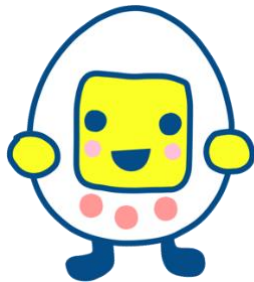


TD4 – Étude de Cas UML : Tamagoshi



On considère une application de Tamagoshi 2.0 accessible par le web. Dans cette application, un internaute peut visiter les galeries de Tamagoshis existants et déposer des messages à leur dresseur s'ils en ont (en laissant leur pseudo). Il peut aussi en choisir un ou plusieurs parmi ceux qui sont libres, pour les dresser (sans rien payer) ; mais pour cela il doit d'abord s'abonner.

Un administrateur s'occupe de lui fournir un compte à partir de son alias et d'un mot de passe, et lui attribue 50 Tamanas en cadeau d'entrée (le Tamana est la monnaie du jeu, taux actuel : 1 tamana = 1 euro)). L'abonné peut aussi acheter un Tamagoshi, cette fois en choisissant ses caractéristiques : âge, race, couleur, caractère, appétit. Il faut pour cela qu'il paye. À tout moment, le joueur peut acheter des Tamanas.

L'abonné peut en tant que dresseur, s'occuper du (ou des) Tamagoshi(s) dont il a la charge. Pour cela, il le nourrit (il faut acheter des ingrédients), il le distrait (promenade, chanson, jeux, lecture, cinéma... toutes les activités engendrent des achats), ou il le berce pour l'endormir. Chaque activité requiert un certain niveau d'énergie pour être effectuée (par ex. faire du tourniquet requiert plus d'énergie que d'aller au cinéma), et fournit une certaine quantité de bonne humeur (par ex. un bowling fournira plus de bonne humeur qu'une promenade). Le Tamagoshi possède un niveau de points de vie, fonction de son âge (fonction croissante avec de 0 à 24 mois, puis décroissante ensuite, sachant que la durée moyenne de vie d'un Tamagoshi est de 3 ans) et de son activité.

Les différents achats se font lors de transactions avec la boutique du jeu, en Tamanas. En plus de cela chaque début de semaine, le système attribue une certaine quantité d'éléments nécessaires au jeu aux abonnés (ingrédients, activités choisis au hasard, en quantité constante définie par le concepteur) : cela constitue leur « inventaire ». Le dresseur utilise ainsi son inventaire pour s'occuper de ses Tamagoshis. Un dresseur peut aussi vendre son Tamagoshi au système ou à un autre dresseur (en général, pour en acheter un autre).

Le Tamagoshi est d'autre part surveillé par l'application qui observe sa satiété, son sommeil, et son humeur (fonction de son ennui). Ces 3 éléments définissent un score correspondant à l'état général du Tamagoshi. Un mauvais état général conduit à des alertes envoyées par mail au dresseur, voir au décès du Tamagoshi. Des points de malus (là aussi, définis par le concepteur) sont affectés aux mauvais dresseurs, ce qui a un impact sur l'attribution hebdomadaire des ingrédients. Au contraire quand le Tamagoshi est très heureux, le système affecte des points de bonus « inventaire ».

Le système souhaite garder une trace de la qualité des dresseurs en enregistrant le taux de Tamagoshis morts pour mauvais soins. Quand ce taux devient trop important, on ne lui confie plus aucun Tamagoshi et le joueur est exclu.

Compléments

- On souhaite enregistrer la date de l'abonnement de tout nouveau Dresseur ; le dresseur choisit aussi un pseudo (affiché pour ses Tamagoshis et les messages).
- L'administrateur peut supprimer des messages, il est notifié de la présence de nouveaux messages pour pouvoir effectuer son activité de modération.
- Les galeries sont créées de façon automatique (et dynamique) par le système en fonction du type de galerie (mignons, bébés, hilarants, bleus...) et des caractéristiques des Tamagoshis ; l'administrateur a toutefois aussi la possibilité de les modifier si nécessaire, tout comme il a la main sur l'inventaire des dresseurs, les alertes, les bonus/malus des dresseurs.

Pistes de réflexions

- *Comment permettre le taux variable du Tamana ?*
- *Les alertes vont fonctionner à partir de seuils (seuil pour la satiété, sommeil, ennui). Comment les gérer ? il y a-t-il d'autres alertes à considérer ? D'autres constantes ?*
- *Quelles sont ici les gestions Métier à considérer ?*
- *Que faire des Tamagoshis morts ?*

Travail à faire

1. Quels seraient les **3 packages** à considérer pour cette application ?
2. Donner le **diagramme des cas d'utilisation** de chaque package.
3. Construire les **diagrammes de classes**, un par package.
4. Donner le **Diagramme d'activités** décrivant le processus « Distraire un Tamagoshi »
5. Donner le **Diagramme de Séquences** décrivant le scénario d'exception concernant l'achat d'un nouveau Tamagoshi par un *mauvais* dresseur (achat qui sera donc refusé).
6. Compléter le diagramme de packages avec les **dépendances** observées dans la modélisation (relation « use »).