

Gothic

E-2: Effects

-
Letzte Änderung: 12.07.99 15:00
Druckdatum: 03.11.98 08:58
Autor: Kai Rosenkranz
-

© 1997,98, 99 Piranha Bytes Software GmbH

Inhaltsverzeichnis

| | | |
|-----------|------------------------|-----------|
| 1. | PARTIKELEFFEKTE | 2 |
| 2. | SOUND | 2 |
| 2.1 | MUSIK | 2 |
| 2.2 | SOUNDEFFEKTE (SFX) | 2 |
| 2.3 | SPRACHE | 11 |
| 3. | SOUNDLISTEN | 12 |
| 3.1 | ANIMATIONS-SFX | 12 |
| 3.2 | UNTERGRUND-SFX | 13 |
| 3.3 | AKTIONS-SFX | 14 |
| 3.4 | KOLLISIONS-SFX | 14 |
| 3.5 | ZERSTÖRUNGS-SFX | 16 |
| 3.6 | ZAUBER-SFX | 16 |
| 3.7 | SKRIPT-SFX | 17 |
| 3.8 | VOB-SFX | 17 |
| 3.9 | GEBIETS-SFX | 17 |

1.

Partikeleffekte

Partikel erscheinen an der Aufprallstelle, fliegen und verschwinden bei Aufprall

- Rennen in flachem Wasser
- Gegenstand fällt ins Wasser
- Blutspritzer
- Holz- und Steinsplitter
- Blutlache breitet sich aus. Klebt an der unteren Seite der Bounding Box. Wenn Neigung $< 10^\circ$, gibt es ein Ausbreiten zu allen Seiten, wenn Neigung steiler, fließt die Lache (andere AnimTex) bergabwärts (bis zum nächsten Knick?)

Technische Details zu den Partikeleffekten findet Ihr in einer separaten Dokumentation.

[..\DOKUZenGin_ParticleFXDoku.doc](#)

In diesem Dokument befinden sich Hinweise zum Aufbau eines PFX, dessen Parametern, Partikelverhalten, globalen Variablen zur Manipulation der Flugbahn und Visualisierung. Es tauchen jedoch keine Hinweise zu Gestaltung, Design oder Anwendung auf.

Die Praxis hat gezeigt, daß sich mit dem Partikelsystem auch weit mehr Effekte realisieren lassen, als man mit dem Begriff „Partikel“ verbindet. Alle grafischen Feinheiten, die man aus einzelnen Elementen zusammensetzen kann, lassen sich prima als Partikeleffekt umsetzen. So z.B. Fliegenschwärme, Fallendes Laub, Gischt, Zerstörungs-Effekte und vieles mehr. Da jeder Partikel separat berechnet wird („schlimmstenfalls“ mit Kollisionskontrolle, Gravitation, Blending), sind Partikeleffekte enorme Ressourcenfresser. Sparsamer Umgang mit der Anzahl und Größe der Partikel ist deshalb ratsam.

2. Sound

2.1 Musik

- Dynamisches Musiksystem
- Übergänge zwischen den Musiken
- Liste der nötigen Musikthemen

2.1.1 Dynamisches Musiksystem

Anders als bei einer linear aufgebauten Musikuntermalung (z.B. CD-Audio-Track) paßt sich die Musik dank des auf DirectMusic aufgebauten Musiksystems dynamisch an das Spielgeschehen an.

Folgende Faktoren können die Intensität und Stilrichtung der Musik beeinflussen:

- Die Location, in der sich der Spieler gerade befindet

- Die Spielsituation, z.B. „nix los“, „bedrohlich“, „Kampf“

Das gesamte Musikmaterial besteht aus kleinen „Puzzlesteinen“ (Patterns), die jeweils ohne hörbaren Übergang aneinander passen (meistens ☺). DirectMusik gibt während des gesamten Spiels einen Taktgeber (vergleichbar mit einem Metronom) vor, die Geschwindigkeit kann aber jederzeit verändert werden. Jeweils am Ende eines Taktes wählt das System, abhängig von den oben genannten Faktoren und einem Zufallsgenerator, das nächste Pattern aus. In zeitkritischen Situationen kann das Abwarten des Taktendes auch unterbunden werden, so daß unmittelbar das Thema gewechselt wird.

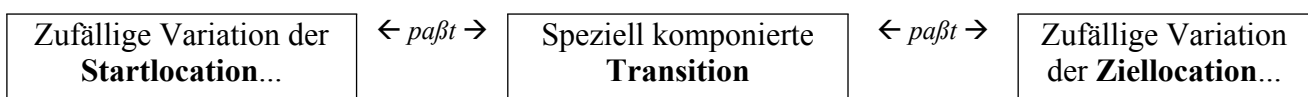
Die „Innereien“ jedes dieser Patterns erinnert an ein MIDI-Arrangement, bietet aber eine handvoll mehr Funktionen. Im Vorfeld wurde ein Pool an Gothic-kompatiblen Instrumenten angelegt, die dann wie ein Synthesizer vom Arrangement angesprochen werden können. Das hält den Speicherbedarf in Grenzen (**optimal: max. 3MB pro Location**) und ermöglicht gleichzeitig endlos viele Variationen, da jede Variation lediglich aus ein paar Bytes Daten besteht, die auf das bestehende Audio-Material zurückgreifen.

Die Klänge sind in Wave-Files gespeichert, die über die folgenden Audioeigenschaften verfügen sollten: **22.05Hz, 16 Bit, Mono**. Diese Konfiguration hat sich als der sinnvollste Kompromis zwischen Speicherplatz und Klangqualität herausgestellt. Stereo-Files werden nicht unterstützt, lassen sich aber nachträglich von DirectMusik aus den Mono-Files zusammensetzen. Weiterhin liegen alle Samples als „Rohstoff“ vor, sämtliche Effekte werden nachträglich von DirectMusic hinzugefügt.

2.1.2 Übergänge zwischen den Musiken

Wie im Abschnitt 2.1.1 beschrieben, liegen für jede Location und Intensität mehrere Puzzlesteine (Patterns) vor, die von einem Zufallsgenerator ausgewählt werden. Darüber hinaus gibt es eine festgelegte Anzahl von speziellen Pattern, die als separat komponierte Übergänge (Transitions) zwischen 2 zufällig gewählte Patterns gespielt werden. Das erhöht den Eindruck, daß es sich um eine Musik ohne Unterbrechungen oder abrupte Übergänge handelt.

Beispiel eines Ortswechsels:



Natürlich muß nicht zwangsläufig für jeden Möglichen Themenwechsel (OC→OW, OW→NC, NC→PC, Ruhig→Kampf, usw...) eine separate Transition komponiert werden. Es reicht wohl aus, für jede Location eine Anzahl von 3 Transitions zur Verfügung zu stellen, und dann festzulegen, welche davon am besten für welchen Übergang paßt.

Probleme können auftreten, wenn mehrere Übergänge von der Spielelogik „angeordnet“ werden, während gerade eine Transition gespielt wird. Schlimmstenfalls kann das zu hörbaren „Einschnitten“ in der Musik führen.

