

Exercice 4 : Stratégies d'attaques

Notions abordées :

- Ajouter des constructeurs à des classes
- Initialiser correctement des objets

But de ce TP :

Nous continuons à améliorer notre jeu vidéo de type RPG en ajoutant la notion de stratégie.

Pour le moment les victoires/défaites sont uniquement liées au hasard et l'utilisateur ne peut faire aucun choix. Nous allons modifier ceci et lui proposer plusieurs choix d'attaques.

A vous de jouer :

Initialisation des objets de type Player

Modifiez votre classe `Player` pour permettre de fournir le pseudo du joueur lors de la création d'un objet de ce type. Vous pouvez laisser des valeurs par défaut pour les autres champs (force, santé, etc.).

Suite à cette modification, vous devrez probablement tenir compte de cette nouvelle initialisation lorsque vous créerez votre objet `Player` en début de partie.

Initialisation des objets de type Bot

Modifiez votre classe `Bot` pour permettre de fournir la force du bot lors de la création d'un objet de ce type. Vous pouvez laisser la valeur par défaut de 100 pour le champ santé.

Suite à cette modification, vous devrez probablement tenir compte de cette nouvelle initialisation lorsque vous créerez des objets de type `Bot` au début de chaque tour d'attaque.

Proposer à l'utilisateur plusieurs choix au moment de l'attaque

Pour rendre le jeu plus intéressant, vous pouvez commencer à proposer des choix à votre utilisateur avant chaque attaque. Vous êtes libre de ces choix, voici quelques pistes si vous êtes en manque d'imagination :

- Proposer plusieurs forces d'attaque (force normale avec 75% de réussite, force doublée avec 50% de réussite, force divisée par 2 avec 100% de réussite)
- Proposer une attaque plus puissante, mais qui peut potentiellement lui coûter de la santé
- Proposer de passer son tour en échange d'une récupération de santé
- etc.

Amélioration de votre code

Essayer d'analyser votre code existant pour voir si vous pouvez le simplifier. Voici des exemples de

simplifications possibles :

- fonctions mal placées qui pourraient être déplacées dans des classes
- fonctions trop longues ou complexes
- code dupliqué à l'identique à plusieurs endroits de votre programme

Vous pouvez résoudre tous ces problèmes en déplaçant du code et en créant des fonctions si nécessaire.