

# **SFX-Workshop**

**COLLABORATORS**

	<i>TITLE :</i> SFX-Workshop		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 29, 2025	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SFX-Workshop</b>	<b>1</b>
1.1	2.1 Workshop	1
1.2	2.1.1 Generierung von Percussionsounds	2
1.3	2.1.2 Generierung von Synthesizersounds	3
1.4	2.1.2 Generierung von Effektsounds	4
1.5	2.1.4 komplexe Effekte	5
1.6	2.1.5 Samplebearbeitung	6

---

# Chapter 1

## SFX-Workshop

### 1.1 2.1 Workshop

#### 2.1 Workshop

-----

In nachfolgenden Kapiteln werden Sie anhand verschiedener Beispiele in die Arbeit mit SFX eingeführt.

- 2.1.1 Generierung von Percussionsounds
- 2.1.2 Generierung von Synthesizersounds
- 2.1.3 Generierung von Effektsounds
- 2.1.4 komplexe Effekte
- 2.1.5 Samplebearbeitung

Die fertigen Samples finden Sie teilweise im Verzeichnis "Workshop" in ihrem Programm-verzeichnis. ←

#### allgemeiner Tip

-----

- \* !!!! AUSPROBIEREN !!!! - es kann nichts kaputtgehen
- \* die Funktionsweise von SoundFX versteht man am besten durch die intensive Benutzung des Programms
- \* Nicht nur die Standarteinstellungen der Operatoren verwenden.  
Beispiel : PitchShifte
  - \* dieser wurde in seiner Qualität stark verbessert
  - \* allerdings ist der WinSize-parameter von entscheidender Bedeutung
  - \* bei tiefenen/dumpferen Samples sollter dieser etwas größer sein  
ca 400-1000
  - \* bei höheren/heller Klängen ca. 200-600
  - \* probieren Sie verschiedene Varianten aus
  - \* ungünstig gewählte Werte hört man daran, daß das Ergebnissample schwankt
- \* Nutzen Sie die Modulationsmöglichkeiten - einige Effekte weden nur dadurch so richtig wirkungsvoll z.B. Detune, Smear
  
- \* Letzter Hinweis :
  - \* wenn Sie Fragen haben - schreiben Sie mir / rufen Sie mich an !!!
  - \* nur so kann Ich sehen, wo etwas zu kompliziert beschrieben ist, wo noch

Schwachstellen sind.

- \* für Ideen, Anregungen, Informationen (Sampleformate) bin ich natürlich jederzeit dankbar

## 1.2 2.1.1 Generierung von Percussionsounds

better Basedrums

-----

- \* laden Sie eine Basedrum die ihnen nicht "bassig" genug ist
- \* starten Sie Synthesize\_Add
  - \* stellen Sie für die Länge den Längenwert des Basedrumsamples ein
  - \* stellen Sie für die Frequenz (Pitch) den niedrigsten Ton auf dem Keyboard ein (65. ...)
  - \* setzen Sie den zweiten Frequenzmodulationswert auf 0.01 (oder noch kleiner)
  - \* wählen Sie als Modus "slowdown"
  - \* setzen Sie den ersten Slider im Reglerfeld auf 1 (ganz nach oben).
  - \* lassen Sie sich das Sample generieren -> jetzt haben Sie einen tiefen Sinus der gegen Ende noch tiefer wird.
  - \* probieren Sie auch mal andere Wellenformen aus (z.B. Saw)
- \* starten Sie Amplify
  - \* Source : Sinus-Sample
  - \* stellen Sie für die Factoren 0 und 100 ein
  - \* Modulationsmodus : "User2"
  - \* Modulationssample : Basedrum-Sample
  - \* "Go" -> jetzt haben Sie die Hüllkurve der Basedrum für den Sinus übernommen.
- \* Mixen Sie jetzt beide Samples zusammen
  - \* Src1: Basedrum
  - \* Src2: Sinus
  - \* Mix : ca. 50-70 % (ausprobieren)

Basedrums

-----

- \* Synthesize\_Add aufrufen
  - \* len=4000
  - \* frq=130.8127....(C-1)
  - \* waveform=sin
  - \* 1. Slider auf max (=1)
  - \* Frequency=1.0 ... 0.0001, Slowdown
- \* ZeroPass
  - \* FadeIn=1
  - \* FadOut=500
- \* ADSR-Envelope
  - \* kurzer Attack
  - \* sonst Defaults
- \* OctaveUp
  - \* interpol=TRUE
- \* Shorten
  - \* treshold=20

```

* ZeroPass
* FadeIn=1
* FadeOut=200
* wer`s ein bisschen härter mag, ...
* Amplify
* amplification=115 ... 80, Linear
* oder
* amplification=180 ... 80, Linear, Wrap=Wrap2