2.1.3 Liste der nötigen Musikthemen

| Kategorie | Benötigte Themen: |
|---|--|
| Outdoor-Locations: <ul style="list-style-type: none"> • Altes Lager • Erzbaronenkaff • Oberwelt • Neues Lager • Psi-Camp • Magischer Ort | <ul style="list-style-type: none"> • Tag Standard • Tag Bedrohung • Tag Kampf • Nacht Standard • Nacht Bedrohung • Nacht Kampf |
| Indoor-Locations: <ul style="list-style-type: none"> • Minen • Höhlen | <ul style="list-style-type: none"> • Standard • Bedrohung • Kampf |
| Ingame, Unabhängig von der Location: <ul style="list-style-type: none"> • Betrunken • Wahnsinnig • Tot→Wiederbeleben | <ul style="list-style-type: none"> • Linear |
| Sonstiges: <ul style="list-style-type: none"> • Titelmusik • Menü • Credits • Diverse Cutscenes | <ul style="list-style-type: none"> • Linear • Bei Menü eventuell mehrere Stufen (z.B. Hauptmenü, Multiplayer, Optionen) |

2.2 Soundeffekte (SFX)

Soundeffekte kann man in folgende Gruppen aufteilen:

- Animations-SFX
- Untergrund-SFX
- Aktions-SFX
- Kollisions-SFX
- Zerstörungs-SFX
- Zauber-SFX
- Skript-SFX
- VOB-SFX

2.2.1 Animations-SFX

Solche SFX werden durch eine Animation einer Figur ausgelöst und werden dann einfach abgespielt. Dazu gehört das „Hepp!“ eines Absprunges, das angestrengte Stöhnen eines Gewichtheberaktes oder aber das „Oops!“ wenn man gegen die Wand läuft.

Da all diese Soundeffekte vom Design her untrennbar mit einer Animation verbunden sind und manche davon sogar Synchronisation mit der Animation benötigen, ist der beste Ort für die Einbindung solcher SFX die entsprechende MDS-Datei der Figur.

Hier könnte die Zuordnung erfolgen, indem für eine Animation genau wie `eventSwapMesh()` ein `eventSFX()` angegeben werden kann. Die Parameter dieses `eventSFX()` wären dann:

- Logischer SFX-Name: Wird an diesen Namen ein „WAV“ angehängt so ist dies auch gleichzeitig der Dateiname für das WAV-File im „DATA\SOUND\SFX“-Verzeichnis.
- Synchronisations-Frame: Nur nötig, wenn auch wirklich synchronisiert werden muß. Das ist aber oft garnicht nötig. Oft genügt es bei Beginn der Animation einfach den SFX abzuspielen.

So könnte z.B. die Einbindung des SFX für das Heben eines tragbaren Objektes (Mensch) aussehen falls der SFX sofort mit Beginn der Animation gestartet werden soll (Parameter „-1“):

```
ani("t_OGet_2_OStand" 1 "s_OStand" 0.1 0.1 M. "Item_Get.asc" F 21 36)
{
    *eventSFX(HUMAN_LIFT_OBJECT, -1)
}
```

oder aber:

```
ani("t_OGet_2_OStand" 1 "s_OStand" 0.1 0.1 M. "Item_Get.asc" F 21 36)
{
    *eventSFX(HUMAN_LIFT_OBJECT, 25)
}
```

falls der SFX beim Frame 25 gestartet werden soll. Dabei ist `HUMAN_LIFT_OBJECT` der logische Animationsname, dessen Verwendung weiter unten näher erläutert ist.

2.2.2 Untergrund-SFX

Diese Soundeffekte sind eine Sonderform der Animations-SFX. Sie werden auf die gleiche Weise eingebunden (im jeweiligen MDS), benötigen beim Auslösen aber noch die Information über den **Untergrund**. Um dies kenntlich zu machen, sollte das MDS-Kommando anders lauten: z.B. `eventSurfaceSFX()`

Nun muß festgestellt werden, auf welchem Material (Level oder VOB) sich die Figur befindet. Für diesen Zweck können die schon existierenden Materialgruppen des statischen Levels benutzt werden:

| Materialgruppe |
|------------------|
| Holz |
| Metall |
| Stein/Geröll |
| Sand |
| Gras |
| Dreck/Matsch |
| niedriges Wasser |
| : |

VORSICHT: Watet die Figur durch **niedriges Wasser**, so muß als Untergrund *Wasser* erkannt werden und nicht das Material des Wassergrundes.

Damit es auch Geräusche gibt, wenn die Figur **auf Objekten** wie z.B. Tischen oder auch falschen Böden (die bei Bedarf einfach wegklappen) geht/rennt, muß das Materialgruppen-Verfahren auch bei MOBs funktionieren.

Anwendung für die Schritt-SFX ist nicht nur **Gehen** und **Rennen**, sondern auch das **MOB-Schieben** sowie alle monsterspezifischen Fortbewegungsmethoden (z.B. Gobbo-Rollen).

Da sich Schrittgeräusche assymetrisch anhören können (ein schlurfendes Monster, ein hinkender Mensch), muß es möglich sein, für **Links** und **Rechts** unterschiedliche SFX angeben zu können. Das setzt voraus, daß die jetzige Auftrennung der Schritt-Animationen in rechts & links bestehen bleibt. Sollte diese Aufteilung (Halbschritte) aus technischen Gründen wieder verworfen werden, muß für die SFX eine andere Lösung gefunden werden!

So könnte z.B. die Einbindung eines SFX für den linken „Fußstapfer“ eines rennenden Menschen aussehen,

```
ani("t_Run_2_RunL" 1 "s_RunL" 0.0 0.0 M. "Run.asc" F 0 11)
{
    *eventSurfaceSFX(HUMAN_RUN_LEFT, 11)
}
```

wobei `HUMAN_RUN_LEFT` der logische SFX-Name und `11` der Synchronisationsframe ist.

Da man hier nicht einfach aus dem Logischen SFX-Namen ein Wave-File ermitteln kann, muß dies in irgendeiner Form in einer **Skriptdatei** (vielleicht „SFX.D“) geschehen. Dort muß dann `HUMAN_RUN_LEFT` und all den anderen Untergrund-SFXs je ein Wavefile für jede Materialgruppe zugeordnet werden.

So könnte ein Eintrag in der „SFX.D“ aussehen:

```

CONST STRING SFX_HUMAN_RUN_LEFT[MATGROUPS_MAX] =
{
    "human_run_left_wood.wav",
    "human_run_left_metal.wav",
    "human_run_left_stone.wav",
    "human_run_left_sand.wav",
    "human_run_left_gras.wav",
    "human_run_left_dirt.wav",
    "human_run_left_water.wav"
};

```

2.2.3 Aktions-SFX

Alle sonstigen Soundeffekte, die durch ein Kommando des Spielers ausgelöst werden, die aber beim besten Willen nicht mit Animationen verbunden werden können (z.B. Kramen im Inventory) müssen ebenfalls in einer **Skriptdatei** angegeben werden können. Hier sollte wieder die gleiche Datei wie oben („SFX.D“) Verwendung finden.

