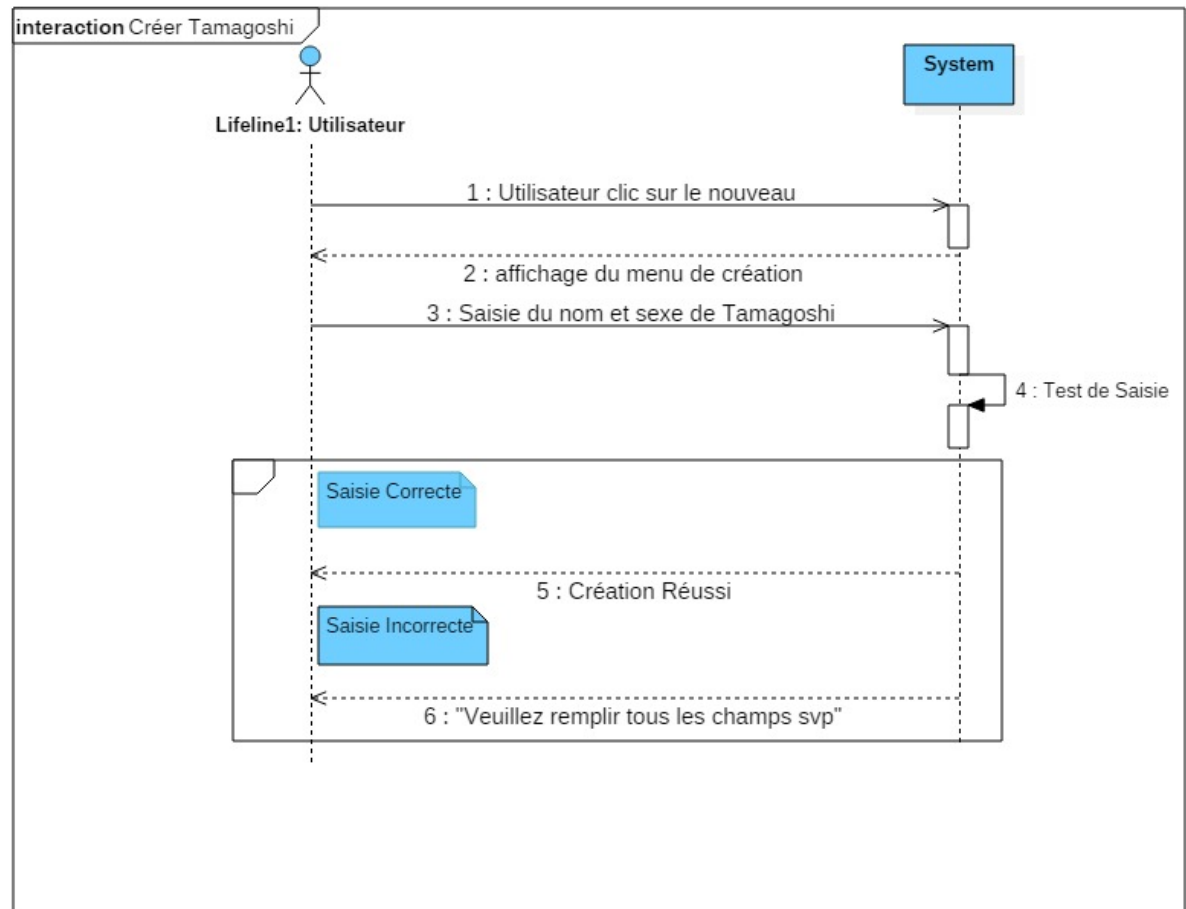
- si les informations sont correct il charge le tamagoshi.

- si les informations sont incorrect où l'utilisateur n'a pas entré le nom de tamagoshi ni définir son sexe et qu'il clic sur le bouton "OK" le système lui renvoie un message d'erreur "vous devez remplir les champs svp".