```

#### HiHats

-----

```

* Noise aufrufen
* len=4000
* Amplify
* amplification=100 ... 0, Linear
* und nochmal
* amplification=100 ... 0, Slowdown
* Version 1 ist fertig
* für Version2 noch ein OctaveUp
* interpol=TRUE

```

#### Snaredrums

-----

```

* Noise aufrufen
* len=6000
* Amplify
* amplification=20 .. 20, None
* Synthesize_Add
* len=6000
* frq=130.8127....(C-1)
* waveform=sin
* 1. Slider auf max (=1)
* Frequency=1.0 ... 0.1, Slowdown
* Slide
* Source=leises Rauschen
* slidedistance=-25000 ... 25000, User 0, ModSource=Sinus
* ADSR-Envelope
* kurzer Attack
* ca. 2/3 der Hüllkurve sehr leise, dabei relativ abrupter
  Wechsel von laut zu leise
* Version1 ist fertig
* weiterhin ...
* Amplify
* amplification=150 ... 100, Linear, Wrap=Wrap2
* oder
* Filter-LowPass
* Effektanteil=-100 ... -100, None
* range/cut-off=9.0 ... 9.0, None

```

## 1.3 2.1.2 Generierung von Synthesizersounds

interessante Strings/Synths

-----

- \* sehr interessante Ergebnisse erhält man bei Flächenklängen wenn man diese mit dem :
  - \* HiPass-Filter bearbeitet (CutOff ca. 3-10)
    - \* mit Amplify danach wieder auf MaxVol bringen
    - \* Klänge sind nicht mehr so aufdringlich / vordergründig
  - \* Chorus-Phaser-Operator bearbeitet
  - \* OctaveDown-Operator bearbeitet (mehrmals)
    - \* dann Swap-Operator ausführt
    - \* wieder OctaveUp
    - \* die letzten beiden Schritte so oft wie man OctaveDown gemacht hat)

## Synthsounds

-----

interessante Tekknosounds

- \* erstellen Sie zwei Samples gleicher Länge mit Synthesize\_Add (können unterschiedlich hoch sein und sollten verschieden klingen)
- \* generieren sie eine Rechteckschwingung der selben Länge mit Synthesize\_Add wählen Sie dafür eine hohe Frequenz (C-4/C-5)
- \* die einzelnen Frequenzen sollten zueinander passen (Akkord/octavenweise gestimmt)
- \* rufen Sie Mix auf
- \* stellen Sie als Sources die ersten beiden Synthsounds ein
- \* setzen Sie die Mixparameter auf 100 und 0
  - Blendshape : User1
  - Modulationsbuffer : Rechtecksound
  - Modulationsmode : Single
- \* -> Go

## 1.4 2.1.2 Generierung von Effektsounds

### Warps

-----

- \* Noise
  - \* len=60000
- \* Chorus/Phaser
  - \* defaults
- \* Synthesize\_Add
  - \* generieren Sie eine halbe Sinus-Periode
- \* Slide
  - \* source=sine
  - \* slidedist=0 ... -30000, Linear
- \* Amplify
  - \* maxvol
- \* Detune
  - \* source=phaser-noise
  - \* detune=0.1 ... 2.0, User 0, ModSource=half\_sine
- \* Amplify
  - \* amplification=0 ... maxvol, User 0, ModSource=half\_sine
- \* eventuell Filter-LowPass (als Boost), Delay, ....
- \* mit Synthesize\_Add nochmal einen halben Sinus generieren
- \* mit Resample/OctaveDown diesen auf Länge ca. 2000 bringen

- \* Amplify
- \* amplification=65 ... 105, User1, ModSource=half\_sine

### Stereo-Warps

-----

- \* lade einen der im vorherigen Abschnitt generierten Sounds
- \* starte ConvertChannels
  - \* num=Stereo
- \* öffne das Optionswindow (Bufferwindow/'P'-Button neben Stringgadget mit Namen des Puffers)
  - \* deselektiere das zweite Checkbox-Gadget (die mit den Haken) (befindet sich bei Misc/Channels)
  - \* jetzt wirken sich alle Operationen nur noch auf den 1. Kanal aus
- \* benutze Amplify
  - \* amplification=120 ... 0, Linear
- \* Optionswindow
  - \* 1. Checkbox deselektieren
  - \* 2. Checkbox selektieren
- \* Amplify
  - \* klicken sie auf den Swap-Button ('<->') und 'Go'
- \* Optionswindow
  - \* 1. Checkbox selektieren
- \* Amplify
  - \* maxvol
- \* und fertig ; jetzt sollten sie ein fx-sound haben der von einer Seite zur anderen fährt (Kopfhörer !)

