

TP3 POO – Création d'un jeu de combat 3/4

TP3 POO - CREATION D'UN JEU DE COMBAT 3/4

DUREE TOTALE DU MODULE : 56H00

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

Travail à faire :

L'objectif de l'ensemble des 4 TP du module est de créer un modèle très simple de jeu de combat. Chaque visiteur pourra créer un nouveau personnage, avec lequel il pourra frapper d'autres personnages. Le personnage frappé se verra infligé un certain nombre de dégâts.

Un personnage est défini selon 2 caractéristiques, son nom et ses dégâts (entre 0 et 100).

Chaque coup porté au personnage lui infligera 5 points de dégâts. Une fois arrivé à 100 points de dégâts, le personnage est mort (supprimé de la BDD).

Notions étudiées durant les 4 TP :

- Les attributs et méthodes ;
- L'instanciation de la classe ;
- L'interconnexion avec la BDD ;
- L'héritage de classes ;

Dans ce troisième TP, nous focaliserons nos efforts sur la **mise en place des interfaces de connexion entre notre front et nos classes**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

Dans un but de simplification de cette étape, nous allons créer une page unique « index.php », qui se chargera de l'affichage et du traitement des données.

La partie d'affichage sera volontairement très simplifiée pour alléger au maximum le travail nécessaire.

Vous êtes libre de scinder cette partie en architecture MVC si vous le souhaitez, et d'améliorer les interfaces mais vous risquez de vous heurter à des problèmes qui ne sont pas le but de ce cours.

1) Formulaire de création/connexion

Dans un premier temps, nous allons créer un formulaire nous permettant de créer de nouveaux personnages, et de nous connecter à un personnage existant.

- Créez un formulaire HTML de méthode POST
- Ajoutez un champ de type text permettant la saisie du nom du personnage
- Ajoutez 2 boutons de validation type submit, ayant pour valeur respectivement « Créer un personnage » et « Utiliser un personnage ».

2) Inclusion des classes et connexion à la BDD

Avant de pouvoir traiter notre formulaire, nous avons besoin de nous connecter à notre BDD, et d'inclure nos classes au sein de notre page index

- Ajoutez une connexion PDO à votre BDD au sein de votre fichier index (\$db = new PDO ...)
- Chargez votre 2 classes Personnage et PersonnagesManager (include())
- Instanciez un nouvel objet PersonnagesManager (\$manager = new Manager...)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

3) Traitement du formulaire

Une fois le nom saisi, et le bouton de création ou connexion validé, nous avons besoin d'ajouter un traitement PHP permettant soit de créer un objet Personnage et de l'enregistrer dans la BDD, soit de récupérer l'objet Personnage depuis la BDD.

Création :

- Ajoutez une logique conditionnelle s'exécutant uniquement si un nom a été soumis, et que vous êtes en mode création (if(isset ...))
- Au sein de cette condition, créez un nouveau personnage prenant comme paramètre l'ID 0, le nom transmis par le formulaire, et des dégâts à 0
- Ajoutez un test permettant de tester si le nom de personnage existe déjà.
 - o Si c'est le cas, affichez un message « Le personnage existe déjà », et supprimez votre objet personnage à l'aide de la méthode unset() appliquée à votre objet
 - o Sinon, Insérez votre Personnage au sein de votre BDD.

Connexion :

- Ajoutez une logique conditionnelle s'exécutant uniquement si un nom a été soumis, et que vous êtes en mode Connexion (if(isset ...))
- Si le personnage existe, créez un nouvel objet Personnage en récupérant ses informations depuis la BDD
- Si ce n'est pas le cas, affichez un message d'erreur « Ce personnage n'existe pas ».

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

4) Ajout de la partie duel

Une fois la partie 3 achevée, vous êtes en mesure de créer de nouveaux personnages, et de vous y connecter.

Il n'est cependant toujours pas possible de faire se combattre notre personnage.

Nous allons à présent ajouter une partie spécifique aux duels, et son traitement.

Une fois le personnage créé ou sélectionné, nous allons donc cacher le formulaire, afficher les informations du personnage sélectionné, la liste des autres personnages et leurs informations, et enfin ajouter la possibilité de cliquer sur l'un d'entre eux pour le frapper.

- Ajouter une condition autour de votre formulaire, testant l'existence d'un objet personnage (if(isset(\$perso)) ...
 - o Si aucun objet Personnage n'est créé, afficher le formulaire de connexion / création
 - o Si un objet est créé, afficher l'interface de duel suivante :

Mes informations

Nom : ron

Dégâts : 10

Qui frapper ?