So könnten Einträge in der „SFX.D“ aussehen, das Wechseln der Inventory-Kategorien hörbar machen:

```

CONST STRING SFX_INV_CAT_LEFT = "inv_cat_left.wav";
CONST STRING SFX_INV_CAT_RIGHT = "inv_cat_right.wav";

```

2.2.4 Kollision-SFX

Eine Kollision findet immer dann statt, wenn ein bewegliches Objekt (Figur, Item, Object → ich nenne es mal den *Verursacher*) auf irgendwas anderes (Figur, Item, Object, statisches Level → sagen wir einfach mal: der *Beteiligte*) in einer vom Versacher ausgelösten *Spielsituation* aufeinander trifft.

Kommt es zu einer Kollision, läuft (SFX-technisch) folgendes ab:

- Die Kategorie von *Verursacher* und *Beteiligter* werden jeweils festgestellt (Figur, Item, Object oder Level)
- Die *Spielsituation* wird festgestellt (Verursacher kollidiert („_HITS_“) einfach nur oder aber attackiert („_ATTACKS_“) den Beteiligten).
- Nun wird mit diesen drei Informationen eine von 20 Subtabellen ausgewählt, um darin den richtigen Kollisionssound zu ermitteln. Diese Subtabellen sind nach dem Schema <VERURSACHER>_HITS_<BETEILIGTEN> bzw. <VERURSACHER>_ATTACKS_<BETEILIGTEN> benannt:

•

| Beteiligter | Verursacher FIGUR | Verursacher ITEM | Verursacher OBJECT |
|--------------|--------------------|-------------------|---------------------|
| Figur | • MODEL_HITS_MODEL | • ITEM_HITS_MODEL | • OBJECT_HITS_MODEL |

| | | | |
|---------------|---|---|--|
| | MODEL_ATTACKS_MODEL | <ul style="list-style-type: none"> ITEM_ATTACKS_MODEL | |
| Item | <ul style="list-style-type: none"> MODEL_HITS_ITEM MODEL_ATTACKS_ITEM | <ul style="list-style-type: none"> ITEM_HITS_ITEM ITEM_ATTACKS_ITEM | <ul style="list-style-type: none"> OBJECT_HITS_ITEM |
| Object | <ul style="list-style-type: none"> MODEL_HITS_OBJECT MODEL_ATTACKS_OBJECT | <ul style="list-style-type: none"> ITEM_HITS_OBJECT ITEM_ATTACKS_OBJECT | <ul style="list-style-type: none"> OBJECT_HITS_OBJECT |
| Level | <ul style="list-style-type: none"> MODEL_HITS_LEVEL MODEL_ATTACKS_LEVEL | <ul style="list-style-type: none"> ITEM_HITS_LEVEL ITEM_ATTACKS_LEVEL | <ul style="list-style-type: none"> OBJECT_HITS_LEVEL |

- Jede dieser 20 Subtabellen sollte wieder in einer Skriptdatei abgelegt werden, am besten auch wieder in der „SFX.D“. Jede dieser Subtabellen ist 2D hat aber unterschiedliche Spalten- und Zeilenanzahlen. Doch es gibt Gemeinsamkeiten:

| Schema | Spaltenbeschriftung | Zeilenbeschriftung | Beispiel |
|--------------------|----------------------|--------------------|---|
| MODEL_HITS_... | Rüstung | - | Fleisch, Leder, Metall,... |
| MODEL_ATTACKS_... | Natürliche Waffenart | - | Faust, Fuß, Maul, Schwanz, ... |
| ITEM_HITS_... | Item-Material | - | Holz, Metall, Stein, Fleisch, Leder ... |
| ITEM_ATTACKS_... | Waffenkategorie | - | Dolch, Schwert, Axt, 2H-Schwert, 2H-Axt, Schild, Bogen, Armbrust. |
| OBJECT_HITS_... | Objekt-Material | - | Holz, Metall, Stein, Fleisch, Leder ... |
| ... | | | |
| ..._HITS_MODEL | - | Rüstung | Fleisch, Leder, Metall,... |
| ..._ATTACKS_MODEL | - | Trefferart | Schaden, Abpraller/Fleisch, Abpraller/Leder, Abpraller/Metall |
| ..._HITS_ITEM | - | Item-Material | Holz, Metall, Stein, Fleisch, Leder ... |
| ..._ATTACKS_ITEM | - | Item-Material | Holz, Metall, Stein, Fleisch, Leder ... |
| ..._HITS_OBJECT | - | Objekt-Material | Holz, Metall, Stein, Fleisch, Leder ... |
| ..._ATTACKS_OBJECT | - | Objekt-Material | Holz, Metall, Stein, Fleisch, Leder ... |
| ..._HITS_LEVEL | - | Level-Material | Holz, Metall, Stein, Gras, Sand, Dreck, Wasser, ... |
| ..._ATTACKS_LEVEL | - | Level-Material | Holz, Metall, Stein, Gras, Sand, Dreck, Wasser, ... |

Puhh, das ist alles ganz schön undurchsichtig und trocken, aber an den folgenden Beispielen wird alles deutlicher :-)

Schauen wir uns mal die Fälle ITEM_HITS_OBJECT sowie ITEM_ATTACKS_OBJECT. Die folgenden zwei Subtabellen zeigen, wie Zeilen und Spalten nach dem obigen Verfahren beschriftet und mit beliebigen WAV-Dateinamen befüllt werden könnten:

| ITEM_HITS_OBJECT | Holz | Metall | Stein | Fleisch | Leder |
|------------------|------------------|-------------------|-------------------|-------------------|-------------------|
| Holz | wood_wood_1.wav | wood_metal_1.wav | wood_stone_1.wav | wood_stone_1.wav | wood_stone_1.wav |
| Metall | wood_metal_1.wav | metal_metal_1.wav | metal_stone_1.wav | metal_stone_1.wav | metal_stone_1.wav |

| | | | | | |
|----------------|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | 2.wav | 1.wav | 1.wav | 1.wav | 1.wav |
| Stein | wood_stone_ 1.wav | metal_stone_ 3.wav | stone_stone_2 .wav | stone_stone_2 .wav | stone_stone_2 .wav |
| Fleisch | flesh.wav | flesh.wav | flesh.wav | flesh.wav | flesh.wav |
| Leder | leather.wav | leather.wav | leather.wav | leather.wav | leather.wav |

| ITEM_ATTACKS_OBJECT | Dolch | 1H-Schwert | 1H-Axt | 2H-Schwert | 2H-Axt | Schild | Bogen | Armbrust |
|---------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Holz | dagger_wood.wav | 1hs_wod.wav | axe_wod.wav | 2hs_wod.wav | axe_wod.wav | shield_wood.wav | arrow_wood.wav | arrow_wood.wav |
| Metall | dagger_metal.wav | 1h_meta1.wav | 1h_meta1.wav | 2h_meta1.wav | 2h_meta1.wav | shield_metal.wav | arrow_metal.wav | arrow_metal.wav |
| Stein | weapon_stone.wav | weapon_stone.wav | weapon_stone.wav | weapon_stone.wav | weapon_stone.wav | weapon_stone.wav | weapon_stone.wav | weapon_stone.wav |
| Gras | : | : | : | : | : | : | : | : |
| Sand | : | : | : | : | : | : | : | : |
| Dreck | : | : | : | : | : | : | : | : |
| Wasser | : | : | : | : | : | : | : | : |

