

POO

Les exos en vrac vous permettent de pouvoir mettre en œuvre les connaissances acquises lors des cours.

Utilisez un depot `poo-php-exosenvrac.bwb` afin de faire cette serie. Utilisez les branches pour les différentes sections.

1. CREATION DE PERSONNAGE

Mettons-nous dans le cadre de développement d'un jeu vidéo tour par tour sur le thème de donjons et dragons.

Nous allons réfléchir sur une partie du jeu, la création de personnage.

Exercice 1 :

Un personnage possède des attributs de base comme la force, l'agilité, l'endurance, les points de vie ainsi que des points de mana (seuls les plus sensibles à la magie pourront extraire l'énergie quantique des milles vérités afin de manipuler cette force). En outre tous pourront interagir avec leur environnement. Les actions seront basiques comme par exemple attaquer un autre personnage, boire une potion, manger un casse-croute, ou bien soigner un autre personnage. Les points d'actions permettent de déterminer les actions possibles à effectuer par tour de jeu.

Les degrés de liberté des personnages sont au nombre de deux selon les axes x et y. La méthode `seDéplacer()` prendra les nouvelles positions en argument.

Pour rendre l'expérience ultime, le personnage est aussi composé d'un attribut `$race` permettant de modifier les attributs de base selon la race affectée au personnage (humain, elfe, orc, nain et lapinCretin). Cet attribut est en effet une instance d'une des races disponible 😊. Sur le même schéma l'attribut `$archetype` est une instance des différents archétypes proposés dans le jeu (voleur, archer, magicien, guerrier et moine). L'archétype, en plus d'apporter ses modifications aux attributs de base du personnage, modifient le comportement de certaines actions (soigner pour un moine crame du mana en lançant un sort, l'attaque de l'archer est à distance, etc).

Afin de pouvoir profiter du jeu en farmant le stuff, un personnage peut s'équiper avec deux types d'équipement : les armures et les armes. Pour plus de facilité les personnages ont une armure qui permet de se protéger d'une attaque en réduisant les dégâts subits, et une arme qui ajoute un modificateur a l'attaque.

- Ecrire les classes de tous les objets utilisés dans le jeu.

- Implémenter les méthodes afin de prendre en compte tous les éléments du jeu.
- Testez votre bouzin ! :D

Exercice 2 :

Ajout de la création de personnage.

Créez une classe PersoGenerator, qui permet de créer tranquillement un personnage. Vu qu'un seul personnage sera créé à la fois vous pouvez définir un attribut \$personnage qui se verra doté de tous ses attributs.

La création du personnage se fera par étape : (choix de la race, choix de l'archétype, armure, arme,...). Par exemple :

```
$pg->addRace(new Orc()); // affectera la race orc au personnage
```

A la fin vous prévoyez une méthode qui retourne le personnage fini en vous assurant que toutes les étapes ont été validées (en effet ce serait idiot d'avoir un personnage sans arme...)

2. ALGORITHME

Petite série d'exercice d'algo histoire de pas perdre la main.

Exercice 1 :

Implémentez la fonction suivante : « fonction addition(int \$a, int \$b) : int »

Exercice 2 :

Initialisez un tableau d'entier avec une dizaine de valeurs aléatoires. Ecrire un algorithme qui cherche une valeur dans ce tableau.

Exercice 3 :

A partir de l'algorithme précédent, implémentez la fonction suivante : « fonction finder(array \$tab, int value) : bool »

Exercice 4 :

```
<?php
$tab = [10, 25, 42, 31, 25];
$find = $tab[0];
for ($i=0; $i<count($tab); $i++) {
    if ($find < $tab[$i]) {
        $find = $tab[$i];
    }
}
echo $find;
```

Commentez le code précédent.

Exercice 5 :

En vous appuyant sur le code précédent, écrivez une fonction qui retourne la valeur trouvée dans le tableau passé en argument. Cette fonction DOIT être documentée.

Exercice 6 :

Implémentez une fonction qui effectue le traitement inverse...