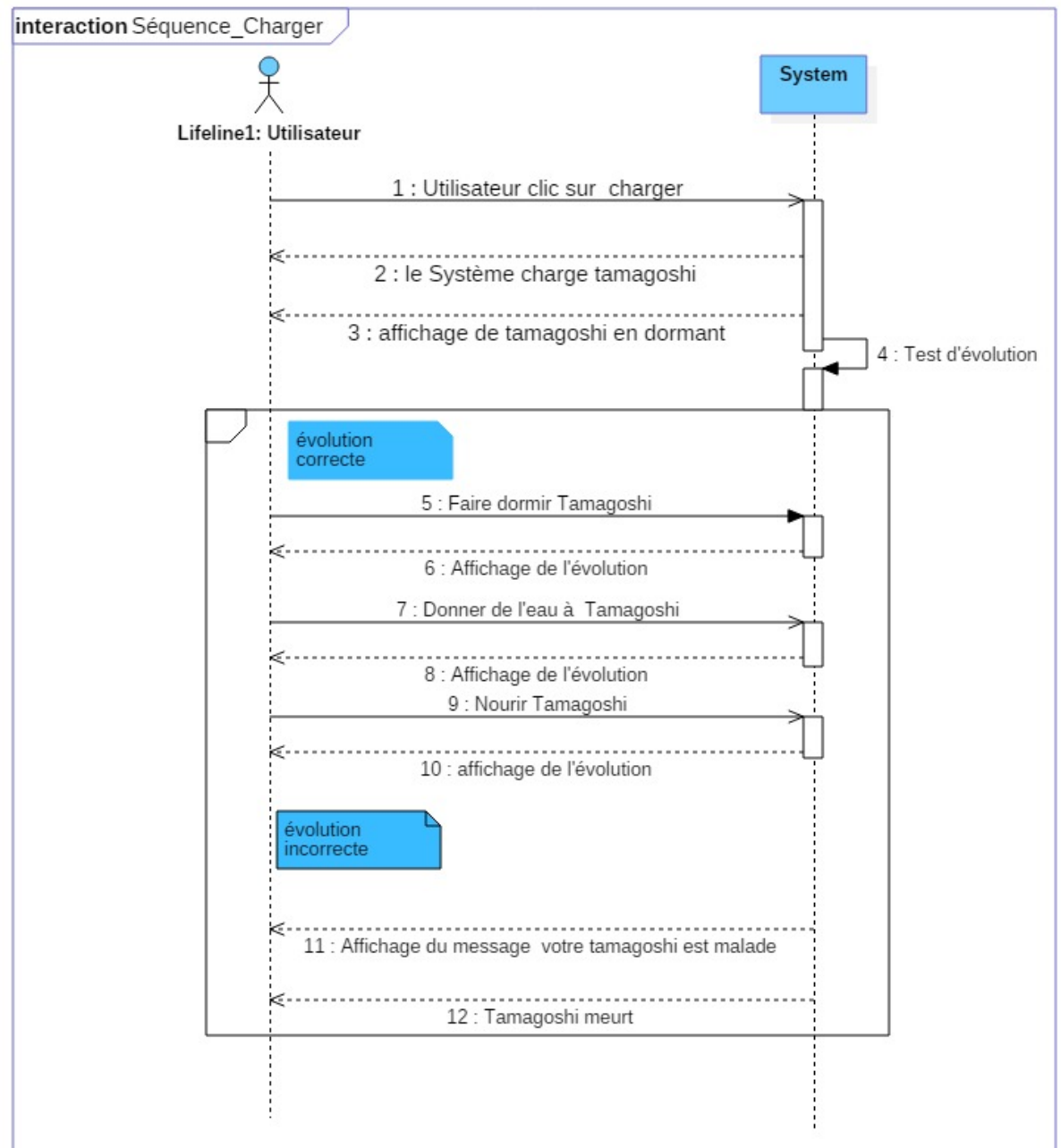
- Diagramme de séquence de cas d'utilisation Charger un tamagoshi :

l'utilisateur clic sur charger le système charge tamagoshi qui a été déjà créer et l'affiche, l'utilisateur peut faire endormir tamagoshi, nourrir tamagoshi, en lui donnant de l'eau, par un clic de bouton sinon le système affiche un message à l'utilisateur que le tamagoshi est malade, il perd ces points de vie. si l'utilisateur ne fait rien à un moment donnée le tamagoshi meurt.

