13.Conversation Function Category

会話を扱うファンクション

Functions

ActionPauseConversation
実行中のconversationを中断します

ActionResumeConversation
中断をかけていたconversationを戻します。

ActionSpeakString
objectに会話させます

ActionSpeakStringByStrRef
クリーチャーに翻訳した文字列を話させます

ActionStartConversation
PCとのconversationをNPCに開始させます

AddJournalQuestEntry
プレイヤーのジャーナルに登録事項を追加します

BeginConversation
conversationを即座に開始しようと試みる

FaceNearestPC
NPCを一番近くの見えているPCの方に向けます

FloatingTextStringOnCreature
テキストをターゲットの頭上に簡潔に表示します

FloatingTextStrRefOnCreature
テキストである文字列のRefをターゲットの頭上に簡潔に表示します

GetDialogSoundLength
Wave(音声)の長さを、秒で取得ます。

GetJournalQuestExperience
ジャーナルカテゴリーで設定した経験値の量を返します。

GetLastSpeaker
最後に会話したクリーチャーを取得します

GetListenPatternNumber
OnConversationで実行中のconversationパターンを調べます。

GetMatchedSubstring
ListenPatternで指定された文字列と一致した文字列を返します。

GetMatchedSubstringsCount
ListenPatternで指定した文字列と一致した文字列の数を返します。

GetPCSpeaker
現在、NPCと会話中のPCを取得します

IsInConversation
オブジェクトが会話中かどうか決めます。

PlayConversationAnimation
実行物をPCの方を向かせ、アニメーションさせます

SetCustomToken
カスタムトークンの値を設定します。

SpeakOneLinerConversation
１行のconversationを話させます

SpeakString
onjectに会話を強制します

StartingConditional
これはカンバセーション¥エディタ用のテンプレート関数です。

See Also

Function Categories

ActionPauseConversation()

実行中のconversationを中断します

```
void ActionPauseConversation();
```

Description

その行動を取らせるスクリプトからこの関数を呼びだした会話ファイル中のノードである実行中のconversationを中断します。

Remarks

空きアクションキューを持った実行者に対して、この関数を使用すべきです。

そうでなければ効果はないでしょう。

ActionPauseConversation()が効果がない実行者にはClearAllActions()を使ってください。

会話でPCの返事の選択肢を中断する場合、PCの選択はActionResumeConversation()が呼び出されるか、PCが会話を中断する、もしくは範囲外へ出るまでハイライトされます。

Version

1.22

See Also

functions: ActionResumeConversation | ActionStartConversation | GetLastSpeaker
categories: Action on Object Functions | Conversation Functions

---
author: Troels Therkelsen, JP team: Rainie

ActionResumeConversation()

中断をかけていたconversationを戻します。

```
void ActionResumeConversation();
```

Description

前もってActionPauseConversation()によって中断していた会話を取り戻します。

Remarks

ポーズ中にPCが会話を中断した場合（ESCキーを押す、魔法を唱える等）、この関数が呼び出され、会話を戻します。この行為が望ましくない場合、"End Conversation Script - Aborted"スクリプトを使うことで独自にコード化できます。

Pcが会話していたobjectの範囲外へ移動した場合、これは取り戻すことができません。

Version

1.22

See Also

functions: ActionPauseConversation | ActionStartConversation | GetLastSpeaker
categories: Action on Object Functions | Conversation Functions

---
author: Troels Therkelsen, editor: Dave Robinson. JP team: Rainie

ActionSpeakString(string, int)

objectに会話させます

```
void ActionSpeakString(
    string sStringToSpeak,
    int nTalkVolume = TALKVOLUME_TALK
);
```

Parameters

sStringToSpeak
会話に使う文字列

nTalkVolume
TALKVOLUME_* (デフォルト：TALKVOLUME_TALK)

Description

実行者のアクションキューに、sStringToSpeakを話すアクションを追加します

nTalkVolumeはアクションの処理する時に文字列をどの大きさで話させるかを決定します

Remarks

この関数は、実行者に直接会話させるSpeakStringとは異なります

実行者は何かを行うためのアクションキューを持ってないといけないことに注意してください

Version

1.22

See Also

functions: ActionSpeakStringByStrRef | ActionStartConversation | SpeakString
categories: Action on Object Functions | Conversation Functions
constants: TALKVOLUME_* Constants

---
author: Charles Feduke, editor: Maximus, additional contributor(s): Drake Coker, JP team: Rainie

ActionStartConversation(object, string, int)

PCとのconversationをNPCに開始させます

```
action ActionStartConversation(
    object oObjectToConverseWith,
    string sDialogResRef = "",
    int bPublicConversation = FALSE
);
```

Parameters

oObjectToConverseWith
対話する対象にするObject

sDialogResRef
参照するconversationのファイル名（デフォルト："）

bPublicConversation
conversationが周りに、またはPCにのみ聞こえるかどうか（デフォルト：FALSE）

Description

この関数をスクリプトの中で使用すると、実行したObjectのアクション・キューにActionStartConversationを加えます

ObjectがActionStartConversationを実行すると、sDialogResRefで設定した対話ファイルを使って、oObjectToConverseWithに設定された対象に話かけるよう試みます
対象がPCでない場合、その対象のonConversationが開始されます

sResRefはの参照するconversationファイルのファイル名です
sResRefが指定されていない場合、PCではない側のObjectのデフォルトのconversationが使用されます（ツールセットのプロパティにあるセット）

bPublicConversationが指定されていない、もしくはFALSEである場合、そのconversationはPCとObjectの間にのみ残ります
そうでなければ、その会話は雑談として放送され、他のプレーヤーにも「聞こえ」ます

