Machine Translated by Google

Tutoriel SuperCollider

Chapitre 5

Par Céleste Hutchins

2005

[www.celesteh.com](http://www.celesteh.com/)

Licence Creative Commons : paternité uniquement

Machine Translated by Google

Portes et enveloppes

Nicole, la programmeuse professionnelle de SuperCollider, utilise son stock

options pour acheter des synthétiseurs analogiques. Elle a découvert que dans le monde analogique,

Il existe une enveloppe très populaire appelée ADSR. Cela signifie Attack, Decay

Maintenir la libération. Lorsque vous frappez une touche de piano, par exemple, ou que vous soufflez dans un

clarinette, le son n'atteint pas immédiatement son plein volume. Il y a un

le temps d'attaque, qui est le temps nécessaire pour que le volume atteigne son maximum. Alors, sur peut­être des instruments, en particulier le piano, le son décroît un peu à cause du

attaque initiale. Le volume atteint alors le volume de maintien . C'est la viande de

la note. (Ou tofu de la note pour les végétariens.) C'est le volume que la note

reste aussi longtemps qu'il est maintenu. Ensuite, la note se termine, mais le son ne continue pas.

mourir immédiatement. Au moment du relâchement, l'amplitude diminue jusqu'à néant, comme le

écho d'une note relâchée au piano.

De toute évidence, les enveloppes réelles de choses comme les clarinettes sont beaucoup plus complexes.

Cependant, dans le matériel analogique, chaque étage d'enveloppe est un autre circuit. C'est

cher, et le cours de l'action de sa startup doit dépasser les 100 $ avant qu'elle puisse

se permettre une enveloppe en 8 étapes. SuperCollider est un logiciel et n'a pas le

mêmes restrictions. Elle peut avoir autant d'étapes d'enveloppe que vous le souhaitez (et autant

son ordinateur peut le gérer), comme vous pouvez le voir dans le fichier d'aide d'Env. Il y en a deux

Machine Translated by Google

coûts de l’étape de l’enveloppe : l’un est l’utilisation du processeur (qui est minuscule), et le l’autre est la complexité. En tant que compositeur et programmeur, vous devez être

capable de gérer la complexité de vos sons. Nicole a travaillé

heures supplémentaires et est stressé et a appris à aimer les ADSR parce qu'ils permettent beaucoup de flexibilité mais ne sont toujours pas très complexes. Ils sont également traditionnels et ont un

un joli son rétro qu'elle creuse.

Une chose très intéressante à propos des enveloppes ADSR est qu'elles ne sont pas fixes durée. Vous n'avez pas besoin de savoir quelle sera la longueur de la note lorsque vous

démarrez­le. Vous pouvez jouer la note aussi longtemps que vous l'aimez, puis la terminer lorsque vous le souhaitez.

sont prêts. Vous contrôlez cela avec une porte. Ces termes proviennent de l'analogique

synthétiseurs, alors imaginons le synthétiseur vintage de Nicole. Elle a un clavier.

Le clavier est en fait un interrupteur. Lorsque la touche est enfoncée, l'interrupteur est

fermé et l'électricité circule. Lorsque la clé est levée, l'interrupteur s'ouvre et

l'électricité ne circule plus. La clé est une porte. Il peut être ouvert ou fermé. Lorsqu'elle appuie sur la touche, l'enveloppe démarre, parcourant les parties ADS. Lorsqu'elle retire son doigt de la touche, l'enveloppe passe sur la partie R. Un

un portail ouvert a une valeur de 0. un portail fermé a une valeur de 1.

Nicole a écrit un ADSR dans un SynthDef :

(

SynthDef.new("exemple 4", {arg out = 0, freq, amp = 0,2, gate = 1,

une, d, s, r ;

était env, le sien ;

env = EnvGen.kr(Env.adsr(a, d, s, r, amp), porte,

doneAction: 2); sin = SinOsc.ar(freq, mul:env);

Machine Translated by Google

Out.ar(out, péché);

}).envoyer(s);

)

a = Synth.new("exemple 4", [\freq, 440, \a, 0.2, \d, 0.1, \s, 0.9,

\r, 0,5, \porte, 1]);

Lorsqu'elle exécute ceci, la note commence à jouer. Quand elle est prête à recevoir la note

A la fin, elle envoie un message au synthé pour mettre le gate à 0. Elle utilise le set message.

a.set(\porte, 0);

Le synthétiseur joue jusqu'à ce qu'il lui envoie un message lui demandant de remettre le gate à zéro.