2.3.5 Diagramme de Séquence Créer un Tamagoshi



2.3.6 Diagramme de Séquence Charger un Tamagoshi



Chapitre 3

Répartition des Taches

3.1 Quelles sont les tâches

Création des images

- création des gifs représentant le dromadaire (3 niveau)
- création du fond (desert,pyramide,cactus)
- création de la gif représentant la pluie
- création de l'image pour le bouton "boire de l'eau"
- création de l'image pour le bouton "manger"
- création de la gif pour le bouton "dormir"

Implémentation de la classe Moteur :

- définition des attributs de la classe
- implémentation du constructeur
- implémentation des fonctions charger et sauvegarder
- implémentation des ascenseur permettant l'accès ou la modification des variables private

Implémentation de la classe Tama1 :

- définition des attributs de la classe
- implémentation du constructeur
- implémentation de chaque fonction représentant une scène particulière du tamagochie,une scène est une suite de gifs,textes,image a affiché
- implémentation de la fonction qui va être appelé chaque demi-seconde par le timer,cette fonction va mettre en avant différentes scènes selon le temps et l'age du tamagochie et elle va aussi gérer la pluie.
- implémentation de la fonction qui va permettre de charger les paramètre paramétrable par l'utilisateur depuis le fichier config.txt
- implémentation d'une fonction qui va prendre en compte le temps écoulé et va décrémenter les variable vitale par rapport au temps qui ces écoulé

Rédaction :

- Diagramme de cas d'utilisation
- Diagramme d'interaction :
- Diagramme de séquence de cas d'utilisation Créer un tamagoshi
- Diagramme de séquence de cas d'utilisation Charger un tamagoshi.
- Diagramme de classe
- Diagramme d'état de transition

Implémentation du Menu :

- Création du fenêtré de Menu.
- Création de Bouton Nouveau.
- Création de Bouton Charger.
- Création d'un formulaire de saisie(nom et sexe).
- Création de Bouton OK.

Composant Graphique :

- Decider qu'elle bibliotheque graphique utiliser
- Definir une Architecture pour ma classe graphique
- Créer un constructeur pour la classe graphique
- Créer une fonction pour inserer des Image
- Créer une fonction pour inserer des Texte
- Créer une fonction pour inserer des Rectangle
- Créer des fonction pour changer la couleur de Rectangle ou Texte
- Créer des fonction pour changer la font des Texte
- Créer une fonction pour dessiner sur la fenetre
- Créer une fonction qui apartir de coordonée renvoie l'identifiant de l'element si il est en mode clickable
- Créer une fonction qui permet de changer le statut d'affichage d'un élément graphique (pour l'afficher ou non)
- Créer une fonction qui permet de changer le statut clickable d'un élément graphique (pour qu'il soit clickable ou non)
- Créer des fonction pour changer la dimension d'un élément graphique
- Créer des fonction pour changer la position d'un élément graphique
- Créer des fonctions qui retourne le mode d'affichage,le mode clickabilité,la position,la dimension, la couleur, et la font d'element graphique
- Réaliser une Javadoc

3.2 Leurs répartitions

3.2.1 BENHACHANI Mohamed

Toute la composante graphique ainsi que un prototype de Moteur,plus un programme de test de la classe graphique.

3.2.2 GOUSSOUMALET Tanguy

Modélisation :

- Modélisation de diagramme de cas d'utilisation
- Modélisation de diagramme d'interaction :
diagramme de séquence de cas d'utilisation Créer un tamagoshi
diagramme de séquence de cas d'utilisation Charger un tamagoshi.

Implémentation du Menu :

- Création du fenêtré de Menu.
- Création de Bouton Nouveau.
- Création de Bouton Charger.
- Création d'un formulaire de saisie(nom et sexe).
- Création de Bouton OK.

3.2.3 KUSTUL Kudret

Création des images

- création des gifs représentant le dromadaire (3 niveau)
- création du fond (desert,pyramide,cactus)
- création de la gif représentant la pluie
- création de l'image pour le bouton "boire de l'eau"
- création de l'image pour le bouton "manger"
- création de la gif pour le bouton "dormir"

Implémentation de la classe Moteur :

- définition des attributs de la classe
- implémentation du constructeur
- implémentation des fonctions charger et sauvegarder
- implémentation des ascenseur permettant l'accès ou la modification des variables private

Implémentation de la classe Tama1 :

- définition des attributs de la classe
- implémentation du constructeur
- implémentation de chaque fonction représentant une scène particulière du tamagochie,une scène est une suite de gifs,textes,image a affiché
- implémentation de la fonction qui va être appelé chaque demi-seconde par le timer,cette fonction va mettre en avant différentes scènes selon le temps et l'âge du tamagochie et elle va aussi gérer la pluie.
- implémentation de la fonction qui va permettre de charger les paramètre paramétrable par l'utilisateur depuis le fichier config.txt
- implémentation d'une fonction qui va prendre en compte le temps écoulé et va décrémenter les variable vitale par rapport au temps qui ces écoulé

