

# PHP POO

## Consignes :

Vous devrez me rendre le code commenté ainsi que le lien Github et un rapport que vous avez utilisé pour le projet dans votre groupe le tout dans un fichier .zip portant votre « nom+prénom+classe-php-Examen.zip ».

Vous devrez envoyer ce fichier à l'adresse mail suivante :  
[j.dubois@ecole-ipssi.net](mailto:j.dubois@ecole-ipssi.net)

Le rendu devra se faire au plus tard vendredi 13h00

Le travail se fera en groupe de 2 à 3.

Attention aux horaires, je ne prendrai pas en compte les contrôles envoyés après l'heure prévu.

Sur ce, bon courage !

## Dragon ball en PHP:

**Objectif :** L'objectif de ce TP est de créer un jeu basé sur l'univers de Dragon Ball en utilisant la programmation orientée objet, avec la possibilité de participer à plusieurs combats et de débloquent de nouvelles attaques.

### Description du Projet :

Vous allez créer un jeu simple basé sur l'univers de Dragon Ball en utilisant PHP et la programmation orientée objet. Le jeu consistera en la création de personnages, la gestion de leurs attributs (niveau de puissance, points de vie, etc.), et des combats entre ces personnages. Après chaque combat, les

personnages gagneront en puissance et débloquent de nouvelles attaques.

## Étapes du Projet :

### 1. Création de classes :

- Créez des classes pour les personnages, y compris les héros (Goku, Vegeta, etc.) et les méchants (Freezer, Cell, etc.). Chaque classe doit avoir des attributs tels que le nom, le niveau de puissance, les points de vie, etc.

### 2. Héritage :

- Utilisez l'héritage pour créer des classes de personnages héroïques et maléfiques qui héritent des attributs et des méthodes de la classe de base "Personnage".

### 3. Méthodes et actions :

- Ajoutez des méthodes aux classes de personnages pour gérer les actions comme "attaquer", "prendre des dégâts" et "mourir". Les méthodes spécifiques à chaque classe peuvent modifier ces actions en fonction du type de personnage (héros ou méchant).

### 4. Système de combat :

- Développez un système de combat qui permettra aux personnages de s'affronter. Le combat doit tenir compte des attributs de chaque personnage (niveau de puissance, points de vie, etc.). Vous pouvez créer des méthodes pour gérer les combats.

## **5. Gestion des statistiques :**

- Affichez les statistiques des personnages pendant le jeu, y compris leur niveau de puissance, leurs points de vie, etc. Vous pouvez créer des méthodes pour afficher ces statistiques.

## **6. Gestion de la victoire et de la défaite :**

- Écrivez du code PHP pour déterminer le vainqueur du combat et mettez en place des conditions de victoire ou de défaite. Après chaque combat, le personnage gagnant gagne en puissance.

## **7. Déblocage de nouvelles attaques :**

- Après le premier combat, le personnage gagnant doit débloquent une nouvelle attaque "Kamehameha". Vous pouvez créer une méthode pour gérer le déblocage de cette nouvelle attaque. Libre à vous de débloquent plus d'attaque.

## **8. Création de plusieurs combats :**

- Créez au moins trois combats différents, avec des personnages différents. Après chaque combat, mettez en place un nouveau combat avec de nouveaux personnages.

## **9. Système d'objectifs :**

- Ajoutez un système d'objectifs. Par exemple, après X combats, vous gagnez le jeu.

## **10. Interface Utilisateur (UI) :**

- Créez des fichiers HTML pour les différentes pages du jeu (par exemple, index.php, combat.php, magasin.php).

- Utilisez des formulaires HTML pour permettre aux utilisateurs de prendre des décisions pendant les combats.
- Intégrez du code PHP pour traiter les actions de l'utilisateur et mettre à jour l'état du jeu.

## **11. Économie de Jeu :**

- Introduisez un système d'économie où les joueurs gagnent de la monnaie virtuelle après chaque combat.
- Ils peuvent utiliser cette monnaie pour acheter des objets spéciaux ou améliorations.

## **12. Bonus :**

Améliorations Fonctionnelles :

Ajoutez des fonctionnalités supplémentaires pour améliorer l'expérience du joueur :

- Système d'expérience avec niveaux.
- Quêtes optionnelles pour gagner des récompenses.
- Personnalisation des personnages avec des accessoires.
- Environnements interactifs dans les combats.
- Éléments (feu, eau, etc.) avec des faiblesses et des forces.

## **13. Documentation et Commentaires :**

- Commentez abondamment votre code pour expliquer le rôle de chaque classe, méthode et variable.