

Inhalt

Lautstärke.....2

Ton abspielen.....3

Stimmen.....3

Geräusche.....3

Playlists.....4

Lautstärke

Du möchtest Musik oder Sounds abspielen. Aber der böse Spieler hat den Ton abgestellt? Was nun? Natürlich kann die Lautstärke im Skript angepasst werden. Anschließend können die Änderungen einfach zurückgedreht werden.

Die Lautstärke kannst Du mit diesen Funktionen beeinflussen:

API.SoundSetVolume

Die Funktion stellt die Gesamtlautstärke ein.

API.SoundSetMusicVolume

Die Funktion stellt die Lautstärke der Musik ein.

API.SoundSetVoiceVolume

Die Funktion stellt die Lautstärke von Sprache ein.

API.SoundSetAtmoVolume

Die Funktion stellt die Lautstärke der Umgebung ein.

API.SoundSetUIVolume

Die Funktion stellt die Lautstärke der Oberfläche ein.

Es wird automatisch eine Sicherung der Einstellungen vorgenommen.

In diesem Beispiel wird alles abgeschaltet, außer der Musik.

```
API.SoundSetVolume(100)
API.SoundSetMusicVolume(50)
API.SoundSetVoiceVolume(0);
API.SoundSetAtmoVolume(0);
API.SoundSetUIVolume(0);
```

So werden die Einstellungen des Spielers wiederhergestellt.

```
API.SoundRestore();
```

Ton abspielen

Du kannst verschiedene Audiodateien abspielen.

Musik, Sounds und Stimmen müssen in ihre entsprechenden Verzeichnisse gepackt werden.

Stimmen

Stimmen sind MP3-Dateien. Sie liegen entweder im Spielverzeichnis oder in der Map. Eigene Stimmen solltest du immer im Verzeichnis der Map ablegen.

So muss das Verzeichnis aufgebaut werden:

```
mapname.s6xmap.unpacked
|-- maps/externalmap/mapname
    |-- voices/*
```

Bedenke, dass `mapname` für den Ordernamen deiner Map steht. Der ist von Map zu Map anders!

Dann kannst du die Stimme abspielen.

Spielen wir nun Stimmen ab!

```
API.PlayVoice("maps/externalmap/myawesomemap/voices/say_hello.mp3");
```

Hier spielst du eine eigene Datei ab. Die Sprachausgabe beendet sich am Ende selbst.

(Die Funktion ist nicht geeignet, um eine Sprachausgabe zu realisieren!)

```
API.StopVoice();
```

Damit kannst du die Sprachausgabe selbst beenden.

Geräusche

Geräusche teilen sich auf in 2D und 3D.

2D-Sounds werden als UI-Sound abgespielt und sind immer zu hören.

3D-Sounds werden als Umgebungsgeräusch an einer Position auf der Map abgespielt.

Ein Geräusch muss immer WAV-Format sein!

Geräusche müssen im Spielverzeichnis liegen. Sonst werden sie nicht gefunden. Es muss eine Low Quality und eine High Quality vorhanden sein.

So muss das Verzeichnis aufgebaut werden:

```
mapname.s6xmap.unpacked
|-- sounds
    |-- high/*
    |-- low/*
```

Bedenke, dass `mapname` für den Ordernamen deiner Map steht. Der ist von Map zu Map anders!

Die Dateien werden in ihre Unterordner abgelegt (z.B. `ui` für `ui/menu_left_gold_pay`).

Du kannst keine neuen Ordner erstellen!

Spielen wir nun Geräusche ab!

```
API.Play2DSound("ui/menu_left_gold_pay");
```

Es wird ein Geräusch in der Benutzeroberfläche abgespielt.

```
API.Play3DSound("Animals/cow_disease", 8500, 35800, 2000);
```

Es wird ein Umgebungsgeräusch an der Position abgespielt.

Playlists

Eine Playlist definiert die abgespielte Musik.

Es ist der vielleicht kompliziertere, dafür aber richtige Weg, Musik abzuspielen.
Die angespielte Musik wird vom Spiel auch als Musik behandelt!

Damit Du eine Playlist verwenden kannst, müssen Playlist und Musikdateien vorhanden sein!

Die Verzeichnisse

So muss das Verzeichnis aufgebaut werden:

```
mapname.s6xmap.unpacked
|-- config/sound/*
|-- maps/externalmap/mapname
    |-- music/*
```

Bedenke, dass `mapname` für den Ordernamen deiner Map steht. Der ist von Map zu Map anders!

Die Playlist kommt nach `config/sound`.

Die Musik kommt nach `maps/externalmap/mapname/music`.

Du kannst keine neuen Ordner erstellen!

Aufbau einer Playlist

Eine Playlist ist eine XML-Datei. In ihr gibst Du die Tracks an, die abgespielt werden.

Eine Playlist kann wie folgt aussehen:

```
<?xml version="1.0" encoding="utf-8"?>
<Playlist>
  <PlaylistEntry>
    <FileName>maps\externalmap\mapname\music\some_music_file.mp3</FileName>
    <Type>Loop</Type>
  </PlaylistEntry>
  <!-- Weitere PlaylistEntry-Einträge -->
</Playlist>
```

In `FileName` steht der Pfad zur Musikdatei.

Mit `Type` wird festgelegt, wie die Musiktatei abgespielt wird.

- `Loop` bedeutet, dass sich der Track endlos wiederholt
- `Normal` bedeutet, dass der Track einmal abgespielt wird

Playlist verwenden

Eine Playlist wird immer für einen bestimmten Spieler abgespielt oder gestoppt.

Lässt du den Spieler weg, wird von Spieler 1 ausgegangen.

Du kannst die Playlist durch einen einfachen Aufruf starten.

```
API.StartEventPlaylist("config/my_playlist.xml");
```

Und natürlich genauso einfach wieder beenden.

```
API.StopEventPlaylist("config/my_playlist.xml");
```