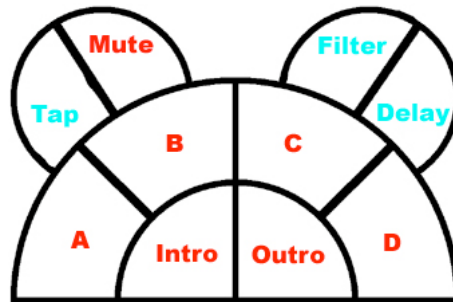


ABLETON als BEWEGLICHES PLAYBACK

Im Gegensatz zu einer festgelegten Audio-Zuspielung ermöglicht die Kombination von Computer/Ableton Live mit einem MIDI-Controller-Pad das "bewegliches" Playback; sprich flexible Arrangements, ein variables Songtempo, und unterstützt somit ein freies agieren und reagieren der Musiker.

Ich erkläre Dir nun meine Vorgehensweise. An Equipment kommt bei mir zum Einsatz:

Mein G4 Powerbook (867 Mhz mit OS X 10.3.9), Live 5, ein kleines *MIDIport 1x1* USB Interface, sowie eine super-kompakte PCMCIA Soundkarte mit vier Ausgängen (*Echo Indigo DJ*). Die Software wird durch ein MIDI-Controller-Pad (*Drumkat EZ*) ferngesteuert.



Ich habe auf dem Drumkat nun ein Arrangement-Schema erstellt, das für alle meine Songs gelten soll. So gibt es je eine Schlagfläche für Intro, Vers (A), Bridge (B), Chorus (C), einen Extrateil (D) und das Outro.

Jeder Song-Part besteht im *Live* wiederum aus verschiedenen Audio-Clips (z.B. Loops, Synths, Vocals usw.), die zu einer Stereo-Spur zusammengefasst werden.

Diese Stereo-Summe (via "Send B" geroutet auf den *Echo* Ausgang 3/4) wird zum FOH Tonmann geschickt.

Parallel zu jedem Song-Part läuft zu meiner Orientierung ein Click Track mit (via "Send A" geroutet auf *Echo* Ausgang 1/2, den Köpfhörerausgang). Zudem gibt es noch die Mute Schlagfläche, bei der nur mein Click spielt, es für das Publikum jedoch nichts zu hören gibt.

Mit den drei verbleibenden "Maus-Ohren" triggere ich Midi-Befehle. So kann ich die Stereo-Summe mit einem Filter ausdünnen oder ich schicke meinen *Impulse* Groove durch ein Filter Delay. Und: ich habe noch eine Tap-Tempo Schlagfläche. Hiermit kann ich in einem gewissen Spielraum in Echtzeit das Tempo verändern, ohne das sich die Tonhöhe meiner Files ändert (= Timestretching). Da der *Ableton* Sequenzer permanent im Hintergrund mitläuft, lassen sich die Trigger-Befehle für meine verschiedenen Song-Parts derart quantisieren, dass ich irgendwann im letzten Takt vor einem angedachten Wechsel auf das nächste Pad schlagen kann.