Hier noch eine Beispieltabelle für den Fall MODEL_ATTACKS_MODEL.

| MODEL_ATTACKS_MODEL | Faust | Fuß | Maul | Schwanz |
|----------------------------|-------------------|-------------------|------------------|-------------------|
| Schaden kommt durch | fist_damage.wav | foot_damage.wav | jaw_damage.wav | tail_damage.wav |
| Abpraller | fist_repelled.wav | foot_repelled.wav | jaw_repelled.wav | tail_repelled.wav |
| Abpraller/Leder | fist_repelled.wav | foot_repelled.wav | jaw_repelled.wav | tail_repelled.wav |
| Abpraller/Metall | fist_repelled.wav | foot_repelled.wav | jaw_repelled.wav | tail_repelled.wav |

2.2.4.1 Item-/Objekt-Material

Diese Materialien haben nichts mit den schon existierenden Level-Materialien (auf Polygonebene) zu tun. Vielmehr muß jedem Objekt, sowie jedem Item EIN Material zugewiesen werden (per Skriptkonstante). Eine Holzkiste mit auftexturierten Eisenbeschlägen würde dann z.B. einfach MAT_WOOD erhalten. Ein Schwert MAT_METAL, eine Statue MAT_STONE. Anhand dieses Materials kann dann nicht nur der Kollisions-SFX sondern auch (nur für Objekte) der Zerstörungs-SFX und vielleicht sogar noch andere Eigenschaften (Art des Partikeleffekts bei Treffer und Zerstörung) festgelegt werden.

2.2.5 Zerstörungs-SFX

Während man das „Aua“ oder „Argh!“ einer verletzten/sterbenden Figur gut an die entsprechende Animation koppeln kann (Animations-SFX), ist dies bei zerstörten Objekten nicht möglich. Da für die Kollisions-SFX von und mit Objekten das Objekt-Material (Holz, Metall, Stein, ...) herangezogen wird, wäre es auch sinnvoll die Objekt-Zerstörungs-SFX so aufzurastern. Daß heißt, daß es für jedes Objekt-Material genau diese Unterteilung gibt. Der Vorteil: Man muß nicht an jede Objekt-Instanz ein Zerstört-SFX anhängen, sondern kann das passende Zerstört-SFX automatisch durch das Objekt-Material ermitteln.

So könnte dann zum Beispiel der entsprechende Eintrag in der SFX.D aussehen:

So könnte ein Eintrag in der „SFX.D“ aussehen:

```
CONST STRING SFX_OBJECT_DESTRUCTION[OBJMAT_MAX] =
{
    "wood_destruction.wav",
    "metal_destruction.wav",
    "stone_destruction.wav",
    „flesh_destruction.wav“,
    „leather_destruction.wav“,
    „clay_destruction.wav“
};
```

2.2.6 Zauber-SFX

Dies ist eine Variante der Aktions-SFX. Da die SFXs der Zauber sehr speziell vom jeweiligen Zauber abhängen, kann dies wohl nur zauberspezifisch festgelegt und dann hardgecodet werden. Zur Vereinfachung der Datenerstellung und -pflege könnte man aber zumindest für jeden Zauberspruch eine Stringtabelle in die SFX.D aufnehmen, die einfach durchnummeriert die verschiedenen SFX des Spruchs enthält.

So könnte vielleicht die Stringtabelle für die SFX des Feuerballs aussehen.

```
CONST STRING SFX_FIREBALL[7] =
{
    „fireball_logo.wav“,           // im Magieauswahlkranz-Fokus
    „fireball_select.wav“,        // im Auswahlkranz selektiert
    „fireball_grows.wav“,         // Wachstumsphase
    „fireball_shot.wav“,          // wird abgeschossen
    „fireball_flies.wav“,         // Fluggeräusch
    „fireball_explodes.wav“,      // der BUMM
    „fireball_canceled.wav“       // Geräusch, wenn Magier beim
casten verletzt wird und der wachsende Feuerball abgebrochen wird
};
```

Die Tabelle für den Lichtspruch ist dann vielleicht nur zwei Einträge groß.

2.2.7 Skript-SFX

Dies ist eine sehr einfache Form der SFX-Ansteuerung und wird benötigt, um Skript-Animationen oder später auch Cutscenes auf die gleiche Weise auch mit SFXs zu versorgen. Dazu ist einfach nur ein Skriptkommando wie z.B.

```
PlaySFX(NSC_DEMON7, „demon_moan.wav“);
```

nötig, welches einen SFX abspielt. Dabei muß neben dem Quell-VOB nur noch der SFX-Dateiname angegeben werden.

2.2.8 VOB-SFX

Um VOBs, die ständig und vor allem von selbst Geräusche von sich geben, zu ermöglichen (Fackel, Lagerfeuer, Sprinklerbrunnen, Wasserfall, ...) sollte man jedem VOB einen SFX zuweisen können. Dies wäre besser, als spezielle SFX-VOBs zu erstellen und an das zu vertonende Objekt anhängen zu müssen.

2.2.9 Allgemeines zu SFX

2.2.9.1 Mehrfach-SFX

Um häufig auftretende Spielsituationen nicht mit ewig gleichen und deshalb tierisch nervenden SFXs zu vermiesen, hier ein supereinfach zu handhabendes System für Mehrfach-SFX:

Immer wenn nach einer SFX-Datei wie z.B. „human_lift_object.wav“ gesucht wird, wird auch überprüft ob es diesen Dateinamen mit angehängten Zahlen gibt. Gibt es bei unserem Beispiel auch „human_lift_object_2.wav“ gesucht. Ist dieser SFX vorhanden, so wird der tatsächlich abgespielte SFX aus diesen beiden zufällig ausgewählt. Ist auch noch „human_lift_object_3.wav“ vorhanden, so wird aus drei ausgewählt.

Der Vorteil dieses Systems: Die Soundtechniker können jederzeit neue Varianten in des \SOUND\SFX-Verzeichnis legen, ohne daß auch nur eine Zeile Quellcode oder Skript geändert werden muß.

2.2.9.2 Pitch

Neben dem Mehrfach-Ablegen von SFX kann auch eine zufällige leichte Frequenzveränderung zur Abwechslung beitragen. Diese Frequenzänderung von z.B. +40Hz wird als Zahl in den Dateinamen hineincodiert. In diesem Beispiel würde der SFX „human_lift_object_3.wav“ dann „human_lift_object_3_40.wav“ heißen.

