Some thoughts on Elektron's Octatrack (Quelques pensées sur Octatrack d'Elektron)

#### Contents

1	Introduction	4		
2	Samples	5		
	2.1 Streaming	5		
	2.2 Ram memory	5		
3	The sample slot lists	6		
	3.1 Sample editing	6		
	3.2 Advantages of the sample slot list	7		
4	The flex machine	8		
	4.1 The flex machine and slices	8		
	4.2 The flex machine and sample locks	8		
	4.3 Slice locks	10		
	4.4 Summary and overview	11		
5	Effects	12		
6	Scenes	12		
	6.1 Summary and overview	13		
7	7 Parts			
	7.1 Parts and what they store	14		
	7.2 Part reloading	15		
	73 Part reloading: usage	17		
	74 Parts and patterns	18		
	75 Scone stacking	20		
	7.6 Using multiple parts	21		
	77 A small but convenient enhancement	25		
	7.7.1 Parts store track volumes	25		
	772 Parts store cranes	25		
	The full store scenes	20		
8	Volume handling	26		
	8.1 Signal flow	26		
	8.2 Volume and scenes	27		
	8.3 Correcting sample volume	27		

9	Sam	pling	28
	9.1	The ot's approach to sampling	28
	9.2	Signal flow	29
		9.2.1 Main outs	30
		9.2.2 Track recorders	31
	9.3	Sampling guidelines	31
	9.4	Track recorders and the sample slot list	32
	9.5	Sampling techniques	33
		9.5.1 Dedicating patterns	33
		9.5.2 Abusing the cue outputs	33
		9.5.3 Abusing a track for live input	34

# 1. Introduction

Cet article fournit une analyse de l'Elektron Octatrack du point de vue des utilisateurs. L'objectif est de regarder l'Ot depuis le début, d'expliquer comment les choses fonctionnent et de développer un aperçu facile à comprendre de la structure de l'Ot.

Plus loin, plusieurs scénarios seront décrits dans lesquels je montrerai / suggérerais comment aborder l'Ot pour les tâches courantes.

Comme nous le verrons, la compréhension de quelques concepts clés réduit considérablement

Le brouillard et la confusion qui semble avoir hanté de nombreux propriétaires de l'Ot.

Quelques notes:

• Cet article a été écrit dans un style cumulatif. Si vous lisez quelque chose sur la page x, je suppose implicitement que tout ce que j'ai abordé avant cela est connu. Vous avez été averti.

• Certaines informations présentées dans cet article seront exclues comme étant «évidentes» ou «rien de nouveau». C'est inévitable.

• L'écriture de cet article n'est pas exactement difficile: l'Ot est ce qu'il est actuellement et il est assez facile d'écrire une histoire sur des choses qui existent déjà.

• Je ne travaille pas pour Elektron et je n'ai aucune idée de leur technologie. Cet article est écrit du point de vue d'un utilisateur. Toute personne de l'équipe de R & D d'Elektron qui lit ceci est probablement en bonne rire;)

• Ce document est un e-ff ort en cours. N'hésitez pas à suggérer et à corriger.

• Pourquoi ai-je écrit ce document? Eh bien ... Ayant obtenu l'ot il y a quelques mois, j'avais un vaste essor de ses possibilités. La fusion de toutes ces possibilités en un flux de travail productif était difficile. J'ai donc décidé d'écrire mes pensées car cela m'a obligé à les exprimer de manière compréhensible. Ce faisant, ce document a augmenté et s'est développé au fil du temps, alors que mes capacités d'intégration de la fonctionnalité de l'Ot dans un bon flux de travail ont augmenté. Je dois donc admettre que j'ai écrit ce document pour des raisons simples. En l'adaptant à un format qui peut être distribué au forum des utilisateurs d'elektron, c'était un peu de travail supplémentaire ... Donc, si cela aide les gens à déchaîner l'Ot, je ne vois aucun motif de ne pas le partager ...

## 2 Samples

Avant de plonger dans l'ot lui-même, il est sage de passer quelques mots sur les samples premier:

Fondamentalement, nous pouvons distinguer deux types de samples:

• Samples courts (formes d'ondes à cycle unique, samples à un seul coup et raisonnablement Petites boucles)

• de longs samples qui peuvent prendre plusieurs minutes

Bien que la distinction ci-dessus soit évidente, il est important de mentionner qu'elle A des conséquences pour la façon dont l'Ot les gère.

2.1 Streaming

Idéalement, l'ot devrait être capable de gérer toutes sortes de samples, à la fois petits et Long, mais. . . Si vous autorisez les gens à utiliser des samples extrêmement longs, vous devez

équipez l'octatrack avec une énorme quantité de mémoire. Il y a une solution à cela: En streaming de fichiers longs directement à partir de la carte cf. Bien que le streaming soit fiable

Technique, elle a ses inconvénients. L'ot a la capacité de jouer 8 échantillons

Simultanément. Donc, si vous chargez 8 machines statiques sur un motif et laissez-les Jouer en même temps, vous sucez efficacement 8 flux d'audio de qualité CD de la carte cf. Cela met un lourd fardeau sur la carte cf et, ce faisant, vous pouvez rencontrer des

problèmes de synchronisation lorsque la carte est bon marché et pas assez rapide.

Par conséquent, bien que le streaming puisse être utilisé, mon attitude personnelle est que j'évite

autant que possible, juste pour être sûr,

D'autre part: Elektron offre le streaming comme une technique valide, donc si vous l'utilisez, il devrait fonctionner comme annoncé, à condition que votre carte cf puisse gérer la vitesse demandé,

# 2.2Ram memory

Une alternative au streaming est la suivante: Équipez-le avec une quantité suffisante d'interne

Mémoire. Une fois que l'on démarre, un projet est automatiquement chargé et une petit collection de samples est lue une fois à partir de la carte cf et met en mémoire ram. Ce faisant, plusieurs avantages sont obtenus:

les échantillons sont lus une fois aucune autre diffusion n'est nécessaire.

Comme les échantillons sont chargés dans la RAM, la manipulation ultérieure de ces échantillons est indépendante de la carte cf et des demandes pour la vitesse et le traitement peuvent être augmentés: lire et écrire en RAM est simplement plus rapide que le streaming d'une carte cf. L'inconvénient est évidemment que la quantité de RAM en mémoire est finalement limite par la quantité d'audio qui peut être mise à la disposition du reste du système.

Elektron a équipé le Ot avec 80Mb de mémoire, ce qui correspond approximativement à 8 Minutes d'audio de qualité CD. 80Mb. . . Hmm. . . Ça ne semble pas grand chose! Conclusion:

Tout ce qui permet à Elektron d'utiliser seulement 80Mb, l'accent n'est pas clairement mis sur

Quantité (c'est-à-dire: stratification et polyphonie excessive), mais sur la qualité et la réalisation

Des choses sérieuses avec un minimum de samples et de mémoire. . .

## 3 The sample slot lists

Dans la section précédente, nous avons vu que Elektron s'attend à ce que vous choisissiez relativement une petite quantité de samples et étirer le plus de ceux-ci.

La sélection d'une petite quantité de samples naturel conduit à la nécessité de certains Genre de liste dans laquelle nous mettons les échantillons que nous voulons utiliser.

Elektron fournit deux Listes pour ce faire: la liste flexible et la liste statique.

Puisque les deux travaillent de la même manière

Je ne décrirai que la liste flex, mais gardez à l'esprit que la liste statique est utilisée pour des samples longs qui sont diffusés, tandis que la liste des flex est utilisée pour des samples plus courts qui sont mis en mémoire RAM.

La liste flexible peut être remplie avec jusqu'à 128 samples. Lors du remplissage de la liste flex

avec des samples, nous avons donc deux conditions:

• Nous ne pouvons pas utiliser plus de 128 samples dans notre projet.

• Que nous utilisions 128 samples ou moins: la somme totale de ces samples ne doit pas dépasser 80Mb.

3.1 Sample editing

Choisir des samples de la carte cf et les mettre dans une liste est très bien, mais nous voulont plus que cela. Nous voulons un contrôle très précis sur la façon dont le samples est joué. Nous voulons "adapter" nos samples afin qu'ils se comportent comme nous le voulons. Ainsi, plusieurs fonctionnalités devraient être proposées:

- définir les points de début et de fin de samples
- définition des points de boucle de samples

• définir si le sample devrait se boucler et comment il devrait boucler: répétitif début de la fin de la boucle ou du ping-pong du début à la fin et du retour, etc.

