DUBUC—TABOUY Amaury L2T5 INFO

30/05/2023

**Avatar**

En Java

AUDIE Cédric L2T5 INFO

**Professeur :** Mme Leschi



Table des matières

[Introduction : 2](#_Toc136267339)

[Contexte : 2](#_Toc136267340)

[Objectifs : 2](#_Toc136267341)

[Nos classes : 2](#_Toc136267342)

[Avatar : 2](#_Toc136267343)

[Admin : 3](#_Toc136267344)

[Question : 3](#_Toc136267345)

[ListeQuestion : 3](#_Toc136267346)

[Matiere : 3](#_Toc136267347)

[Bulletin : 4](#_Toc136267348)

[Defi : 4](#_Toc136267349)

[Test : 4](#_Toc136267350)

[Tickets : 5](#_Toc136267351)

[Fichier2 : 5](#_Toc136267352)

[Main : 5](#_Toc136267353)

[Problèmes et solutions : 5](#_Toc136267354)

[Conclusion : 6](#_Toc136267355)

# Introduction :

## Contexte :

Nous devions réaliser à 2, un jeu permettant de répondre à des questions tout cela grâce à un avatar créer pour chaque étudiant. En y ajoutant des particularités comme des points de vie ou des niveaux.

## Objectifs :

1. Réaliser ce qui nous étiez demander
2. Ajouter nos propres touches
3. Faire en sorte que c’est l’air d’un jeu
4. Pouvoir sauvegarder en mémoire les avatars et les questions

# Nos classes :

## Avatar :

Pour créer un avatar il fallait donc une classe Avatar.

Un avatar est définis par son pseudo, son level, ses points de vie, le mot de passe pour se connecter, d’un bulletin de note, et de 3 attribues permettant de recevoir des questions.

Un avatar une fois connecté peut faire ceci :

* Voir ses points de vie
* Faire une question
* Faire un test
* Voir son niveau
* Ajouter des notes
* Modifier ses notes
* Voir ses notes
* Lancer un défie
* Faire un ticket à l’administrateur
* Ce déconnecté

Il y a quelques méthodes qui sont présente mais pas utilisable par l’avatar comme :

* Supprimer un test ou une question
* Recevoir un test ou un défi
* Ajouter une question

## Admin :

Un admin à lui tous les droits. C’est-à-dire qu’il peut supprimer un avatar ou créer des questions et recevoir des tickets des avatars.

Il peut faire exactement ceci :

* Ajouter un avatar à la liste des avatars
* Ajouter ou supprimer un ticket
* Recevoir des tickets
* Supprimé ou modifié un avatar
* Ajouter, supprimer et modifier une question

## Question :

Une question est définis par son intitulé, ses réponses possibles, la réponse correct, le nombre de points quel rapporte et niveau de difficulté

Cette classe n’a pas de méthode particulière à pars les gets et les sets.

## ListeQuestion :

Une liste de question est composée de questions. Toutes les questions créées sont mis dans cette liste.

Cette classe à comme méthode :

* Retourner la taille de la liste
* Ajouter une question
* Ajouter une question à un test
* Supprimer une question
* Modifier une question

Elle permet de regrouper toutes les questions au même endroit et donc de choisir aléatoirement des questions.

## Matiere :

Une matière est composée d’un nom et d’un coefficient.

Elle permet juste de faire le lien entre les notes et le bulletin. Car une note attribuée à une matière.

Cette classe n’a pas de méthode particulière à pars les gets et les sets.

## Bulletin :

Le bulletin est un dictionnaire contenant en clé : la matière et en valeur : une liste d’entier correspondant à des notes. Chaque Avatar a un et un seul bulletin. Une note a son propre coefficient, mais la matière à elle aussi un coefficient.

Exemple : Les points de vie ajoutée sont tous les points d’une note au-dessus de la moyenne.

Donc un 15 coeff 2 donne 10 points de vie, mais si la matière est aussi coeff 2 alors j’ai 20 points de vie en plus au lieu de 10.

Les méthodes associées à cette classe sont les suivantes :

* Ajouter une note
* Modifier une note
* Supprimer une note

## Defi :

Un défi se joue à 2, un avatar lance un défi à un autre avatar. Pour cela un défi est donc composé de deux Avatars. La question envoyée est choisi aléatoirement parmi la liste de question. Elle est ensuite envoyée à l’adversaire, qui lui aura une méthode pour recevoir une question.

Cette classe a donc ces méthodes :

* Choix d’une question aléatoire
* Envoie de la question

De son côté l’adversaire aura 2 jours pour répondre à la question sinon il perdra des points de vie. S’il répond juste alors il en gagne et l’Avatar qui aura envoyé la question en perdra.

Suivant le niveau de la question et de si il répond juste, l’adversaire gagne ou perd des niveaux.

## Test :

Un test est un jeu solo, donc qui est fait pour se tester soit même. Il est définis par l’avatar qui veut se tester et d’une liste de questions.

La liste est composée de 5 questions choisit aléatoirement. Suivant le taux de réussite au test, on gagne des points de vie ou on en perd. Pareil pour les niveaux.

Cette classe n’a pas de méthode particulière à pars les gets et les sets.

## Tickets :

Un ticket est défini par un intitulé qui peut être en l’occurrence soit « pseudo », soit « mdp », un mot qui est le nouveau pseudo ou le nouveau mot de passe, l’Avatar qui fais le ticket et l’Admin qui va le recevoir.

Cette classe n’a pas de méthode particulière à pars les gets et les sets.

Il y a par contre 2 méthodes dans la classe avatar permettant de faire un ticket soit de changement de pseudo, soit de changement de mot de passe.

L’Admin lui peut consulter une liste où sont envoyés les tickets et peut faire le changement nécessaire.

## Fichier2 :

Ce fichier est le point de départ du programme. Il contient toutes les questions préalablement créé, certains avatars, un ticket et le seul et l’unique administrateur.

Toutes ces informations sont ainsi sauvegarder en mémoire, ce qui évite de repartir de zéro à chaque fois.

Cette classe n’a pas de méthode étant donné que c’est un main.

## Main :

Le fichier main, lui sert à gérer les principales exceptions et de faire le menu pour les Avatars et l’admin. Il met en relation toutes les classes expliquées plutôt.

# Problèmes et solutions :

Nous avons eu des problèmes pour les sauvegardes, car on avait créé trop de question et le fichier ne pouvait en contenir autant. Avant d’avoir le cours sur les inputs et les exceptions, on avait eu quelques problèmes.

# Conclusion :

Pour conclure, nous avons quasiment réussi tout ce que l’on c’était fixé, à pars que certaines méthodes crée ne sont pas utilisée. C’était très sympa comme projet car il nous permet de toucher à beaucoup de chose autour du java. Le fait que ce soit un jeu, nous donne d’avantage d’envie à le réaliser.