2.2.9.3 verschiedene Stimme

Da es bei allen SFX, die den Stimmbändern eines Menschen entspringen verschiedene Stimmvarianten geben muß, sollte auch die Stimme, ähnlich wie schon die Mehrfach-SFX in den Dateinamen hineincodiert werden.

2.2.10 Technische Evaluation

Die erste Stufe der SFX-Integrierung ist die technische Evaluation:

- Soundsystem GALAXY von Carlo Vogelsang? Wenn ja, welche Version?
- Dateiformat für SFX (WAV?)
- Abspielen von SFX mit individueller Lautstärke (Lautstärke-Faktor um die WAV-Lautstärke zu „skalieren“)

- Leiser werden bei Entfernen der Kamera von der Geräuschquelle
- Stereo-Effekt (linker/rechter Lautsprecher) bei Drehung der Kamera
- Probleme durch mehrere gleichzeitige SFX (Mehrkanal), Abspielperformance (Ruckler, SFX-Aussetzer)
- Hall-Möglichkeiten abschätzen
- Echtzeitmodifikation der Samples (verschiedene Stimmen mit gleichen Dateien?)
- 3D-Sound: 3D-Dolby-Surround, Q-Sound, Support 3D-Hardware (Aureal?)

2.2.11 Beispiele

| Thema | Situation | Beispiel | SFX-Art |
|---------------------------------------|---|------------------------------------|------------------------|
| <i>Bewegung:</i> | | | |
| Gehen/Rennen vor/seitl./zurück/drehen | auf verschiedenen Untergründen (Stein, Holz, Matsch, Gras, niedriges Wasser, halbhohes Wasser) → Sync! | „Tipp, tapp“ | Untergrund |
| Kollision | ...mit Wand beim Rennen/Springen/Fallen | „Boing!“ (Wand) & „Oops!“ (SC) | Kollision & Animation |
| Hinfallen | ...mit Schaden und Liegenbleiben | „Boing!“ (Wand) & „Argh!“ (SC) | Kollision & Animation |
| Springen vor/hoch | Abspringen | „Luftkomprimieren“ | Animation |
| Springen vor/hoch | Aufkommen | | Kollision |
| Springen halbhoch | JumpTable | „Hepp!“ | Animation |
| Springen seitlich | | „Hepp!“ | Animation |
| Springen zurück | | „Hepp!“ | Animation |
| Klettern | beim Ergreifen der Kante | „kurzes kratzendes Geräusch“ | Animation |
| Klettern | beim sich Hochziehen | „Stöhnen!“ | Animation |
| Wasser | Hineinspringen/-fallen | „Platsch!“ | Kollision |
| Wasser | Herausklettern | „Tröpfel!“ | Animation |
| Wasser | Schwimmen | „strömendes Wasser“ | Animation |
| Wasser | Tauchen | „strömendes Wasser“ aber dumpfer | Animation |
| <i>Action:</i> | | | |
| Item | Aufheben | „kurzes kratzendes Geräusch“ | Animation |
| Item | Werfen | „hörbarer starker Luftzug“ | Animation |
| Item | Fackel werfen | „lodernde Flamme“ | Hardcoded |
| Item | kollidiert mit Level (Ablegen, Werfen, Schießen) | | Kollision |
| MOB | Aufheben | „erfolgrch. Gewichtheber-Stöhnen“ | Animation |
| MOB | Aufheben, aber MOB zu schwer | „erfolgloses Gewichtheber-Stöhnen“ | Animation |
| MOB | Werfen | „angestregtes Luftpressen“ | Animation |
| MOB | Schieben/Ziehen | „Gewichtheber-Stöhnen + Poltern“ | Animation & Untergrund |
| MOB | Schieben/Ziehen, aber MOB zu schwer | „erfolgloses Gewichtheber-Stöhnen“ | Animation |

| | | | |
|---|--|-------------------------------------|------------|
| MOB | kollidiert mit Level (Werfen) | | Kollision |
| MOB | Benutzen von MOBs (z.B.: öffnen/schließen) | (var) | Skript |
| <i>Inventory:</i> (des SCs, von Leichen und von Container-MOBs) | | | |
| Inventory | Aufrufen | „Rucksackklappe auf“ | Aktion |
| Inventory | Schließen | „Rucksackklappe zu“ | Aktion |
| Inventory | Kategorie wechseln | „kurzes Herumkramen im Rucksack“ | Aktion |
| Item | Equippen | „Festschnallen/-zurren“ | Aktion |
| Item | Benutzen | (var) | Skript |
| Item | ablegen aus dem Inventory | „kurzes Herumkramen im Rucksack“ | Animation |
| <i>Kampf:</i> | | | |
| Waffenauswahl | beim Wechsel auf Nah-/Fernkampfwaffe | „Herumnesteln an Lederbändern“ | Animation |
| Waffe ziehen | Schwert/Dolch | „Metallklinge aus Metallscheide“ | Animation |
| Waffe ziehen | sonstige Nahkampfwaffe/Fernkampfwaffe | „Holzgriff aus Lederschlaufe“ | Animation |
| Waffe wegstecken | Schwert/Dolch | „Metallklinge in Metallscheide“ | Animation |
| Waffe wegstecken | sonstige Nahkampfwaffe/Fernkampfwaffe | „Herumnesteln an Lederbändern“ | Animation |
| Schlag | Sync!: wird bei Treffer unterbrochen | „Luftzug“ | Animation |
| Schlag | aber 2H-Waffe zu schwer | „Luftzug mit vorherigen Stöhnen“ | Animation |
| Bogenschuß | | „Zosch!“ | Animation |
| Armbrustschuß | | „Zock!“ | Animation |
| Schlag/Schuß | spezielle/magische/individuelle Waffe | (var) | Skript |
| Treffer | auf NSC, unabhängig von Waffe | | Kollision |
| Treffer | auf Item/MOB | | Kollision |
| Treffer | auf statisches Level | | Kollision |
| Parade | Schwert/Axt/Dolch auf Schwert/Axt/Dolch | „metallisches Klöng“ | Animation |
| Parade | alle anderen Fälle (auch Holz auf Metall) | „hölzernes Klock“ | Animation |
| Parade | Waffe auf Schild | „Schild-Klong“ | Animation |
| Kampfsteps vor/seitl./zurück | Sync!: wie normale Schritte nur anders synchronisiert | | Untergrund |
| <i>Magie:</i> | | | |
| Waffenauswahl | beim Wechsel auf Magieauswahlkranz | | Zauber |
| Zauberauswahl | beim Rotieren des Magieauswahlkranzes | | Zauber |
| Zauber sprechen | je nach Zauber völlig individuell | | Zauber |
| Zauber sprechen | wenn kein Mana/Wille mehr übrig | „zersplitterndes Glas“ | Zauber |
| <i>Selbständig:</i> | | | |
| Item | gibt regelmäßige Töne von sich (z.B. Fackel, Flammenschwert) | | VOB |
| MOB | gibt regelmäßige Töne von sich (z.B. Lagerfeuer, Brunnen, Wasserfall) | | VOB |

2.3 Sprache

- alles vertont ?

