

Une erreur est survenue lors du chargement de la version complète de ce site. Veuillez vider le cache de votre navigateur et rafraîchir cette page pour corriger cette erreur.

## Fonctionnement général du système de script

**Galathil**

**Important :** Ce tutoriel a été rédigé par Nobody pour le site wow-emu.fr. Je le recopie ici à l'identique, Il se peut que certaines informations soient obsolètes à ce jour. Bonne lecture !

# Fonctionnement général du système de script

**Prélude :** Ce sujet n'est pas original, il a été transféré d'un précédent forum d'émulation en date d'Octobre 2013. J'en suis l'auteur, je le transfère donc pour les besoins de ce forum. Bien qu'il soit en partie ré-écrit, il se peut que certains termes soient inadaptés ou ne soient plus d'actualité.

**Ce tutoriel n'a pas été directement écrit par moi-même, mais par un des membres de l'équipe Mist-Eria. ce sujet est posté avec l'accord de son auteur. Des modifications peuvent y être apportées dans un but de clarification.**

Bonjour à tous !

Aujourd'hui nous allons voir ensemble comment fonctionne le système de script de la plupart des émulateurs WoW. Vous allez voir que le principe est relativement simple à comprendre et plutôt pas mal fichu !

## 1) Le principe général : un évènement déclenche une action

Tous les scripts du serveur reposent sur le principe de la programmation événementiel. Derrière ce mot un peu barbare se cache un principe tout simple : un évènement déclenche une action. Cela signifie que l'exécution du script est entièrement contrôlée par le serveur. Dès qu'un évènement se produit, le serveur appelle la fonction correspondante dans le script qui va être chargé d'ajuster le comportement du boss / spell / item / commande. Ce système de fonctionnement est particulièrement bien adapté car il nous permet d'obtenir plein d'info sur l'évènement et d'ajuster très finement l'action déclenchée.

Exemple: Voici comment on pourrait implémenter un boss qui renvoie la moitié des dégâts qu'il a subi au groupe entier.

```
//Fonction qui sera appelée automatiquement par le serveur
//dès que le boss recevra des dégâts
fonction DamageTaken(Player attacker, int damage)
{
    int degatPourJoueurs = damage / 2;

    Pour chaque membre de attacker->Group
    {
        //On inflige des dégâts au joueur
        //en précisant que c'est le boss qui les inflige
        membre.DealDamage(boss, degatPourJoueurs);
    }
}
```

## 2) Jusqu'où un tel système peut nous permettre d'aller ?

Le système de script nous permet de personnaliser complètement le serveur. Je vais détailler les scripts de boss puis je vous donnerai un aperçu de tout ce que l'on peut scripter.

Voici les différents évènements que l'on peut personnaliser pour un boss :

- L'aggro : Cela permet de, par exemple, dire une phrase quand le boss entre en combat.
- La réinitialisation : Permet de réinitialiser le script du boss pour un nouveau combat (remise à zéro des timers, despawn des adds, etc.)

La reinitialisation permet de réinitialiser le script du boss pour un nouveau combat (remise à zéro des timers, des spawn des adds, etc.).

Cet évènement est également déclenché au spawn du boss

- Le boss n'a plus de cible : Cela se produit quand tous les membres du raid sont mort, ou que le groupe a fui. Cet évènement renvoi le boss à sa position initiale. Cela permet de dire une dernière phrase pour narguer le groupe, par exemple.
- Le boss tue une cible : Cet évènement est déclenché quand le boss tue une cible. On s'en sert pour lui faire dire une phrase ou maintenir à jour un compteur, entre autre.
- Tick d'horloge : Cet évènement alerte le script qu'un nouveau tick d'horloge c'est produit. On peut ainsi tenir à jour des timers qui permettent de d'exécuter une action toutes les X secondes. Cette évènement est appelé plusieurs fois par secondes et c'est le cœur de tout script de boss.
- Le boss vient de mourir : Le boss a été tué par un joueur. Cette fonction permet par exemple de signaler à l'instance qu'il faut ouvrir les grilles de la salle suivante ou qu'il faut spawn un coffre de butin.

Avec ces quelques évènements on peut scripter un boss de manière à ce qu'il représente un véritable défi pour les joueurs. Noter que tous les mobs du jeu ont un script associé par défaut, qui ne leur fait rien faire du tout : ce sont boss "poteaux". Au niveau code, on crée un script de boss en dérivant une classe d'AI de base et en surchargeant les fonctions virtuelles correspondantes à chaque évènement. Dans un autre tutoriel, nous étudierons en détail le script d'un boss fun appelé l'Epouvantard<sup>1</sup>.

Pour terminer ce tutoriel, nous allons voir d'autre utilisation des script que pour les boss.

- Les scripts d'instances : Ils permettent de contrôler le déroulement global d'une instance. Par exemple quand un boss meurt, on ouvre les portes vers le suivant. On peut s'en servir aussi pour déclencher des évènements aléatoires comme une musique ou une explosion. Bref ils vous permettent d'avoir un contrôle total sur votre zone instanciée.
- Les scripts de sort : Ils permettent de définir des comportements particuliers à des sorts. Par exemple on s'en sert pour un sort du type : Soigne d'un montant égal 5% des dégâts subis durant les 5 dernières secondes.
- Les scripts d'item : Ils permettent de personnaliser un item. On peut ainsi créer des pierres de téléportation par exemple.
- Les scripts de joueurs : On peut les utiliser pour scripter des spells très complexes. Je pense notamment à la Plaie de tumulte du démoniste ([Tumulte](#)). Mais ce n'est pas tout. On peut aussi s'en servir pour des systèmes de surveillances très poussées, comme l'affichage en direct aux MJs de ce qu'un joueur dit.
- Les scripts de commande : On les utilise pour créer de nouvelles commandes en jeu (par exemple quand on implante un nouveau système comme le marché noir)

Ce n'est pas tout il y en a pleins d'autres que je ne détaillerai pas car ils ne sont quasiment jamais utilisés. Cependant il faut savoir qu'ils existent.

- Script de groupe
- Script de guild
- Script d'hôtel des ventes
- Script serveur
- Script de gameobject
- Script de champs de bataille
- Script de météo (si si, vous avez bien lu)
- Script de véhicule
- Script de condition
- Script de transport (zeppelin, etc...)
- Script de haut-fait
- Script de zone
- Script de formule
- Script d'area trigger

Voilà, j'espère que ce tutoriel vous a plu et vous a été utile. Je reste à votre disposition si vous avez des questions, des critiques ou des suggestions.

Vous pouvez tester l'Epouvantard sur Mist-Eria. À l'heure où j'écris ces lignes il n'a encore jamais été vaincu. N'hésitez pas à tenter votre chance:

