

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.

## Gossip NPC

iThorgrim

# Gossip NPC

## Tutoriel par @Killit5

Dans ce tutoriel nous allons voir comment créer simplement un GOSSIP NPC en C++.

## 1 - La bibliothèque

```
#include "ScriptMgr.h"
#include "ScriptedGossip.h"
```

Il ne faut pas oublier la bibliothèque ScriptedGossip.h, disponible dans les sources,

```
src\server\game\AI\ScriptedAI\
```

On y retrouve diverses fonctions utiles :

- AddGossipItemFor
- SendGossipMenuFor
- CloseGossipMenuFor

## 2 - La création

On commence par créer la structure de notre script (class,type,nom, etc...).

```
class GossipTutroial : public CreatureScript
{
public:
    GossipTutroial() : CreatureScript("GossipTutroial") { }

    struct MyAI : public ScriptedAI
    {
        MyAI(Creature* m_creature) : ScriptedAI(m_creature) { }

        bool GossipHello(Player* player) override
        {
            return OnGossipHello(player, me);
        }

        bool GossipSelect(Player* player, uint32 menuId, uint32 gossipListId) override
        {
            uint32 action = player->PlayerTalkClass->GetGossipOptionAction(gossipListId);
            return OnGossipSelect(player, me, action);
        }
    };

    CreatureAI* GetAI(Creature* creature) const override
```

```

{
    return new MyAI(creature);
}
};

```

On crée ensuite notre première fonction, celle que l'on verra au moment de parler au pnj.

Nous allons utiliser les deux fonctions suivantes, que nous pouvons retrouver dans le fichier `ScriptedGossip.h` :

```

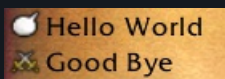
AddGossipItemFor(Player* player, uint32 icon, std::string const&& text, uint32 sender, uint32 action)
SendGossipMenuFor(Player* player, uint32 npcTextID, ObjectGuid const&& guid)
static bool OnGossipHello(Player * player, Creature * creature)
{
    AddGossipItemFor(player,GOSSIP_ICON_CHAT, "Hello World", GOSSIP_SENDER_MAIN, 1);
    AddGossipItemFor(player,GOSSIP_ICON_BATTLE, "Good Bye", GOSSIP_SENDER_MAIN, 2);
    SendGossipMenuFor(player, player->GetGossipTextId(creature), creature->GetGUID());
    return true;
}

```

L'appel de la fonction `AddGossipItemFor` va nous permettre d'afficher un texte qui exécutera une action lorsque l'on cliquera dessus. Il faut penser à changer à chaque fois le numéro de l'action, pour que l'exécution soit différente ici 1 et 2.

Ne surtout pas oublier d'utiliser la fonction `SendGossipMenuFor` , sinon le pnj ne va rien afficher.

En somme, voilà le résultat sur le pnj :



```

static bool OnGossipSelect(Player* player, Creature* creature , uint32 Action)
{
    switch (Action)
    {
        case 1:
            creature->TextEmote("Bienvenue sur Open-WOW", player);
            CloseGossipMenuFor(player);
            break;
        case 2:
            CloseGossipMenuFor(player);
            break;
    }
    return true;
}

```

Si nous choisissons l'action 1, nous demandons au pnj de lancer une émote.

**Bienvenue sur Open-WOW**

Si nous choisissons l'action 2, le gossip se ferme.

L'appel de la fonction `CloseGossipMenuFor` permet de fermer le gossip.

Il nous manque juste la dernière ligne pour appeler le script.

```

void AddSC_GossipTutroial()
{
    new GossipTutroial();
}

```

Il nous manque plus que le PNJ à créer :

```

SET

```

```

@Entry = 50000,
@Name = "GossipTutroial",
@ScriptName = "GossipTutroial";

INSERT INTO `creature_template` (`entry`, `modelid1`, `modelid2`, `name`, `subname`, `IconName`, `gossip_menu_id`, `minlevel`,
(@Entry, 21665, 0, @Name, '', NULL, 0, 80, 80, 2, 35, 1, 1, 0, 0, 2000, 0, 1, 0, 7, 138936390, 0, 0, 0, '', 0, 1, 0, 0, 1,

```

Si tous se passe bien le script final ressemble à ça :

```

#include "ScriptMgr.h"
#include "ScriptedGossip.h"

class GossipTutroial : public CreatureScript
{
public:
    GossipTutroial() : CreatureScript("GossipTutroial") { }
    static bool OnGossipHello(Player * player, Creature * creature)
    {
        AddGossipItemFor(player,GOSSIP_ICON_CHAT, "Hello World", GOSSIP_SENDER_MAIN, 1);
        AddGossipItemFor(player,GOSSIP_ICON_BATTLE, "Good Bye", GOSSIP_SENDER_MAIN, 2);
        SendGossipMenuFor(player, player->GetGossipTextId(creature), creature->GetGUID());
        return true;
    }

    static bool OnGossipSelect(Player* player, Creature* creature , uint32 Action)
    {
        switch (Action)
        {
            case 1:
                creature->TextEmote("Bienvenue sur Open-WOW", player);
                CloseGossipMenuFor(player);
                break;
            case 2:
                CloseGossipMenuFor(player);
                break;
        }
        return true;
    }

    struct MyAI : public ScriptedAI
    {
        MyAI(Creature* m_creature) : ScriptedAI(m_creature) { }
        bool GossipHello(Player* player) override
        {
            return OnGossipHello(player, me);
        }
        bool GossipSelect(Player* player, uint32 menuId, uint32 gossipListId) override
        {
            uint32 action = player->PlayerTalkClass->GetGossipOptionAction(gossipListId);
            return OnGossipSelect(player, me, action);
        }
    };

    CreatureAI* GetAI(Creature* creature) const override
    {
        return new MyAI(creature);
    }
};

```

```
void AddSC_GossipTutorial()  
{  
    new GossipTutorial();  
}
```

Enjoy !

---