

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.

## Réaliser des Gossips en LUA

kazuma

# Les Gossips

## Tutoriel par Kazuma

### 1 - Introduction

Salut à tous, aujourd'hui je vous met à disposition un tutoriel sur les gossip , vous verrez au final que c'est très simple et accessible à tous !

#### Tout d'abord qu'est ce que c'est un "Gossip" ?

Un gossip ("potin" en français) , c'est le menu qui apparaît quand vous vous adressez à un PNJ , cette fenêtre qui apparaît à gauche de votre écran avec une image de parchemin : c'est un gossip !

Grâce à ce que je vais vous montrer, vous allez avoir la possibilité de faire apparaître un gossip, d'y inscrire ce que vous voulez, vous pourrez y mettre en place des choix qui pourront faire apparaître des "sous-menu" mais aussi effectuer n'importe quelle traitement que vous voudrez ! Vous pourrez téléporter le joueur ou bien encore lui voler tout son argent , à vous de voir !

### 2 - Comment ça marche ?

Dans notre exemple on va lier l'utile à l'agréable , je vais vous présenter comment créer un téléporteur !

Dans notre script nous auront besoin de créer *deux* fonctions :

- **OnGossipHello** : *Fréquemment appelé de cette manière à quelques détails près, cette fonction devra s'exécuter au moment où on s'adressera au Téléporteur.*
- **OnGossipSelect** : *Fréquemment appelé de cette manière à quelques détails près, cette fonction devra s'exécuter au moment où sélectionne un choix dans le gossip du Téléporteur.*

### 3 - OnGossipHello

#### A - Première partie et analyse

Tout de suite un premier aperçu du début de notre script :

```
1  local NPCId = 16128
2  local MenuId = 600 -- Nécessaire pour les joueurs
3
4
5  -- GESTION GOSSIP
6
7  local function OnGossipHello(event, player, object)
8
9  end
10
11 local function OnGossipSelect(event, player, object, sender, intid, code, menuid)
12
13 end
14
```

```
local NPCId = 16128
```

Ici on crée une variable locale qui contiendra le numéro / l'ID (Entry) correspondant au PNJ qui fera office de téléporteur dans notre exemple. Dorénavant, il nous suffira d'écrire "NPCId" dans le script au lieu de 16128.

```
local MenuId = 600 -- Nécessaire pour les joueurs
```

Ici on crée une variable locale qui contiendra un numéro que l'on peut choisir au hasard, il est important dans la situation où l'unité qui

envoie le gossip et l'unité qui reçoit le gossip est la même personne : le joueur. Même si on ne fait pas ça souvent, je vous recommande tout de même d'avoir un MenuId.

```
local function OnGossipHello(event, player, object)

end

local function OnGossipSelect(event, player, object, sender, intid, code, menuid)

end
```

Enfin, voici les deux fonctions du script dont j'ai déjà parlé et expliqué. On remarque que ce sont des fonctions locales : c'est-à-dire que ces fonctions ne sont accessibles que dans CE script et ne pourra interférer avec vos autres scripts.

**Il est extrêmement important d'utiliser le plus possible les variables locales pour ne pas faire bosser votre serveur plus qu'il ne le devrait !**

## B - Deuxième partie et analyse

Tout de suite un deuxième aperçu de notre script :

```
5  -- GESTION GOSSIP
6
7  local function OnGossipHello(event, player, object)
8      player:GossipClearMenu() -- Met à vide l'éventuel gossip déjà ouvert.
9      player:GossipSetText("Salut Open-Wow, ceci est un texte mis en place grâce à une fonction custom,\nAinsi, n'importe quel texte peut-être inscrit ici.")
10     player:GossipMenuAddItem(0, "Accéder au sous-menu", 1, 1) -- sous-menu
11     player:GossipSendMenu(0x7FFFFFFF, object, MenuId) -- MenuId nécessaire pour les joueurs
12 end
13
14 local function OnGossipSelect(event, player, object, sender, intid, code, menuid)
15     if (intid == 1) then
16         player:GossipClearMenu() -- Met à vide l'éventuel gossip déjà ouvert.
17         player:GossipSetText("Re-bonjour Open-wow, vous êtes arrivé dans le sous-menu.")
18     end
19 end
20 end
21
22
```

Dans la fonction OnGossipHello :

```
player:GossipClearMenu() -- Met à vide l'éventuel gossip déjà ouvert.
```

Ici c'est ligne est importante car elle remet à zéro un éventuel gossip déjà ouvert, c'est indispensable si vous voulez donner l'illusion qu'un choix fasse accéder le joueur à un "sous-menu".

