

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.

Mini Boss

iThorgrim

Mini Boss C++

Tutoriel par @killit5

Dans ce tutoriel nous allons voir comment créer simplement un petit boss.

Le script de départ se constitue comme suit :

```
#include "ScriptMgr.h"

enum Spells
{
};

class World_Boss : public CreatureScript
{
public:
    World_Boss() : CreatureScript("World_Boss") { } // script name onto the DB

    struct World_BossAI : public ScriptedAI
    {
        World_BossAI(Creature* creature) : ScriptedAI(creature) {}

        void JustEngagedWith(Unit* /*who*/) override
        {
        }

        void UpdateAI(uint32 diff) override
        {
            if (!UpdateVictim())
                return;

            DoMeleeAttackIfReady();
        }
    };

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

void AddSC_World_Boss()
{
    new World_Boss();
}
```

Dans le type enums Spells nous allons prendre quelque sort simple d'utilisation, un cleave, et un enrage, rien de plus simple.

```
enum Spells
{
    SPELL_CLEAVE          = 15284,
    SPELL_ENRAGE          = 27680
};
```

Ensuite il existe plusieurs fonction disponible comme celle que nous allons utiliser :

```
void JustEngagedWith(Unit* /*who*/) //Fonction appelé lorsque le combat est engagé
void JustDied(Unit* /*killer*/) //Fonction appelée quand la créature est morte
void KilledUnit(Unit* /*victim*/) //Fonction appelée quand la créature tue quelqu'un
```

Nous allons commencer par utiliser la fonction principale.

```
void UpdateAI(uint32 diff)
```

Cette fonction est exécutée toutes les 100ms ou plus environ. C'est ici que le script va se dérouler.

On commence par y ajouter 2-3 lignes nécessaire.

```
void UpdateAI(uint32 diff) override
{
    if (!UpdateVictim()) //Nécessaire, sinon la créature ne va pas vous capter
        return;

    DoMeleeAttackIfReady(); //Pour que la créature effectue des attaques en mêlée
}
```

Comme la fonction `void UpdateAI(uint32 diff)` est "*un timer*" nous allons l'utiliser pour créer "*une temporisation/compteur*" pour dire à quelle moment je souhaite lancer un sort.

Commençons par la technique enrage pourquoi ne pas le lancer à 20% de vie de la créature -> facile.

```
if(HealthBelowPct(20)) //SI 20% de vie
{
    DoCast(me, SPELL_ENRAGE); //Il lance sur lui un sort
}
```

Attention ! Il nous faut rajouter une condition sinon la créature cast le sort sans s'arrêter.

```
bool enraged;

void JustEngagedWith(Unit* /*who*/) override
{
    enraged = false;
}
```

```
void UpdateAI(uint32 diff) override
{
    if (!UpdateVictim())
        return;

    if(HealthBelowPct(20) && !enraged)
    {
        DoCast(me, SPELL_ENRAGE);
        enraged = true;
    }
}
```

```
DoMeleeAttackIfReady();  
}
```

Ensuite nous allons demander à la créature de faire quelque technique. On commence par créer une nouvelle variable qui contiendra une valeur en milliseconde.

```
uint32 spell_cleave_timer;  
bool enraged;  
  
void JustEngagedWith(Unit* /*who*/) override  
{  
    spell_cleave_timer = 10000; //10sec  
    enraged = false;  
}
```

Comme nous le savons tous 1000ms = 1 sec, donc 10000ms = 10 sec. Donc au moment ou le combat commenceras la créature lanceras le sort environ 10 sec après.

```
void UpdateAI(uint32 diff) override  
{  
    if (!UpdateVictim())  
        return;  
    if(spell_cleave_timer <= diff)  
    {  
        DoCastVictim(SPELL_CLEAVE);  
        spell_cleave_timer = 10000;  
    }  
    else spell_cleave_timer -= diff;  
    if(HealthBelowPct(20) && !enraged)  
    {  
        DoCast(me, SPELL_ENRAGE);  
        enraged = true;  
    }  
    DoMeleeAttackIfReady();  
}
```

On utilise ici, simplement la fonction :

```
DoCastVictim(SPELL_CLEAVE);
```

La technique seras lancé sur ça cible.

On aurait aussi bien pu faire d'une autre façon :

```
Unit* target = SelectTarget(SELECT_TARGET_RANDOM, 0);  
DoCast(target, SPELL_ENRAGE);
```

Ici il sélection une cible au hasard, mais d'autre possibilité s'oppose a nous :

```
SELECT_TARGET_RANDOM , // choisir une cible aléatoire  
SELECT_TARGET_MAXTHREAT, // Choisir une cible avec un niveau de menace élevée  
SELECT_TARGET_MINTHREAT, // Choisir une cible avec un niveau de menace faible  
SELECT_TARGET_MAXDISTANCE, // Choisir une cible plus loin  
SELECT_TARGET_MINDISTANCE // Choisir une cible plus proche
```

Bien sur le "0" à côté de SELECT_TARGET_RANDOM n'est pas insignifiant, exemple :

```
(SELECT_TARGET_RANDOM, 0) //Choisir une cible aléatoire
(SELECT_TARGET_RANDOM, 1) //Choisir une cible aléatoire à courte distance
(SELECT_TARGET_RANDOM, 2) //Choisir une cible aléatoire à moyenne distance
(SELECT_TARGET_RANDOM, 3) //Choisir une cible aléatoire à grande distance
(SELECT_TARGET_RANDOM, 4) //Choisir une cible aléatoire avec du mana
(SELECT_TARGET_RANDOM, 5) //Choisir une cible aléatoire avec de la rage
(SELECT_TARGET_RANDOM, 6) //Choisir une cible aléatoire avec de l'énergie
```

Nous pouvons compléter aussi ce script par l'appel d'une nouvelle créature lors du combat.