Elle peut envoyer des messages à n'importe quel paramètre en utilisant set. Essayez de lancer ceci, en faisant une pause

entre chaque commande :

a = Synth.new("exemple 4", [\freq, 440, \a, 0.2, \d, 0.1, \s, 0.9,

\r, 0,5, \porte, 1]);

a.set(\freq, 330);

a.set(\porte, 0);

N'oubliez pas que l'attaque, le maintien et le déclin ont tous lieu lorsque la porte est

ouvert, ou 1. Le déverrouillage se produit après que le portail soit éteint, ou zéro. L'enveloppe

Le générateur utilise le portail. Lorsque la porte s'ouvre, le générateur d'enveloppe commence

pour jouer via l'adsr. Lorsqu'il arrive à la partie sustain, il y reste jusqu'à ce que

la porte se ferme et joue la partie de libération de l'adsr. Quand c'est terminé, il regarde le doneAction. Puisque doneAction est 2, il supprime le

Machine Translated by Google

synthé. Nous n'avons cependant pas besoin de retirer le synthé. Nous pourrions le garder

et continuez à lui transmettre de nouvelles valeurs. Essayez d'exécuter l'exemple suivant. Se souvenir de

faites une pause entre les messages définis.

(

SynthDef.new("exemple 5", {arg out = 0, freq, amp = 0,2,

porte = 1, a, d, s, r ;

était env, le sien ;

env = EnvGen.kr(Env.adsr(a, d, s, r, amp), porte, doneAction: 0);

sin = SinOsc.ar(freq, mul:env); Out.ar(out, péché);

}).envoyer(s);

)

a = Synth.new("exemple 5", [\freq, 440, \a, 0.2, \d, 0.1, \s, 0.9,

\r, 0,5, \porte, 1]);

a.set(\porte, 0);

a.set(\freq, 330, \a, 0.2, \d, 0.1, \s, 0.9, \r, 0.5, \gate, 1);

a.set(\porte, 0);

Voyez que lorsque vous regardez la fenêtre du serveur, il est écrit "Synthés : 1. Ce synthétiseur est

toujours là, attendant que nous le remettions en place. Exécutez­le à nouveau, mais cette fois, changez le

hauteur au milieu d'une note.

Machine Translated by Google

a.set(\freq, 330, \a, 0.2, \d, 0.1, \s, 0.9, \r, 0.5, \gate, 1);

a.set(\freq, 587);

a.set(\porte, 0);

Lorsque vous souhaitez que ce synthé disparaisse, rappelez­vous que vous pouvez le libérer en

en appuyant sur la pomme­. N'oubliez pas également qu'une doneAction de 0 est la valeur par défaut.

Le patron de Nicole entend son code et l'apprécie. Le contrat qu'elle a signé dit que tout de son code SuperCollider appartient à l'entreprise, alors il lui dit de garder

travailler là­dessus, mais pour adoucir le changement de hauteur. Elle intègre un ugen appelé Slew :

(

SynthDef.new("exemple 6", {arg out = 0, freq = 440, amp = 0,2,

porte = 1, a, d, s, r ;

var env, péché, massacre;

env = EnvGen.kr(Env.adsr(a, d, s, r, amp), gate);

slew = Slew.kr (fréquence, 4000, 4000); sin = SinOsc.ar(slew, mul:env); Out.ar(out, péché);

}).envoyer(s);

)

a = Synth.new("exemple 6", [\freq, 440, \a, 0.2, \d, 0.1, \s, 0.9, \r,

0,5, \porte, 1]);

Machine Translated by Google

a.set(\freq, 587);

a.set(\freq, 330);

a.set(\porte, 0);

Slew est un Ugen. Le .kr signifie qu'il fonctionne au taux de contrôle. Slew ne le fait pas produire du son. Il produit ce que, sur les synthétiseurs de Nicole, on appellerait un

"tension de commande". C'est un nombre qui change lentement (bien en dessous du niveau audio). taux) pour modifier un autre Ugen.