Composante Grpahique :

Modélisation :

-Diagramme de classe

Chapitre 4

Planning

	semaine n°1	semaine n°2	semaine n°3	semaine n°4	semaine n°5	semaine n°6	semaine n°7	semaine n°8	semaine n°9	semaine n°10
-Composer une équipe de trois personnes										
-Choisir un chef de projet										
-Choisir un environnement										
-Choisir une méthode de développement										
-Définir les fonctionnalités potentielles										
-Faire le planning										
-Définir l'architecture global										
-Définir le modèle										
-Attribuer les différentes parties aux membres de l'équipe										
-Mise en commun du code réaliser par chacun										
-Vérifier si le code de chacun correspond aux objectifs fixer										
-1 ère version de l'aplication										
-test de la 1 ère version										
-rédaction du rapport des test										
-Definir les fonctionnalités supplémentaires										
-Attribuer les différentes fonctionnalités aux membres de l'équipe										
-Mise en commun du code réaliser par chacun										
-Vérifier si le code de chacun correspond aux objectifs fixer										
-Dernière version de l'aplication										
-test de la dernière version										
-rédaction du rapport des test										
-Developement de l'application										
-rédaction du rapport final										

Chapitre 5

Premier compte rendu

5.1 Compte rendu global sur l'avancement du projet

Une premiere Version du Tamagochie a etait faite, mais elle n'interagit pas avec l'utilisateur.

5.2 Compte rendu individuel de l'avancement du projet

5.2.1 BENHACHANI Mohamed

Après de nombreuse recherche, j'ai decidé d'utiliser les bibliotheque graphique inclue par default dans java et de faire de graphique une classe qui herite de la classe java Jpanel, pour pouvoir implementer la fonction paintComponent(java.awt.Graphics g). Tache effectuer :

- Decider qu'elle bibliotheque graphique utiliser
- Definir une Architecture pour ma classe graphique
- Créer un constructeur pour la classe graphique
- Créer une fonction pour inserer des Image
- Créer une fonction pour inserer des Texte
- Créer une fonction pour inserer des Rectangle
- Créer une fonction pour dessiner sur la fenetre
- Créer une fonction qui apartir de coordoonée renvoie l'identifiant de l'element si il est en mode clickable
- Réalisation d'une partie de la Javadoc

N'ayant pas pas implementer les fonction pour changer, et connaitre le mode d'affichage, de clickabilité,les dimension et la position des objet graphique, j'ai decidé de mettre temporairement certaine variable en public.

Deux difficulté majeure, la premiere étant de faire en sorte que les gif s'anime, car

par default elle ne s'anime pas, mais grâce a ma ténacité j'ai réussi a élaborer un protocole pour quelle marche. La deuxième est la cadence aléatoire d'affichage des images, en effet les bibliothèque par default de java, ne gère pas le nombre d'image par seconde dans la fenetre il était inconstant c'est pourquoi j'ai fait appel a un timer pour assuré un nombre minimum d'image par seconde.

5.2.2 GOUSSOUMALET Tanguy :

Etant reçu mes tâches, j'ai commencé à faire des recherches concernant la création de menu et l'affichage de fenêtre qui permet d'implémenter le Tam1 créée par mes collègues de groupes. la fenêtre était belle et bien conçue. après j'ai commencé à modéliser le système en créant des diagrammes. le bouton Charger qui permet de charger le tam1 marche normalement il m'implémente le tam1. Tandis que le bouton Nouveau qui permet de créer le tamagoshi qui ne marchait pas bien quand je clic sur le Nouveau le système affiche le formulaire de création de tamagoshi mais dans une autre fenêtre et il ferme l'autre fenêtre du coût quand je valide avec le bouton "OK" il me charge pas le Tam1.

5.2.3 KUSTUL Kudret

La classe moteur a été implémenté en première ,nous avons d'abord défini les attributs de cette classe, par la suite j'ai implémenté les fonction charger et sauvegarder.En parallèle ,mon ami BENHACHANI Mohamed(a.K.a la machine)a push sur gitHub une première version de la classe Graphique avec les fonctions nécessaires et suffisante a la création d'une première version de la la classe Tama1.J'ai donc implémenté une première version du tamagochie sans bouton ,sans évolution .Cette première version avait pour but de tester le bon fonctionnement des classes Graphique et Moteur.Les résultat se sont avéré concluant j'ai eu juste a demandé M BENHACHANI d'implémenté des ascenseur dans la classe Graphique pour pouvoir mettre tous les attributs de cette classe qui étaient en public auparavant.

