

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.

## Changer et Synchroniser l'écoulement du temps en jeu

EIntemporel

### Changer dynamiquement l'écoulement du temps et le synchroniser pour chaque joueur

(Conçu pour un serveur qui tourne avec TrinityCore-Eluna 3.3.5, ce qui est à mon avis sensiblement pareil sur un autre framework)

Je vous partage aujourd'hui mes scripts qui me permette de modifier le temps en jeu, quand je parle de temps je parle de la vitesse d'écoulement.

**Ainsi 1 journée en jeu correspond à 1 heure dans la vraie vie.**

Pour ce faire nous allons tout d'abord devoir implémenter une nouvelle fonction (en cpp) afin de récupérer l'heure réelle de la machine sur laquelle le serveur tourne, pour que quelque soit le fuseau horaire ou le réglage de l'heure de l'ordinateur du joueur, celui-ci ai la même heure que tout le monde.

**Les fichiers que nous allons modifier sont les suivants :**

- \src\server\game\LuaEngine\GlobalMethods.h
- \src\server\game\LuaEngine\LuaFunctions.cpp

#### GlobalMethods.h

Ici nous allons ajouter une nouvelle fonction pour Eluna, afin de pouvoir récupérer l'heure de la machine du serveur.

Pour cela nous allons nous rendre juste après **[namespace LuaGlobalFunctions]** pour moi c'est directement à la **ligne 26**.

Voici la fonction que j'ai implémenter pour mon code :

```
int GetLocalTime(lua_State* L)
{
    time_t arg = time(NULL);
    tm now;

    localtime_r(&arg, &now);

    lua_newtable(L);
    int _time = lua_gettop(L);

    Eluna::Push(L, now.tm_hour);
    lua_rawseti(L, _time, 0);
    Eluna::Push(L, now.tm_min);
    lua_rawseti(L, _time, 1);
    Eluna::Push(L, now.tm_sec);
    lua_rawseti(L, _time, 2);

    lua_settop(L, _time);
    return 1;
}
```

#### localtime\_r

Ainsi lorsque nous appellerons *GetLocalTime()* dans nos scripts côté Lua, on aura en retour une table qui suivra le model suivant :

[0] = hour,  
[1] = min,

[2] = sec

## LuaFunctions.cpp

Ici il nous faut juste déclarer notre nouvelle fonction, du coup pour moi dans **[GlobalMethods[]]**, **ligne 77** après le commentaire on ajoute :

```
{ "GetLocalTime", &LuaGlobalFunctions::GetLocalTime },
```

Maintenant : **Recompilation !**

Ensuite vous n'avez plus qu'à mettre mes deux fichiers .lua dans votre dossier lua\_scripts et voilà ! ( Tout est normalement expliqué dans les commentaires des fichiers )

Les fichiers utilisent le paquet : **SMSG\_LOGIN\_SET\_TIME\_SPEED** qui est défini ici : \src\server\game\Server\Packets\MiscPackets.h - **ligne 120**

```
class LoginSetTimeSpeed final : public ServerPacket
{
public:
    LoginSetTimeSpeed() : ServerPacket(SMSG_LOGIN_SET_TIME_SPEED, 12) {}

    WorldPacket const* Write() override;

    float NewSpeed = 0.0f;
    uint32 GameTime = 0;
    int32 GameTimeHolidayOffset = 0;
};
```

Grâce a mon script vous pouvez appeler la fonction **Global.isDay()** (*oubliez pas le [require "GlobalMethods"]*), pour savoir si vous êtes le jour ou la nuit, et ainsi paramétrer si vous le souhaitez des évènement durant la journée ou la nuit uniquement.

Pour modifier l'heure du début et de fin de journée, modifié simplement les valeurs dans la fonction **[Global.isDay()]** - **ligne 39** - *GlobalMethods.lua*

**Les valeurs suivent la forme suivante :**

hour \* 60 + (coef hour) + min

avec [coef hour] qui prend pour valeur [hour x 4] si [min] est égal à 0, sinon [coef hour] prend en valeur [(hour-1) x 4]

Exemple :

14h36 = 14 \* 60 + ( 13 \* 4 ) + 36 = 928

14h00 = 14 \* 60 + ( 14 \* 4 ) + 0 = 896

*Un décalage léger au niveau des minutes en jeu peut se faire ressentir, néanmoins ayant implémenter une fonction de correction a chaque heure réelle, ce problème n'en ai presque plus un (la différence est par moment 1 à 2 minutes en jeu grand maximum, ce qui correspond a quelques petites secondes réellement) donc ceci n'impacte en rien le gameplay que vous pourriez instaurer*

Le lien vers les 2 fichiers .lua :



settimespeed.zip

2kB

J'espère que ça vous sera utile !

E-Intemporel.

---

**noc**

Waouh que voila un tutoriel tres rare, bravo et merci pour le partage

## EIntemporel

Petite coquille dans la définition de ma fonction GetLocalTime() du côté cpp, la déclaration de la variable i en temps que uint32 est inutile puisqu'elle n'est pas utilisée, donc vous pouvez supprimer la ligne :

```
uint32 i = 0;
```

---

**noc**

Merci pour cette precision

---