

Exercice 2 : Créer et détruire des Bots

Notions abordées :

- Utiliser plusieurs types d'objets
- Passer des objets en paramètres de fonctions
- Organiser son code

But de ce TP :

Nous continuons notre RPG en donnant la capacité de répondre à notre **Bot**.

A vous de jouer :

Partez de la fin du TP précédent (si vous ne l'avez pas terminé, prenez la correction).

Optionel : Essayez de garder l'historique de vos versions en utilisant Git ou en dupliquant le TP 101 avant de commencer le 102.

Créez un nouveau type "Player"

Un **Player** possède les propriétés suivantes :

- le pseudo est une chaîne de caractère standard
- la force est un entier qui sera utilisé pour chacune de ses attaques
- la santé est un entier qui représente le pourcentage de vie du joueur

Créez un objet Player qui représentera le joueur en cours

Dans votre fonction principale, créez un objet de type **Player**, donnez le nom que vous voulez à cette variable. Initialisez cette nouvelle instance avec une **force** de 1 et une **santé** de 100.

Le pseudo que vous avez demandé à l'utilisateur au lancement de votre jeu doit maintenant être stocké dans cet objet ; modifiez le code de votre fonction principale pour en tenir compte.

Le Bot contre-attaque

Après chaque coup du joueur, appliquez maintenant une attaque du **Bot** vers le joueur.

Utilisez les dés (sans avoir à appuyer sur Entrée) pour déterminer la force du coup que **Bot** assènera sur le joueur.

Affichez dans la console les messages nécessaires pour que l'on comprenne ce qu'il se passe (dés, coups, santé, etc.)

A la fin de la partie, affichez si le joueur a gagné ou si il a perdu.

Organisez votre code

Votre fonction principale commence à être trop grande et trop complexe ; il est temps de créer des sous-fonctions :

- Une procédure permettant d'afficher les infos essentielles d'un **Player** reçu en paramètre (pseudo, santé, force).
- Une procédure permettant d'afficher les infos essentielles d'un **Bot** reçu en paramètre (santé, force).
- Une procédure d'attaque du **Player** vers le **Bot**, qui reçoit les 2 objets en paramètres et lance l'attaque entre eux.
- Une procédure d'attaque du **Bot** vers le **Player**, qui reçoit les 2 objets en paramètres et lance l'attaque entre eux.

Utilisez ces procédures pour simplifier le code de votre fonction principale.

Equilibrez la partie

Pour l'instant, le joueur frappe toujours en premier, cela pourrait trop l'aider en cas de match serré.

Au début de chaque tour, trouvez un moyen pour tirer au sort et déterminer celui qui attaquera en premier.