

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.

Script de boss [Facile]

EIntemporel

Bonjour, aujourd'hui je vais vous expliquer comment très facilement on peut faire un petit script de boss avec quelque sorts, quelques évènements, et même l'utilisation d'un objet.

Réfléchir au script

Ici c'est pas compliqué on va écrire notre script à la main, un peu comme si on faisait le guild de stratégie du combat comme sur Retail, alors les différents sorts utilisés, et les différentes phases, etc... ça nous permet de pas faire du n'importe quoi et de rester focus.

Script d'un mini boss de niveau 15

Nom : Stone

Faction : Défias (17)

Niveau : 15

Point de vie : environ 700

Equipement : Epée 1 main et bouclier

Sort :

- Charge

- Etourdissement (5 secondes)

- Pourfendre

- Coup de tonnerre

- Brise genou

- Mur protecteur

Objet :

- Potion inférieure

Stratégie :

Création du boss en SQL

Ici on va pas trop si attarder, je vous passe juste la requête SQL pour avoir le même boss que moi.

(A exécuter dans la base [world])

L'entête de notre script

J'ai pour habitude de toujours énumérer tout les items/sort/phase que je vais utiliser dans ce que j'appelle l'entête de notre script, c'est sous forme de tableau ou de variable constante qui vont me permettre de rapidement modifier les différents points clef de notre script, par exemple je vais rapidement pouvoir modifier le pourcentage de point de vie avant le déclenchement de tel phase directement dans notre entête.

Bon pour commencer je déclare la constante qui contiendra l'ID du boss, pour permettre d'utiliser les `RegisterEvent`.

```
local STONE_ID = 75000
```

Ensuite je déclare mon tableau de phase, tableau qui va me permettre de définir les pourcentages de point de vie pour les déclenchements, ainsi que de pouvoir stocker si oui ou non la phase est déjà passée, pour éviter de répéter à l'infini une phase qui rendrait le combat impossible (*notamment ici la phase de heal avec l'[Etourdissement (5 secondes)] et le soin avec la [Potion inférieure]*)

```
--[[
    Ici, par exemple pour obtenir le déclenchement de la phase "Utiliser la potion" on aura qu'à faire
    STONE_PHASE.UTILISER_LA_POTION.POURCENTAGE_DE_DECLANCHEMENT ( ça nous renvoie 40 )
    Au moins plus tard dans le code, les variables seront claires de compréhension, on ne se perdra pas
]]
local STONE_PHASE = {

--[[ PREND EN COMPTE LE POURCENTAGE DE VIE DU BOSS ]]

    UTILISER_LA_POTION = {
        POURCENTAGE_DE_DECLANCHEMENT = 40,
        DEJA_DECLANCHER = false
    },
    UTILISER_LE_MUR_PROTECTEUR = {
        POURCENTAGE_DE_DECLANCHEMENT = 20,
        DEJA_DECLANCHER = false
    },

--[[ PREND EN COMPTE LE POURCENTAGE DE VIE DE LA CIBLE ACTUELLE ]]

    UTILISER_LE_BRISE_GENOU = {
        POURCENTAGE_DE_DECLANCHEMENT = 30,
        DEJA_DECLANCHER = false
    }
}
```

Une fois le tableau des phases déclaré je passe au tableau qui énumère les sorts que le boss pourra utiliser, ici c'est le même principe.

```
local STONE_SPELL = {
    CHARGE = 100,
    ETOURDISSEMENT = 5726,
    POURFENDRE = 6546,
    COUP_DE_TONNERRE = 6343,
    BRISE_GENOU = 1715,
    MUR_PROTECTEUR = 871
}
```

Puis on déclare le tableau des objets que le boss va utiliser (*en soit c'est aussi un sort mais je préfère les différencier*). Ici on ne va pas faire utiliser directement un objet par le boss, mais le sort qui est fixé sur cet objet (*ici notre potion*), donc pour récupérer l'ID du sort, on récupère d'abord l'ID de l'objet en question, on va dans la base de données [item_template] à la ligne de notre objet puis on récupère l'ID du sort à la colonne [spellid_1]. (*Généralement il y aura qu'un seul sort défini par item du moins ici pour des potions/bandage/etc...*)

```
local STONE_SPELL_ITEM = {  
    POTION_DE_SOIN_INFERIEURE = 440  
}
```

Voilà notre partie d'entête est terminé, maintenant on va passé au code en lui même.