Remarks

PCは、NPCやplaceables（配置用オブジェクト）、トリガ及びドアとの対話を行うことができますがアイテムとは対話することができません
（しかしアイテムと同じ名前を持たせた'透明なオブジェクト'とは可能です）

元々、BioWareの原型ではbPublicConversationはbPrivateConversationと名付けられていましたが、このパラメーター名がその意図したものとは逆の働きをすることに混同していることに気をつけてください

会話をしようとするどのような静的ObjectもPCの視界範囲内にあるでしょう（カメラが自動拡大中でも約10m以内）

もしドアとPCの会話を始めようとして、そのドアが10m以上離れていれば、会話ウィンドウは簡潔に現われて即座に閉じてしまいますが、ドアの最初の文章はPCのチャットウィンドウに表示されるでしょう

アクションを実行するObjectが動的ならば、それはoObjectToConverseWithに走り寄り会話をしようと試みますが、別のアクションがそのキューに加えられるならやめるでしょう

ActionStartConversation関数を登録しようとすると結果は不定です
プレイヤーが通常の移動をしている場合には行動を止める可能性が非常に高いです

sResRefはモジュール内容一覧の「カンバーセーション」の項目にあるconversationのファイル名です

PCはOnConversationがないので、デフォルトのconvasationファイルを持ちません

カメラのZoomは操作できませんが、角度とモードはScriptで指定できます

Known Bugs

v1.28以前のpatchでは、ActionStartConversationをエリアやモジュールに割り当てた場合は問題を引き起こします。

Version

1.28

Example

```
// ---    NPCs   ----
/*
 *これはNPCのOnPerception用で, NPCが気付いた
 * 最初のPCに対し、デフォルトのconversationとして
 * "q_dragnbone"を開始させる。
 * そのconversationはそのPCにのみ聞こえる
 */
ActionStartConversation( GetLastPerceived(), "q_dragnbone");

// --- Placeables ---
/*
 * これは静的ではない配置用オブジェクトのOnUsed用で
 * その配置オブジェクトを使用したPCに対し
 * デフォルトのconversationを開始す
 * (配置用オブジェクトのプロパティの詳細設定タブで設定します)
 */
ActionStartConversation( GetLastUsedBy() );

/*
 * これは静的ではない配置用オブジェクトの
 * OnUsed用で、それを使用したPCに対して
 * "offer_healing"というファイルの会話を開始する
 * 他のプレイヤーもその会話を聞くことができる
 */
ActionStartConversation( GetLastUsedBy(), "offer_healing", TRUE );

// --- Triggers ---
/*
 * これはトリガーのOnEnter用Scriptで
 * 入ったPCとトリガーの間に"odd_voice"という
 * 会話を開始する。（トリガーに）入ったPCだけが
 * 会話を聞くことができる。しかし、会話が始まる前に、
 * プレイヤーの移動は会話を取り消すかもしれない
 */
ActionStartConversation( GetLastEntering(), "odd_voice" );

/*
 * これはトリガーのOnEnter用Scriptで
 * "rat_boy"というタグを持つNPCはPCに駆け寄り
 * 彼のデフォルトのconversationを開始する
 * またOnConversationも始まるが、表示した場合
 * BeginConversationで終わるべきである
 * onConversationの実行の中に異なるconversationのresrefが
 * セットされると、その会話が始められる
 */
object oBoy = GetObjectByTag("rat_boy");
object oPC = GetEnteringObject();
AssignCommand(oBoy, ActionStartConversation(oPC, "", TRUE));
```

See Also

functions: ActionPauseConversation | ActionResumeConversation | ActionSpeakString | BeginConversation | FloatingTextStringOnCreature | GetLastSpeaker | GetPCSpeaker
categories: Action on Object Functions | Conversation Functions

---

author: Iskander Merriman, editor: Charles Feduke, additional contributor(s): Michael Mason, John Detwiler, Jotham, JP team: marshall, Rainie

AddJournalQuestEntry(string, int, object, int, int, int)

プレイヤーのジャーナルに登録事項を追加します。
（日記エディタで、エントリーを作成するのが先決です。）

```
void AddJournalQuestEntry(
string sCategoryTag,
int nEntryID,
object oCreature,
int bAllPartyMembers = TRUE,
int bAllPlayers = FALSE,
int bAllowOverrideHigher = FALSE
);
```

Parameters

sCategoryTag
ジャーナルカテゴリーのタグ（ファイル名での大文字と小文字は区別されます）

nEntryID
ジャーナルエントリーのID

oCreature
ジャーナルエントリーを受け取らせたいPC

bAllPartyMembers
TRUEならば、エントリーはoCreatureのいるパーティのジャーナルに追加されます。
（デフォルト：TRUE）

bAllPlayers
TRUEならば、エントリーはモジュール内の全てのプレイヤーのジャーナルに表示されます。
（デフォルト：FALSE）

bAllowOverrideHigher
TRUEならば、nState（エントリーのID）を現在のジャーナルエントリーに可逆に上書きするに違いないでしょう。（デフォルト：FALSE）