Slew, prend quelques arguments, le premier est la chose à massacrer, les deux suivants sont la pente ascendante et la pente descendante, puis il faut mul et ajouter.

Nous pouvons faire basculer n'importe lequel des arguments vers un SynthDef.

Vous pouvez voir sur Slew que les Ugens peuvent se modifier. Vous pouvez passer un contrôlez le taux ugen ou un taux audio ugen dans l'entrée de et audio ugen et utilisez avec n'importe quel argument tel que la fréquence, la phase, le mul, l'ajout, peu importe.

Lorsque vous expérimentez cela, assurez­vous de ne pas utiliser l'argument add de l'oscillateur qui est le dernier d'une chaîne, car cela ajouterait

quelque chose appelé biais DC. Cela peut endommager vos haut­parleurs. Le biais DC signifie que votre forme d'onde n'est pas centrée sur zéro ; il est décalé pour se centrer autour d'un autre nombre. Cela signifie que les aimants de vos enceintes doivent tirer vers

ce nombre augmente et cela stresse vos enceintes. A part ça, essaye de changer

ugens. Envoyez des oscillateurs comme arguments aux oscillateurs. Nous parlerons davantage du des noms pour les différentes façons de procéder à l’avenir.

Poêle

Machine Translated by Google

Le patron de Nicole est content, mais il souhaite que le son soit stéréo. Un peu comme une poêle Bouton sur une table de mixage, l'ugen Pan2 peut déplacer un signal entre deux

chaînes. Alors elle ajoute pan :

(

SynthDef.new("exemple 7", {arg out = 0, freq = 440, amp = 0,2,

porte = 1, a, d, s, r, pan = 0 ;

était env, sin, slew, panner ;

env = EnvGen.kr(Env.adsr(a, d, s, r, amp), gate); slew = Slew.kr (fréquence, 4000, 4000);

sin = SinOsc.ar(slew, mul:env); casseroles = Pan2.ar(sin, casserole); Out.ar(out, panner);

}).envoyer(s);

)

Pan2.ar est un constructeur de débit audio. Pan est le débit audio car il produit un signal sonore. La première entrée est le signal à analyser. La deuxième entrée est le

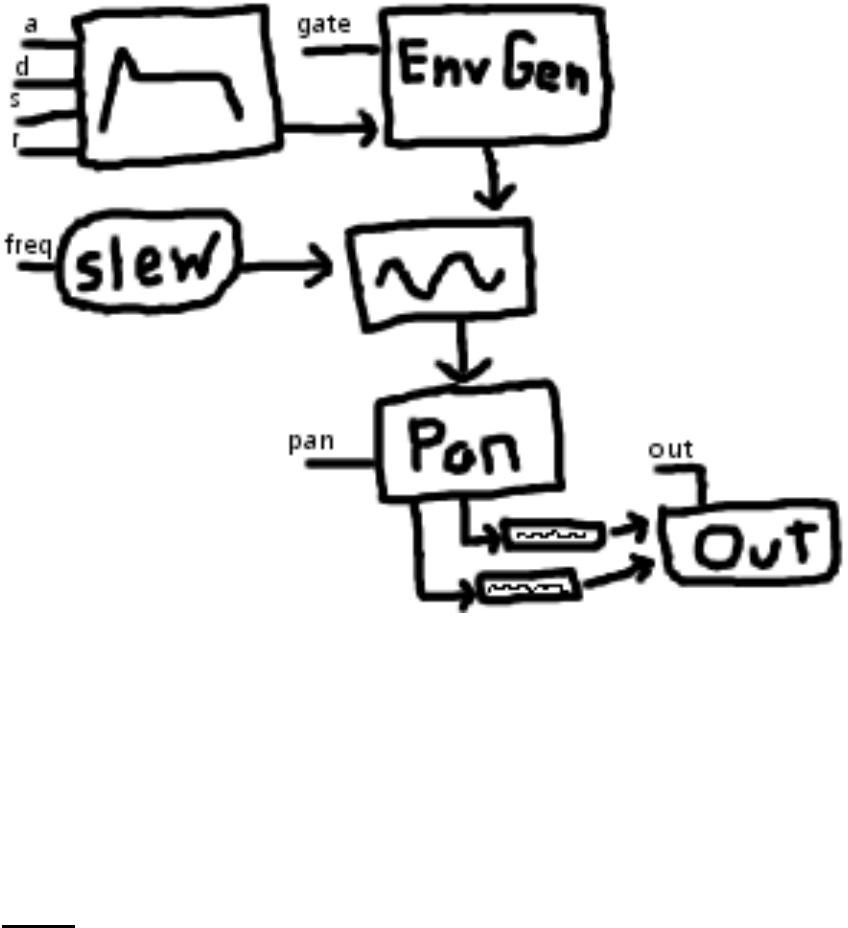
position vers laquelle effectuer un panoramique. ­1 est l'extrême gauche, 0 est le milieu et 1 est l'extrême droite. Un numéro dans

