

103 - Gain d'XP

Notions abordées :

- Ajouter des fonctions aux objets
- Permettre aux objets de modifier leurs propres données

But de ce TP :

Nous continuons à améliorer notre jeu vidéo de type RPG en ajoutant la notion d'amélioration des personnages.

A vous de jouer :

Déplacer les fonctions d'affichage

Notre app possède 2 fonctions permettant d'afficher soit un `Bot` soit un `Player` ; ces fonctions sont situées dans notre fichier principal.

Faites en sorte de déplacer ces fonctions à l'intérieur de leurs classes respectives, pour pouvoir demander à n'importer quel objet de type `Bot` ou `Player` de s'afficher lui même dans la console.

Vous pouvez modifier ou supprimer les paramètres reçus par ces fonctions si nécessaire.

Déplacer les fonctions d'attaque

En suivant le même principe que pour les fonctions d'affichage, déplacez les fonctions d'attaque à l'intérieur de leurs classes respectives.

L'objectif consiste à pouvoir demander à un objet de type `Bot` d'attaquer un objet de type `Player` avec une fonction située dans la classe `Bot` .

Inversément, on doit pouvoir demander à un `Player` d'attaquer un `Bot` avec une fonction située dans la classe `Player` .

Vous pouvez modifier ou supprimer les paramètres reçus par ces fonctions si nécessaire.

Remarque : la fonction de lancer de dés devrait rester dans notre fichier principal. Si vous réalisez ce TP dans un langage ne supportant pas les fonctions sans objets (Java ou C# par exemple), elle ne sera plus disponible dans les classes `Bot` et `Player` .

Pour contourner ce problème vous pouvez exceptionnellement dupliquer la fonction de lancer de dés et la copier dans `Bot` et dans `Player` . Nous verrons lors des prochains TP comment trouver une solution plus élégante.

Améliorer les fonctions d'attaque

Utilisez la variable `force` de l'objet qui attaque pour multiplier l'effet de la valeur des dés ; ceci s'applique aux `Bot` comme aux `Player` .

Ajouter une fonction de victoire pour les objets de type Player

Vous pourrez appeler cette fonction à la fin de chaque partie gagnée.

Faites en sorte d'améliorer le joueur après chaque victoire. Voici des exemples d'améliorations possibles :

- Augmenter la force d'un montant fixe (variable et tiré au sort)
- Augmenter la force d'une pourcentage (variable et tiré au sort). Vous pouvez changer le type de la variable `force` pour utiliser un réel à la place d'un entier.
- Ajouter une variable XP qui pourrait augmenter proportionnellement à la force du `Bot` vaincu.
- etc.

Gérer plusieurs tours de jeu

Dans votre fonction principale, veillez à ce que le joueur qui vient de gagner un `Bot` , puisse en affronter un autre, juste après. Tant qu'il est victorieux, il continue d'affronter des `Bot` , dès qu'il perd, vous lui affichez son résultat (nombre de bots vaincus ou un score que vous avez calculé).

Pour rendre le jeu plus intéressant, vous pouvez ajouter plusieurs fonctionnalités :

- Augmenter progressivement la force des bots
- Augmenter le nombre de bots qui attaquent simultanément le joueur
- Faire en sorte que l'amélioration du joueur à la fin de chaque partie gagnée soit proportionnelle à la difficulté du combat qu'il vient de remporter.