Description

AddJournalQuestEntry()はまさにその名の通りに、プレイヤーの日記画面の「クエスト」または「完了したクエスト」のページにエントリーを追加します。
ただ1人のPC、PCのパーティメンバーの全て、またはモジュール内の全ての駐留者に対して、エントリーを追加しフラグを立てることができます。
関数はジャーナルエントリーを参照する為に、szPlotIDとnStateという、２つのパラメーターを参照します。
sCategoryTagはジャーナルカテゴリーのタグです。例えば、"ボブの素敵なブチ蹴り靴"という名前のクエストを作り、それに"isk_jrnl_bbbbk"というタグを与え、ジャーナルエントリーに割り当てる場合、そのタグを利用してください。

iEntryIDは特定のジャーナルエントリーのIDです。これは日記エディタのID欄に入力した数字で、ジャーナルエントリーを表示する順番を表す数字でもあります。デフォルトでは、AddJournalQuestEntry()を使用した場合、ジャーナルエントリーの数字は増えなければいけません。

このコマンドのジャーナル更新が行われたとき、ジャーナルカテゴリーの名前を無記名のチャットメッセージとして表示します。

Remarks

conversation内でジャーナルエントリーを更新するための、スクリプトを作る必要はありません。「その他のアクション」タブで、望みのジャーナルカテゴリーとジャーナルエントリーを選択するだけでいいのです。

bAllowOverrideHigherをTRUEにする際は注意してください。最終段階である「完了したクエスト」エントリーを、それより前の段階である「クエスト」エントリーへと、矛盾して移行することが簡単にできてしまいます。

日記エディタ内の「カテゴリー終了」チェックボックスで、日記タブのエントリーは固定されて表示されます。完遂した「クエスト」タブの他のものは「完了したクエスト」タブに移ります。「ユーザーメモ」タブにはメッセージを追加することは出来ません。

前にあったジャーナルエントリーは自動的に除去されます。同タイトルのクエストの効果を使いたいならば、異なるタグをカテゴリーに使って、名前だけを同じにします。

PvPトーナメントでの戦いの軌跡を残す為に、上書きフラグは有効に利用出来るかもしれません。

このコマンドはXPの指定は「行いません」。指定したいのであれば、GetJournalQuestExperience()、もしくはGiveXPToCreature()を使ってください。

AddJouralQuestEntryは大/小文字の区別されないsCategoryTagを見つけるでしょう。しかし、同じsCategoryTagではあるものの、異なる大/小文字を使用したクエストは、個々の異なるクエストを追加するでしょう。（同じクエストではありません）

PCのアイテム取得で、ジャーナルエントリーを追加させるようにするにはモジュールのOnItemAcquiredで、この関数を使う必要があります。下記のサンプルを見てください。

ジャーナルエントリーをPCにセットするときに、LocalIntegerは"NW_JOURNAL_ENTRY*"（*は追加されたエントリー名）という名前で、同一のPCに記憶させられます。integer value（整数値）は記憶したジャーナルエントリーのIDです。

Version

1.28

Example

```
// ジャーナルカテゴリー"isk_jrnl_bbbbk"に３のエントリーを登録:
// 1   - 始動メッセージ
// 100 - 中盤メッセージ
// 200 - 終了メッセージ (エディターの「カテゴリー終了」にチェック)
// oPCを有効なPC objectと仮定する
// 実行の順序:

// 参加してきたPCの日記に、ジャーナルカテゴリー"isk_jrnl_bbbbk"の
// ID 1のジャーナルエントリーを追加する
// 他人のジャーナルは影響されない
AddJournalQuestEntry("isk_jrnl_bbbbk", 1, oPC, FALSE, FALSE, FALSE);

// これはそのPCのジャーナルだけをID 100のエントリーに更新する
AddJournalQuestEntry("isk_jrnl_bbbbk", 100, GetEnteringObject(), FALSE, FALSE, FALSE);

// ジャーナルは既にエントリー100になっているため、これらは何も起きない
AddJournalQuestEntry("isk_jrnl_bbbbk", 100, oPC, FALSE, FALSE, FALSE);
AddJournalQuestEntry("isk_jrnl_bbbbk", 1, oPC, FALSE, FALSE, FALSE);

// これはPCの日記画面の「完了したクエスト」タブに
// 終了メッセージを追加する
AddJournalQuestEntry("isk_jrnl_bbbbk", 200, oPC, FALSE, FALSE, FALSE);

// これはクエストの終了メッセージを取り消し
// 「クエスト」タブに中盤メッセージを追加する
AddJournalQuestEntry("isk_jrnl_bbbbk", 100, oPC, FALSE, FALSE, TRUE);

// ----- Completely different example -----

// アイテムを拾う際にエントリーを追加する
// スクリプトをOnAcquireItemに入れる:
// (Thomas Daugaard氏の好意に感謝)
//
// "my_item_tag"と"my_item_quest"とが、
// チェックするアイテムのタグと、適切なジャーナルエントリーのカテゴリー・タグ、
// それぞれが一致するように修正する
void main()
{
    // 拾ったobjectを取得する
    object itemAcquired = GetModuleItemAcquired();

    if(itemAcquired != OBJECT_INVALID) {
        // 拾ったアイテムのタグを取得する
        // "my_item_tag"であれば、正しいアイテムである
        if ( GetTag(itemAcquired) == "my_item_tag") {
            // アイテムを持っているオブジェクト（プレーヤー）を取得する
            object oPC = GetItemPossessor(itemAcquired);
            // そのプレーヤーのジャーナルに適切なエントリーを追加する
            AddJournalQuestEntry ("my_item_quest", 100, oPC);
        }
    }
}
```