**Dans le cas contraire, les choix dans le gossip se succéderont tous les uns après les autres.**

```
player:GossipSetText("Salut Open-Wow, ceci est un texte mis en place grâce à une fonction custom,\nAinsi, n'importe quel texte peut-être inscrit ici.")
```

GossipSetText est une nouvelle fonction qui n'est pas fournis dans le LuaEngine de base, il faut aller la chercher sur le Github de Eluna :

[Fichier GossipTextExtension](#)

Vous devrez créer un fichier lua , y copier-coller le script que vous trouverez sur Github et la mettre dans votre dossier lua\_scripts pour que vous puissiez vous servir de GossipSetText .

Cette fonction comme son nom l'indique permet de pouvoir saisir le texte que vous voulez en haut du gossip. *Plus besoin de passer par la DB.*

```
player:GossipMenuAddItem(0, "Accéder au sous-menu", 1, 1) -- sous-menu
```

Cette fonction est celle qui constitue en grande partie le gossip, cette fonction représente un choix dans le gossip.

- **0** : Il correspond à l'icone de la ligne, il en existe plusieurs, je vous laisse les découvrir, vous pouvez très bien y mettre 1, 2 ou 3.
- **"Accéder au sous-menu"** : Tout bêtement ceci est le texte qui apparaît sur la ligne du choix, libre à vous d'y mettre ce que vous voulez.
- **1** : Laissez ce numéro comme il est, il n'est pas si important, mais vous pouvez toujours regarder sur le doc Eluna si vous avez soif de connaissance.
- **1 (intid)** : On appelle ce numéro le intid , il est déterminant , c'est grâce à ce numéro que ensuite vous pourrez expliquer quel traitement effectué en fonction du choix du jour. Par exemple : pour la ligne à l'intid numéro 1 vous faites le traitement A, pour la ligne à l'intid numéro 2 vous faites le traitement B.

Pour chaque choix que vous voudrez mettre dans le gossip, vous devrez rajouter cette fonction, exemple :

```

player:GossipMenuAddItem(0, "Téléportation vers Dalaran", 1, 1) -- Choix Dalaran
player:GossipMenuAddItem(0, "Téléportation vers Hurlevent", 1, 2) -- Choix Hurlevent (intid différent)

```

**Attention à toujours veiller à ce que le intid de chaque GossipMenuAddItem soit différent, ils doivent être unique sinon il y aura des conflits !**

```

player:GossipSendMenu(0x7FFFFFFF, object, MenuId) -- MenuId nécessaire pour les joueurs

```

Enfin, cette dernière ligne permet de faire apparaître aux yeux du joueur le gossip et ses choix, sans cela le gossip n'apparaîtra pas.

- **0x7FFFFFFF** : Ceci est spécifiquement dû à l'utilisation de la fonction `GossipSetText` cela indique que vous voulez utiliser le texte de `GossipSetText`, dans le cas où vous n'utilisez pas `GossipSetText` il vous suffit de saisir le numéro d'un texte présent en DB de la table `npc_text`.
- **object** : Ceci est le sender, c'est-à-dire l'unité qui envoie ce gossip, vous pouvez laisser "object" mais dans l'éventuel cas où c'est un joueur qui envoie ce gossip vous devez absolument renseigner le `MenuId` en troisième paramètre sinon cela ne fonctionnera pas.
- **MenuId** : Ceci est indispensable dans le cas où l'envoyeur du gossip est un joueur, autrement vous pouvez vous en passer, cela dit le laisser ne dérange pas non plus...

