Projet 2 : Un petit RPG

Thibault PENNING

Le but de ce projet est de se familiariser avec la programmation orienté objet en créant de nous même un petit RPG de combat au tour par tour.

Introduction

L'un des paradigmes les plus utilisés (et de loin) dans la création de jeu vidéo est celui de la POO. Chaque caractère, accessoires, décors, ..., visible et même souvent invisible sont des instances d'une classe.

Le but de ce projet consistera donc en la création d'un jeu vidéo. Le genre de jeu-vidéo dont nous allons nous intéresser est un jeu de rôle, plus précisément un RPG de combat tour par tour utilisant des combats de monstre (Pokémon, DigiMon, Dragon Quest ou encore TemTem).

> **Avertissement**
>
> Vous ne serez noté ici que sur la partie code, et plus précisément POO du projet. Toute création graphique, histoire, équilibrage, touchant au gameplay, voir même de programmation n'entrant pas dans le projet (ex: réseau). Vous ne serez évaluer QUE sur la bonne utilisation de la POO. Ainsi ne vous focalisez pas trop sur le jeu en lui-même.

Structure

Voici les points des principales classes à créer :

- Monstres
- Attaques
- Objets (boost, soin, ...)
- Combats
- Terrain
- Dresseurs/Humain

D'autres classes peuvent être créées.

Par exemple un `Monstre` contiendra surement une liste d'attaque possible, un ensemble de point de vie, un état ("normal", "endormie", …), un type, …

La classe `Combat` quant à elle s'occuperas de la gestion du tour par tour, des monstres en jeu, …

> **Astuce**
>
> Essayé de trouver un univers unique et qui vous parle. Amusez-vous à créer, c'est souvent plus marquant que copier !

Toutes les techniques présentées en classe peuvent être utilisées. Évité au maximum les choses trop complexes, rester simple.

N'hésitez pas à créer plusieurs fichiers. Si vous en ressentez le besoin vous pouvez utiliser des librairies de stockage des données de manière textuelle (afin de stocker les statistiques et les différents monstres, vous permettant de rendre un code plus propre), comme configparser, csv, TOML, ou encore JSON. La librairie Pickle peut aussi être utile même si elle est complexe à manipuler (mais peut servir de système de sauvegarde simplifié).

Seul la partie combat seras ici évalué. Toutes autres parties ne seront pas regardée.

Le gameplay n'est volontairement pas définie. La plupart d'entre vous se baserons (à juste titre) sur celui de Pokémon. Ne vous définissez pas un but trop compliqué au début. Faite simple et agrémenté le gameplay au fur et à mesure. N'oubliez pas que même si la forme est sympathique ici, le seul but et de vous faire pratiquer la POO. Ainsi préférer réaliser

Déroulement

> **Avertissement**
>
> Comme la dernière fois préférez la sécurité à la complétude.
> Rappeler vous que des points sont dédiés exclusivement à la justesse du projet. Construisez donc sur des bases solides et faites des tests !

Commencé petit, et ajouté au besoin de la modularité au fur et à mesure (combat à plusieurs, différente statistique, expérience, différents objets, …)

Par **groupe de 2 ou 3**.

S'il vous plait faites des **groupes mixtes en niveau**. Toutes les personnes doivent participer au projet. **L'investissement est un part de la note**. Ainsi il est conseillé au plus expérimenté de ne pas être toujours derrière le clavier, ou de dicter, mais d'être une aide technique pour

les autres. De la même manière que ne pas participer est dévalorisé, ne pas laisser les autres s'exprimer seras aussi dévalué.

Les rendus attendus sont les suivants :

1. Un fichier **README.md** (ou autre), décrivant le programme brièvement, son fonctionnement brièvement et surtout les entrée et sortie : comment exécuter votre programme (arguments, fonction utilisable en API, …) qui me permettent d'exécuter votre code.
2. Un fichier **SPECS.md** décrivant vos classes, avec chaque attribut et méthodes. Il est conseillé de créer un texte avant afin d'expliquer les règles de vos combats, de manière suffisamment détaillé pour comprendre toutes les subtilitésé de votre implémentation
3. Un fichier, sous quelconque forme que vous souhaitez, me permettant d'apprécier le **travail individuel de chacun et l'organisation du groupe**. Cela peut inclure un cahier de projet, un Gantt, un tableau des taches format Kaban, …
4. Les **fichiers en Python* (vous pouvez viser Python 3.11) sources qui permettent le fonctionnement de votre projet.

En plus de la production de code, il sera attendu des élèves une explication sous forme d'une **présentation orale**, du fonctionnement de leur projet et de la raison derrière leur choix. Les erreurs qu'ils ont faites face est une part non négligeable dans un projet, ainsi présenter les erreurs (résolu ou non) avec une explication est possiblement tout aussi valorisant qu'une explication d'un choix qui fonctionne.

Tout code de teste susceptible d'être laissé, s'il permet de s'assurer du bon fonctionnement du programme (cela est même valorisé).

Tout autres documents servant la compréhension du projet (cahier des charges, schéma, …) sera valorisé.

> **Note**
>
> Comme je vous ai précisé précédemment, l'interface graphique, si vous en réaliser une ne seras pas noté. Je vous déconseille fortement donc d'en réaliser une (à moindre d'avoir fini le projet).
> Cependant si vous souhaitez réaliser une interface minimaliste, n'utilisez pas TKinter ou PyGame. NCurse est une librairie intégrée dans Python permettant de créer des interfaces en terminal de manière très simple. Cela peut vous permettre de créer rapidement une interface pour gérer et tester les combats. Ces GUI ne seront tout de même pas noter, alors ne le faite pas si vous n'en voyez pas l'intérêt !