Le Combat

Tout d'abord il faut bien pensez qu'à chaque fois qu'on utilisera un `RegisterCreatureEvent` il faudra toujours le mettre après la fonction vers laquelle il pointe, pour ne pas s'embêter je les regroupe tous en toute fin de fichier.

Liste des RegisterCreatureEvent

On passe à notre premier évènement : `OnEnterCombat` avec `ID = 1` .

C'est par cette évènement qu'on va initialiser le combat, et donc lancer les hostilités, si on reprend notre script écrit de la première partie, le boss va en premier lieux essayer d'utiliser la charge, donc c'est partie :

Les fonctions vu ici :

- [GetDistance](#)
- [IsWithinLoS](#)
- [CastSpell](#)

```
-- stone : représente notre boss, target : la cible  
local function OnStoneEnterCombat(event, stone, target)  
  
    -- Ici on regarde si la distance entre notre boss et la cible se situe entre 8 et 25 ( yards )  
    if ( stone:GetDistance(target) > 8 and stone:GetDistance(target) < 25 ) then  
  
        -- Pour éviter les erreurs, on vérifie aussi si le boss est dans la ligne de vue de la cible  
        -- ça évite de lancer charge alors qu'il y a un bâtiment entre les deux  
        if ( stone:IsWithinLoS(target) ) then  
  
            -- Si tout es OK, on lance le sort, sur notre cible.  
            -- Le dernier argument passé à "true" permet d'ignorer la temps d'incantation et la ressources nécessaire.  
            stone:CastSpell(target, STONE_SPELL.CHARGE, true)  
  
        end  
    end  
  
end  
  
-- On oublie pas d'enregistrer l'évènement  
-- Le premier argument est la variable qui correspond à l'ID du boss  
-- Le deuxième argument correspond à l'ID de l'évènement ( OnEnterCombat )  
-- Le troisième argument et simplement le nom vers notre fonction ici : OnStoneEnterCombat ( c'est plus explicite )  
RegisterCreatureEvent(STONE_ID, 1, OnStoneEnterCombat)
```

Dans la suite des choses le combat s'entame, on passe donc à l'utilisation des différents sorts basique durant ce combat, qui sont : [Pourfendre] et [Coup de tonnerre] (*[Attaque à l'épée] se fait de manière automatique*)

Ici on va quasiment devoir créer nos 2 propres évènements, le premier pour lancer [Pourfendre] à intervalle régulier, le second pour lancer [Coup de tonnerre] dans un interface un peut plus aléatoire. Pour ce faire on va rester dans la même fonction `OnStoneEnterCombat` , mais on attribut au boss les deux nouveaux évènements grâce à la fonction

- [RegisterEvent](#)
- [GetVictim](#)

```
local function OnStoneEnterCombat(event, stone, target)
```

```
local function OnStoneEnterCombat(event, stone, target)
```

```
if ( stone:GetDistance(target) > 8 and stone:GetDistance(target) < 25 ) then
    if ( stone:IsWithinLoS(target) ) then
        stone:CastSpell(target, STONE_SPELL.CHARGE, true)
    end
end
```

```
– Ici on appelle la fonction 'CastPourfendre' à un intervalle régulier de 10000 ms soit, 10 secondes
– le troisième argument étant définis à 0, permet de l'appeler indéfiniment toute les 10 secondes
stone:RegisterEvent(CastPourfendre, 10000, 0)
```

```
– Ici on appelle la fonction 'CastCoupDeTonnerre' dans un intervalle aléatoire compris entre 6 et 12 secondes, indéfiniment
stone:RegisterEvent(CastCoupDeTonnerre, {6000,12000}, 0)
end
```

```
--[[
Fonction qui est appelé via un 'RegisterEvent' (ligne:107), elle nous permet donc de lancer le sort 'Pourfendre'
défini dans les tables en tout début de fichier, autant de fois que le 'RegisterEvent' l'appelle.
```

```
]]
function CastPourfendre(eventid, delay, repeats, stone)
    – Ici pour récupérer notre cible, on utiliser la fonction GetVictim() sur le boss
    stone:CastSpell(stone:GetVictim(), STONE_SPELL.POURFENDRE, true)
end
```

```
--[[
Fonction qui est appelé via un 'RegisterEvent' (ligne:108), elle nous permet donc de lancer le sort 'Coup de tonnerre'
défini dans les tables en tout début de fichier, autant de fois que le 'RegisterEvent' l'appelle.
```

```
]]
function CastCoupDeTonnerre(eventid, delay, repeats, stone)
    stone:CastSpell(stone:GetVictim(), STONE_SPELL.COUP_DE_TONNERRE, true)
end
```