See Also

functions: GetJournalQuestExperience | GiveXPToCreature | RemoveJournalQuestEntry | RewardPartyXP
categories: Conversation Functions | Experience Functions | Journal Functions
events: OnAcquireItem Event | OnActivateItem Event | OnAreaTransitionClick Event | OnEnter Event | OnUsed Event | OnUserDefined Event

---

author: Iskander Merriman, editor: Charles Feduke, additional contributor(s): Dave Dursley, Kenneth Cummins, Gabriel Weiss, JP team: geshi, Rainie

BeginConversation(string, object)

conversationを即座に開始しようと試みる

```
int BeginConversation(
    string sResRef = "",
    object oObjectToDialog = OBJECT_INVALID
);
```

Parameters

sResRef
これが指定されない場合、デフォルトの対話ファイルが使用される（デフォルト："）

oObjectToDialog
これが指定されない場合、イベントトリガーを引き起こした人が使用される（デフォルト：OBJECT_INVALID）

Description

この機能はOnDialogの（論理的）終了において希望のconversationを開始させる為に通常使用される。しかしActionStartConversationはより適切かもしれない。

もしsResRefが無指定の場合、このコマンドを実行するobjectのプロパティを参照する。conversationプロパティがあるならば、そのconversationが使用される。

デフォルトでは、この機能は第2のパラメーターであるイベントを含むトリガーオブジェクトを使用する。

Remarks

実行オブジェクトもしくはoOnjectToDialogのいずれかはPCであるに違いないだろう。

Version

1.22

Example

```
// NPCのonConversationの中のこのスクリプトは、 NPCが見つけたエリアに
// 依存する異なるconversationファイルを選択する
void main() {
  // エリア名を取得する
  string sAreaName = GetTag(GetArea(OBJECT_SELF));
  // conversation ResRefのための変数を宣言する
  string sConversation;
  // エリアによる正しいconversationを割り当てる
  if ("isk_a_rangersrest" ==  sAreaName)
    sConversation = "rangers_potboy";
  else if ("isk_a_goldengoblin" ==  sAreaName)
    sConversation = "ggoblin_thief";
  else if ("isk_a_sewer1" ==  sAreaName)
    sConversation = "assassin_servant";

  // 適切なconversationを開始する
  BeginConversation(sConversation);
}
```

See Also

functions: ActionStartConversation
categories: Action on Object Functions | Conversation Functions
events: OnConversation Event

---

author: Iskander Merriman, editor: Jeremy Spilinek、JP team: Rainie

FaceNearestPC()

NPCを一番近くの見えているPCの方に向けます

```
void FaceNearestPC();
```

Description

NPCを一番近くの見えているPCの方に向けます

Requirements

#include "nw_i0_2q4luskan"

Version

1.22

Example

```
// 1.24のソースはこの関数用
void FaceNearestPC()
{
   vector vFace = GetPosition(GetNearestCreature(CREATURE_TYPE_PLAYER_CHAR,
PLAYER_CHAR_IS_PC, OBJECT_SELF, 1, CREATURE_TYPE_PERCEPTION,
PERCEPTION_SEEN));
   SetFacingPoint(vFace);
}
```

See Also

categories: Conversation Functions

---

author: Tom Cassiotis, JP team: Rainie

FloatingTextStringOnCreature(string, object, int)

テキストをターゲットの頭上に簡潔に表示します

```
void FloatingTextStringOnCreature(
   string sStringToDisplay,
   object oCreatureToFloatAbove,
   int bBroadcastToFaction = TRUE
);
```

Parameters

sStringToDisplay
表示するメッセージ

oCreatureToFloatAbove
メッセージを表示させるクリーチャー

bBroadcastToFaction
これがTRUEであるならば、oCreatureToFloatAboveと同じファクション且つ30mの範囲内のクリーチャーだけがテキストを見ます。（デフォルト：TRUE）

Description

目標となるクリーチャーの頭上の空間にテキストを簡潔に表示させて、フェードアウトさせます。SpeakStringと異なり、テキストの前にクリーチャーの名前を表示しません。しかしながらテキストが見えるかどうかは限定されます（以下のRemarksを見てください）。

bBroadcastToFactionがTRUEでなのであれば、目標のクリーチャーと、またその30m以内の同一ファクションを持つクリーチャーがテキストを視認します。bBroadcastToFactionがFALSEならば、ターゲットクリーチャーのみが視認します。

Remarks

この関数はゲーム効果（探知や周囲の知覚など）でPCや可能ならばパーティに通知するのに、最も役立つように思えますが、PCとクリーチャー間のコミュニケーションを促す方法としてはそうでもありません。例えば、PCが聞き耳チェックを成功した場合、このテキストによって通知することができるでしょう（「ホールへと降りてくる足音を聞いた」など）。他の用途としてはPCが発する擬音（「ポキッ！」、「こほっこほっ！」「ヒック！」等）や簡潔なゲーム出力（立ち聞こえ、日時計の時間、Itemの能力などのシミュレート）などです。.