• tranchage: la capacité de découper un échantillon en plusieurs tranches, permettant des moyens ridicules de reprendre ces tranches intimes.

Toutes les opérations ci-dessus sont effectuées dans le menu d'édition de sample . Une fois que vous Tranché(sliced) un sample ou des points de boucle définis, les décisions que vous avez prises sont tocké dans la liste des emplacements de samples.

C'est un concept important: la liste ci-dessus des opérations sont stockées avec le sample dans la liste des emplacements des samples et n'ont rien à faire avec les machines flexibles qui finissent par jouer le sample. Cela signifie que vous pouvez ajouter un sample à la liste des emplacements des samples, la découper, l'ajuster, etc.

Sans assigner à une machine flex. Vous pourriez même le faire sans utiliser le sample partout dans votre projet. Cela serait évidemment inutile, mais il illustre clairement que la liste de samples slots(traces) vit à sa guise et existe tout au long du projet,

Si les machines flexibles utilisent quelque chose de cette liste ou non.

Le train de pensée le plus important est donc facile: vous sélectionnez des échantillons De la carte cf et les mettre dans une liste. Une fois qu'un échantillon est dans la liste, vous définissez les propriétés de base de ce sample et ainsi déterminer comment le reste du système devrait le gérer,

3.2 Advantages of the sample slot list(Avantages de la liste des fentes d'échantillons)

La mise de samples dans une liste de slot et définissant quelques façons de base que le reste du système devrait manipuler le sample a plusieurs avantages:

**SPEDD** Vous préparez un échantillon en définissant et en le slice comme vous le souhaitez et une fois que vous avez fait cela, l'échantillon et ses paramètres sont disponibles dans n'importe quel piste tenant une machine flexible dans n'importe quel motif dans n'importe quelle banque tout au long du projet.

Train de pensée de base: vous faites le travail dur une fois et réutilisez-le partout où vous en avez besoin.

**FLEXIBILITY** Le même sample peut être mise plusieurs fois dans la liste. Ceci permet pour manipuler le même samples de façons complètement différentes. Par exemple: mettez une sorte de loop dans slot1 et slice cela à 16 tranches. Choisissez maintenant le même samples et mettez-le dans slot2. Vous pouvez maintenant le slice cela jusqu'à 64 tranches, choisir le start différent et des points de fin, etc. Chargez le même échantillon dans le slot 3 et vous pouvez de nouveau faire des choix différents. Vous avez maintenant défini 3 façons différentes de manipuler le même sample.

**ADAPTIVITY** Depuis la création de la musique c'est un processus fortement émotionnel et artistique,

the sample slot list ne doit pas être remplie tout droit du début :

Comme votre projet progresse, de plus en plus de samples peuvent être injectés dans la liste et les samples inutilisés peuvent être enlevés comme le temps passe. Il suffit de faire attention aux conditions décrit plus tôt : pas plus que 128 échantillons, Ne pas manger plus de 80Mb de mémoire.

## 4 The flex machine

Jusqu'ici nous avons vu que les samples sont tirés de la cf-carte et cachés dans une liste. Chacun de ces échantillons peut être préparé pour une utilisation ultérieure en utilisant l'opérations dans le menu d'édition de l'échantillon.

Pour jouer des samples à partir de la liste flexible, la machine flex est nécessaire. Une fois que vous placez une machine flexible sur une piste, l'ot vous demande de sélectionner un sample dans la liste.

À ce stade, vous pouvez sélectionner un sample déjà existant ou vous pouvez sélectionner un free sample slot emplacement après quoi l'Ot vous permet de sélectionner un sample de la carte cf.

Le sample sélectionné est ensuite ajouté à la liste et affecté à la machine flexible. Une chose importante à comprendre ici est que Elektron vous fournit un utilisateur Interface dans laquelle l'ajout de sample à la liste des samples et l'attribution des samples à Une machine flexible va de pair. Gardez à l'esprit cependant que ce sont deux procédures séparées.

Maintenant qu'un sample est assigné à une machine flexible, vous pouvez le trigger par La programmation trig sur le séquenceur de cette piste ou en enregistrant en direct ces triggers.

4.1 The flex machine and slices.

L'affectation d'un sample à une machine flexible semble facile et simple, mais là sont quelques détails à garder à l'esprit: les samples qui vivent dans la liste de samples peuvent être préparée comme décrit précédemment. Cela signifie que certains samples ce slice, tandis que d'autres ne le feront pas. Cela a des conséquences sur la façon dont les machines flex les traitent.

Si un un sample est slicé, vous devez demander à la machine à utiliser d'utiliser ces slices. Plongez dans le menu de configuration de lecture de la machine flex et réglez l'option tranche sur "ON". Avec ce réglage, le paramètre de démarrage sur la page de lecture peut maintenant être et Choisir quelle partie doit être jouée.

En bout de ligne(bottom lines): puisques les samples peuvent être préparés de différentes façons, les machines flexibles doivent être adaptées / mesure afin d'utiliser ce qui a été préparé.

Les fonctionnalités que vous avez dans le menu d'édition de l'échantillon forment la base sur laquelle les machines flex fonctionne.

4.2 The flex machine and sample

Avant de lire plus loin: je suppose que vous comprenez le concept de paramètre **LOCK.** Les paramètre **LOCK** forment une caractéristique très puissante lorsqu'ils sont utilisés sur le machine flex. Pour expliquer leur pouvoir, je me tournerai vers un exemple simple. Assumer:

• La liste des samples contient trois samples loop, simplement appelé loop1, loop2 et loop3.

• La liste d'échantillons contient également deux samples loop qui ont été sliced.

Appelons-les sliced1 et sliced2.

- Une machine flexible vit sur la TRACK 1 et a été affectée à loop1.
- Une autre machine flex vit sur TRACK 5 et a été affecté à sliced2.

Schématiquement, le modèle a été configuré comme ceci:



Ceci est une configuration assez simple. Regardons la machine flexible 1... Machine Flex 1 joue une boucle simple. Par conséquent, le séquenceur n'a besoin que d'un triggerer pour le jouer et ce trigger est placé sur la première step. Rien de mal Ici, mais supposons que j'aimerai jouer un autre sample. Vous pouvez maintenant tenir le trigger Et tournez le bouton de niveau. L'ot vous présente la liste sample slot et vous pouvez Sélectionnez l'un des samples dans cette liste. Supposons que je sélectionne loop2. Le résultat est que, bien que loop1 ait été (et soit) affecté à la machine flex, j'ai remplacé ceci en utilisant un sample lock sur cette étape. Schématiquement parlant:



Et là, il est: en utilisant des samples lock, je peux jouer n'importe quel samples de la liste de samples à n'importe quelle step sur n'importe quelle track contenant une machine flexible dans n'importe quel pattern banque. C'est une manière assez puissante de traiter les échantillons. Cette approche montre la différence de fonctionnement de ot par rapport aux autres sampleurs: au lieu des couches et une polyphonie excessives, Elektron met le pouvoir dans l'ccessibilité des samples,

### 4.3 Slice locks

Encore une fois, regardez l'exemple tel qu'utilisé dans la section précédente.

La machine Flex 5 a été affectée à un sample en tranches.

Pour utiliser les sliced, la machine flexible se déroule comme décrit précédemment. Dans la section précédente, nous avons vu qu'en utilisant des samples locks, vous pouvez sélectionner n'importe quel sample dans votre listes de samples.Jusqu'à présent, si bon, mais tout comme le sample peut être verrouillé, peut donc le paramètre de démarrage sur la page de lecture. Le start paramèters détermine quelle slice est jouée. Ainsi, vous pouvez non seulement déterminer quel sample vous voulez jouer, mais qui aussi slice dans ce samples spécifique en bloquant le start parameter. À titre d'exemple, supposons que nous fermons la machine flex 5 à sliced2. Schématiquement:



Étant donné que le sample sliced2 contient également des slices, je peux maintenant verrouiller le start parameter pour sélectionner la slice (tranche) dont j'ai besoin. En bout de ligne(bottom lines): toute machine flexible configurée pour traiter des tranches peut jouer toute slice de tout échantillons contenant des slices.

Vous sélectionnez le samples en utilisant des samples locs et vous sélectionnez la slice en verrouillant le start parameter sur la page de lecture.

Mettons cela à l'extrême et faisons un peu de mathématiques: la liste samples slot peut contenir jusqu'à 128 échantillons. Chacun de ces échantillons peut contenir 64 tranches. Nous avons donc  $128 \times 64 = 8192$  tranches disponibles.

