



LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de la Culture, de l'Enseignement
supérieur et de la Recherche

Les séries dodécaphoniques vues sous MaxMSP

Arthur Stammet

Janvier 2007

Editions Noise Watchers

1, rue de la Forêt, L-1534 Luxembourg
now@pt.lu - www.nowa.lu

© Janvier 2007

Tables

Table des matières

Les séries dodécaphoniques vues sous MaxMSP	1
Tables	3
Table des matières	3
Table des illustrations	3
Les séries dodécaphoniques vues sous MaxMSP	4

Table des illustrations

Figure 1 : 12-Tone Series Random Generator (février 2006) - vue lock	4
Figure 2 : 12-Tone Series Random Generator (extrait) - vue edit	4
Figure 3 : Une gamme chromatique préparée pour servir dans un objet « Picture based control »	5
Figure 4 : L'inspecteur du « Picture based control » alimenté avec la gamme chromatique.....	5
Figure 5 : Note Slider Tutorial (test des relations entre les nombres et les résultats visuels obtenus)	5
Figure 6 : Le module « dodeca » (février 2006) - vue lock	6
Figure 7 : Le patch « Dodeca Tutor » (février 2006) - vue lock	6
Figure 8 : Le patch « Dodeca Note Tutor » (janvier 2007) - vue lock	7
Figure 9 : Le patch « Dodeca Note Tutor 2 » avec le patcher CreateSeries (janvier 2007) - vues lock.....	7

Les séries dodécaphoniques vues sous MaxMSP

Tout compositeur voulant utiliser des séries dodécaphoniques est confronté à une double difficulté : trouver toutes les notes et les écouter. Pour venir à bout de ce défi il suffit de bien savoir compter et de jouer ces notes sur un instrument, ou bien : utiliser mon patch !

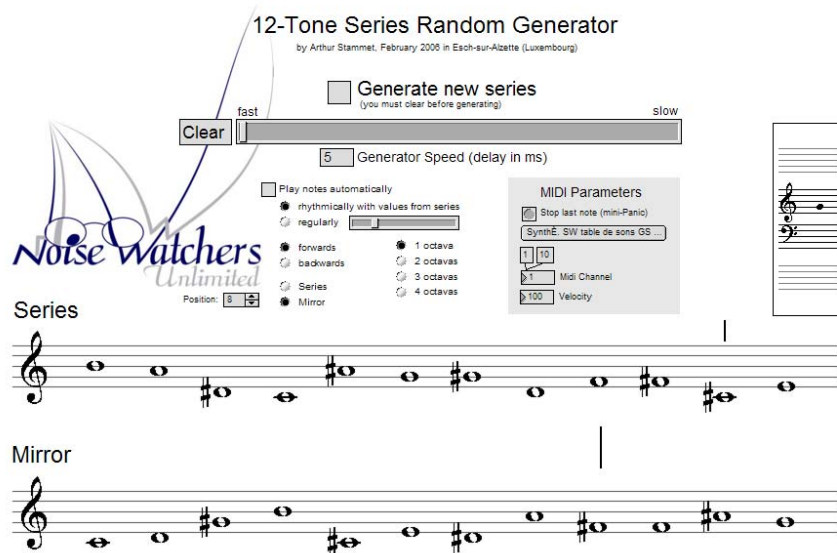


Figure 1 : 12-Tone Series Random Generator (février 2006) - vue lock

Réalisé avant tout pour faire quelques tests sommaires de certains objets dans MaxMSP, ce patch est rapidement devenu très (trop !) complexe, tel que l'illustre l'extrait suivant :

