

TP1 – Introduction Java

Langage Objet, conteneurs et exceptions

Grâce à un voyage dans la DeLorean du docteur Emmet Brown, vous vous retrouvez seul(e) avec votre ordinateur en l'an 1975. Après avoir soufflé les bougies de Jeanne Calment qui fêtait ses 100 ans cette année-là, vous décidez de changer le cours de l'histoire en devenant le créateur d'un des tous premiers jeux de type shoot them up.

En effet, prenant de cours la société Taito qui l'inventa en 1978, vous décidez de développer le jeu « Space Invaders », afin de pérenniser votre nom et votre héritage avant votre retour dans le 21^{ème} siècle. Tout ça parce que oui, vous êtes cupide, et vous voulez vous faire du pognon.

Puisque vous avez votre ordinateur avec ce qu'il faut, vous décidez de développer ce jeu en Java !

(On cherchera plus tard comment une borne arcade de 1975 peut faire tourner une application en Java, arrêtez d'être si premier degré, c'est un sujet de TP, pas un article pour « Sciences Magazine », alors mettez-vous au boulot maintenant)

1. Créer la classe « Joueur »

Que ce soit dit : un jeu qui n'a pas de joueurs a autant de raison d'exister qu'un batteur à œufs sur un Vespa. Nous allons donc commencer par nous occuper des joueurs :

- 1) Créer une classe **Joueur** avec les attributs nom, prénom et pseudo.
- 2) Créer un constructeur public prenant ces différentes informations en paramètres.
- 3) Faire en sorte que les attributs nom, prénom ne puissent pas être modifiés par la suite grâce au mot clé final.
- 4) Créer une première méthode privée statique permettant de formater le prénom et le nom, de telle manière à ce qu'ils commencent par une majuscule et qu'ils ne contiennent ensuite que des minuscules. Vous pouvez utiliser les méthodes de la classe String (voir JavaDoc).
- 5) Faire appel à cette fonction dans le constructeur afin de formater les différents attributs de classe lors de la création d'un nouveau joueur.
- 6) Générer les getters du nom et du prénom via l'interface d'Eclipse (menu Source).
- 7) Modifier les getters afin de n'en laisser qu'un seul permettant d'obtenir une chaîne de caractères comprenant le prénom suivi du nom du joueur.
- 8) Générer le getters et setter du pseudo du joueur.
- 9) Pour votre classe Joueur, redéfinir la méthode toString() de la classe **Object** (utilisez le mot clé « override ») afin que cette méthode renvoie désormais le pseudo du joueur, suivi de son prénom et nom entre parenthèses.

- 10) De la même façon, redéfinir la méthode *equals(Object)* de la classe **Object** afin de comparer deux joueurs (comparer leur pseudo).

2. Créer la classe « SpaceInvaders »

Maintenant qu'on a nos joueurs, on peut commencer à construire le « jeu » à proprement parler. Parce que s'il n'y a pas de jeu, il n'y a pas de joueur. Pas de joueurs, pas de TP. Pas de TP, pas de note. Pas de note, pas de diplôme. Pas de diplôme ... pas de diplôme.

- 1) Créer une classe **SpaceInvaders** possédant un constructeur vide.
- 2) Ajouter la méthode *main()* à la classe. Ce sera le point d'entrée de notre jeu.
- 3) Créer une fonction privée d'initialisation créant 3 joueurs.
- 4) Appeler la fonction *init()* depuis le constructeur de la classe, puis instancier un nouveau SpaceInvaders dans la méthode *main()*.
- 5) Afficher la liste des joueurs à l'écran en utilisant un appel implicite à la fonction *toString()* de la classe **Joueur**.
- 6) Lancer le programme et vérifier l'affichage.

3. Créer la classe « Vaisseau »

Qui dit bataille spatiale, dit forcément vaisseaux spatiaux. Bah oui ... On ne va pas se battre dans l'espace avec une trottinette. Ça n'aurait aucun sens.

- 1) Créer une classe **Vaisseau** permettant de stocker les points de structure maximum de celui-ci, les points de bouclier maximum et s'il est détruit.
- 2) Créer les accesseurs (getters et setters) adéquats.

4. Créer la classe « Arme »

Nous avons donc maintenant des vaisseaux, c'est déjà pas mal, mais s'ils ne sont pas capables de se défendre, ils sont un peu inutiles. Ce serait comme envoyer un Airbus A380 pour faire la guerre, on est d'accord que ce n'est pas le meilleur plan de bataille de l'Histoire.

- 1) Créer une classe **Arme** comprenant un nom, les dégâts minimums et les dégâts maximum occasionnés.
- 2) Ajouter un enum **Type** avec différentes valeurs : Direct, Explosif, Guidé.
- 3) Ajouter une instance de cet enum en attribut de la classe **Arme**.

5. Créer la classe « Armurerie »

Pour ce qui est du stockage de vos armes, vous avez le choix : faire une classe « Armurerie », ou tout mettre dans vos poches. Je vous conseille la première option. Non, pardon, je vous l'impose.

- 1) Créer une classe **Armurerie** possédant une liste d'armes (utiliser la classe ArrayList).
- 2) Créer une fonction init(), appelée depuis le constructeur et qui a pour but de remplir l'armurerie (une arme de chaque type par exemple).
- 3) Compléter la classe avec les visibilitées, les méthodes et les mots clés que vous jugez nécessaires.

6. Mettre à jour la classe « Vaisseau »

Bien, nous avons maintenant nos vaisseaux d'un côté, et nos armes de l'autre.

Ça va, il n'y a rien qui vous choque ? Non ?

VOUS NE PENSEZ PAS QU'IL FAUDRAIT ACCROCHER LES ARMES AUX VAISSEAUX ????

- 1) Mettre à jour la classe **Vaisseau** afin d'y stocker une liste d'armes (maximum trois). Chaque arme possède un niveau d'amélioration.
- 2) Ajouter une fonction afin d'ajouter une arme à un vaisseau et une autre pour enlever l'arme.
- 3) Ajouter une fonction permettant d'afficher les armes présentes sur un vaisseau, ainsi qu'une fonction calculant le total de dégâts moyens infligés par le vaisseau.

7. Mettre à jour la classe « Joueur »

Un vaisseau sans pilote, c'est un peu comme une voiture sans permis : ça peut avancer, mais ça n'ira jamais bien loin, et souvent ça embête les gens qui sont autour.

- 1) Mettre à jour la classe **Joueur** afin qu'il puisse utiliser un vaisseau par défaut.

8. Mettre à jour la classe « SpaceInvaders »

On a bien bossé, on a créé plein de trucs super cool, maintenant on va commencer à les utiliser dans notre jeu !

- 1) Compléter la classe **SpaceInvaders** afin d'utiliser les dernières classes écrites.
- 2) Afficher l'armurerie.
- 3) Afficher toutes les informations disponibles concernant le vaisseau du joueur.

9. Pour aller plus loin (et essayer de gratter des points bonus)

9.a. Classe `ArmurerieException`

Créer une classe **`ArmurerieException`** héritant de la classe `Exception`. Lever cette exception lorsqu'on ajoute une arme n'étant pas liée à une arme de l'armurerie à un vaisseau.

Une attention particulière devra être portée à la visibilité des fonctions et attributs, ainsi qu'à la propreté du code.