Ajoutez à cela que nous pouvons faire la même chose avec la liste sample statiques et machines statiques et ce nombre double à un massif de 16384 slice disponible.

4.4 Summary and overview(4.4 Résumé et aperçu)

Jusqu'à ce point, la structure globale de l'Ot est, à mon avis, encore relativement Facile à saisir:

• Les samples sont sélectionnés à partir de la carte cf et mettent une liste de samples disponible dans le projet.

• Les samples qui figurent dans la liste peuvent être préparés pour une manipulation accrue. Cela se fait dans les menus d'édition de sample.

• Machines Flex jouent des samples dans la liste des samples. Lock et slice le sample, il Peut être utilisé pour accéder à n'importe quelle slice de n'importe quel samples dans la liste où et quand vous avez besoin.

## 5 Effects

Dans la section précédente, nous avons vu comment les samples peuvent être préparés et joués de manière différente.

Bien que cela crée déjà beaucoup de possibilités d'accueil, il serait bien que nous puissions également supprimer les samples grâce à certains effets.

Contrairement aux modèles précédents, comme le MNM et md, Elektron nous offre deux slots effet dans lequel on peut insérer un effet. En d'autres termes: Le son généré par une machine est alimenté en deux processeurs fx de notre choix. Schématiquement parlant:



L'avantage de cette approche est que nous pouvons sélectionner les effets que nous voulons appliquer à un samples spécifique. C'est beaucoup plus souple que d'être obligé d'utiliser un ensemble fixe d'effets.

6 Scenes

Dans cette section, je vais décrire les scènes, mais pas en gaspillant beaucoup de pages sur la façon de les configurer car je finis par réécrire le manuel d'Elektron. Au lieu de cela, je vais regarder les scènes d'un point de vue un peu plus conceptuel et vous fournir un modèle simple de la façon de les regarder.

Fondamentalement, une scène vous permet d'utiliser le crossfader pour contrôler plusieurs paramètres sur plusieurs pages sur plusieurs pistes à la fois. L'idée d'attacher plusieurs paramètres à un crossfader n'est pas exactement nouvelle. Par exemple, si nous utilisons Ableton Live avec le contrôleur apc40 d'Akai, nous obtenons les mêmes fonctionnalités. Le crossfader sur l'apc40 peut être attaché à plusieurs paramètres éparpillés partout et un fondu simple les modifie tous.

Si Elektron a étendu cette fonctionnalité à un niveau supérieur: des collections complètes de paramètres peuvent être stockées dans une scène et pas moins de 16 scènes peuvent être chargées dans les deux espaces de scène vivant sur le côté gauche et droit du crossfader. À son point, il est important de regarder quelques scénarios:

**Fade de la valeur par défaut à une scène** Supposons que le crossfader est sur la gauche et que l'emplacement de la scène A est coupé. Cela désactive efficacement la scène et le pattern joue selon ses paramètres de paramètres par défaut. Une scène a été chargée du slot de la scène B. Maintenant, quand je déplace le crossfader, je Fading des paramètres par défaut aux réglages de la scène. **Fading back** me ramène aux paramètres par défaut du motif.

**Fade de la scène à la scène un autre scénario:** alors que le crossfader se trouve à l'emplacement de la scène B, je peux charger une autre scène sur l'emplacement de scène A. La déviation vers la gauche déplace maintenant les paramètres de la scène B vers la scène A ..

**Muting a scene :** Arrêt d'une scène Enfin, lorsque je mute une scène, l'Ot retourne immédiatement à ses réglages par défaut ou se déplace peu à peu si je me déforme.

Pour résumer, il existe trois choses clés que vous pouvez faire avec des scènes:

- fondu(fade) par défaut sur scène fade from default to scene
- fondu(fade) d'une scène à l'autre fade from scene to scene
- disparaître de la scène par défaut **fade from scene to default** Schématiquement parlant:



Métaphoriquement parlant:

Une scène vous permet d'entreprendre une expédition du camp de base (paramètres par défaut) dans un nouveau territoire (scène) et aller plus loin et beaucoup plus loin que vous gardez le chargement de nouvelles scènes et Fade pour eux. Revenir en arrière au camp de base est toujours possible en annulant une scène et en évitant. Bien sûr, vous pouvez également sauter le fading et activer / désactiver une scène alors que le crossfader est là.

Dans ce cas, vous créez un saut soudain par défaut ou une scène ou vice versa

6.1 Summary and overview

Jusqu'à présent, la structure générale de l'Ot est encore assez simple:

• Nous tirons des samples de la carte cf, les placer dans une liste et les préparer à une utilisation ultérieure.

• On charge des machines à des pistes un assign (certaines) nos samples préparés pour eux.

• Nous mettons en place des effets sur les sons mangle « Mangle Sound »provenant des machines.

• Nous configurons des scènes pour permettre de modifier plusieurs paramètres à la fois, ce qui nous permet de dériver des paramètres par défaut (et de revenir) de manière pratique. Une chose cruciale à comprendre ici est la façon dont les choses dépendent les unes des autres: Vous avez d'abord sélectionné des samples, puis configurez les effets dont vous en avez besoin et une fois que cette base se trouve, vous construisez des scènes pour effectuer sur cette plate-forme.

## 7 Parts

Le système partiel est probablement l'aspect le plus compliqué de l'Ot.

Bien que le système de pièces soit très puissant et permet des configurations et des possibilités très intrigantes, il est également une source énorme de confusion tant pour les nouveaux arrivants que pour les utilisateurs plus expérimentés. Décrire quelles parties sont difficiles car elles peuvent être utilisées de différentes façons.

Par conséquent, simplement décrire ce qu'est une partie ne sera pas utile car cela aboutit à une description plutôt abstraite et formelle qui permettra à beaucoup de gens de se pencher sur les parties et comment les utiliser.

J'ai donc décidé de prendre un itinéraire différent.

Au lieu de trouver une définition formelle, je vais expliquer quelles sont les parties et ce qu'elles font en décrivant quelques expériences simples.

En suivant simplement cette approche étape par étape, notre compréhension des parties augmentera et (espérons-le) évoluera jusqu'àu niveau où tout le brouillard aura disparu. . . Mais d'abord, nous devons plonger dans un peu de théorie sèche sur une question simple: quelles informations sont stockées où.

7.1 Parts and what they store (Pièces et ce qu'elles stockent)

Question: Quelles informations a-t-il besoin pour jouer un pattern comme prévu? Voyons voir. . . Tout d'abord, il faut connaître les samples que nous voulons utiliser. Les samples sont stockés dans la liste des samples comme décrit précédemment. Cette liste, cependant, est disponible dans l'ensemble du projet.

Il n'est donc pas spécifique à notre pattern. Ainsi, j'ignore la listes des samples.

Suivant: L'Ot veut savoir quelles machines ont été chargées sur les pistes et comment nous les avons configurées. Donc, nous avons ici la **première information** valable dont vous avez besoin: les machines et leurs paramètres.

**Ensuite:** l'Ot souhaite également connaître les effets que nous utilisons sur ces machines et la façon dont nous les avons configurés. Voici donc l**e numéro deux:** les effets et leurs paramètres.

**Ensuite:** nous avons des machines et leurs effets. . . Et nous avons configuré plusieurs scènes que nous utilisons pour effectuer. Les scènes sont donc le **numéro trois**.

Suivant: toutes les pistes jouent leur audio sur leur propre volume. Les paramètres de volume des pistes sont donc **numéro quatre.** Enfin: Ot veut savoir quand trigger les samples et si les paramètre lock ont été utilisés pour ajouter des cloches et des sifflets, alors voici le **numéro cinq.** 

Pour résumer: l'ot a besoin de cinq morceaux d'information pour jouer un modèle comme nous le voulons:

- triggers and parameter locks
- scenes
- effects and their settings (et leur réglages)
- machines and their settings (et leur réglages)
- track volume settings (les réglages de volume de piste)

Prochaine question: Où toutes ces choses doivent-elles être stockées? Le plus simple La manière serait de stocker toutes ces choses dans un pattern. . . Et c'est là que ça devient intéressant. . . Elektron a décidé de faire quelque chose de différent. Au lieu de stocker toutes ces informations à l'intérieur d'un patterns, Elektron a diffusé cette information sur des modèles et ce qu'Ellktron appelle des parts. Alors, qu'est-ce qui est stocké là où? Bien:

Patterns store triggers and parameter locks

Parts store machines (+settings), effects (+settings), scenes and track volumes.

Alors, pourquoi cette distinction? Au premier abord, cela semble compliquer les choses inutilement.