- Satzvariablen müssen...
 - ...der Betonung wegen stets am Anfang oder am Ende eines Satzes stehen
 - ...im Nominativ verwendet werden
 - ...den bestimmten Nominativ-Artikel „der/die/das“ tragen
- Liste aller Sätze (Herkunft/Kategorie erfassen → Grundlage Lokalisationstool)

3. Soundlisten

3.1 Animations-SFX

| SFX-Datei | Animation | Anmerkung |
|-----------------|--|---|
| HUMAN | | |
| Jump | Springen | |
| Hang | An Kante festhalten | Kurzes kratzendes Geräusch |
| ClimbUp | Hochklettern | Starkes Ausatmen |
| ClimbDown | | |
| IGet | Item aufheben | kratzendes Geräusch (je nach Material?) |
| IThrow | Item werfen | Starker Luftzug |
| IDrop | Item ablegen (fallenlassen) | (evtl Kollisions-Sound?) |
| OGet | | Sehr starkes Ausatmen |
| OThrow | | Kurzes Atmen beim Anheben |
| ODrop | | |
| Oops | Mit Hindernis kollidieren | „Uff“ (zusätzlicher Kollisions-sound?) |
| Stumble | Rückwärts-Stolpern nach Treffer | Schritte |
| Slide | Rutschen auf Schrägen (Loop) | (je nach Untergrundmaterial?) |
| Fall | Tiefer Fall | Schrei |
| Fallen | Aufprall auf dem Boden | (Kollisions-sound?) |
| GetUp | Aufstehen | |
| Swim | | |
| Swim2Dive | Untertauchen | Stärkeres Plätschern |
| Dive | Tauchen | Ambient Sounds |
| Drown | Ertrinken | Gurgelndes Geräusch |
| Swim2Hang | An Kante festhalten | Starkes Plätschern |
| t_Hang_2_Stand | An Kante hochziehen aus dem Wasser | !gleiche Ani wie an Land |
| Die | | Todesschrei, Umfallen |
| DieAssassinate | | Gurgelnde Laute |
| TorchToggle | Fackel von einer in die andere Hand wechseln | |
| TorchThrow | Fackel werfen (Flamme im Luftzug) | !gleiche Ani wie alle Items |
| Ready1hS | 1h-Schwert ziehen | |
| Ready1hA | 1h-Axt ziehen | |
| Ready2hS | 2H-Schwert ziehen | |
| Ready2hA | 2H-Axt ziehen | |
| ReadyBow | | |
| Ready Crossbow | | |
| Ready Fist | | |
| 1hAttack | | |
| 1hAttack talent | | |
| 1hParade | | |

| | | |
|--------------------|------------------------------------|---|
| 2hAttack | | |
| 2hAttack_talent | | |
| 2hParade | | |
| BowShoot | | |
| BowReload | | |
| CrossbowShoot | | |
| CrossbowReload | | |
| Hurt | | Schmerzensschrei |
| Surface | | Luftschnappen |
| GOBBO | | |
| GOB_Jump | Springen | |
| GOB_Hang | An Kante festhalten | Kurzes kratzendes Geräusch |
| GOB_ClimbUp | Hochklettern | Starkes Ausatmen |
| GOB_ClimbDown | | |
| GOB_IGet | Item aufheben | kratzendes Geräusch (je nach Material?) |
| GOB_IThrow | Item werfen | Starker Luftzug |
| GOB_IDrop | Item ablegen (fallenlassen) | (evtl Kollisions-Sound?) |
| GOB_Oops | Mit Hindernis kollidieren | „Uff“ (zusätzlicher Kollisions-sound?) |
| GOB_Stumble | Rückwärts-Stolpern nach Treffer | Schritte |
| GOB_Slide | Rutschen auf Schrägen (Loop) | (je nach Untergrundmaterial?) |
| GOB_Fall | Tiefer Fall | Schrei |
| GOB_Fallen | Aufprall auf dem Boden | (Kollisions-sound?) |
| GOB_GetUp | Aufstehen | |
| GOB_Swim | | |
| GOB_Swim2Dive | Untertauchen | Stärkeres Plätschern |
| GOB_Dive | Tauchen | Ambient Sounds |
| GOB_Drown | Ertrinken | Gurgelndes Geräusch |
| GOB_Swim2Hang | An Kante festhalten | Starkes Plätschern |
| GOB_t_Hang_2_Stand | An Kante hochziehen aus dem Wasser | !gleiche Ani wie an Land |
| GOB_Die | | Todesschrei, Umfallen |
| GOB_DieAssassinate | | Gurgelnde Laute |
| GOB_1hAttack | | |
| GOB_1hParade | | |
| GOB_Alert | schlägt Alarm | |
| GOB_Fly | fliegt durch die Luft | |
| GOB_Fun | freut sich über Item | |
| GOB_Hurt | | Schmerzensschrei |
| GOB_Surface | | Luftschnappen |
| | | |

3.2 Untergrund-SFX

| SFX-Datei | Animation | Anmerkung |
|--------------|-----------|-----------|
| HUMAN | | |
| Walk | Gehen | |
| RUN_EARTH | | |
| RUN_METAL | | |
| RUN_STONE | | |
| RUN_WATER | | |
| RUN_WOOD | | |
| RUN_UNDEF | | |
| STRAFE_EARTH | ? | |

| | | |
|------------------|---------------|------------|
| STRAFE_METAL | ? | |
| STRAFE_STONE | ? | |
| STRAFE_WATER | ? | |
| STRAFE_WOOD | ? | |
| STRAFE_UNDEF | ? | |
| Sneak | Schleichen | Sehr leise |
| Push | MOB Schieben | |
| Pull | MOB Ziehen | |
| | | |
| GOBBO | | |
| GOB_Walk | | |
| GOB_Run_Earth | | |
| GOB_Run_Metal | | |
| GOB_Run_Stone | | |
| GOB_Run_Water | | |
| GOB_Run_Wood | | |
| GOB_Run_Undef | | |
| GOB_Strafe_Earth | ? | |
| GOB_Strafe_Metal | ? | |
| GOB_Strafe_Stone | ? | |
| GOB_Strafe_Water | ? | |
| GOB_Strafe_Wood | ? | |
| GOB_Strafe_Undef | ? | |
| GOB_Sneak_Earth | Rollanimation | leise |
| GOB_Sneak_Metal | Rollanimation | leise |
| GOB_Sneak_Stone | Rollanimation | leise |
| GOB_Sneak_Water | Rollanimation | leise |
| GOB_Sneak_Wood | Rollanimation | leise |
| GOB_Sneak_Undef | Rollanimation | leise |