Rebelote, nous redéclarons une nouvelle variable :

```
uint32 Spawn_Timer;
Spawn_Timer = 25000; //25sec
if(Spawn_Timer <= diff)
{
    me->SummonCreature(17895, me->GetPositionX(), me->GetPositionY(), me->GetPositionZ(), 0, TEMPSUMMON_TIMED_DESPAWN_OUT_
    Spawn_Timer = 25000;
}
else Spawn_Timer -= diff;
```

la fonction `TEMPSUMMON_TIMED_DESPAWN_OUT_OF_COMBAT` permet de faire disparaître la créature 1 sec après la fin du combat. D'autre proposition s'offre encore à nous :

<code>TEMPSUMMON_TIMED_OR_DEAD_DESPAWN</code>	<code>//Despawns après un temps spécifique ou quand le boss disparaît</code>
<code>TEMPSUMMON_TIMED_OR_CORPSE_DESPAWN</code>	<code>//Despawns après un temps spécifique ou quand la créature meurt</code>
<code>TEMPSUMMON_TIMED_DESPAWN</code>	<code>//Despawns après un temps spécifique</code>
<code>TEMPSUMMON_TIMED_DESPAWN_OUT_OF_COMBAT</code>	<code>//Despawns après un temps spécifique ou quand le combat finit</code>
<code>TEMPSUMMON_CORPSE_DESPAWN</code>	<code>//Despawns après la mort</code>
<code>TEMPSUMMON_CORPSE_TIMED_DESPAWN</code>	<code>//Despawns après un temps spécifique après la mort</code>
<code>TEMPSUMMON_DEAD_DESPAWN</code>	<code>//Despawns quand la créature disparaît</code>
<code>TEMPSUMMON_MANUAL_DESPAWN</code>	<code>//Despawns quand la fonction UnSummon() est utilisée</code>

Nous pouvons finir ce tutoriel en étoffant tout ça par quelque parole en utilisant la fonction "*Talk(ID)*", comme suit :

Il nous faut créer un nouveau texte dans la DB, utilisons "*creature_text*".

Pour en savoir plus sur cette DB je vous laisse check [ici](#).

Je vous donne un petit script SQL comme exemple :

```
SET
@CreatureID = 45000,
@Text = "Que trépassé si je faiblis !",
@comment = "Boos Tutoriel";

INSERT INTO `creature_text` (`CreatureID`, `GroupID`, `ID`, `Text`, `Type`, `Language`, `Probability`, `Emote`, `Duration`
(@CreatureID, '0', '0', @Text, '12', '0', '100', '0', '0', '0', '0', '0', @comment);
```

Il nous suffit de mettre notre fonction dans le script.

```
void JustEngagedWith(Unit* /*who*/) override
{
    Spawn_Timer = 25000;
    spell_cleave_timer = 10000; //10sec
    enraged = false;
    Talk(0);
}
```

Au final, nous allons avoir se petit code :

```
#include "ScriptMgr.h"

enum Spells
{
    SPELL_CLEAVE = 15284,
    SPELL_ENRAGE = 27680
};

class World_Boss : public CreatureScript
{
public:
    World_Boss() : CreatureScript("World_Boss") { } // script name onto the DB

    struct World_BossAI : public ScriptedAI
    {
        World_BossAI(Creature* creature) : ScriptedAI(creature) {}

        uint32 spell_cleave_timer;
        uint32 Spawn_Timer;
        bool enraged;

        void JustEngagedWith(Unit* /*who*/) override
        {
            Spawn_Timer = 25000;
            spell_cleave_timer = 10000; //10sec
            enraged = false;
            Talk(0);
        }

        void UpdateAI(uint32 diff) override
        {
            if (!UpdateVictim())
                return;

            if(spell_cleave_timer <= diff)
            {
                DoCastVictim(SPELL_CLEAVE);
                spell_cleave_timer = 10000;
            }
            else spell_cleave_timer -= diff;
            if(Spawn_Timer <= diff)
            {
                me->SummonCreature(17895, me->GetPositionX(), me->GetPositionY(), me->GetPositionZ(), 0, TEMPSUMMON_TIMED_DESPAWN_OUT_C
                Spawn_Timer = 25000;
            }
            else Spawn_Timer -= diff;
            if(HealthBelowPct(20) && !enraged)
            {
                DoCast(me, SPELL_ENRAGE);
                enraged = true;
            }
            DoMeleeAttackIfReady();
        }
    };
};
```

```
CreatureAI* GetAI(Creature* creature) const
{
    return new World_BossAI(creature);
}

};

void AddSC_World_Boss()
{
    new World_Boss();
}
```