Eh bien, le temps pour une action. . .

7.2 Part reloading (Partie rechargeant)

Tout d'abord: Construisons un modèle simple pour travailler.

Le pattern que j'ai créé est configuré comme suit:

La track 1 contient un bass -rum. Pour lui donner une morsure, je place un compresseur sur la piste, suivi d'un délai qui sera utilisé lors de la construction d'une accumulation.

La track 2 tient un clap. Pour le sculpter, je pose un filtre et une réverbération derrière lui.

La track 3 contient une boucle avec des trucs hi-hat. Effets: filtre et retard.

La track 4 contient une loop bass line, suivie du filtre et du lo-fi.

La Track 5 contient un pad avec filtre et réverbération.

Plus loin, j'ai mis en place quelques scènes pour créer un mouvement.

Maintenant, sans expliquer pourquoi, essayez ce qui suit:

• Appuyez sur [FUNCTION] + [PART] pour accéder au parts menu. Vous verrez quatre parties.

La première est en surbrillance et montre une petite étoile. Cette étoile est importante, plus à ce sujet plus tard.

• Appuyez sur [FUNCTION] + [EDIT]. Un menu apparaît ce qui vous permet de sauvegarder la pièce. Enregistrez et quittez le menu.

• Jouez le pattern. Si tout s'est bien passé, rien n'a changé et le pattern joue comme d'habitude.

# **Ensuite:**

• Pendant la lecture, modifiez votre pattern en changeant de manière aléatoire les paramètres sur n'importe quelle track. Remplacez également les effets par des différents et supprimez certaines machines ou lui attribuez des samples différents.

**Soyez brutal ici:** modifiez tellement le motif que le motif original est complètement détruit. **Remarque importante:** Ne pas enregistrer les parametres locks et ne pas ajouter ou supprimer des triggers.

Pour résumer: modifiez tout ce qui est stocké dans une parts mais laissez intacte ce qui est stocké dans un pattern ((trigs and parameter locks)).

• Félicitations: vous venez de transformer un beau pattern en un complet désordre.

Ensuite:

• Pendant la lecture de votre pattern détruit, appuyez sur [FUNCTION] + [RELOAD]. Une fois que vous faites cela, toute destruction que vous avez causée est annulée et le pattern est instantanément inversé à l'état où il se trouvait lorsque vous avez enregistré la partie.

• Répétez ce qui précède: détruisez votre pattern, modifiez les paramètres, remplacez les machines, quoi que ce soit, et appuyez à nouveau sur [FUNCTION] + [RELOAD] pour retourner à la base.

Métaphoriquement parlant: en sauvegardant une partie, vous avez mis en place un camp de base.

Une fois mis en place, vous pouvez vous promener et voir ce qui se passe en expérimentant différentes machines, effets et scènes.

Supprimez, modifiez et remplacez ce que vous voulez. Une fois que vous vous êtes perdu, appuyez sur [FUNCTION] + [RELOAD] et vous êtes automatiquement ramener à la base.

Schématiquement:



Comme vous pouvez le voir, le rechargement d'une partie a une certaine similitude avec l'utilisation de scènes:

les deux peuvent être utilisé pour dériver des paramètres standard et vous permet de retourner à la base. Les scènes, cependant, ne sont utilisées que pour changer les paramètres et permettre la décoloration entre les valeurs des paramètres.

Les parties, en revanche, sont beaucoup plus radicales. Ils n'autorisent pas la décoloration, mais ils permettent de remplacer les machines, les effets et les scènes tout en conservant la possibilité de renvoyer tout au dernier état enregistré.

Alors, quelle était cette petite étoile? Facile: une fois que vous avez sauvegardé une partie, l'état actuel du motif est exactement le même que celui enregistré dans la partie.

Une fois que vous avez changé (en tournant un bouton de paramètre ou en remplaçant une machine entière ou autre), l'état actuel a changé de ce qui a été enregistré.

L'étoile indique que des modifications ont été apportées à l'état initialement enregistré.

En d'autres termes: lorsque vous voyez cette étoile, ce que vous entendez actuellement est en quelque sorte différent de ce que vous avez enregistré dans la partie.

## 7.3 Part reloading: usage

Il existe plusieurs situations dans lesquelles le rechargement d'une pièce peut être utile. Certains suggestions:

**Expérimentation indolore(Painless experimentation)** Ceci a été décrit dans la dernière section: Après avoir sauvegardé la partie, vous pouvez remplacer, jeter et sculpter ce que vous voulez, mais le camp de base est toujours à portée de main.

**Développement incrémentiel** Supposons que vous avez un pattern exécuté avec une partie sauvegardée.

Comme vous l'écoutez de manière critique, vous trouvez que le hit hat est un peu trop fort. Vous le corrigez en réduisant son volume. Vous trouvez également que le bass-drum que vous avez utilisé est complètement déchet et devrait être remplacé. Donc, vous trouvez un bass drum approprié et composez-le. Au fil du temps, vous continuez à régler le pattern jusqu'à ce qu'il ressemble profondément mieux que ce que vous avez commencé. Une fois que vous êtes satisfait: enregistrez la piartie! C'est-à-dire: déplacer votre camp de base vers ce meilleur emplacement de son.

### 7.4 Parts and patterns

Dans la dernière section, nous avons vu une utilisation de base des parties et avons compris de quelle façon elles fonctionnaient.

Ce n'est pas encore la grande image. Dans cette section, je me concentrerai sur la relation entre une partie et une collection de patterns.

Je vais commencer par le même exemple que j'ai utilisé plus tôt. Supposons que le pattern vit sur a1, qu'il utilise la partie 1 et que la partie a été enregistrée.

Cela me donne un bon point de départ. Avant d'aller plus loin, il est important de comprendre à quoi ressemble la mémoire de l'Ot.

Le schéma suivant montre la situation présente:



La flèche illustre que le pattern utilise la partie. Rappelles toi:

Patterns store triggers and lock, tandis que la partie stocke l'infrastructure sous-jacente:

Machines, effets, scènes et volumes de pistes.

Que se passe-t-il lorsque nous copions le pattern a1 à a2? Bien. . . Il est évident que tous les triggers et locks paramètres qui sont sur le pattern a1 sont maintenant copiés sur le pattern a2. Mais qu'est-ce qui arrive à la partie? Est-ce qu'une copie papier de la partie est également faite?

Nan. . . La partie est laissée inchangée. Au lieu de cela, le modèle a2 utilise également la partie 1. Schématiquement:



Pensez-y un moment aux implications de cela. . .

Si je change quelque chose dans la partie pendant que le pattern a2 est en cours d'exécution, cette modification sera également reflétée dans le pattern a1, puisqu'elle utilise la même partie. C'est une énorme source de confusion pour de nombreux utilisateurs, car ils pensent à tort qu'un modèle stocke toutes ses informations localement. Ce n'est pas comme je l'ai décrit précédemment.

Nous concluons donc quelque chose de très important: tout changement que vous apporterez à une partie sera reflété dans tous les modèles qui utilisent cette partie.

Nous allons étendre un peu plus loin. Je retourne au pattern a1 et le copie dans Pattern a3.

### Schématiquement:



À ce stade, vous devriez savoir où cela se passe. Je peux créer diverses variantes d'un pattern en copiant et en collant à volonté et tous ces patterns utiliseront les mêmes machines, effets et scènes sous-jacents. En modifiant quelque chose sur les machines, les effets ou les scènes dans l'un de ces patterns, cela se reflète dans tous les autres modèles qui utilisent cette partie.

Alors, quel est le but de ceci? Facile: commodité et cohérence. Par exemple: Supposons que la résonance de mon filtre dans la Track 5 soit un peu trop agressive. Donc, je le domine un peu. Si chaque pattern stockait ses informations localement, je devrais changer cela sur le pattern a1, puis passer au pattern a2 et le changer à nouveau, puis passer au pattern a3 et le changer pour la troisième fois. . . Ce qui m'énerve profondément.

Avec plusieurs pattern utilisant la même partie, je n'ai besoin que de faire des modifications une fois et tous les patterns utilisant cette partie suivent avec elle. Cela m'économise beaucoup de travail terne et garantit que tous les patterns forment une collection cohérente de variations, toutes utilisant des machines qui semblent identiques, des effets et des scènes qui fonctionnent de la même manière sur tous ces patterns.