3.3 Aktions-SFX (erst mal komplett ignorieren!)

| SFX-Datei | Spielsituation | Anmerkung |
|---------------|-------------------------------------|-----------------------|
| INV_OPEN | Inventory öffnen | |
| INV_CLOSE | Inventory schließen | |
| INV_CHANGE | Inventorykategorie wechseln | |
| | | |
| TRADE_OPEN | Handelsscreen öffnen | |
| TRADE_CLOSE | Handelsscreen schließen | |
| | | |
| PROFILE_OPEN | SC-Profil öffnen | |
| PROFILE_CLOSE | SC-Profil schließen | |
| | | |
| ASK_YES | JA-NEIN-Fragebox: Daumen nach oben | |
| AKS_NO | JA-NEIN-Fragebox: Daumen nach unten | |
| | | |
| HELP_ON | Tastaturhilfe-Fenster einblenden | |
| HELP_OFF | Tastaturhilfe-Fenster einblenden | |
| | | |
| - | (Optionen) | Spiel laden/speichern |
| - | (Hauptmenü) | |

3.4 Kollisions-SFX

| SFX-Datei | Spielsituation | Anmerkung |
|-----------|---|---------------------------------------|
| MHM_FL_FL | Figur rennt in Figur | dumpfer Aufprall |
| MHM_ME_ME | gerüstete Figur kracht in gerüstete Figur | „Schepper!“ |
| MAM_FI_DM | Faustattacke verursacht Schaden | dumpfer Schlag mit „innerem Knacken“ |
| MAM_FI_BM | Faustattacke prallt von Metallrüstung ab | dumpfes „Klong“ |
| MAM_JA_DM | Beißattacke verursacht Schaden | Zähne graben sich satt ins Fleisch |
| MAM_JA_BM | Beißattacke prallt von Metallrüstung ab | Zähne ratzen unangenehm über Metall |
| MAM_FI_BL | Faustattacke prallt von Lederrüstung ab | |
| MHI_FL_WO | Figur rennt gegen Gegenstand | dumpfer, dezenter Aufprall |
| MAI_FI_WO | Faustattacke auf Gegenstand | dumpfer, dezenter Aufprall |
| MHO_FL_WO | Figur rennt gegen Objekt | dumpfer Aufprall |
| MAO_FI_WO | Faustattacke auf Objekt aus Holz | dumpfer, hölzerner Aufprall |
| MAO_FI_FL | Faustattacke auf Objekt aus Fleisch | matschiger Aufprall |
| MHL_FL_ST | Figur rennt gegen Wand | dumpfer Aufprall |
| MHL_ME_ST | gerüstete Figur rennt gegen Wand | scheppernder Aufprall |
| MHL_FL_WA | Figur fällt ins Wasser | großes, breites „Platsch!“ |
| MAL_FI_ST | Faustattacke gegen Felswand | sehr dumpfes und leises Geräusch |
| MAL_FI_WO | Faustattacke gegen Holzwand | hölzerner Aufprall |
| MAL_FI_WA | Faustattacke ins Wasser | „Schlag ins Wasser“ wörtlich genommen |
| MAL_FI_ME | Faustattacke gegen Metallwand | |
| IHM_WO_FL | Gegenstand fliegt gegen Figur | dumpfer Aufprall |
| IHM_WO_ME | Gegenstand fliegt gegen gerüstete Figur | scheppernder Aufprall |
| IHM_ST_ME | Stein fliegt gegen gerüstete Figur | |
| IAM_BO_DM | Pfeil verwundet Figur | |
| IAM_1H_DM | Klinge verwundet eine Figur | Klinge gräbt sich satt ins Fleisch |
| IAM_2H_DM | Klinge verwundet eine Figur | Klinge gräbt sich satt ins Fleisch |
| IAM_1H_BL | Klinge prallt an Lederrüstung einer Figur ab | dumpfes Abgleiten |
| IAM_1H_BM | Klinge prallt an Metallrüstung einer Figur ab | schepperndes, schleifendes Abgleiten |
| IAM_2H_BM | Klinge prallt an Metallrüstung einer Figur ab | schepperndes, schleifendes Abgleiten |
| IAM_BO_BM | Pfeil prallt an Metallrüstung einer Figur ab | |
| IHI_WO_WO | zwei kleine Holzgegenstände kollidieren | sehr dezentes, hölzernes „Tock!“ |
| IAI_1H_BW | | |
| IAI_1H_WO | Klinge attackiert kleinen Holzgegenstand | dezentes, hölzernes „Zack!“ |
| IAI_ME_ME | Klinge attackiert kleinen Metallgegenstand | |
| IHO_WO_WO | kleines Holzitem trifft großes Holzobjekt | deutliches, hölzernes „Tock!“ |
| IHO_BO_ME | Pfeil trifft größeres Metallobjekt | |
| IAO_1H_WO | Klinge attackiert großes Holzobjekt | sattes, hölzernes „Zack!“ |
| IHO_ST_ME | Stein trifft größeres Metallobjekt | |
| IHO_ME_ME | Klinge trifft größeres Metallobjekt | |
| IHL_ST_WO | Stein fliegt gegen Holzwand | hölzernes Aufprallgeräusch |
| IHL_ST_ST | Stein fliegt gegen Steinwand | dezentes „Klok!“ |
| IHL_ST_WA | Stein fällt ins Wasser | dezentes „Platsch!“ |
| IHL_WO_WA | kleines Holzitem fällt ins Wasser | |
| IAL_1H_ST | Klinge attackiert Steinwand | vibrierendes „Kläng!“ |
| IAL_2H_ST | große Klinge attackiert Steinwand | vibrierendes „Kläng!“ |
| IAL_1H_WO | Klinge attackiert Holzwand | hölzernes „Tock!“ |

| | | |
|-----------|--|---|
| IAL_1H_WA | Klinge haut ins Wasser | kleines, dezentes „Platsch!“ |
| IAL_BO_ST | Pfeil trifft Steinwand | zerbröselt mit dezentem Geräusch |
| IAL_BO_WO | Pfeil trifft Holzwand und bleibt stecken | hölzernes „Tock!“ |
| IAL_BO_ME | Pfeil trifft Metallwand | |
| IAO_BO_GL | Pfeil trifft Flasche / Krug | |
| IAO_1H_WO | | |
| IAO_ME_WO | | |
| IAO_ME_ME | | |
| IAO_BO_ME | | |
| IAO_2H_ME | | |
| OHM_WO_FL | großes Holzobjekt trifft Figur | dumpfes „Dumm!“ |
| OHM_WO_ME | großes Holzobjekt trifft gerüstete Figur | schepperndes „Dong!“ |
| OHI_ST_WO | großer Felsbrocken trifft kleines Holzitem | |
| OHI_WO_WO | großes Holzobjekt trifft kleines Holzitem | sehr dezentes, hölzernes „Tock!“ |
| | | |
| OHO_WO_WO | zwei große Holzobjekte treffen aufeinander | lautes, hölzernes „Tock!“ |
| OHO_ST_ST | zwei große Felsbrocken treffen aufeinander | lautes „Klock!“ |
| OHO_ST_WO | großer Fels trifft großes Holzobjekt | |
| OHL_WO_ST | großes Holzobjekt trifft auf Steinwand | lauter Aufprall mit etwas Holzsplittern |
| OHL_ST_ST | Felsbrocken trifft auf Steinwand | lautes, schabendes „Klock!“ |
| OHL_ST_WA | Felsbrocken fällt ins Wasser | lautes, sattes „Platsch!“ |
| OHL_WO_WA | großes Holzobjekt fällt ins Wasser | |
| OHL_WO_DI | großes Holzobjekt fällt auf Erdboden | sehr tiefes und sehr dumpfes „Fump!“ |