Et voilà, désormais [Pourfendre] et [Coup de tonnerre] vont s'utiliser de façon régulière ou aléatoirement sur la cible actuelle du boss, indéfiniment du moment que le boss est en combat. Mais maintenant qu'on a défini nos `RegisterEvent` il faut penser que même quand le boss n'est plus en combat, les événements continueront de s'exécuter, pour palier à ça il va falloir stopper les événements lorsque le boss n'est plus en combat, ou alors lorsqu'il meurt, etc...

On va donc créer une fonction qui va nous permettre de réinitialiser notre boss , autant les phases que les événements.

```
– On met cette fonction tout en haut du fichier juste avant l'entête pour pouvoir l'utiliser depuis n'importe où en dessous par la suite
local function ReinitialisationDesPhases(event, stone, ...)
```

```
– On remet toutes les phases à FALSE
STONE_PHASE.UTILISER_LA_POTION.DEJA_DECLANCHER = false
STONE_PHASE.UTILISER_LE_MUR_PROTECTEUR.DEJA_DECLANCHER = false
STONE_PHASE.UTILISER_LE_BRISE_GENOU.DEJA_DECLANCHER = false
```

```
– On supprime tout les événements lié à notre boss
stone:RemoveEvents()
end
```

```
local function OnStoneEnterCombat(event, stone, target)
```

```
– On l'utilise aussi ici puisque c'est notre point de départ du combat, en prévention il vaut mieux réinitialiser des le début, juste au cas où !
ReinitialisationDesPhases(event, stone)
```

```
...
end
```

```
– Ici on appelle la fonction de réinitialisation sur les évènements suivant :  
– On Died, ID 4 : Quand le boss meurt  
– On Spawn, ID 5 : Quand le boss spawn  
– On Reach Home, ID 24 : Quand le boss retourne à son point initiale ( c'est quand il retourne de là où il vient en sortant d'un combat )  
RegisterCreatureEvent(STONE_ID, 4, ReinitialisationDesPhases) – On Died  
RegisterCreatureEvent(STONE_ID, 5, ReinitialisationDesPhases) – On Spawn  
RegisterCreatureEvent(STONE_ID, 24, ReinitialisationDesPhases) – On Reach Home
```

Maintenant on passe à la suite, on va maintenant scripté la prochaine phase, qui consiste à applique le sort [Brise genou] à la cible lorsqu'elle passe sous la barre des 30% de points de vie.

Pour ce faire on va se servir de l'évènement `OnSpellHitTarget` avec `ID = 15`. Cet évènement se déclenche à chaque fois que le boss lance un sort sur une cible (même une attaque basique compte comme un sort), c'est donc à se moment là qu'on regardera si notre cible passe sous la barre des 30% de points de vie. Et lorsque la cible meurt, on actualise l'évènement pour pouvoir à nouveau l'utiliser.