Cela se développe pour jouer avec de multiples patterns: Supposons al est en cours d'exécution et que je modifiee divers paramètres, le volume des pistes, etc. Lorsque je passe à a2 (ou tout autre pattern à l'aide de cette partie), les paramètres que j'ai changés continueront à être modifiés (tant que la partie n'est pas rechargée), ce qui permet un changement naturel de al à a2. Si chaque motif stockait ces paramètres localement, la commutation d'un motif entraînerait tous les types de sauts de paramètres.

La seule différence réelle entre les patterns a1 et a2 réside dans les trigger et paramètres locks. Bien que cela puisse paraître un peu stupide à son point, rappelez-vous que les paramètres lock peuvent être utilisés pour jouer n'importe quelle slice de n'importe quel sample dans la liste des samples afin que les triggers et locks seuls créent déjà assez de chaos pour transformer le motif a2 en quelque chose qui est très différent de a1 Et a3.

Pour compléter, considérez un exemple simple et réel:

• Le Pattern al contient mon Pattern de base. Je peux utiliser des scènes pour effectuer quelques astuces sur ce Pattern et utiliser également la fonction Partts reload que j'ai décrite plus tôt.

• Le Pattern a2 contient une copie du Pattern a1, mais ici, j'ai ajouté des triggers supplémentaires sur mon bassdrum à la fin du Pattern. J'ai également utilisé certains

paramètres locks pour lancer le bass-drum en reverze. J'ai également utilisé des locks pour redéfinir(locks to retrigger) quelques hit hat et, enfin, j'ai utilisé des locks pour ajouter de la saleté à ma ligne de basse sur la piste 4.

• Comme le pattern a1 finit, je demande à l'Ot de passer à a2.

A2 contient l'accumulation et j'attends patiemment qu'il se développe. Comme a2 finit, je dirige l'ot à pattern a3. . .

• Le pattern a3 sert de break: tous les triggers sur le bass -drum ont été supprimés. Au lieu de cela, j'ai mis un trigger sur l'étape 1 et l'ai locké à un sample dans la liste qui contient un son de bruit massif qui augmente lentement en volume et nettoie tout. Au fur et à mesure que ce pattern approche, je dirige l'ot au pattern a1 et tout recommence. . .

#### 7.5 Scene stacking(Empilement de scène)

Dans la dernière section, nous avons vu comment les patterns multiples peuvent utiliser la même partie. Les scènes sont stockées dans la partie. Cela signifie qu'une fois qu'une scène est activée, elle reste active même si je passe au pattern suivant. En d'autres termes:

16 scènes peuvent être utilisées librement sur tous les patterns qui utilisent la même partie. Cela peut être utilisé pour passer d'une scène à l'autre et d'un pattern à un autre de manière naturelle. Un léger problème émerge cependant:

Je peux préparer 16 scènes à utiliser avec mes motifs, mais comment me souvenir quelle scène fait quoi? D'accord. . . Je pourrais l'écrire, mais il existe un moyen simple de laisser le mouvement à travers les patterns aller de pair avec le mouvement à travers les scènes. Je l'appelle l'empilement de scène.

L'idée est simple:

• Lorsque le premier pattern est exécuté, il se produit sans être affecté par aucune scène.

• Je déplace le crossfader vers la droite et charge la scène 1 vers la gauche.

Fading vers la gauche met maintenant la scène 1 en plein contrôle(full control).

• Je copie la scène 1 sur la scène 9 et charge la scène 9 vers la droite. Le crossfading vers le droite n'a aucun effet, puisque la scène 9 contient les mêmes valeurs de paramètres.

• Je fais tous les changements que je veux de la scène 9. Puisque les paramètres de la scène 1 sont encore stockés dans la scène 9, je peux me déplacer naturellement de 1 à 9. Les paramètres qui ont changé dans la scène 9 sont empilés sur ce qui a déjà été modifié dans scène 1.

• Je copie la scène 9 dans la scène 2 qui est chargée à gauche. Encore une fois, je modifie les changements apportés à la scène.

- copy 2 to 10, modify and fade
- copy 10 to 3, modify and fade
- copy 3 to 11, modify and fade

• . . .

En mettant des scènes de cette façon, je peux passer à travers de façon naturelle: Les scènes 1 à 8 sont toujours sur la gauche tandis que les scènes 9 à 16 sont toujours à droite.

Fading de scène en scène est facile:

• Si le crossfader est à gauche, j'appuie sur le bouton de scène droit, regarde les leds (9-16) et je charge la scène qui vient après celle chargée actuellement.

• Si le crossfader est à droite, j'appuie sur le bouton gauche de la scène, regarde les leds (1-9) et je charge la scène qui vient après celle actuellement chargée.

Comme le passage des scènes est devenu facile, passer d'un pattern à l'autre est également devenu facile. Au fur et à mesure que je dépasse les patterns, je me fondais de scène en scène. Ces scènes ont été configurées de telle manière qu'elles créent une progression fluide à travers les patterns. . .

#### 7.6 Using multiple parts

Comme nous l'avons vu dans la section précédente, les parts store machines, les effets, les scènes et le volume. Plusieurs patterns peuvent utiliser cette partie et séquencer leurs machines de diverses façons en jouant avec des triggers and parameter locks.

Jusqu'à présent, nous n'avons utilisé qu'une seule partie et, comme vous l'avez probablement déjà pensé, l'ot peut contenir jusqu'à quatre parties par banque. Quatre parties, signifie que nous avons quatre «plates-formes» individuelles (machines, effets, scènes et volume), nous pouvons alimenter nos patterns.

Dans cette section, je vais couvrir largement un exemple qui montre comment utiliser plusieurs patterns, scènes et parties qui peuvent être utilisés pour construire une transition entre deux motifs.

Pour commencer, prendre les éléments suivants, très basiques, configuration:

- Un pattern vit la première piste.
- Le pattern utilise la première partie.
- La partie est sauvegardée: basecamp est sécurisé.

Au lieu de résumer quel genre de sons ont été chargés sur quel piste, je l'ai mis dans le schéma suivant:



Je peux jouer un peu avec ce pattern en modifiant certains paramètres, faire des mutations et des mutations, déclencher manuellement des échantillons, etc. Rappelez-vous que je peux toujours revenir au **basecamp (en arrière au pattern d'origine, avant les modifications)** en utilisant la fonction de recharge de pièce comme Décrit plus tôt.

Heure d'un changement: je veux construire une accumulation. Je le fais en configurant une scène sur l'emplacement de scène b. La scène est facilement construite:

• Les BassDrum feeds(alimentante) un delay, ce qui se traduit par une grosse caisse qui rebondit dans tous les sens.

• Le filtre sur le clap est ouvert, la résonance augmente un peu et la réverbération qui est ajoutée.

• Les rate reduce et de distorsion sur le hi-hat sont utilisés pour créer des artefacts et des destructions.

• Le paramètre de distorsion est augmenté sur la ligne de basse pour la laisser bouffée.

Avec cette scène chargée, un simple et lent fade du crossfader de gauche à droite pompe de plus en plus de folie dans le motif. Comme je continue à déplacer le fader, l'accumulation grandit de plus en plus. . . à quoi?! Bien. . . Ehh. . . Injecter une accumulation est bien, mais tôt ou tard, une accumulation doit être suivie d'un passage vers une nouvelle direction. Je peux passer à un nouveau pattern ou à un fade sur une nouvelle scène comme cela a été décrit dans la dernière section, mais le problème est que, ce que je fais, les machines et les effets resteront les mêmes. Pour vraiment faire passer une nouvelle direction, il faut faire des choses plus radicales:

• Je copie le pattern a1 à a2. Comme décrit précédemment, nous avons maintenant deux patterns utilisant la même partie.

• Dans le pattern a2, je me tourne vers le menu parts et copie la partie 1 vers la partie 2.

• La copie de la partie 1 à la partie 2 ne signifie toujours pas que a2 utilise la partie 2.

Je le sélectionne et appuyez sur [YES].

• Je sauvegarde ALL PARTS.

• Je regarde l'écran de l'Ot et vérifie que a2 utilise effectivement la partie 2.

Après les opérations ci-dessus, la mémoire de l'Ot ressemble comme suit:



Comme vous pouvez le voir, j'ai maintenant deux patterns, tous deux utilisant leur propre partie. Puisque a2 est une copie de a1 et que la partie 2 est une copie de la partie 1, les deux patterns ressemblent exactement à la même chose.

Pourtant, la modification de la partie 2 est maintenant entièrement indépendante de la partie 1. Temps pour certains changements: je veux que a2 sonne beaucoup plus agressif que a1. Pour ce faire, je modifie plusieurs éléments dans la partie 2:

• Le bass drum est remplacé par un plus agressif. Je remplacer également les effets: lofi et la réverbération sont maintenant en place.