この関数はどんな種類の配置用オブジェクトの上にも表示できません。またNPCを目標としたテキストはPCには見えることはありません。

看板や、日時計、もしくはテキストを表示する時にPCがobjectの近くにいる間は通知したいPCとしてoCreatureToFloatAboveに割り当てられたその他の配置用オブジェクトから、聞こえるテキストを擬制できます。

Version

1.28

Example

```
// これは日時計のOnUsed用のスクリプトである
// 日時計は日中のみ機能する
// もし日中なら、日時計はPCの頭上に
// 時間を表示する

void main()
{
    // 使用者を取得し、それがPCであると定義する
    object oPC = GetLastUsedBy();
    // 日中であるかどうかを調べる
    if (GetIsDay())
    {
        // 変数を初期化
        int nHour = GetTimeHour();
        string sTime = IntToString(nHour) + " o'clock";
        // そのPCの頭上にのみ時間を表示する
        FloatingTextStringOnCreature(sTime, oPC, FALSE);
    }
    else
    {
         // 夜間ならば、時間が特定できないとプレイヤーに告げる
        FloatingTextStringOnCreature("There is no sun to cast "  + "a shadow on the sundial.",
            oPC,
            FALSE
        );
    }
}
```

See Also

functions: ActionStartConversation | FloatingTextStrRefOnCreature | SpeakString
categories: Conversation Functions | PC Only Functions | Visual Effects Functions

---

author: Ryan Hunt, editor: Charles Feduke, additional contributor(s): Erik Jones, Douglas Appelt, JP team: Rainie

FloatingTextStrRefOnCreature(int, object, int)

テキストである文字列のRefをターゲットの頭上に簡潔に表示します

```
void FloatingTextStrRefOnCreature(
   int nStrRefToDisplay,
   object oCreatureToFloatAbove,
   int bBroadcastToFaction = TRUE
);
```

Parameters

nStrRefToDisplay
文字列のRef（故にテキストは翻訳されます）

oCreatureToFloatAbove
ターゲットクリーチャーとなるPCまたはNPC

bBroadcastToFaction
これがTRUEであるならば、oCreatureToFloatAboveと同じファクション且つ30mの範囲内のクリーチャーだけがテキストを見ます。（デフォルト：TRUE）

Description

目標となるクリーチャーの頭上の空間にテキストとしての文字列Refを簡潔に表示させて、フェードアウトさせます。SpeakStringと異なり、テキストの前にクリーチャーの名前を表示しません。しかしながらテキストが見えるかどうかは限定されます（以下のRemarksを見てください）。

bBroadcastToFactionがTRUEでなのであれば、目標のクリーチャーと、またその30m以内の同一ファクションを持つクリーチャーがテキストを視認します。bBroadcastToFactionがFALSEならば、ターゲットクリーチャーのみが視認します。

Remarks

この関数はゲーム効果（探知や周囲の知覚など）でPCや可能ならばパーティに通知するのに、最も役立つように思えますが、PCとクリーチャー間のコミュニケーションを促す方法としてはそうでもありません。例えば、PCが聞き耳チェックを成功した場合、このテキストによって通知することができるでしょう（「ホールへと降りてくる足音を聞いた」など）。他の用途としてはPCが発する擬音（「ポキッ！」、「こほっこほっ！」「ヒック！」等）や簡潔なゲーム出力（立ち聞こえ、日時計の時間、Itemの能力などのシミュレート）などです。

現在、Str Refに変更を加えるトークテーブルを更新する方法がないように思われます。またデフォルトのStr Refコードの公式のリストも存在しません。

Known Bugs

大したバグではないですが制限があります：

この関数はどんな種類の配置用オブジェクトの上にも表示できません。またNPCを目標としたテキストはPCには見えることはありません。

Version

1.22

See Also

functions: ActionSpeakStringByStrRef | FloatingTextStringOnCreature
categories: Conversation Functions | PC Only Functions

---
author: Ryan Hunt, editor: Jeremy Spilinek, JP team: Rainie

GetDialogSoundLength(int)

Wave（音声）の長さを、秒で取得ます。

```
float GetDialogSoundLength(
    int nStrRef
);
```

Parameters

nStrRef
会話オーディオの参照する文字列番号（StrRef）

Description

音声wave（*.wav）ファイルの長さを、秒で取得します。会話で使う音声にのみ動作します。

Version

1.30

See Also

categories: Conversation Functions | Cut-Scene Functions | Sound Effects Functions

---

author: Charles Feduke, JP team: Rainie

GetJournalQuestExperience(string)

ジャーナルカテゴリーで設定した経験値の量を返します。

```
int GetJournalQuestExperience(
    string szPlotID
);
```

Parameters

szPlotID
ジャーナルカテゴリーのタグ

Description

日記上のクエストに経験値を与えるコマンドとして使用します。日記エディタ内のカテゴリーの「XP」欄に入力した経験値の量を返します。

日記エディタの日記上のクエストにXPの数値を設定できますがAddJournalQuestEntry()では、クエスト完了時でも、このXPを与えることはできません。代わりにGetJournalQuestExperience()を使用してください。