- [GetHealthPct](#)
- [SelectVictim](#)

```
local function OnStoneTargetDied(event, stone, victim)  
    – Réinitialise la phase du brise genou pour pouvoir l'utiliser sur une autre cible si besoin  
    STONE_PHASE.UTILISER_LE_BRISE_GENOU.DEJA_DECLANCHER = false  
  
    – Force le boss à se trouver une nouvelle cible si possible  
    stone:SelectVictim()  
end  
  
local function OnStoneSpellHitTarget(event, stone, target, spellid)  
  
    –[[  
    Phase : UTILISER_LE_BRISE_GENOU  
    Ici on regarde si la cible à moins de 30% de point de vie, et si la phase n'a pas déjà eu lieu  
    ]]  
    if ( target:GetHealthPct() < STONE_PHASE.UTILISER_LE_BRISE_GENOU.POURCENTAGE_DE_DECLANCHEMENT and not STONE_PHASE.UTILISER_LE_BRISE_GENOU.DEJA_DECLANCHER )  
  
        – Si tout est OK, alors on lance le sort [Brise genou] sur la cible  
        stone:CastSpell(target, STONE_SPELL.BRISE_GENOU, true)  
  
        – Et on défini la phase comme étant passé  
        STONE_PHASE.UTILISER_LE_BRISE_GENOU.DEJA_DECLANCHER = true  
  
    end  
end  
  
– Sans oublier d'enregistrer l'évènement pour que les fonctions puissent se déclencher  
RegisterCreatureEvent(STONE_ID, 3, OnStoneTargetDied)  
RegisterCreatureEvent(STONE_ID, 15, OnStoneSpellHitTarget)
```

Maintenant que cela est fait, on passe au deux phases qui concernent directement la vie du boss, je parle de la phase où il étourdi la cible, puis utilise la potion, ainsi que de la phase où il utilise le sort [Mur protecteur], ici on va donc se servir de l'évènement `OnDamageTaken` qui permet de vérifier à chaque coup que subit le boss, si celui-ci passe en dessous des seuils des différentes phases pour les déclencher :

SendUnitSay

```
local function OnStoneDamageTaken(event, stone, attacker, damage)  
  
    –[[  
    Phase : UTILISER LA POTION
```

```

]]
if ( stone:GetHealthPct() < STONE_PHASE.UTILISER_LA_POTION.POURCENTAGE_DE_DECLANCHEMENT and not STONE_PHASE.UTILISER_

    -- On utilise le sort d'étourdissement sur la cible
    stone:CastSpell(attacker, STONE_SPELL.ETOURDISSEMENT, true)
    -- Puis on utilise le sort de la potion sur le boss
    stone:CastSpell(stone, STONE_SPELL.ITEM.POTION_DE_SOIN_INFERIEURE)

    -- On fait dire au boss une phrase pour annoncer au joueur qu'il vient d'utiliser une potion
    stone:SendUnitSay("Heureusement que j'ai pensé à prendre une potion !", 0)

    -- On indique ici que la phase viens d'être effectué
    STONE_PHASE.UTILISER_LA_POTION.DEJA_DECLANCHER = true

end

--[
    Phase : UTILISER_LE_MUR_PROTECTEUR
]]
if ( stone:GetHealthPct() < STONE_PHASE.UTILISER_LE_MUR_PROTECTEUR.POURCENTAGE_DE_DECLANCHEMENT and not STONE_PHAS

    -- On utilise le sort [Mur protecteur] sur le boss
    stone:CastSpell(stone, STONE_SPELL.MUR_PROTECTEUR, true)

    -- Puis on indique que la phase viens d'être effectué
    STONE_PHASE.UTILISER_LE_MUR_PROTECTEUR.DEJA_DECLANCHER = true

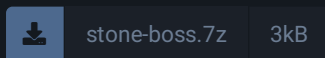
end
end

-- On enregistre l'évènement OnDamageTaken
RegisterCreatureEvent(STONE_ID, 9, OnStoneDamageTaken)

```

Et voilà après tout ça notre script est terminé, le boss est fin prêt !

Le script en entier



Le combat en vidéo

iThorgrim

Merci beaucoup pour ce tutoriel, franchement il est ultra complet !
Ca me donne des idées pour mon petit projet tout ça ☺

Doctor

Super tuto ! Chapeau Camarad ☺☺