Tâche réalisé :

Création des images

- création des gifs représentant le dromadaire (3 niveau)
- création du fond (desert,pyramide,cactus)
- création de la gif représentant la pluie
- création de l'image pour le bouton "boire de l'eau"
- création de l'image pour le bouton "manger"
- création de la gif pour le bouton "dormir"

Implémentation de la classe Moteur :

- définition des attributs de la classe
- implémentation du constructeur

- implémentation des fonctions charger et sauvegarder
- implémentation des ascenseur permettant l'accès ou la modification des variables private

Implémentation de la classe Tama1 :

- définition des attributs de la classe
- implémentation du constructeur

Chapitre 6

Compte rendu final

6.1 Compte rendu global

Le Tamagochie est fini est malheureusement le Menu ne permet pas la creation d'un nouveau Tamagochie personnalisé et ne possede pas de javado.

6.2 Compte rendu individuel

6.2.1 BENHACHANI Mohamed

Kudret souhaité une fonction qui apartir d'une gif, n'affiche qu'une image, j'ai par consequent fait une fonction qui affiche une gif, mais ne l'anime pas.

Tache effectuer :

- Créer des fonction pour changer la couleur de Rectangle ou Texte
- Créer des fonction pour changer la font des Texte
- Créer une fonction qui apartir de coordoonée renvoie l'identifiant de l'element si il est en mode clickable
- Créer une fonction qui permet de changer le statut d'affichage d'un élément graphique (pour l'afficher ou non)
- Créer une fonction qui permet de changer le statut clickable d'un élément graphique (pour qu'il soit clickable ou non)
- Créer des fonction pour changer la dimension d'un élément graphique
- Créer des fonction pour changer la position d'un élément graphique
- Créer des fonctions qui retourne le mode d'affichage,le mode clickabilité,la position,la dimension, la couleur, et la font d'element graphique
- Finir la Javadoc.

6.2.2 GOUSSOUMALET Tanguy

le menu a été implémenté correctement. le bouton charger permet de charger le Tam1 implémenté par mon ami KUSTUL Kudret me permet de l'appel pour

l'afficher dans ma fenêtre. le bouton nouveau qui permet de créer le tamagoshi qui ne marchait pas bien quand je clic sur le nouveau le système affiche le formulaire de création de tamagoshi mais dans une autre fenêtre et il ferme l'autre fenêtre du coup quand je valide avec le bouton "OK" il me charge pas le Tam1.

6.2.3 KUSTUL Kudret

La classe Tama1 a été implémenté a temps. Tous les bouton qui permettent a l'utilisateur d'interagir avec le tama ont été correctement implémenté .Une fonction qui permet de prendre en compte le temps écoulé lorsque l'utilisateur ferme puis charge sont tama après une durée a été correctement implémenté et est fonctionnel. La fonctionnalité de pluie a aussi été incrémenté .J'ai donc implémenté toutes les fonctionnalités que je devais faire. La partie la plus dur a été de prendre en compte le temps qui c'est écoulé hors ligne et de traité cette information. J'ai aussi implémenté une fonction "charger" qui va prendre des paramètres du fichier config.txt. Les paramètres sont : le temps de chargement ,les temps inter niveau ,le vitesse a laquelle le dromadaire meurt etc ...

Tâches réalisés :

Implémentation de la classe Tama1 :

- implémentation de chaque fonction représentant une scène particulière du tamagochie, une scène est une suite de gifs, textes, image a affiché
- implémentation de la fonction qui va être appelé chaque demi-seconde par le timer, cette fonction va mettre en avant différentes scènes selon le temps et l'age du tamagochie et elle va aussi gérer la pluie.
- implémentation de la fonction qui va permettre de charger les paramètres paramétrable par l'utilisateur depuis le fichier config.txt
- implémentation d'une fonction qui va prendre en compte le temps écoulé et va décrémenter les variable vitale par rapport au temps qui ces écoulé

Chapitre 7

Conclusion

Pour conclure on peut dire que le Tamagochie est une réussite, bien que mitigé car même si le Tamagochie, le moteur et Graphique on largement répondu au attente, Menu lui est incomplet est ne permet pas la création personnaliser d'un nouveau Tamagochie.