Remarks

パーティの経験値を割り当てたい場合は、 Biowareのnw_i0_toolに含まれるRewardPartyXP()を使っても良いでしょう。

スクリプトになれるまでは、カンバーセーション¥エディタのスクリプトウィザードでめんどうな作業をやってしまっても良いでしょう。

Version

1.22

Example

```
// oPCという単体のPCに対して、ジャーナルカテゴリータグが
// "isk_jrnl_bbbbk"であるクエストの経験値を全て与える
GiveXpToCreature(oPC, GetJournalQuestExperience("isk_jrnl_bbbbk");

// クエストはPCにobjectを集める事を要求して
// 得たobjectの総数を格納する
// 最大値に比例してPCにどれくらい与えるかを計算する
// (NUM_HORSESHOES)
int iFound = GetLocalInt(oPC, "iFound");
int iXP = iFound  * GetJournalQuestExperience("shoe_hunt") / NUM_HORSESHOES;
```

See Also

functions: AddJournalQuestEntry | GiveXPToCreature
categories: Conversation Functions | Experience Functions | Journal Functions | PC Only Functions

---
author: Iskander Merriman, editor: Kristian Markon, JP team: Rainie

GetLastSpeaker()

最後に会話したクリーチャーを取得します

```
object GetLastSpeaker();
```

Description

PCが会話している相手の人物を所得する為にこの関数をconversationスクリプトで使用してください

呼び出されたのが有効なクリーチャーでない場合は、OBJECT_INVALIDを返します

Version

1.22

See Also

functions: ActionPauseConversation | ActionResumeConversation | ActionStartConversation | EventConversation | GetPCSpeaker | IsInConversation
categories: Conversation Functions
events: OnConversation Event

---
author: Jody Fletcher, JP team: Rainie

GetListenPatternNumber()

OnConversationで実行中のconversationパターンを調べます。

```
int GetListenPatternNumber();
```

Description

実行されたlistenパターンに記述された整数を返します。これは有効な整数値（-1より大きい）で表れされます。この関数はOnConversationの中でのみ使う事ができます。

listenパターンはSetListenPattern関数を使って設定し、それからObjectが聞いたかどうかをチェックします。関数SetListenPatternが数値10を持った文字列ATTACKを持っていたら、文字列ATTACKが叫ばされたとき(そのために、OnConversationのイベントは起きます)、関数GetListenPatternNumberは10を返すでしょう。

多数のパターンが一致した場合、最も高い数値が返されます。

Remarks

この関数は、NPCがグループでのアクション（例. 衛兵の攻撃）やその他のアクション（例. 追尾）を実行するのを支援する為に主に使われるように思います。

Version

1.22

Example

```
// NPCにこれらの機能を付与する
// ゲームに入る

// WOOTと叫ぶと、メッセージ「YEAH IT WORKS」を呼び出す

// OnSpawn
void main()
{
   SetListening(OBJECT_SELF,TRUE);
   SetListenPattern(OBJECT_SELF,"WOOT",2001);
}

// OnConversation
void main()
{
   int nMatch = GetListenPatternNumber();
   if(nMatch == 2001)
      SpeakString("YEAH IT WORKS");
}
```

See Also

functions: GetMatchedSubstring | GetMatchedSubstringsCount | SetListening | SetListeningPatterns | SetListenPattern
categories: Conversation Functions | Core AI Functions | Henchmen/Familiars/Summoned Functions | Party Functions

---

author: GoLeM, editor: Jochem van 't Hull, JP team: Rainie

GetMatchedSubstring(int)

ListenPatternで指定された文字列と一致した文字列を返します。

```
string GetMatchedSubstring(
    int nString
);
```

Parameters

nString
一致した文字列のインデックス。0から始まります。

Description

ListenPatternで指定された文字列と一致した文字列を返します。
一致した文字列はnStringでインデックス化されて返されます（後述）。
この関数はSetListenPatternで文字列を指定する際ワイルドカードで指定された場合に、OnConversationでのみ有効となります。
（例えば：SetListenPattern(OBJECT_SELF, "FOO**", 500)では"FOO", "FOOTER" "FOOBAR, それに"FOOT"が一致します。
文字列は大文字・小文字の区別はなく"FoOBaR""fooBAR"も同様に一致します。）

例として、SetListenPattern(OBJECT_SELF, "FOO**", 500)と指定し、NPCは ” foobar ” と聞いたとします：

まず文字列が一致しているかどうか比較されます。
GetListenPatternNumber()で500が返されます。

次に一致した文字列が決定されます。
GetMatichedSubstringsCount()で2が返されます。

インデックスは0から始まります（Cの配列と同様です）。
例えば：
GetMatchedSubstring(0) で "foo" を返します
GetMatchedSubstring(1) で "bar" を返します

また他の例として、SetListenPattern(OBJECT_SELF, "**foo**bar**", 500)と指定し、NPCが ” foofoofoobarbarbar ” と聞いたとします：

GetMatchedSubstring(0) で "" を返します
GetMatchedSubstring(1) で "foo" を返します
GetMatchedSubstring(2) で "foofoo" を返します
GetMatchedSubstring(3) で "bar" を返します
GetMatchedSubstring(4) で "barbar" を返します

Remarks

使い方を教えてくださったTemple氏に感謝します。

NWCJに登録されている「謎かけ墓地」で多く使われています（sPassword = GetMatchedSubstring(0);として答えを求めています）。

Version

1.22

Example

```
// このスクリプトではNPCは「woot」という言葉を聞いたら返答します。
// OnSpawnに入れます
void main()
{
   SetListening(OBJECT_SELF,TRUE);
   SetListenPattern(OBJECT_SELF,"Woot**",2001);
}

// OnConversationに入れます
void main()
{
   int i = 0;
   int nMatch = GetListenPatternNumber();
   if(nMatch == 2001)
      SpeakString("YEAH IT WORKS");

   nMatch = GetMatchedSubstringsCount();
   SpeakString(IntToString(nMatch));
   while(i<nMatch)
   {
      SpeakString(IntToString(i) + ". " + GetMatchedSubstring(i));
      i++;
   }
}
```