3.5 Zerstörungs-SFX

| SFX-Datei | Spielsituation | Anmerkung |
|---------------|--|-----------|
| CRASH_WOOD | z.B. für Tisch, Holztür, Stützbalken,... | |
| CRASH_CLAY | z.B. für Statue, Krug | |
| CRASH_STONE | z.B. für Statue, Findling | |
| CRASH_METAL | z.B. Metalltür | |
| CRASH_FLESH | z.B. Leichen, Körperteile | |
| CRASH_LEATHER | z.B. Ledervorhänge, | |
| CRASH_JAR | z.B. Tonkrüge | |
| CRASH_MINE | z.B. Mega-Minen-Einsturz | |
| CRASH_ITEM | z.B. Itemcrash | |
| CRASH_BONE | | |
| CRASH_BIG | Mega-Crash | |

3.6 Zauber-SFX

| SFX-Datei | Spielsituation | Anmerkung |
|-------------------|--|-------------|
| LIGHT_LOGO | <i>Licht</i> im Fokus des Auswahlkranzes | |
| LIGHT_SELECTED | <i>Licht</i> wird im Auswahlkranz aktiviert | |
| LIGHT_CASTED | <i>Licht</i> wird gesprochen | |
| LIGHT_SPHERE | <i>Licht</i> : Das permanente Geräusch der Lichtsphäre | geloopt !!! |
| LIGHT_FADES | <i>Licht</i> : Zeitdauer läuft aus | |
| LIGHT_CANCELED | <i>Licht</i> : Zauberabbruch während Casten | |
| | | |
| FIREBALL_LOGO | <i>Feuerball</i> im Fokus des Auswahlkranzes | |
| FIREBALL_SELECTED | <i>Feuerball</i> wird im Auswahlkranz aktiviert | |
| FIREBALL_GROWS | <i>Feuerball</i> wird aufgeblasen | geloopt !!! |
| FIREBALL_SHOT | <i>Feuerball</i> : abgeschossen | |
| FIREBALL_FLIES | <i>Feuerball</i> : ...fliegt durch die Luft | geloopt !!! |

| | | |
|---------------------|---|---------------------------------|
| FIREBALL_EXPLODES | <i>Feuerball</i> : explodiert beim Aufprall | |
| FIREBALL_CANCELED | <i>Feuerball</i> : Zauberabbruch während Casten | |
| | | |
| CONTROL_LOGO | <i>Kontrolle</i> im Fokus des Auswahlkranzens | |
| CONTROL_SELECTED | <i>Kontrolle</i> wird im Auswahlkranz aktiviert | |
| CONTROL_STRUGGLE | <i>Kontrolle</i> : Zauber ringt mit dem Opfer | geloopt !!! |
| CONTROL_FAILED | <i>Kontrolle</i> : Struggle abgebrochen | Energie aus oder User-Interrupt |
| CONTROL_SUCCESS | <i>Kontrolle</i> : Opfer wird erfolgreich übernommen | Kameraflug zum Opfer |
| CONTROL_CONTROL | <i>Kontrolle</i> : Opfer gibt dezentes Geräusch von sich | geloopt !!! |
| CONTROL_RETURN | <i>Kontrolle</i> : Zauberer kehrt in seinen Körper zurück | freiwillig oder durch Treffer |
| | | |
| SHAPESHIFT_LOGO | <i>Verwandlung</i> im Fokus des Auswahlkranzens | |
| SHAPESHIFT_SELECTED | <i>Verwandlung</i> wird im Auswahlkranz aktiviert | |
| SHAPESHIFT_TOGGLE | <i>Verwandlung</i> : Neuer Gestaltumriß sichtbar | geloopt !!! |
| SHAPESHIFT_SHIFT | <i>Verwandlung</i> : Gestalt wird angenommen | |
| SHAPESHIFT_RETURN | <i>Verwandlung</i> : Zauberer verwandelt sich zurück | |
| | | |

3.7 Skript-SFX

| SFX-Datei | Spielsituation | Anmerkung |
|----------------|----------------------------------|---|
| CHEST_OPEN | Türe/Kiste/Schrank öffnet sich | Klacken, dann Knarzen |
| CHEST_CLOSE | Türe/Kiste/Schrank schließt sich | Knarzen mit abschließendem zuschnappen |
| BIGCHEST_OPEN | große Truhe öffnet sich | |
| BIGCHEST_CLOSE | große Truhe schließt sich | |
| LEVEL_USE | Hebel wird betätigt | Schleifgeräusche + „Tock“ für Erreichen der Endstellung |
| PORTCULLI_UP | Fallgitter hebt sich | Schleifgeräusche + Kettenrasseln |
| PORTCULLI_DOWN | Fallgitter senkt sich | Schleifgeräusche + Kettenrasseln |
| PORTCULLI_FALL | Fallgitter fällt schnell zu | Schleifgeräusche + Knall |
| STONEHEAP_USE | Stein aus Steinhaufen aufheben | nachrutschende Steine |
| KEY_OPEN | Schlüsselklappen und Schließen | |
| KEY_CLOSE | Schlüsselklappen und Schließen | |
| | | |

3.8 VOB-SFX

| SFX-Datei | VOB | Anmerkung |
|---------------|----------------------|-----------|
| TORCH_BURN | brennende Fackel | |
| CAMPFIRE_BURN | Lagerfeuer/Feueresse | |
| WATERFALL | Wasserfall | |
| FOUNTAIN | plätschernde Quelle | |
| GONG | Gong | |
| ANVIL | Amboß | |
| BALLISTA | | |
| KARAFFE | | |
| WATERWHEEL | | |
| WATERPUMP | | |
| DRAWBRIDGE | | |
| SARCOPHAG | steinerner Deckel | |

3.9 Gebiets-SFX

| SFX-Datei | Spielsituation | Anmerkung |
|---------------------|------------------------------------|-------------|
| AMBIENT_WINDYCAVE | Wind pfeift durch Höhlen | geloopt !!! |
| AMBIENT_UNDERWATER | Unterwasserhintergrundgeräusch | geloopt !!! |
| AMBIENT_NIGHTMEADOW | nächtliches Grillenkonzert | geloopt !!! |
| AMBIENT_DARKNIGHT | dunkle Geräusche in der Nacht | geloopt !!! |
| AMBIENT_SUNRISE | Vogelgezwitscher bei Sonnenaufgang | geloopt !!! |
| AMBIENT_RAIN | Regen | geloopt !!! |
| AMBIENT_MINE | diverse Minengeräusche | |
| | | |