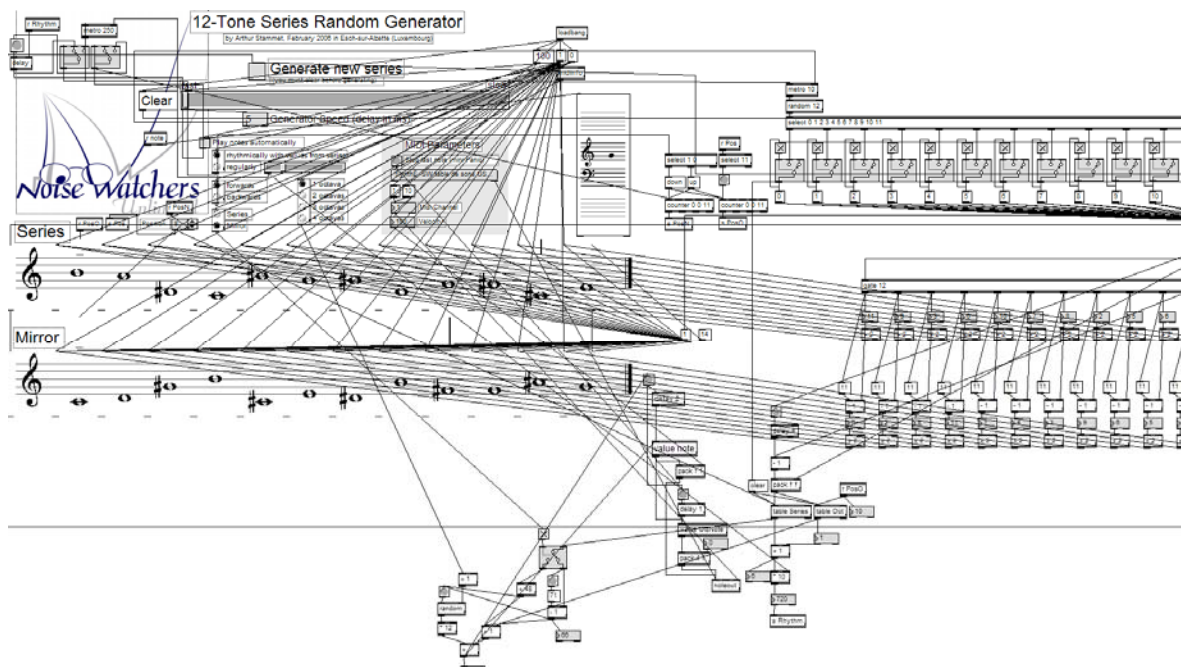


Figure 2 : 12-Tone Series Random Generator (extrait) - vue edit

Ce patch peut maintenant servir de logiciel complet et fini qui fonctionne bien et fait que j'attends de lui : construire rapidement des séries dodécaphoniques qu'il illustre tant sur le plan visuel qu'auditif dans leurs formes originales et leurs permutations (miroir et écrevisse). Un mode de transposition utilisant les degrés de la série elle-même est aussi prévu et fonctionnel.

En tant que programme il n'est qu'un prototype difficile à gérer et à amplifier¹ : trop de câbles, trop de groupes d'objets.

Il fait en outre partie de ma recherche sur la notation en ligne dont il présente un premier aspect : l'illustration d'une série de nombres à l'aide des notes qui leur correspondent.

Pour y arriver il faut utiliser une panoplie plus ou moins large de logiciels parmi lesquels Max ne présente que le stade final.

Dans un logiciel de notation musicale, en l'occurrence Capella 2004, j'ai tout d'abord créé une gamme chromatique que j'ai ensuite exportée au format TIFF avant de la trafiquer dans un logiciel de traitement d'image (Photoshop) pour lui donner les dimensions dont j'avais besoin pour les utiliser en tant que « Picture based control » (objet Max). Voici l'image obtenue :



Figure 3 : Une gamme chromatique préparée pour servir dans un objet « Picture based control »

Le petit patch « Note Slider Tutorial » sert de test pour les images correspondant aux valeurs reçues dans l'entrée du « Picture based control » (pictctrl) paramétré comme suit :

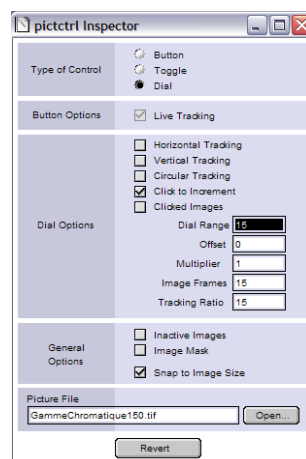


Figure 4 : L'inspecteur du « Picture based control » alimenté avec la gamme chromatique

Voici les résultats obtenus en envoyant des nombres dans son entrée :

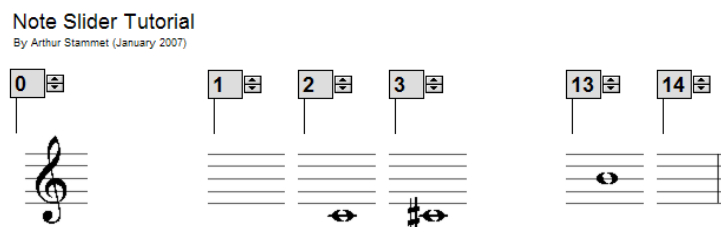


Figure 5 : Note Slider Tutorial (test des relations entre les nombres et les résultats visuels obtenus)

Il est dorénavant possible de copier-coller cet objet dans ses travaux ultérieurs.