See Also

functions: GetListenPatternNumber | GetMatchedSubstringsCount | SetListening | SetListeningPatterns | SetListenPattern
categories: Conversation Functions | Henchmen/Familiars/Summoned Functions
events: OnConversation Event

---

author: GoLeM, editor: Kristian Markon, JP team: geshi

GetMatchedSubstringsCount()

ListenPatternで指定した文字列と一致した文字列の数を返します。

```
int GetMatchedSubstringsCount();
```

Description

SetListenPatternでワイルドカードで指定した文字列と一致した文字列の数を返します。
返される値は有効な整数値です；一致したものがない場合は1を返します。
このファンクションはOnConversationでのみ有効です。

例えば、SetListenPattern(OBJECT_SELF, "FOO**", 500)では"FOO", "FOOTER" "FOOBAR, それに"FOOT"が一致します。
文字列は大文字・小文字の区別はなく"FoOBaR""fooBAR"も同様に一致します。）

この場合、NPCが"foobar"と聞くと、GetMatchedSubstringsCountはサブ文字列を調べ、2を返します（"foo"とbar"）。そして返された(2)でGetMatchedSubstringを使って一致したサブ文字列を得ることが出来ます。

Remarks

使い方を教えてくださったTemple氏に感謝します。

Version

1.22

Example

```
// このスクリプトはNPCがwootと聞いたら返答するようにします
// OnSpawn
void main()
{
   SetListening(OBJECT_SELF,TRUE);
   SetListenPattern(OBJECT_SELF,"Woot**",2001);
}

// OnConversation
void main()
{
   int i = 0;
   int nMatch = GetListenPatternNumber();
   if(nMatch == 2001)
      SpeakString("YEAH IT WORKS");

   nMatch = GetMatchedSubstringsCount();
   SpeakString(IntToString(nMatch));
   while(i<nMatch)
   {
      SpeakString(IntToString(i) + ". " + GetMatchedSubstring(i));
      i++;
   }
}
```

See Also

functions: GetListenPatternNumber | GetMatchedSubstring | SetListening | SetListeningPatterns | SetListenPattern
categories: Conversation Functions | Henchmen/Familiars/Summoned Functions
events: OnConversation Event

---

author: GoLeM, editor: Kristian Markon, JP team: geshi

GetPCSpeaker()

現在、NPCと会話中のPCを取得します

```
object GetPCSpeaker();
```

Description

現在、NPCと会話中のPCを取得します

Remarks

この関数は会話の一部として呼び出されたスクリプトにのみ使用でき、常に会話中のPCを返します。

Version

1.28

Example

```
// "Actions Taken"スクリプトはNPCのconversationで
// PCに対してクエスト変数を設定する
void main() {
  object oPC = GetPCSpeaker();
  SetLocalInt( oPC, "golden_duck", 100);
}
// --- 例'Actions Taken'の終了

// StartingConditionalはNPC conversationである
// "Text Appears When"用スクリプトである
// このスクリプトが付属する会話の選択肢（node、ノード）は、
// PCの変数である"golden_duck"が100に設定されたときのみ表示されます。
int StartingConditional() {
  object oPC = GetPCSpeaker();
  int nDuckState = GetLocalInt( oPC, "golden_duck");
  if (100 == nDuckState) return TRUE;
  return FALSE;
}
```

See Also

functions: ActionStartConversation | GetLastSpeaker
categories: Conversation Functions | PC Only Functions
events: OnConversation Event

---

author: Iskander Merriman, editor: Charles Feduke, JP team: Rainie

IsInConversation(object)

オブジェクトが会話中かどうか決めます。

```
int IsInConversation(
    object oTarget
);
```

Parameters

oTarget
チェックするobject

Description

この関数は、目標の会話状態に基づいて、TRUEもしくはFALSEを返します。

目標の会話状態に基づいた行動を条件付きで実行させるような場合にこの関数を使用してください。

Version

1.22

Example

```
if (IsInConversation(OBJECT_SELF))
{
  // ここで会話中の行動を指定します
}
else
{
  // ここで会話中で無いときの行動を指定します
}
```

See Also

functions: GetLastSpeaker
categories: Conversation Functions

---
author: Michael Nork, editor: Jeff Lindsey, JP team: geshi, Rainie

PlayConversationAnimation(int, object)

実行物をPCの方を向かせ、アニメーションさせます

```
void PlayConversationAnimation(
   int nAnimationConstant,
   object oTarget
);
```

Parameters

nAnimationConstant
実行させるアニメーション（ANIMATION_*）

oTarget
正面向かさせるobject

Description

まず最初に、実行物はoTargetの方を向きます。oTargetが有効なobjectではない場合、その代わりとして一番近くのPCの方を向きます。それから、実行者はnAnimationConstantをアニメーションします。もしこれがループするアニメーションであるなら、2秒間だけ実行されます。