entre ces chiffres, il se déplace entre ces positions. Pan fonctionne par

créant un tableau d'ugens. Le premier élément du tableau est le canal gauche. Le le deuxième élément est le canal droit.

Combien y a­t­il d'ugens dans ce SynthDef ? Nicole le dessine dans un carnet :

Machine Translated by Google

Les sorties des opérations ugen sont des ugens. Vous pouvez voir qu'il y a 8 UGens dans

notre SynthDef. Six d'entre eux ont été créés avec des constructeurs et Pan en a créé deux.

Interface graphique

Nicole utilise son croquis pour suivre ce qui se passe. Cependant, tout cela est défini

les messages ne sont pas très conviviaux. Elle veut une interface graphique pour faciliter ses synthés

utiliser. Une GUI est une interface utilisateur graphique. Elle recherche une classe appelée Display.

Le fichier d'aide lui indique que « Display est un ControlEnvironment qui fournit une interface graphique ».

Après avoir lu le fichier d'aide, elle écrit un simple Display :

(

Display.make({arg thisDisplay, freq, amp, pan;

freq.spec\_(\freq); amp.spec\_(\amp); pan.spec\_(\pan);

Machine Translated by Google

thisDisplay.synthDef\_({arg freq, amp, pan;

c'était le sien, les casseroles ;

sin = SinOsc.ar(freq, mul: amp); casseroles = Pan2.ar(sin, casserole); Out.ar(0, casseroles);

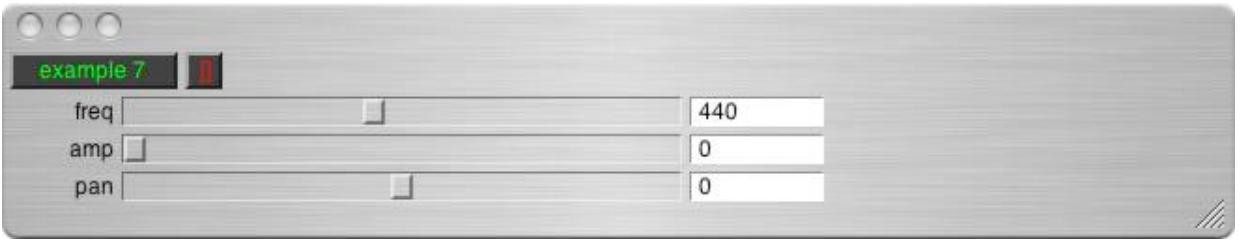
}, [\freq, freq, \amp, amp, \pan, pan]);

thisDisplay.name\_("exemple 7");

}).montrer;

)

Une nouvelle fenêtre s'ouvre :



Appuyez sur le bouton vert « exemple 7 » pour démarrer l’affichage. Le bouton devrait devenir jaune lorsqu'il fonctionne. Il ne fait aucun bruit car l'ampli est refusé. Utilisez votre souris pour parcourir le fader pour augmenter le amplitude.

