Voice Handler

Overview

The voice handler is basically a simplified way in which to play complex sound events that deal with voices. It is also a way to easily handle subtitles along with the voices and play special effects track along with them. It has a queue system and keeps track of how many times a voice has been played.

Note that while often used along with the event database, these are two completely separate systems, so they should not be confused with one another.

Setup

Before calling a subject the source of the sound should be set up. This is done with SetVoiceSource() in cVoiceHandler, or better the helper function Voice_SetSource(). This sets up the entity (which can be something moving) which will emit the voice coming from the specified character. It will also set up the distance that the voice can be heard. Do this in OnEnter in the map script file.

Properties

Character

The character is the agent that is speaking the lines. Normally this is a person of some sort but it can really be whatever. It mostly used used as a way to give debug data on what character is saying what line, and also used to place the voice in the world (the position of character can be set in script).

ID
Internal Id

Name
Name of the character

AutoGenVoice
If the voices should be autogenerated (jrpg style).

AutoGenVoiceFile
The sample that should be played when the character speaks.

AutoGenVoiceSoundFreq
Determines the frequences (times per second) that AutoGenVoiceFile is played.

Scene

This does not have to mean the level that the voice are played in, but what subtext that they are used in. A scene is NOT the same as the level the voices belong to, but a level can (and should have) have many scenes. It is a way to group subjects and also to setup what voices should be in focus (by using FocusPrio).

ID
Internal Id

Name
Name of the scene

FocusPrio
This is the priority for a voice to into focus, meaning that is subtittles are shown and any other voice playing gets its volume lowered. To beat a currently playing voice, FocusPrio must be higher.
When a subject starts playing it is checked if another sound is playing and FocusPrio determines if the new subject gets into focus. Also when a subject stops playing a certain amount of time is waited (default 2 seconds) and then there is a check to see if there is any playing voice that should come into focus. If there are many playing, FocusPrio determines which one. The reason for waiting a litte while is in case there is a conversiation in one scene containing many subjects, and if so it would not sound good if the focus was constantly switched (ie unrelated subtitles popping up for a second).

Effect

This defines certain effect that should be played on the voice, be that echo, noise or whatnot.
Note: Right now there is not much done with effect, so it is pretty useless

ID
Internal Id

Name
Name of the effect

GlobalStartEffectFile
A file that is played each time a new line starts.

GlobalEndEffectFile
A file that is played each time a new line is over.

GlobalVoiceOffset
Not used.

Subject

A subject is what you call when you want to voice to be played. A subject is the basic data structure of the voice data and it contains sound files, subtitle texts and various options.

A subject contain one or more **Lines** (more on these below). Depending on current state and settings, one (or none) of these lines will be played when a certain subject is called.

ID
Internal Id

Name
Name of the subject

Prio
The priority of the subject. This will depend if a subject voice will be blocked when another voice in the same scene is already playing. If prio is higher than the one currently played, the current is stopped and the new one plays.

BlockedAction
This is what happens in case a line is already playing, is part of the same scene and the new line has same or lower Prio has the current. Possible values are:
Enqueue: The line is queued and played as soon as the currently played (and any other already queued lines) has finished playing.
Skip: The line is not played. If this happens it counts as a playback and thus affects MaxPlaybacks (see below).
StandBy. The line is not played, but neither does it count as a playback. It is as if a Voice Subject was never called to be played.

RandomizeOrder
If the Lines should be chosen randomly. If false, the line declared first plays first.

Trigger
This is the trigger that the subject responds to. This can either be a standard trigger (as outlined here) or it can be a custom trigger. A custom trigger is either created through script or with the ResponseTrigger property (see Line properties below).

Criteria
This is the criteria that needs to be correct for the subject to played. Remember that if several Subjects (triggered by the same Trigger) all have matching criteria, then the Subject with the most Critera is the one that is played. This makes it possible to have default trigger responses by having no Criteria at all, and these will only be used when no other subject match their criteria.
More info on syntax and such here

Actions
Actions are used to set change the facts in the working memory of the event database. This memory is then checked with evaluating Criteria and therefore the contents of the actions help decide what Subjects are picked. The actions are out at the moment the voice starts playing (meaning when the subject is picked and OutputDelay has passed). Note that should the voice not play the actions are still carried out!
More info on syntax and such here

OutputDelay
This is the delay between an event being picked by a trigger and the voice being played and actions performed.

CharacterId
Id of character connected to the subject.

SceneId
Id of scene connected to the subject.

EffectId
Id of the effect used for the subject.

Line

When a voice is called to be played the entirety of a line is always played. The line contains one or more **Sounds** (more on these below) that make up what is played.

ID
Internal Id

The name of the line
Name of the line

MaxPlaybacks
Max number of times this line can be played. If 0, then the number of times is unlimited.

ResponseTrigger
When the line has finished playing, this custom trigger is sent back into the event system (which the voice handler uses).
This makes a for a very nice way of linking subjects in a dynamic way. An example:
When “Subject A” has finished playing it can send “RespondToA” as a trigger. Both subjects B and C than then have “RespondToA” as their trigger and the subject with the mpst fitting criteria will be picked (if any). For instance subject B might have the Criteria “Bananas>2” as a Criteria, while C's Criteria is empty (“”). This means that after Subject A has been playing AND the fact Bananas is over 2, then Subject B will start playing. If the fact Bananas is not above 2, then Subject C will play. Note that when “Bananas>2” is true, both subject match, but since the one with the most matching criteria (1 vs 0) is picked, Subject B is wins and is played.

Sound

This is the lowest level data structure for a subject. It contains properties for the voice, (opitonal) effect sound, and subtitle.

The file name for the voice sound is generated based on the names of the higher level strucutres. The syntax is:
[scene]_ [character] _[subject] _[line] _[sound index in 3 digits]

Example, the firs sound of a subject can be:
Arthur_Camelot_AngryAtMerlin_Insults_001

The lang file entry for the text is also generated, with this syntax:
Category: “Voices”
Entry: *[scene]_ [character] _[subject] _[line] _[sound index in 3 digits]* (exactly the same as for the voice file!)

Finally, if “AutogenerateLangFiles” in “Main” is set to true in the user settings, then the lang file entries will be autogenerated with the text specified in the sound's properties.

Text
The text for the voices, not really saved by added to a lang file (if AutogenerateLangFiles is true i settings, see above)

HasVoice
If the sound has a voice at all. This is only meaningful false if there is an Effect filed specificed.

Volume
Volume of the sound and effect file.

VoiceOffset
Number of seconds before voice starts to play.

TextOffset
Number of seconds before the subtitles show up.

EffectOffset
Number of seconds before the effect file starts to play.

EffectFile
File path to an effect file that is played along with the voice.

EndsAfterEffects
If the sound should end when the effect file is done playing.

EndPadding
Number of seconds of padding in the length of voice file. If this is over 0, then that means the next sound will start that many seconds earlier (while the current sound is still playing).

File Format

```
<VoiceData>
  <Characters>
    <Character [Properties] />
    ...
  </Characters>
  <Effects>
    <Effect [Properties] />
    ...
  </Effects>
  <Scenes>
    <Scene [Properties] />
    ...
  </Scenes>
  <Subjects>
    <Subject [Properties] >
      <Line [Properties] >
```

```
        <Sound [Properties] />
        ...
      </Line>
      ...
    </Subject>
  </Subjects>
</VoiceData>
```

For information on the Properties that can be used, see Properties section above.