• La loop hi-hat est remplacée par un agressif 909 hi-hat + lofi + filter.

• La ligne de basse est remplacée par un très profond et une pression de 1 + eq + delay.

(The bassline is replaced by a really deep and pumping one+eq+ delay.)

• Clap et pad sont laissés inchangés.

J'ajoute également des triggers and parameter locks sur différentes tracks et tweak et finetune jusqu'à ce que j'ai la boucle que je souhaite. Pour rendre ces modifications permanentes, je sauvegarde toutes les parties. Après les opérations ci-dessus, la mémoire ot ressemble à ceci:



Les différences entre la partie 1 et la partie 2 ont été colorées en rouge. Droite: tout a été réglé. . . Essayons:

• Je passe(switch) à a1 et appuyez sur Play.

• Je charge ma scène d'accumulation (my buildup scene) sur l'emplacement de scène b.

• Temps pour l'accumulation(Time for the buildup): je déplace lentement le crossfader et le motif commence à construire et à construire. . .

• À un moment de mon choix, j'ai frappé [PATTERN] + [A2]. a ce moment, quand je fait cela, je sais déjà que lorsque a1 finira, il sera suivi de la sonorité plus agressive a2.

• J'attends jusqu'à ce que a1 se termine et, au moment où a2 frappe, je coupe l'emplacement de la scène b.

En bloquant la scène b, a2 joue sans être affecté par la scène.

• A2 joue maintenant proprement: le nouveau bass-drum et bassline frappent vraiment Et le 909 hi-hat coupe les airs. Le padoriginal et le claquement de

La partie 1 est encore présente comme ils l'étaient: ce nouveau patterns contient encore le restes de son prédécesseur, mais les éléments nouvellement ajoutés ont poussé ce pattern dans une nouvelle direction.

Je suis maintenant partit avec a2 qui utilise sa propre partie. Je peux donc configurer des scènes adaptées à ce patterns, fonctionner avec des paramètres et la fonction de reload part, muet, un-mute, etc. Je peux également copier a2 sur d'autres patterns et générer des variations, sachant que toutes ces variations utilisent la même partie. En effet, le jeu recommence.

#### 7.7 A small but convenient enhancement. . .(Une amélioration petite mais pratique. . .)

Dans la section précédente, j'ai construit un pattern, utilisé une scène pour créer une accumulation(buildup) et déplacé vers le prochain pattern, qui utilise sa propre partie. Une fois que a2 est rentré, je mute la scène et a2 est complètement nettoyé.

Effectuer un nettoyage dans ce cas signifie que la ligne de basse agressive que j'ai installée sur la partie 2 est immédiatement entendue en plein volume au moment où a2 frappe. Rien de mal à cela, mais supposons que je ne veux pas entendre la ligne de basse au moment où a2 kick entre et que je veux la mettre en forme en tant que 2 joue dessus. . . Comment faire ça?

Suppose I don't want to hear the bassline at the moment a2 kicks in and that I want to fade it in myself as a2 plays on... How to do that? Au premier coup d'œil, un mute de la piste de la bass ligne pourrait être réglé au moment où a2 kick entre.

Bien que cela soit possible, il me faut être très précis dans mon chronométrage. Pour aggraver les choses, je devais m'occuper du timing depuis que je devais muter la scène B au moment exact a2 kicks in. Ce n'est évidemment pas pratique. Examinons certaines solutions. . .

#### 7.7.1 Parts store track volumes

Une partie stocke les machines, les effets, les scènes et le volume auquel les pistes alimentent leur audio vers les sorties principales. Par conséquent, régler le volume de la piste sur la piste de la ligne de basse sur 0 et enregistrer la pièce, "muet" la piste une fois que a2 kick in. Je peux ensuite sélectionner la piste et augmenter le volume à mesure que le pattern joue sur. (pattern play on)

Il y a cependant un inconvénient à cela: chaque fois que la partie se charge, le volume de cette piste est réglé sur 0.

Chaque fois que j'utilise la fonction reolad part, la piste tombe à 0.

Donc, tant que la partie ne reload pas, je suis en sécurité, mais ce n'est pas idéal.

7.7.2 Parts store scenes

Rappelez-vous qu'une partie stocke les machines, les effets, les scènes et le volume.

Le fait qu'une partie stocke des scènes est très important ici.

Pendant que a1 joue, les scènes de la partie 1 peuvent être utilisées pour faire ce que vous voulez.

Pour créer l'accumulation(buildup) vers a2, j'ai chargé la scène 9 dans l'emplacement de scène b et je l'ai lentement fade. Alors. . . Une fois que a1 finit, le fader est poussé plein vers la droite et la scène 9 est en plein contrôle.

Ensuite, a2 kick rentre et charge sa propre partie.

Le fait que a2 charge sa propre part, implique qu'elle charge aussi ses propres scènes! Elektron a également franchi une étape: une partie stocke également les scènes chargées sur les emplacements de scène a et b et aussi si une scène est activée ou désactivée. Utilisons cela à notre avantage:

• Je me déplace à a2 et supprime la scène 9.

• Je charge une nouvelle scène 9 sur l'emplacement de la scène b. En commençant par une scène propre, je m'assure qu'aucun paramètre inattendu ne soit affecté par cette scène.

• Je configure la scène 9 de telle sorte que le volume de la ligne de basse agressive soit réglé sur 0.

• Je sauvegarde la partie pour la rendre permanente.

Et là, il est: Une fois que a2 kick intervient, je n'ai plus besoin de muter la scène b à ce moment précis. En fait, je n'ai besoin de rien faire:

Au moment où l'on passe de a1 à a2, la partie 2 est chargée et la scène 9 de la partie 2 est également chargée.

Cette nouvelle scène tue le volume de la ligne de basse. En remontant le fader vers la gauche (où aucune scène ne vit du tout) s'efface dans la ligne de basse à son niveau normal.

Le pattern fonctionne maintenant proprement.

Métaphoriquement parlant: j'ai quitté mon précédent basecamp pour de bon et je suis arrivé à ma nouvelle maison.

Puisque j'ai vraiment aimé le clap et le pad, je les ai emmenés avec moi, mais un jour, quand je me déplace à la prochaine basecamp, je pourrais les laisser derrière. . .

#### 8 Volume handling(Traitement du volume)

Une fois qu'une machine joue un samples, plusieurs paramètres affectent son volume. Comprendre quels paramètres vivent où et comment ils interagissent avec des scènes évitera plusieurs pièges de manipulation de volume.

#### 8.1 Signal flow(Débit du Signal)

Le signal qu'une machine génèré, est d'abord détruit par l'amplificateur enveloppe. Sur la page de l'ampli, vous trouverez le **paramètre vol** qui est un paramètre bipolaire, ce qui signifie que sa portée passe de -64 à +63.

Le paramètre par défaut est 0. La génération d'effets de tremolo peut être effectuée en laissant un lfo moduler ce paramètre.

L'imitation de side chain compression latérale se fait en configurant un lfo de telle sorte qu'il diminue le volume sur ces moments où un kick frappe.

Les paramètres lock peuvent également être utilisés pour cela.

Après l'enveloppe de l'amplificateur, le signal passe à travers deux processeurs fx. Enfin, le paramètre de volume de track détermine le volume global de toute la track (Samples + fx). Schématiquement:



Comme on peut le voir sur le schéma, l'ot offre la gestion du volume pré et post fx sur chaque piste.

### 8.2 Volume and scenes

En plus des paramètres de volume réguliers, ot offre deux paramètres supplémentaires qui sont exclusivement disponibles pour les scènes:

Track volume: xlev peut être réglé sur min ou max. Le paramètre est affiché une fois que vous maintenez l'un des boutons de la scène.

Min équivaut simplement à 0 et max correspond au volume actuellement réglé (c'est-à-dire: si la scène est muté).

Cela vous permet donc de "muter" une piste lorsque la scène est active orto dans une piste pour le volume de piste "habituel" (usual ).

Volume de l'amplificateur: xvol se trouve dans la page de l'enveloppe de l'ampli lorsque vous maintenez un bouton de scène. Il fonctionne tout comme le paramètre xlev, mais cette fois cela affecte le paramètre de volume de l'ampli.

Schematically:



Les paramètres de xvol et xlev sont très utiles car ils permettent aux scènes de fanent et les pistes à la fois des pré et post fx.

