プログラマー向けスクリプトドキュメント

作成日時：2013/11/18

文責：保坂元八

■スクリプターさん用のWIKI

以下の場所にスクリプターさんさん用のドキュメントがあります。

レッドマイン->wiki->ゲームデザイナー＞スクリプト＞スクリプト参考資料

スクリプト作業の前にこちらに目を通してください。

■スクリプト関数の追加手順１　追加先funcsetと関数名を決める

スクリプト関数は種類ごとのfuncsetに分かれています。

新規のスクリプト関数を追加する場合、まずはfuncset\_xxx.incのどこに追加するかを決めてください。以下に、主なfuncsetとその意味合いを列挙します。

funcset\_init.inc:

特殊（初期化）スクリプトで使える関数群です。

特殊スクリプト以外でも使うことが出来ます。

includeは必要ありません。

funcset\_common.inc

汎用のスクリプト関数群です。

特殊スクリプト以外のすべての場所で使うことが出来ます。

includeは必要ありません。

　funcset\_prog.inc

プログラマ専用のグローバルな関数群置き場です。

includeは必要ありません。

　funcset\_prog\_field.inc

プログラマ専用のフィールド限定のグローバルな関数群置き場です。

フィールドのDLLが読まれた状態でないと使用できません。

includeする必要があります。

funcset\_gimmick.inc

ギミック関係のスクリプト関数群です。

必要な.p上でincludeを行ってください。

　Etc….

□追加時の注意点

・プログラマ専用の関数の先頭には\_をつけてください！

■スクリプト関数の追加手順２　関数の定義

　以下の書式で追加先のfuncset\_xxx.incに追記してください。

　native 【関数名】( 引数… );

　例：native NewFunc( arg1, arg2 );

　またこの際、doxygen形式でコメントをつけてください。

　スクリプトサイドの定義は以上で完了です。

■スクリプト関数の追加手順３　ランタイム側の実装

　次に、ランタイム側に関数の実体を定義します。

　以下のパスにScriptFuncSetで始まる、funcset\_xxx.incに対応した.hがありますので、

　そちらを開いてください。

sango\_project/prog/src/field/script

　private領域に以下の書式で関数を追加します。

　static cell AMX\_NATIVE\_CALL 【関数名】(AMX \* amx, const cell \* ptr );

次に、対応した.cppを開きます。

GetTableという関数の中にs\_FuncTableというテーブルがありますので、

こちらに新しく追加した関数を以下の書式で登録します。

GFL\_PAWN\_FUNC( 【関数名】 )

最後に関数の中身を以下の書式で定義します。

Cell AMX\_NATIVE\_CALL 【FUNCSET名】::【関数名】( AMX\* amx, const cell \* ptr ){}

関数の中身を記述すれば実装完了です。

■TIPS

□引数について

引数のptr は配列なっており、0番目にはセルのサイズ \* 引数の数が入ります。

1番目移行はスクリプトの引数が順次入ります。Amxは管理クラスのポインタになっており、以下の関数でScriptObjectを取得できます。

ScriptObject\* p\_obj = ScriptObject::GetClass<ScriptObject\*>(amx);

Pawnは内部で固定小数点を使っているため、小数点を受け取る時は

amx\_ctofでfloatに変換する必要があります。

amx\_ctof(ptr[1]);

□Suspendについて

関数実行後、一度スクリプトを中断して処理を回したい場合は、以下の関数を読んでください。

System\* pSys = System::GetInstance();

　pSYs->Suspend();

以上になります。