

## TP4 POO – Création d'un jeu de combat 4/4

### *TP4 POO - CREATION D'UN JEU DE COMBAT 4/4*

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.	

## TP4 POO – Création d'un jeu de combat 4/4

### 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 quatrième et dernier TP, nous allons ajouter une spécification de Personnage à l'aide des notions d'héritage. Nous créerons 2 classes héritées de Personnage, une classe Guerrier, et une classe Magicien.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Benoît Avenel</u>		11/2022	
		<i>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.</i>	

## TP4 POO – Création d'un jeu de combat 4/4

### Cahier des charges

Nous allons améliorer notre jeu de combat, pour ajouter la possibilité de sélectionner des classes de personnages, qui auront des avantages différents.

Il ne sera plus possible de créer un simple personnage, mais obligatoirement un personnage possédant une classe spécifique :

- Magicien : Il possèdera une nouvelle fonctionnalité « lancer un sort », qui permettra d'endormir un personnage pendant \$atout \* 6 heures (\$atout représentant la dose de magie du personnage)
- Guerrier : Lorsqu'un coup lui est porté, il pourra parer le coup en fonction de sa protection (atout)
- 

*NB : Ceci n'est qu'une petite liste de départ. Libre à vous de créer d'autres personnages. ;)*

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.	

## TP4 POO – Création d'un jeu de combat 4/4

### Nouvelles fonctionnalités à implémenter

Étant donné qu'un magicien peut endormir un personnage, il est nécessaire d'implémenter deux nouvelles fonctionnalités :

Celle consistant à savoir si un personnage est endormi ou non (nécessaire lorsque ledit personnage voudra en frapper un autre : s'il est endormi, ça ne doit pas être possible).

Celle consistant à obtenir la date du réveil du personnage sous la forme « XX heures, YY minutes et ZZ secondes », qui s'affichera dans le cadre d'information du personnage s'il est endormi.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Benoît Avenel</u></b>		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.	

## TP4 POO – Création d'un jeu de combat 4/4

Etape préliminaire : Mise à jour de la BDD

Pour pouvoir implémenter nos nouvelles classes de personnages, et leurs spécificités, nous allons devoir modifier notre table créée précédemment.

- Créez une nouvelle table « Personnages\_v2 » ayant pour attributs
  - Id – int – PK auto increment
  - Nom – Varchar
  - Degats – int
  - timeEndormi – int
  - Type – Varchar
  - Atout – int

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Benoît Avenel</u></b>		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.	

## TP4 POO – Création d'un jeu de combat 4/4

### Mise en route

Pour simplifier le travail nécessaire à la mise en place de ces nouvelles fonctionnalités, une partie avancée du code est disponible sur le Github suivant :

<https://github.com/benoitavenel/sourcesJeuCombat.git>

Les classe Personnage et PersonnagesManagers ont été actualisées pour répondre à notre nouvelle version.

La page d'index a également été mise à jour avec une première partie des modifications nécessaires.

**CEPENDANT**, certaines modifications devront être ajoutées à cette page une fois l'ajout des classes Magicien et Guerrier effectuée.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Benoît Avenel</u></b>		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.	

## TP4 POO – Création d'un jeu de combat 4/4

### Explications de code

- Hydratation des attributs d'une classe

Dans un but de simplification de mise en place des différents constructeurs, une méthode d'hydratation des données a été mise en place. Cette méthode permet d'instancier un objet sans que l'ensemble de ses attributs soient passés en paramètre. Cette façon de coder, non nécessaire, permet d'améliorer la propreté et la rapidité de codage des classes.

- Récupération de la classe appelante

Dans notre classe Personnage, nous avons besoin d'instancier le type de notre personnage.

2 possibilités s'offrent alors à nous :

- Passer le type du personnage en attribut lors de l'appel au constructeur, ce qui implique de rajouter un attribut lors de chaque appel.
- Utiliser la fonction `get_class()`, qui permet de récupérer le nom de la classe hérité faisant appel au constructeur de la classe parent.

Par raison de propreté, c'est cette dernière qui a été utilisée ici.

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.	

## TP4 POO – Création d'un jeu de combat 4/4

### 1) Mise en place de la classe Guerrier

La classe guerrier fonctionne de manière analogue à la classe Personnage.

Cette dernière héritera donc de l'ensemble des attributs et méthodes de Personnage.

- Mettez en place la structure de la classe Guerrier, héritant de la classe Personnage

Cependant, nous allons devoir spécifier la méthode recevoirDegats.

En effet, la classe guerrier recevra des dégâts en fonctions de ses atouts, selon la logique suivante :

Si les dégâts du personnage sont compris entre 0 et 25, sa capacité d'atout sera de 4.

Si les dégâts du personnage sont compris entre 25 et 50, sa capacité d'atout sera de 3.

Si les dégâts du personnage sont compris entre 50 et 75, sa capacité d'atout sera de 2.

Si les dégâts du personnage sont compris entre 75 et 90, sa capacité d'atout sera de 1.

Si les dégâts du personnage sont compris entre 90 et 100, sa capacité d'atout sera de 0.

Les nouveaux dégâts subis par le personnage seront égaux aux dégâts reçus – sa capacité d'atout.

Ses dégâts reçus seront donc compris entre 1 et 5, en fonction de sa capacité d'atout.

En fonction des dégâts finaux, la méthode retournera enfin si le personnage a été tué ou a uniquement subi des dégâts.

- Mettez en place la fonction recevoirDegats(), appliquant les dégâts finaux au guerrier, et retournant son état ( blessé ou mort )

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Benoît Avenel</u></b>		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.	



## TP4 POO – Création d'un jeu de combat 4/4

### 2) Mise en place de la classe Magicien

La classe Magicien fonctionne de manière analogue à la classe Personnage.

Cette dernière héritera donc de l'ensemble des attributs et méthodes de Personnage.

- Mettez en place la structure de la classe Magicien, héritant de la classe Personnage

Cependant, nous allons devoir spécifier la méthode lancerUnSort().

En effet, la classe Magicien pourra lancer un sort selon la logique suivante :

Si les dégats du personnage sont compris entre 0 et 25, sa capacité d'atout sera de 4.

Si les dégats du personnage sont compris entre 25 et 50, sa capacité d'atout sera de 3.

Si les dégats du personnage sont compris entre 50 et 75, sa capacité d'atout sera de 2.

Si les dégats du personnage sont compris entre 75 et 90, sa capacité d'atout sera de 1.

Si les dégats du personnage sont compris entre 90 et 100, sa capacité d'atout sera de 0.

Le magicien ne pourra pas s'ensorceler lui-même, lancer de sort si sa magie est vide, ou s'il est endormi.

Si toutes ses conditions sont vérifiées, il pourra ensorceler le Personnage pour une durée de 21600\*atout (21600 représentant 6 heures en secondes)

- Mettez en place la fonction lancerUnSort(), permettant au magicien de lancer son sortilège

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Benoît Avenel</u></b>		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.	

## TP4 POO – Création d'un jeu de combat 4/4

### 3) Utilisation des classes

A ce stade, votre application est presque finalisée. L'ensemble de vos classes Personnage, Guerrier et Magicien sont opérationnelles, ainsi que votre manager de classe.

Cependant, votre application n'est toujours pas utilisable depuis votre index.php.

- Modifiez votre index pour y ajouter l'inclusion de vos classes Guerrier et Magicien, AINSI QUE l'instanciation d'objets lors de la soumissions du formulaire

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Benoît Avenel</u></b>		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.	