AVENEL (dégâts : 10)

sdfg (dégâts : 50)

- La première partie de l'interface représente les informations de notre personnage. Pour la créer, nous aurons besoin d'un fieldset, avec un champ légende « Mes Informations »
A l'intérieur, nous utiliserons les méthodes nom et dégâts sur notre personnage pour récupérer ses informations, que nous afficherons à l'intérieur d'un paragraphe.
- La seconde partie correspond aux informations sur les autres personnages, que nous allons pouvoir combattre.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

La récupération de cette liste se fait grâce à la méthode getList de notre manager.

- Si cette liste est vide, nous afficherons un message « Personne à frapper ! »
- Si ce n'est pas le cas, nous procurerons l'ensemble des personnages à l'aide d'une boucle foreach()
 - Au sein de cette boucle, nous afficherons un lien, permettant de frapper le personnage à l'aide du code suivant :

```
echo '<a href="?frapper=', $unPerso->id(), '>',  
htmlspecialchars($unPerso->nom()), '  
</a> (dégâts : ', $unPerso->degats(), '<br />';
```

5) Connexion persistante, et déconnexion

A cette phase, vous devez être en mesure de vous connecter, de créer de nouveaux personnages, et d'afficher les différentes interfaces connexion/création et combat.

Si ce n'est pas le cas, n'hésitez pas à reprendre le fonctionnement de vos classes Personnage et PersonnagesManager pour ajuster vos codes.

Cependant, si vous rechargez votre navigateur, vous être obligé de vous reconnecter à votre personnage à chaque fois.

De plus, il est pour l'instant impossible de changer de personnage pour lancer un autre duel.

Nous allons donc ajouter une sauvegarde persistante de notre connexion à l'aide de variables de session, et ajoutez la possibilité de se déconnecter pour lancer de nouveau duel.

- Ajoutez une initialisation de session en haut de fichier (session_start())

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

- Ajoutez une condition sur la variable de session « perso ». Si celle-ci existe, instanciez un nouvel objet personnage (Attention, cette action doit être effectuée avant l'affichage conditionnelle du formulaire ou de la zone de combat).
- Modifiez la création et la sélection de personnage, pour que ces fonctions instancient une nouvelle session perso, stockant votre objet personnage.
- Ajoutez un lien de déconnexion sur l'interface de combat.
- Ajoutez un test sur le getter de déconnexion. S'il est actif, utilisez une `session_destroy()`, et rechargez la page à l'aide de la méthode `header()`

6) Ajout du système de combat

Il nous reste une dernière étape avant que notre interface soit totalement fonctionnelle et non des moindres : l'ajout du système de combat !

Comment doit se passer la phase de traitement ? Avant toute chose, il faut bien vérifier que le joueur est connecté et que la variable `$perso` existe, sinon nous n'irons pas bien loin. Seconde vérification : il faut demander à notre manager si le personnage que l'on veut frapper existe bien. Si ces deux conditions sont vérifiées, alors on peut lancer l'attaque.

Pour lancer l'attaque, il va falloir récupérer le personnage à frapper grâce à notre manager. Ensuite, il suffira d'invoquer la méthode permettant de frapper le personnage.

- Ajoutez une condition s'exécutant uniquement si le getter « frapper » est actif, et ajoutez-y :
 - o Un test sur la variable `$perso`. Si celle-ci n'est pas définie, on affiche un message d'erreur et on arrête le traitement
 - o Un test sur la personne à frapper (`$_GET['frapper']`). Si celle-ci n'existe pas au sein la BDD, on affiche un message d'erreur et on arrête le traitement.
- Si toutes ces vérifications sont vérifiées, faites frapper le personnage sélectionné par votre personnage, et stockez le retour au sein d'une variable `$retour`

```
$retour = $perso->frapper($persoAfrapper);
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	

TP3 POO – Création d'un jeu de combat 3/4

- Ajoutez un switch case suivante la valeur de votre variable retour. Si retour vaut :
 - « ErreurCible », affichez un message « Impossible de se frapper » et arrêtez le traitement.
 - « PersonneFrappee », affichez un message « Le personnage a bien été frappé », modifiez votre personnage, et modifiez le personnage ciblé

```
$manager->update($perso);
```

- « PersonneTuee », affichez un message « Le personnage a été tué », modifiez votre personnage, et supprimez le personnage ciblé

Félicitations, votre jeu de combat est maintenant opérationnel !

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	