Si, par exemple, un signal est traité par un retard avec beaucoup de commentaires ou une réverbération avec une longue queue de réverbération, la décoloration d'une track à l'aide du paramètre xvol permet au retard / réverbération de disparaître naturellement.

De là, il s'ensuit que pendant une performance, la plupart des mixages de tracks sont idéalement réalisés en programmant les paramètres xlev et xvol dans les scènes.

### 8.3 Correcting sample volume

Supposons que vous avez un kick fonctionnant sur le pattern a1.

Le sample du kick correspond au reste du pattern et les amplis et les volumes de piste de l'amplitude ont le niveau souhaité.

Les paramètres sont enregistrés sur la partie.

Le motif a2 est une copie du motif a1 avec une exception: le kick a été remplacée par une autre en utilisant des paramètres locks. Bien que cette nouvelle basse ait le caractère que vous voulez, il est malheureusement trop fort.

Comment corriger cela?

Au premier coup d'œil, il est tentant de diminuer le volume de la track, mais cela crée un problème: une fois que vous réduisez le volume de la piste, le kick qui est sur al devient trop douce. Apparemment, nous nous sommes heurtés à une situation d'exclusivité mutuelle:

Soit la boussole a1 est trop douple, soit la grosse batterie a2 est trop forte.

La modification du volume de la piste n'est évidemment pas une solution, de sorte que je laisse le bouton de volume de la piste intact.

Peut-être que l'utilisation de paramètres locks sur le paramètre de volume de l'amplificateur fait le travail.

En fait, mais maintenant je dois me rappeler que le kick devrait être locké à chaque étape que je programme dans les patterns qui utilisent ce kick. Bien qu'il s'agisse d'une solution, elle n'est toujours pas très pratique.

Effectivement, le problème est le suivant: tout sur vos patterns était correct, jusqu'à ce que ce kick ait été composé.

Le problème est causé par le sample lui-même et, par conséquent, il doit être corrigé sur le level sample et non pas en perturbant les paramètres exsistant qui affectent également d'autres éléments. **Train de pensée de base: l'échantillon devrait s'adapter au système et non l'inverse.** 

Heureusement, l'Ot offre un moyen simple de corriger les samples "mal comportés": Sur la page des attributs de l'éditeur de sample, un paramètre de gain est trouvé.

Ce paramètre vous permet d'augmenter ou de diminuer le volume de l'échantillon avec un 24dB massif dans les deux sens. Il suffit de démarrer le motif et de configurer le volume dans la page des attributs: de nombreux problèmes avec des track volumes incohérents ou des lock paramètres sont maintenant historiques car ce paramètre est considéré comme un attibut du sample et stocké dans la liste des emplacements (slot) samples.

#### 9 Sampling

Jusqu'à présent, j'ai complètement ignoré les capacités de l'Ot pour le sampling. J'ai eu une simple raison de le faire: expliquer la structure globale de l'Ot est assez difficile et parler de sampling n'aurait pas changé la vue d'ensemble. Afin d'éviter de rendre les choses plus compliquées que nécessaire, j'ai évité le problème de sampling dans son ensemble.

#### 9.1 The ot's approach to sampling

Quand je regarde les sampleurs du passé, je peux voir que la plupart des sampleurs ont la capacité d'enregistrer de nouveaux matériaux du monde extérieur et de le reproduire. En effet, cela est évident, mais le point est que sur la plupart des sampleurs, ces deux caractéristiques ont été complètement séparées.

La plupart des sampleurs offrent à l'utilisateur un «sampling mode » et un «mode de lecture(playbackmode)» et pour de bonnes raisons: lors de l'enregistrement de nouveaux matériaux, vous entrez en sampling mode, appuyez sur START et enregistrez. Une fois l'enregistrement pris Place, certaines installations sont proposées pour le découpage et la préparation de sample avant, enfin, peuvent être utilisées pour la lecture. Tout au long des années, plusieurs améliorations ont été ajoutées.

Par exemple: midi peut être utilisé pour synchroniser l'échantillonnage et la lecture à l'équipement externe, etc.

Puis, en 2004, Ableton Live 4 a frappé les marchés.

Ableton a pris un autre échantillon de visualisation: il offre un système dans lequel les samples samplés et de lecture peuvent être effectués sans arrêter le séquenceur ou passer d'un sampleing à l'autre.

En d'autres termes: Ableton intégré à la fois dans un système qui est sous le contrôle direct d'un séquenceur et d'un utilisateur qui le manipule.

La plus grande différence entre Live 4 et Live 3 réside dans le fait que Live 4 a été la première version qui a permis l'utilisation de synthétiseurs et d'effets vst, Reliant le monde du sampling au monde de la production et du flou de la distinction entre les deux.

À l'époque, Elektron devait envisager d'intégrer déjà des sampleurs dans leurs produits. En 2005, le module a été introduit et cette unité prend plus ou moins la même approche de sampling que celle d'Ableton: le sampling et la lecture peuvent être effectués sans arrêter du séquenceur. Ce faisant, le sampling de nouveaux matériaux est devenu aussi trivial que de lire des samples. Cette approche permet une intégration transparente du sampling dans une performance et c'est un grand pas en avant de ce qui existait déjà à cette époque.

9.2 Signal flow(Débit du Signal)

Avant de plonger dans le processus de sampling lui-même, quelques mots devraient être consacrés au flux de signaux.

Peut-être un peu terne, mais il est important que la compréhension du flux du signal évite les problèmes de volume.

Tout d'abord: Ot propose deux entrées de ligne stéréo, simplement appelées entrée a / b Et entrée c / d. Lorsque l'équipement externe est connecté à l'Ot, un petit problème apparaît: certains synthétiseurs fournissent un signal très chaud tandis que d'autres fournissent un signal trop doux. L'Ot devrait donc offrir une installation pour augmenter ou supprimer un signal entrant avant que ce signal ne frappe le reste du système.

Le ot offre des paramètres de gain simples pour a / b et c / d dans la page du mixer. Le gain permet de booster un signal entrant avec 12dB ou de le supprimer à 0.

Le rôle de ces paramètres devrait être clair:

Vous utilisez le gain pour corriger le volume des signaux entrants afin que le reste de l'utilisateur reçoive un signal correct.

Idéalement, le gain devrait être configuré une fois et être laissé tel quel tout au long du reste du projet.

Réglez et oubliez. Les paramètres de gain ne peuvent donc pas être automatisés, séquencés ou autres.

Ils existent pour fournir à l'Ot des signaux correct au monde extérieur et c'est tout. Toutes les astuces sur le volume, la fade in ou out pendant la performance, etc. devraient être faites autrement. . .

Schematically:



Maintenant que nous avons un signal sain, nous pouvons faire deux choses avec lui:

• send le directement aux principales sorties

• l'alimenter(feed) à différents enregistreurs de piste(feed it to various track recorders)

#### 9.2.1 Main outs

(sending singal)L'envoi de signaux aux principales sorties est probablement la partie la plus simple: les DIR paramètres dans la page du mixer sont utilisés pour cela. En augmentant les DIR paramètres, vous déterminez à quel point le signal entrant est alimenté directement sur les sorties principales.

D'où le nom "dir". Toutes les autres choses qui se produisent à l'intérieur de l'Ot sont complètement séparées de ceci, avec une exception importante:

Alimenter un signal entrant directement aux sorties principales est tout bien, mais pendant une performance, j'aimerais faire disparaître les entrées et sorties en direct, quand j'en ai besoin. Au cours d'une performance, cependant, je suis probablement déjà en train de jouer avec différents paramètres de tracks, des scènes, etc. Chaque fois que je veux faire disparaître les direct in, je dois me tourner vers l'écran du mixeur pour les contrôler. Pas très pratique. . .



#### 9.2.2 Track recorders

Les direct in peuvent être envoyées directement au main out, mais nous pouvons également les transmettre aux enregistreurs de pistes(track recorder) pour le sample. Schématiquement:



La conception du flux de signal(signal flow) de cette façon est assez intelligente car elle permet de fade in des direct in, indépendamment du processus de samping.

Je peux donc sampler sans entendre les entrées, ou vice versa ou toute combinaison.

#### 9.3 Sampling guidelines

The ot dispose de huit enregistreurs entièrement indépendants sur chacun de ses tracks. Les thermocouples sont construits "en haut" d'une piste et sont toujours disponibles, indépendemment de se qui se nesse sur une niete

indépendamment de ce qui se passe sur une piste.

Chacun de ces enregistreurs peut enregistrer à partir de n'importe quelle combinaison de trois sources:

Les direct in a / b et / ou c / d et / ou une troisième source, qui peuvent être définies sur n'importe quelle track, la sortie principale ou la cue out.