Display.make est un constructeur. Cela prend une fonction car c'est un argument. Le la fonction prend un nombre arbitraire d’arguments. Le premier est l'affichage

objet renvoyé par le constructeur. Nicole l'a appelé "thisDisplay" parce qu'il

fait littéralement référence à cet affichage. (c'est un mot réservé dans SuperCollider.) Le les arguments supplémentaires sont ControlValues. Elle a appelé ses ControlValues "freq", "ampli" et "pan".

Machine Translated by Google

Les ControlValues ont une propriété appelée spec. Les messages qui se terminent par un

les traits de soulignement sont appelés messages de définition . Mettez en surbrillance le mot ControlValue et puis appuyez sur apple­j pour voir le code source de la classe. Les classes SuperCollider sont écrit en SuperCollider. Regardez en haut du fichier, sous les commentaires. "var

<spécification ; » signifie que ControlValue a une variable interne appelée spec et que

les utilisateurs peuvent modifier la valeur de cette variable avec « spec\_ ». Nous reviendrons plus au fonctionnement des cours plus tard. Dans ce cas, Nicole utilise des valeurs prédéfinies,

elle passe donc quelques littéraux (\freq, par exemple) comme arguments, lui indiquant lequel valeur prédéfinie à utiliser.

Ensuite, elle a défini le SynthDef associé à thisDisplay. Ce setter envoie des messages prend deux arguments. L'une est une fonction que vous passeriez à un

SynthDef. Le deuxième argument est le tableau que vous utiliseriez pour créer un nouveau instance de Synth. \freq correspond à la fréquence de l'argument SynthDef. \amp correspond l'ampli d'argument SynthDef. la fréquence (sans la barre oblique) correspond à la ControlValue fréquence. amp correspond à l’ampli ControlValue. Ce qui se passe là, c'est qu'elle est

dire au preset de créer un SynthDef et quand il crée un Synth, de l'utiliser Variables ControlValue pour contrôler les arguments Synth.

Elle donne ensuite un nom à l'affichage, en l'appelant « exemple 7 ».

Ensuite, elle prend l'affichage créé par le constructeur et transmet un message à il lui dit de se montrer.

Nicole se rend compte que c'est un excellent moyen de tester les différentes valeurs des synthés. ressemble. Elle propose un moyen de tester ses enveloppes ADSR :

(

Display.make({arg thisDisplay, freq, amp, pan, a, d, s, r, gate;

Machine Translated by Google

était un bouton ;

freq.spec\_(\freq); amp.spec\_(\amp); pan.spec\_(\pan);

une.sp(0,5, 0, 10, 0);

d.sp(0,5, 0, 10, 0);

s.spec\_(\amp); r.sp(0,5, 0, 10, 0);

bouton = ControlButton (porte); bouton.states\_([["off"], ["on"]]);

thisDisplay.guiItems\_([ thisDisplay.clientgroup,

fréquence, ampli, panoramique, a, d, s, r,

bouton

]);

thisDisplay.synthDef\_({arg freq, amp, pan, a, d, s, r, gate;

était péché, panner, env;

env = EnvGen.kr(Env.adsr(a, d, s, r, amp), gate); sin = SinOsc.ar(freq, mul: env);

casseroles = Pan2.ar(sin, casserole);

Out.ar(0, casseroles);

