



PYTHON

BATTLE GAME



BATTLE GAME

UN RPG HORS DU COMMUN

ÉNONCÉ

Le but de ce projet est de créer un jeu de rôle, qui permet de créer un héros traversant un couloir en combattant des monstres. Le héros sera défini par une classe, barbare, magicien...

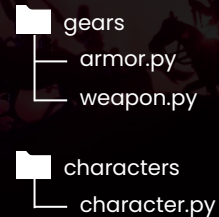
ARBORESCENCE DE PROJET

BIEN SE PRÉPARER

DOSSIERS & FICHIERS

Créer un dossier characters qui contient un fichier character.py.
Créer un dossier gears qui contient un fichier weapon.py et un fichier armor.py.
Sans oublier de créer le fichier main.py à la racine de du projet.

battle-game



ARMES & ARMURES

ÉQUIPEMENTS



IMPLÉMENTATION DES CLASSES WEAPON ET ARMOR

La classe Weapon possède deux attributs, name et damage.

La classe Armor possède deux attributs, name et defense.

Instancier au moins deux armes et deux armures dans le fichier main.py.

PERSONNAGE

HÉROS

IMPLÉMENTATION DE LA CLASSE CHARACTER

La classe Character va gérer les types de personnage
Les magiciens, les barbares... hériteront de cette classe.

attributs

hp
name
weapon
armor

méthodes

attack(...)

TEST DES ATTAQUES

AFFICHER LES INFORMATIONS

ATTAQUE D'UN PERSONNAGE SUR UN AUTRE

Dans le fichier main.py, instancier deux personnages différents, John et Jane par exemple, puis procéder aux attaques.



```
john.attack(jane)
print("Results: ")
print(john.name, john.hp, "HP")
print(jane.name, jane.hp, "HP")
input("-----")
```

```
jane.attack(john)
print("Results: ")
print(john.name, john.hp, "HP")
print(jane.name, jane.hp, "HP")
input("-----")
```

main.py



```
John attack Jane with Sword
Results:
John 100 HP
Jane 92 HP
-----
```

```
Jane attack John with Axe
Results:
John 85 HP
Jane 92 HP
-----
```

SORTIE CONSOLE

BARBARE & MAGICIEN

SPÉCIALISATION

IMPLÉMENTATION DES CLASSES BARBARIAN ET WIZARD

Dans le dossier characters, créer deux nouveaux fichiers, barbarian.py et wizard.py qui contiennent respectivement les classes Barbarian et Wizard.

Ces deux nouvelles classes héritent de la classe Character.

--

La classe barbare surcharge la méthode "attack" de manière à frapper deux fois.

La classe Wizard a deux nouveaux attributs, sorts et mana.

Créer une classe Spell avec les attributs name, damage et mana.

La méthode "attack" sera également surchargée afin de donner le choix au magicien entre lancer un sort ou attaquer avec son arme.



GESTION DU COMBAT

IL EST TEMPS DE RENTRER DANS L'ARÈNE

IMPLÉMENTATION DE LA CLASSE ARENA

La classe Arena gère le combat en faisant se battre deux personnages jusqu'à ce que l'un des deux meurt. Elle permet également l'affichage de mini-bilans après chaque "round" [une attaque chacun], ainsi qu'un bilan à la fin du combat.

attributs

first_caracter

second_character

méthodes

fight(...)





BONUS

BESTIAIRE

IMAGINER DES ENNEMIS

IMPLÉMENTATION DE NOUVEAUX MONSTRES

Créer quatre nouvelles classes, afin de pouvoir créer quatre types de monstres.
Ces classes héritent de la classe Character car elles aussi peuvent attaquer.



CARTE & SALLE

DIFFICULTÉ PROGRESSIVE

Salle 1

Tirage aléatoire
Monstre A

Salle 2

Tirage aléatoire
Monstre B

Salle du boss

Type boss
Monstre C

LE COULOIR

Créer une classe Map et une classe Room. Une carte a plusieurs salles. Chaque salle comprend un monstre du bestiaire tiré aléatoirement excepté la dernière salle qui sera dédiée au boss.

Afin de finir le jeu, l'utilisateur devra choisir entre jouer un barbare ou un magicien, battre toutes les salles et tuer le boss.

GESTION DE LA SAUVEGARDE

POUVOIR FAIRE UNE PAUSE À TOUT MOMENT

LIBRAIRIE PICKLE

À l'aide de la librairie pickle et de vos recherches, implémenter la sauvegarde lorsque le joueur quitte en milieu de partie. Au lancement du jeu, vérifier qu'une sauvegarde existe, si c'est le cas alors proposer au joueur de reprendre la sauvegarde ou de commencer une nouvelle partie.

<https://pypi.org/project/pypickle/>



À VOUS DE JOUER



MERCI DE VOTRE ATTENTION