SpellScript : le sort "Eclat lunaire"

iThorgrim

SpellScript : le sort "Eclat lunaire"

Tutoriel par @Galathil

Ceci étant mes premiers pas dans les SpellScript, il se peut que le code ne soit pas totalement correct et les explications imprécises.

Lecteurs avertis, soyez indulgents et préférez m'expliquer en privé pour rendre ce tuto plus clair par la suite.

Développement d'un SpellScript : Eclat Lunaire

Sommaire :

Introduction

Ce tutoriel vise à vous donner quelques bonnes pratiques afin de réaliser vos SpellScript au travers d'un exemple.

La démarche de réalisation est toute aussi importante que le résultat final, soyez attentifs.

La version de TrinityCore utilisée est la suivante : **TrinityCore 6.2.3**

Symptômes

Ceci est un rapport que vous pouvez recevoir d'un joueur :

```
Bonjour,

Le sort "Eclat Lunaire" de mon druide ne fonctionne pas. Je choppe bien l'aggro de la créature mais le dot n'est pas appli

Bien cordialement,
un joueur
```

La recherche d'informations

A première vue, il semble que le sort ne soit pas traité par le serveur ou partiellement ! Nous allons repérer l'ID du sort concerné.Pour cela deux méthodes :

- **wowhead** : On recherche les sorts du druide pour isoler "Eclat Lunaire", l'id se trouve dans l'url
- **Commande lookup** : Sur un personnage druide avec le sort de connu, grâce à la commande ".lookup spell Eclat lunaire". L'un des ids portera la notion "know", c'est celui que l'on recherche.

Dans les deux cas, nous tombons sur un premier id : <mark>8921</mark>.

Pour savoir si le sort est déjà traité par le core (ou tout au moins une portion) faites une recherche dans les sources avec les mots clefs "moonfire" ou "<mark>8921</mark>". De même, recherchez dans la table **world.spell_script_names**.

Vous ne trouverez rien de concluant.

Pour avoir un peu plus d'informations sur le sort, nous allons utiliser **SpellWork **proposé par TrinityCore ici : **SpellWork** .

Il vous suffit de lancer la solution "SpellWork.sln" et de la générer (Nécessite Visual Studio 2015 si vous êtes sous Windows).

L'exécutable sera positionné dans "bin/debug".
Au démarrage, sélectionnez le dossier de vos "DBC/votre_langue".

En effectuant une recherche sur l'id de sort <mark>8921</mark>, nous obtenons un tas d'informations dans la colonne de gauche.Vous pouvez observer que le sort à un effet DUMMY, c'est à dire qu'il ne fait rien du tout lui même (en bas du cadre).

En regardant de plus prêt la description du sort, on peut voir un autre ID apparaitre : <mark>164812</mark> qui semble être le bon sort !

Comment être sûr de cela?

En jeu, sélectionnez un monstre et tapez la commande ".cast id_du_sort" pour tester les deux ids. Observez comme 164812 semble faire le job, c'est tout bon !

Nous allons maintenant créer notre SpellScript !

Le script

Script à ajouter dans "\src\server\scripts\Spells\spell_druid.cpp" :

```
// 8921 - Moonfire (Dummy)
class spell_dru_moonfire : public SpellScriptLoader
{
public:
spell_dru_moonfire() : SpellScriptLoader("spell_dru_moonfire") { }

class spell_dru_moonfire_SpellScript : public SpellScript
{
PrepareSpellScript(spell_dru_moonfire_SpellScript);

enum
{
SPELL_DRUID_MOONFIRE_DAMAGE = 164812
};

bool Load() override
{
if (GetCaster() && GetCaster()->GetTypeId() != TYPEID_PLAYER)
return false;

return true;
}

void HandleOnHit()
{
Unit* caster = GetCaster();
Unit* target = GetHitUnit();

if (!caster || !target)
return;
```

```
caster->CastSpell(target, SPELL_DRUID_MOONFIRE_DAMAGE, true);
}

void Register() override
{
OnHit += SpellHitFn(spell_dru_moonfire_SpellScript::HandleOnHit);
}
};

SpellScript* GetSpellScript() const override
{
return new spell_dru_moonfire_SpellScript();
}
};
```

N'oubliez pas d'ajouter l'initialisation tout en bas du fichier (pour que le core charge bien le script !)

```
void AddSC_druid_spell_scripts()
{
(...)
new spell_dru_moonfire();
(...)
}
```

Enfin, pour que le core puisse lier votre script au sort <mark>8921</mark>, ajoutez une ligne dans la table **world.spell_script_names** :

```
REPLACE INTO `spell_script_names` (`spell_id`, `ScriptName`) VALUES (8921, 'spell_dru_moonfire');
```

Vous ne comprenez rien? Pas de soucis, passons à l'explication.

Explications

Dans cette partie, nous allons analyser le code composé précédemment :

- **La structure globale** : Comme vous le voyez, nous avons une première classe héritant de SpellScriptLoader et une classe interne héritant de SpellScript. Cette architecture au premier abord déroutante vise à simplifier les choses dans le cas ou il faudrait ajouter un AuraScript associé au SpellScript. Dans ce cas il suffira de créer une seconde classe interne.
- **Aura ou SpellScript ?** Un SpellScript est un script de sort simple sans effet de durée (dot ou buff), contrairement à un AuraScript qui donne accès à des méthodes pour gérer les sors sur la durée.
- **Le constructeur** : construction de l'objet parent en indiquant le nom de la classe enfant.
- **L'enum** : De manière générale, quand on manipule des ids, il est commun d'utiliser des ENUM, cela rend le code plus lisible et si l'id change, vous ne le changez qu'a un seul endroit dans le code !
- **Méthode load()** : doit retourner un booléen pour déterminer si toutes les conditions requises sont réunies pour exécuter le script. Ici, nous vérifions que le lanceur de sort est bien un joueur.
- **Méthode HandleOnHit()** : le nom de la méthode importe peu, vous verrez pourquoi plus bas. On va tout simplement lancer le bon sort (164812) sur la victime qui est la cible sélectionnée. Notez que l'on vérifie l'existence des pointeurs, c'est plus propre et évite les exceptions et donc un crash potentiel du serveur.
- **Méthode register()** : comme son nom l'indique, elle enregistre la pile d'évènements à lancer. On demande donc d'ajouter la méthode précédemment créé à la pile d'appel "onHit".
- **Méthode GetSpellScript()** : qui est simplement le point d'appel pour instancier notre classe interne. Nécessaire pour le SpellScriptLoader.

- On ajoute l'instanciation de la classe tout en bas de "spell_druid.cpp", sans quoi votre code ne sera jamais chargé !
- On ajoute une ligne dans la table **world.spell_script_names** pour indiquer au serveur de lancer notre script quand le sort "Eclat lunaire" est appelé.
- Compilez, testez et admirez le résultat !

Remerciements

Je remercie chaleureusement @Traesh pour son aide et ses conseils qui m'ont permis de développer ce script et vous le proposer ici.

Aide, idées, suggestions, critiques

Si vous avez des soucis avec le script, merci de créer une demande d'aide dans la section appropriée, en indiquant que vous avez suivi ce tutoriel.

Pour les idées, suggestions et critiques, préférez me contacter par MP? j'éditerai le tutoriel en conséquence.

Encore une fois, ce sont mes premiers pas, soyez indulgents !