## 4 - OnGossipSelect

Maintenant on va s'attaquer à la deuxième fonction "OnGossipSelect", cette fois-ci on va mettre en place les traitements à faire en fonction des choix dans le gossip.

```

14 local function OnGossipSelect(event, player, object, sender, intid, code, menuid)
15     if (intid == 1) then
16         player:GossipClearMenu() -- Met à vide l'éventuel gossip déjà ouvert.
17         player:GossipSetText("Re-bonjour Open-wow, vous êtes arrivé dans le sous-menu.")
18         player:GossipSendMenu(0x7FFFFFFF, object, MenuId) -- MenuId nécessaire pour les joueurs
19     elseif (intid == 2) then
20         player:Teleport(0, -9065.594727, 433.876373, 93.056351, 3.918453) -- Téléportation Hurlevent
21         player:GossipComplete() -- Ferme le gossip
22     end
23 end
24 end

```

Le code a été commenté de manière à être suffisamment explicite pour ceux ayant un minimum de connaissance. Maintenant on va voir plus en détail qu'est-ce qu'il se passe exactement.

Comme je l'ai dit le "intid" est fonction `GossipMenuAddItem` est déterminant car c'est grâce à ça que tout fonctionne. Ainsi, on utilise la condition IF ("SI" en français).

Le bloc d'instruction IF permet de traiter une multitude de cas, on vérifie la valeur de l'intid et en fonction du numéro, si il y a un traitement à effectué, il est effectué.

```

if (intid == 1) then
    player:GossipClearMenu() -- Met à vide l'éventuel gossip déjà ouvert.
    player:GossipSetText("Re-bonjour Open-wow, vous êtes arrivé dans le sous-menu.")
    player:GossipSendMenu(0x7FFFFFFF, object, MenuId) -- MenuId nécessaire pour les joueurs

```

Voilà la traduction en français :

- Si le choix 1 a été sélectionné.
- On efface ce que affiche actuellement le gossip.
- On décide que le texte à afficher sera "e-bonjour Open-wow, vous êtes arrivé dans le sous-menu."
- On fait apparaître aux yeux du joueur le nouveaux texte (et éventuel choix si il y en a...).

```

elseif (intid == 2) then
    player:Teleport(0, -9065.594727, 433.876373, 93.056351, 3.918453) -- Téléportation Hurlevent
    player:GossipComplete() -- Ferme le gossip
end
end

```

Là, on prolonge la structure du bloc d'instruction IF grâce au mot clé "elseif" ("sinon si" traduit en français).

La, on prolonge la structure du bloc d'instruction n° grâce au mot-clé `elseif` (sinon si traduit en français).

Ainsi, dans le cas où le choix numéro 1 n'as pas été sélectionné on vérifie si le choix 2 lui n'a pas été sélectionné. Si il a été sélectionné alors on fais tel traitement, en l'occurrence ici c'est une téléportation du joueur vers Hurlevent.

- **GossipComplete** : Cette fonction permet de fermer le gossip, la fenêtre est fermé puisqu'elle ne nous est d'aucune utilité désormais.

## 5 - Déclencher les fonctions

Eh oui, ce n'est pas encore terminé, car jusqu'à maintenant les deux fonctions ne sont jamais utilisé, nous devons les faire se déclencher aux moments propice !

**OnGossipHello**, doit se déclencher quand on parle au PNJ souhaité.

**OnGossipSelect** doit se déclencher quand on interagit avec le gossip envoyer par le PNJ en question.

```
27 RegisterCreatureGossipEvent(NpcId, 1, OnGossipHello)
28 RegisterCreatureGossipEvent(NpcId, 2, OnGossipSelect)
```

Il faut utiliser des fonctions Register pour "Enregistrer" une fonction à un événement (aussi appelé "**Hook**" ou "**event**" en anglais). La fonction *register* qui nous intéressent est :

**RegisterCreatureGossipEvent** car on interagit avec une créature (NPC/PNJ).

```
RegisterCreatureGossipEvent(NpcId, 1, OnGossipHello)
RegisterCreatureGossipEvent(NpcId, 2, OnGossipSelect)
```

- **NpcId** : Ici nous devons renseigné l'entry (id) du NPC qui aura le Gossip.
- **1** : Ici nous devons renseigné le numéro de l'événement qui déclenchera la fonction, le 1 correspond au moment où on parle au NPC. Voir plus
- **OnGossipHello**: Ici nous devons renseigné le nom de la fonction à appelé.

Désormais vous n'avez plus qu'à place le script LUA dans votre dossier lua\_scripts et aller parler au PNJ avec l'id 16128 (Rhonin, à Dalaran).

En espérant que ce tutoriel puisse en aider plus d'un notamment les plus grand débutants, bonne chance dans vos projets de serveur World of Warcraft !