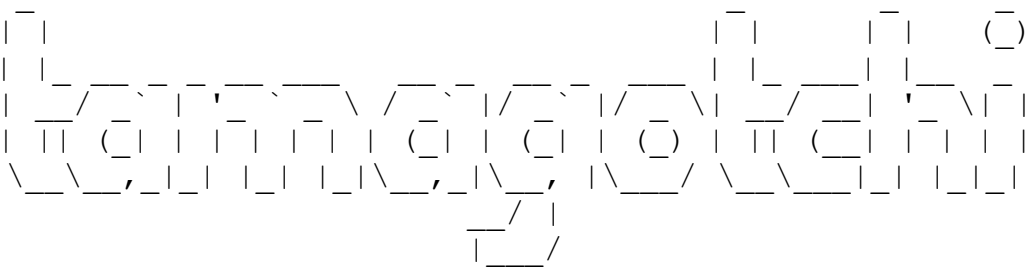
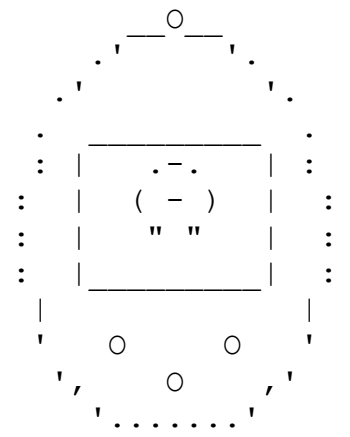
JOLIVET THIBAULT





Programmation Objet - JAVA

1/ Rappel

Une première version du Tamagotchi a été remise pour le 22 octobre, programmé en JAVA via le logiciel Eclipse.
Le but de ce projet était de nous familiariser avec la programmation objet via la création d'une classe objet Tamagotchi.

Certaines exigences définies en début de projets devaient être respectées :

- action : manger,
- action : aller aux toilettes,
- action : se teindre les cheveux, mais pas en roux,
- action : se reposer,
- caractéristique : couleur,
- action : se reproduire,
- action : exprimer son humeur ex : sauter de joie,
- action : voir des amis,
- caractéristique : santé (ex : s'il fume sa santé diminue et sa faim augmente),
- action : faire du sport (augmente la santé),
- action : mourir,
- caractéristique : taille (en cm ou en mètre)
- action : il peut enlever le masque (caractéristique : masqué ou non),
- caractéristique : espérance de vie,
- action : se laver.
Bonus :
- Caractéristique : Nom,
- Action : Choisir un type (Tamagotchi de type chien / Chat ... Dragon) à la création.

Pour cela, j'ai d'abord créé ma classe objet Tamagotchi, avec l'ensemble des méthodes nécessaires au bon déroulement du programme.

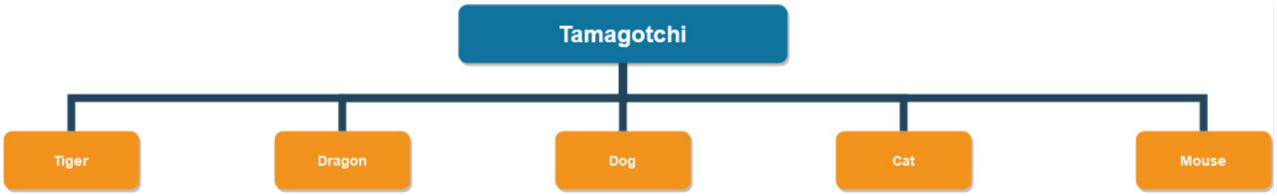
J'ai utilisé, pour une meilleure expérience utilisateur (UX), JOptionPane. Il s'agit d'une classe JAVA qui permet de créer facilement des boîtes de dialogue. Cela permet pour ce projet d'avoir un rendu plus visuel et plus original qu'avec la « console » JAVA.

Concernant le jeu en lui-même, le joueur pouvait choisir son type de tamagotchi parmi 5 propositions (Chat, Chien, Dragon, Tigre, Souris). Les résultats des méthodes pouvaient varier en fonction du type de tamagotchi (par ex : le chien mange un os, le chat mange une souris …). La gestion des différences entre type était faite via des boucles « if ».
Dès lors que nous avons des sous-catégories, il est utile de créer des « sous-classes » pour la réalisation du programme. On parle alors d'héritage (voir partie 2).

Le jeu se déroule sur un nombre de jour qui dépend de l'espérance de vie du tamagotchi ainsi qu'un nombre limité d'action par jour (8).

2/ Modifications apportées

Le but de la version 2 du programme est d'utiliser la notion d'héritage.
Ainsi, j'ai créé des « sous-classes » Dragon, Chien, Chat, Souris et Tigre, héritages de la classe objet Tamagotchi.



Au niveau des améliorations pour donner suite aux corrections et commentaires sur le V1 :

- Modification de la fonction d'apparition de l'image (problème de chemin absolu).
- Choix multiples proposés à l'utilisateur => plus d'entrée donc plus de casse à gérer et plus d'erreurs.
- arrondis à deux chiffres réalisé.

Pas de bugs mais des cas non gérés qu'on pourrait prendre en compte :
- pas réussi à faire fonctionner l'apparition des images
- gérer la casse quand on tappe l'animal par exemple
- pas de gestion d'erreurs si on tape n'importe quoi a la place d'un chiffre dans les actions => le programme est KO
- gérer les arrondis a deux chiffres pour la taille (c'est plus propre :))

Feedback général

Ca serait bien si les entrées dans la console ignoraient les majuscules.

- Modification de la méthode masque.
- Rajout de messages aux joueurs lorsque certaines valeurs vitales sont mauvaises.

Au dela des attentes

Point d'amélioration :
- gérer le fait qu'on puisse enlever et remettre le masque (et pas seulement l'enlever).
- Ajouter des msg aux joueurs lui indiquant quelles sont les actions qu'il devrait prochainement effectuer (ex : donne lui a manger sinon il va perdre de la vie).

- Rajout du « 12 ».

Petit oubli: pas de "12" devant "Montrer son humeur" du menu principal, mais ça marche donc pas de points enlevés.
j'ai pu lui faire faire toutes les actions sans bug.

J'ai, de plus, amélioré l'UX en utilisant uniquement des JOptionPane avec des choix multiples pour que l'utilisateurs n'ait pas à utiliser son clavier, sauf pour choisir le nom du Tamagotchi.
Cela évite également les erreurs de frappe et de ce fait sa gestion.

Diagramme de class Tamagotchi Version 2