L'enregistrement de l'une de ces sources peut être triggé par le séquenceur ou manuellement et je ne vais pas plonger dans les détails de cela ici, puisque je terminerais réécriture du manuel Elektron. Je vous présenterai cependant quelques brèves lignes directrices:

Sequenced Sampling En configurant les triggeurs de l'enregistreur, le processus d'enregistrement est entièrement contrôlé par le séquenceur.

Le trigger des enregistreurs du séquenceur présente un avantage par rapport à sampling manuel: vous garantirez que le sampling est toujours synchronisé avec le séquenceur.

Ainsi, la coupe(trimming) ou la préparation ultérieure de sample n'est habituellement pas nécessaire.

Il peut être utilisé directement dans le reste du système sans arrêter le séquenceur.

**One shot triggers(un déclencheur déclenche)** Si vous souhaitez sampler une fois sans écraser le sample la prochaine fois que les patterns loop, vous pouvez utiliser les **One shot triggers**. Ils déclenchent l'enregistreur une fois mais pas plus.

Les déclencheurs d'un coup peuvent être réarmés à volonté, donc c'est une belle technique si vous voulez un **Sequenced Sampling**, mais seulement lorsque vous en avez besoin. Cela équivaut plus ou moins à trigger un recorder dans Ableton Live.

**Manual Sampling** En déclenchant manuellement les échantillonneurs, vous jettez les avantages temporels du **Sequenced Sampling**, mais les insuffisances de votre propre timing peuvent entraîner des accidents heureux.

**Quantized sampling** Il s'agit d'une forme de trigger manuel des enregistreurs, mais en définissant les paramètres **qrec** et **qpl** dans la page de configuration des enregistrements, les enregistreurs ne sont pas trigger instantanément, mais sont quantifiés dans le séquenceur de l'Ot. Cela équivaut plus ou moins à déclencher un enregistreur dans Ableton Live.

#### 9.4 Track recorders and the sample slot list

Alors, où ces enregistreurs écrivent-ils leur audio? Bien. . . Chaque enregistreur de track possède son propre buffer d'enregistrement dédié. Cela signifie qu'un enregistreur sur la track écrit toujours son audio pour enregistrer un recordbuffer. Ceci est câblé dans le système.

Alors, où vivent les enregistreurs? Réponse: dans la liste des samples slots. Pensez-y un moment aux implications de cela.

Comme décrit précédemment, la liste des slots des samples est disponible dans l'ensemble du projet.

En d'autres termes: une fois qu'un enregistreur écrit quelque chose à son recordbuffer, ce sample est instantanément disponible sur n'importe quelle track tenant une machine flexible dans n'importe quel pattern dans n'importe quelle bank, à condition que l'enregistreur ne remplace pas le sample. . . Et même si cela se produit, ce nouvel échantillon est, encore une fois, immédiatement disponible partout.

Enfin, quelques détails utiles sur les blocs d'enregistrement:

Éditing Il n'y a pas de différence entre un recordbuffer et regular sample slot.

Donc, tout ce qui est mis dans un tampon d'enregistrement peut être adapté à l'éditeur du sample. Cela comprend le slicing. . . **Saving to cf-card** Si vous souhaitez sécuriser un sample qui se trouve actuellement dans le recordbuffer, vous pouvez plonger dans le sample editor et vous déplacer dans la page du fichier. Vous y trouverez l'option "save and assign sample".

En sélectionnant cela, le contenu du recordbuffertampon) est écrit sur la carte cf.

Vous décidez comment il doit être nommé et après, l'Ot attribue automatiquement ce sample à un emplacements free sample slot.

Ceci est pratique: si vous souhaitez save un recorred sample de la(from) destruction et le rendre immédiatement disponible dans votre projet, c'est le chemin à parcourir.

### 9.5 Sampling techniques

Je vais maintenant décrire quelques stratégies de sampling.

### 9.5.1 Dedicating patterns (Patterns dédiés)

Le fait qu'un pattern enregistré soit instantanément disponible partout, crée de nombreuses possibilités.

On pourrait, par exemple, définir quelques record trigger dans un pattern et passer immédiatement au pattern suivant qui n'a pas ces triggers. De cette façon, un modèle pourrait être considéré comme un enregistreur fly-by:

Passez au pattern pour enregistrer ce dont vous avez besoin et laissez immédiatement au prochain pattern où (certains de) ces samples sont utilisés, entre autres,

Bien que sacrifier un pattern complet pour contenir quelques record trigger, peut sembler un gaspillage de motifs, nous en avons assez, donc ce n'est pas grave.

Bien sûr, on pourrait également utiliser des triggers en un coup pour la configuration des fly-bys de l'enregistreur.

### 9.5.2 Abusing the cue outputs (Abus des sorties cue)

Chaque track dans l'Ot peut être envoyée au sortie master out, mais aussi aux sorties Cue.

Les sorties CUE out sont indépendantes de la sortie master out.

Comme décrit précédemment, un track recorder peut enregistrer les direct in(entrée direct) ainsi qu' une troisième source.

Cette troisième source peut être n'importe quelle track, les sorties master out et la sortie cue out.

Alors. . . Configurez track recorder pour enregistrer la sortie de la cue out.

Quelle que soit la combinaison des entrées en direct in et des pistes dont vous avez besoin, vous pouvez l'envoyer en les ciblant, indépendamment de ce qui se passe sur les sorties master out. Si, par exemple, vous avez besoin d'un enregistrement du bass-drums, hihats, claps et in c / d, indiquez ces quatre, déclenchez l'enregistreur comme vous le souhaitez et vous obtenez un samples de ce que vous avez sélectionné.

Si vous avez besoin d'une forme de pré-écoute, branchez un casque et affectez le casque à la sortie de la touche dans la page du mixeur.

Cette technique de sampling est méchante et n'est pas décrite dans le manuel comme une technique distincte, mais cela fonctionne et c'est I.m.h.o. Très utilisable, donc je le présente ici.

Comme décrit précédemment, les tracks recorder et les tracks sont indépendants.

Une machine flexible fonctionnant sur une track ne mordra pas le tracks recorder sur la même piste et vice versa. . .

Il existe toutefois un moyen de configurer un tracks recorder et une machine flexible de telle sorte qu'ils se mordent de manière utile.

Je l'ai configuré comme suit:

• Configurez un tracks recorder et laissez-le enregistrer l'entrée en direct in. La longueur de l'enregistrement doit correspondre à la longueur du pattern.

• Placez un full trigger au step 1. Ce faisant, le recorder continue d'enregistrer, écrasant audacieusement le dernier enregistrement chaque fois que le pattern reboucle. Bien que l'enregistreur fonctionne, je ne peux pas entendre les entrées en direct.

• Mettre en place une machine flexible sur la même track.

• Assignez la machine flexible à son propre recordbuffer. En d'autres termes: si le recorder de la track 4 a été utilisé, nous mettons maintenant une machine flexible sur la track 4 et l'assignons à Recordbuffer 4.

• Trigger la machine flexible au step 1. En effet, c'est la même étape sur laquelle l'enregistreur à été trigger.

Nous avons mis en place une situation étrange: l'enregistreur de piste est en train d'écrire dans la mémoire tampon alors que le flexmachine est en train de lire en même temps. . .

Alors, qu'est-ce qui se passe? Eh bien, cela est documenté dans le manuel, dans une telle situation, vous entendrez les entrées en direct. Cela présente quelques avantages:

• La suppression du trigger du trigger recorder situation de lecture / écriture.

Résultat: le flexplayer joue simplement le contenu de l'enregistreur de piste. L'entrée en direct est passée.

• Que vous entendiez l'entrée en direct ou l'échantillon, ils sont tous deux affectés par les effets de la piste.

• Le volume de la piste peut être utilisé pour disparaître ou sortir de la piste comme d'habitude, qu'il joue le tampon ou passe l'entrée en direct. Oh, et j'ai-je mentionné des scènes?

• La suppression du déclencheur de l'enregistreur déclenche la situation de lecture / écriture.

Résultat: le flexplayer joue simplement le contenu de l'enregistreur de piste.

L'entrée en direct est passée.(Live input has gone.)

• Que vous entendiez l'entrée en direct in ou le sample, ils sont tous deux affectés par les effets de la track.

• Le volume de la piste peut être utilisé pour des fade in de la track comme d'habitude, si elle joue le tampon(buffer) ou passe l'entrée en direct in. Oh, et ai-je mentionné des scènes?