Remarks

宿屋のスクリプトや2人のNPCを会話させあう場合のようなスクリプトに非常に適しています。

Requirements

#include "nw_i0_2q4luskan"

Version

1.29

Example

```
//"drunk_man"というタグを付けたNPCの周辺にトリガーをペイントする
#include "nw_i0_2q4luskan"

void main()
{
//「酔っ払い」に近寄ったのは誰か？
object oTarget=GetEnteringObject();

//トリガーの中心の「酔っ払い」
object oDrunk=GetObjectByTag("drunk_man");

//入ってきたobjectが「酔っ払い」である場合、実行しない
if (oTarget==oDrunk) return;

//３種のアニメーションからランダムに１つを選択する
int nAnimation;
switch (d3())
{
case 1: nAnimation=ANIMATION_LOOPING_TALK_LAUGHING; break;
case 2: nAnimation=ANIMATION_LOOPING_PAUSE_DRUNK; break;
case 3: nAnimation=ANIMATION_FIREFORGET_DRINK; break;
}

//oDrunkをoTargetに向け、酔っ払ったような行動を取らせる
AssignCommand(oDrunk, PlayConversationAnimation(nAnimation, oTarget));

}
```

See Also

categories: Action on Object Functions | Conversation Functions | Visual Effects Functions
constants: ANIMATION_* Constants

---
author: Lilac Soul, editor: Charles Feduke, JP team: Rainie

SetCustomToken(int, string)

カスタムトークンの値を設定します。

```
action SetCustomToken(
   int nCustomTokenNumber,
   string sTokenValue
);
```

Parameters

nCustomTokenNumber
カスタムトークンの番号です。

sTokenValue
カスタムトークンの値です。

Description

conversationで使われるカスタムトークンの値を設定します。

どのスクリプトのカスタムトークンでも設定できます。しかしながら、条件を含ませておくStartingConditionalスクリプトのトークンを設定を考えるべきかもしれません。

Remarks

カスタムトークン０－９はBiowareによって使われており、使用出来ないはずです。

設定した同じカスタムトークンを設定したスクリプトの構成を、再利用するのは危険です。これを避ける為には、conversationの直前にカスタムトークンを設定してください（conversatonの内の新しいトークンを作成せず、会話の初めにそれらをすべて作成します）。

カスタムトークンを使用するには、conversationのどこかに<CUSTOMxxxx>を配置します。xxxxはnCustomTokenNumberで与えられた値です。<CUSTOM100>のようにです。

Version

1.28

Example

```
// ここに作成したカスタムトークンは、<CUSTOM100>
// のタイピングにより、会話の中で使用する事ができる
int StartingConditional()
{
  SetCustomToken(100, GetName(GetPCSpeaker()));
  return TRUE;
}
```

See Also

categories: Conversation Functions

---

author: Tom Cassiotis, editor: Charles Feduke, additional contributor(s): Vincent Bairan, JP team: Rainie

SpeakOneLinerConversation(string, object)

１行のconversationを話させます

```
void SpeakOneLinerConversation(
    string sDialogResRef = "",
    object oTokenTarget = OBJECT_TYPE_INVALID
);
```

Parameters

sDialogResRef
(Default: "")

oTokenTarget
文字列に特定のクリーチャーのトークンがあるのならば指定しなければなりません
(デフォルト：OBJECT_TYPE_INVALID)

Description

実行objectの会話ダイアログに、選択肢のない会話行を即座に話させます。conversationに2番目以降のパーティはいないので、oTokenTargetは１行に含まれるトークンを設定する為に使用することができます。

Version

1.22

See Also

categories: Conversation Functions
constants: OBJECT_TYPE_* Constants

---
author: Jeff Lindsey, JP team: Rainie

SpeakString(string, int)

onjectに会話を強制します

```
void SpeakString(
    string sStringToSpeak,
    int nTalkVolume = TALKVOLUME_TALK
);
```

Parameters

sStringToSpeak
会話させるテキストです

nTalkVolume
TALKVOLUME_* (デフォルト: TALKVOLUME_TALK)

Description

実行者は直接sStringToSpeakを話します

nTalkVolumeは文字列を会話する声の大きさを決定します

Remarks

実行者は一般トリガーにはなれません。それがそうである場合、何も起こりません。

もしトリガーでテキストを表示させたい場合は、（例えば、FloatingTextStringOnCreatureで）PCの頭上に、もしくはテキストを話すことができる静的ではない透明な配置物を作ってトリガーにテキストを表示させます。

この関数はアクションキューにアクションを追加するActionSpeakStringとは異なります。

Version

1.22

See Also

functions: ActionSpeakString | ActionSpeakStringByStrRef | DebugSpeak | FloatingTextStringOnCreature
categories: Conversation Functions
constants: TALKVOLUME_* Constants

---
author: Charles Feduke, additional contributor(s): Drake Coker, JP team: Rainie

StartingConditional()

これはカンバセーション・エディタ用のテンプレート関数です。

```
int StartingConditional();
```

Description

カンバセーション・エディタの「テキスト表示の条件」タブで、StartingConditional():intスクリプトを作成する為に、この関数は利用されます。

Remarks

書かれたスクリプトがコンパイルできない通り、これはテンプレートでのみ有効です。
nw_d2_racesnh.nss

Requirements

#include "nw_d2_racesnh.nss"

Version

1.22

See Also

categories: Conversation Functions

---

author: Michael Nork, JP team: Rainie