}, [\freq, freq, \amp, amp, \pan, pan, \a, a, \d, d,

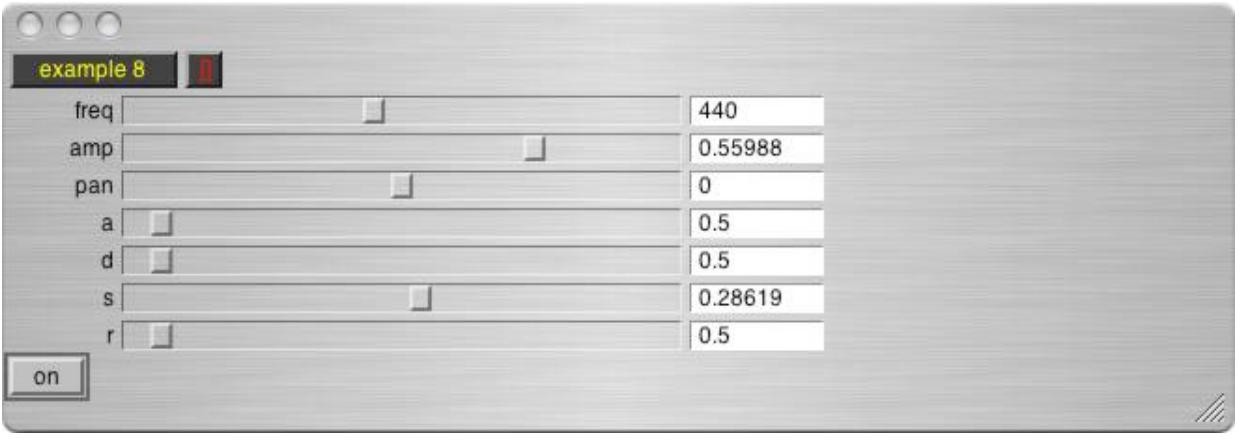
\s, s, \r, r, \porte, porte]);

thisDisplay.name\_("exemple 8");

Machine Translated by Google

}).montrer;

)



Montez l'ampli et le s. Cliquez sur le bouton « exemple 8 » pour le démarrer.

Cliquez ensuite plusieurs fois sur le bouton marche/arrêt pour faire fonctionner le portail. Comment Nicole

tu sais comment écrire ça ?

Elle a consulté le fichier d'aide de ControlValue et a découvert tous les différents ControlSpecs par défaut qu'elle pourrait utiliser. Il n'y en avait aucun qui était approprié pour les timings de l'ADSR, elle a donc utilisé la méthode sp pour en créer avec un valeur initiale de 0,5, une valeur minimale de 0 et une valeur maximale de 10. Elle

s'est souvenue que s est l'amplitude du sustain, elle a donc utilisé la valeur par défaut de /amp.

Le fichier d'aide Display mentionne ControlButton et donne un exemple

code. Elle a copié le code, créant un bouton et lui donnant deux états. Elle a mis

le premier état à désactiver, car il est à la position du tableau 0 et une porte hors tension est 0.

Le fichier d'aide Display explique que guiItems « est un tableau d'objets qui répondent à la méthode draw(window). En copiant l'exemple de code, elle a inclus

thisDisplay.clientgroup dans le tableau guiItems, puis elle a répertorié tous les autres ControlValues et son bouton.

Machine Translated by Google

Elle a mis l'enveloppe adsr dans le SynthDef, puis a ajouté l'adsr et la porte à

le tableau sur les arguments de synthétiseur et les valeurs de contrôle.

# Problèmes

1. Écrivez un SynthDef qui modifie l'amplitude. Utilisez set pour lui en envoyer changements d'amplitude. Faites de même pour la poêle.
2. Esquissez votre SynthDef. Est­ce que le nombre d'UGens affichés dans le

La fenêtre du serveur correspond au numéro que vous attendiez de votre croquis ?

1. Écrivez une routine qui crée plusieurs synthés et les contrôle avec set messages. Les synthés peuvent être des instances du même SynthDef.
2. Si vous disposez d'une configuration à quatre canaux, expérimentez avec Pan4.ar. Comment

Pan4.ar(sin, x, y) diffère de Pan2.ar(Pan2.ar(sin, x), y) ?

1. Créez un affichage en utilisant différents outils, notamment Saw et Pulse. Utiliser un ControlSpec pour contrôler la largeur de l'impulsion.

# Projet

Vous pouvez mélanger plusieurs signaux en les additionnant tous et en les divisant par le nombre de signaux. Par exemple:

résultat = (sine1 + sinus2 + sinus3 + sinus4 + sinus5) / 5 ;

Créez un affichage avec plusieurs oscillateurs, chaque fréquence et amplitude

contrôlés séparément. Écrivez un morceau d'une ou deux minutes que vous jouez en bougeant les curseurs.