## 1.5 2.1.4 komplexe Effekte

### Robotize

-----

verleiht einem Sprachsample einen Roboterstimmenklang

- \* Sprachsample laden
- \* Variante 1
  - \* Delay aufrufen
    - \* Effektanteil ca. 95%
    - \* Feedback ca. 85 %
    - \* Delay 32,64,128,256 (Delay 4 mal auf Ausgangssample ausführen)
  - \* Mix-ZPlane aufrufen
    - \* Src1-Src2 vier Delay-Samples
    - \* zu gleichen Teilen mixen (x-axis=z-axis=50)
- \* Variante 2
  - \* Echo aufrufen
    - \* Delay=200, keine Modulation
    - \* Amplitude 80..10 linear
    - \* Num=12
    - \* Start=0
    - \* Ampf=70

### Hall (Echo)

-----

verleiht dem Klang räumlichkeit

- \* Echo aufrufen
- \* Delay=200..1400 linear
- \* Amplitude 80..10 linear
- \* Num=12
- \* Start=0
- \* Ampf=65

Chord

-----

generiert einen Akkord mit dem PitchShifter  
Vorteil gegenüber z.B. Chord im Protracker :  
die zu mixenden Samples sind alle gleichlang

- \* Sample laden
- \* Faktoren berechnen
  - \* z.B. C-Dur C 1.000000
  - E 1.259893
  - G 1.498331
- \* einfact Resample aufrufen
  - \* die Note bei der Periode eintragen und Factor ablesen
  - \* Resample verlassen (Close-Gadget)
- \* Pitchshift aufrufen und Faktor für E eintragen (Source Originalsample)
- \* PitchShift aufrufen und Factor für G eintragen (Source Originalsample)
- \* Mix aufrufen
  - \* beide gepitchshifteten Samples mit 50:50 mixen
- \* Mix aufrufen
  - \* Originalsample und Mix mit 33:67 mixen
- \* Amplify maxvol

Ghost Echo

-----

erzeugt Geisterechos

- \* Sample laden (z.B. Sprachsample)
- \* Reverse aufrufen
  - \* Effektanteil=100%
- \* Optionsfenster aufrufen
  - \* das Ergebnissample ca. um 5000-10000 Werte verlängern
- \* Echo aufrufen
  - \* Standart-Einstellungen benutzen
  - \* eventuell Echoanz erhöhen
  - \* eventuell Amp. auf kleineren Wert setzen (bei Übersteuerung)
- \* Reverse aufrufen
  - \* Effektanteil=100%
- \* und fertig - jetzt hört man ein Sprachsample bei dem die Reflexionen vor dem eigentlichen Schallereignis kommen

## 1.6 2.1.5 Samplebearbeitung

## better Basssounds/Drumloops/Basedrums

-----

verbessert den Bassbereich eines Samples

- \* Sample laden
- \* falls es ein 'altes Tracker-Sample' ist erst mal resampeln und zwar auf mind. 22050 Hz (besser 44100 od 48000 Hz)  
(Interpolation einschalten !!)
- \* jetzt Filter-LowPass
  - \* Effektanteil=-100 ... -100, None
  - \* range/cut-off=3.0 ... 3.0, None
  - \* wirkt als Höhen-Booster !!!
- \* Amplify
  - \* Amplification ca 70 ... 70, None
- \* Filter-HiPass
  - \* Effektanteil=-80 ... -80, None
  - \* range/cut-off=3.0 ... 3.0, None
  - \* wirkt als Bass-Booster !!!

## Enhance

-----

erweitert das Frequenzspectrum eines Klangs

- \* Sample laden
- \* mit PitchShift bearbeiten
  - \* einmal mit Faktor 2.0 und einmal mit 0.5
- \* beide ge"pitchshift"ete Versionen mit 50 % zusammenmischen
- \* Ergebnis mit Ausgangssample (auch 50 %) zusammenmischen

## Remove Quantizenoise

-----

vermindert das Quantisierungsrauschen von 8bit Samples

- \* 8bit Sample laden
  - \* OctaveDown,OctaveUp ca 3 mal abwechselnd hintereinander aufrufen.
  - \* Filter-LowPass (Effektanteil negativ -> Boost) aufrufen
  - \* als 16bit Sample abspeichern
-