¹ Il s'agit ici d'une espèce de « code spaghetti » dans un environnement orienté vers des objets. Il est impératif, dans le développement de patches Max d'utiliser des « sub-Patchers » (correspondant aux procédures dans les langages de programmation), des « externals » (correspondant aux bibliothèques dans les langages de programmation) et des objets « send » et « receive » (correspondant aux variables dans les langages de programmation).

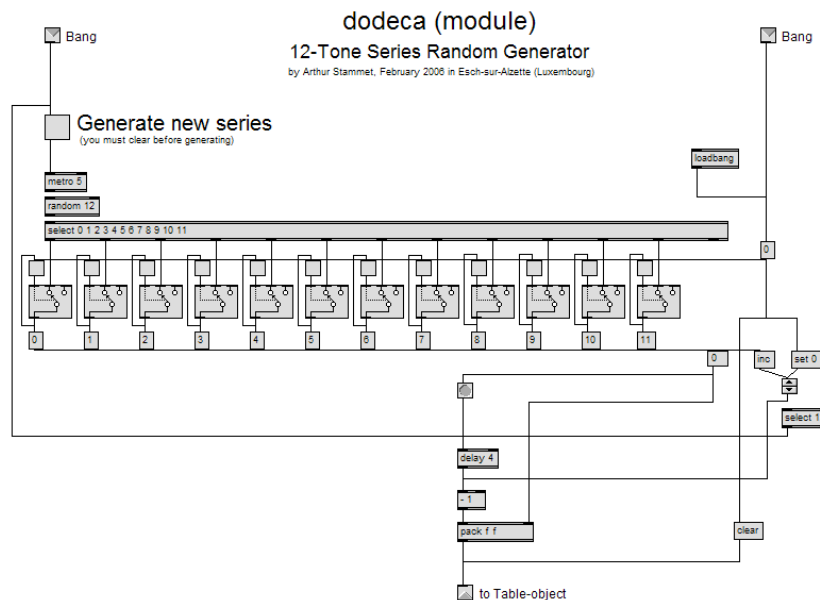


Figure 6 : Le module « dodeca » (février 2006) - vue lock

Ce patch émet une série de valeurs allant de 0 à 11 parmi lesquelles aucune répétition n'est possible. Ce principe pouvant être appliqué à des séries dodécaphoniques, il est aisé de construire par la suite des modules similaires émettant des séries de nombres dont le total ne sera pas 12 mais n'importe quelle autre valeur inférieure ou supérieure à ce chiffre.

Comme toujours je développe des petits tutoriaux pour mes modules. En voici un qui utilise un objet « table » pour stocker les valeurs obtenues² :

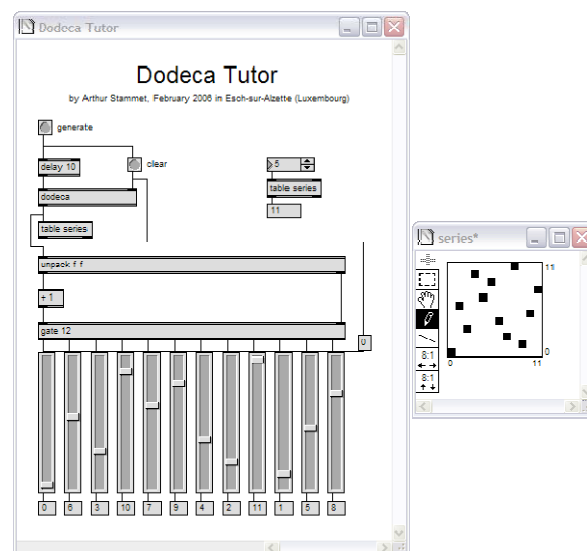


Figure 7 : Le patch « Dodeca Tutor » (février 2006) - vue lock

² J'ai choisi de ne pas intégrer une table dans mon module pour permettre à l'utilisateur d'en utiliser plusieurs qui peuvent différer entre elles en les nommant différemment.

Mon deuxième patch pédagogique illustre l'aspect d'une notation en ligne utilisant les technologies développées ci-dessus. Une brève séance de copier-coller alliée à une analyse des données accumulées ci-dessus fut nécessaire pour obtenir de bons résultats parfaitement prévisibles :

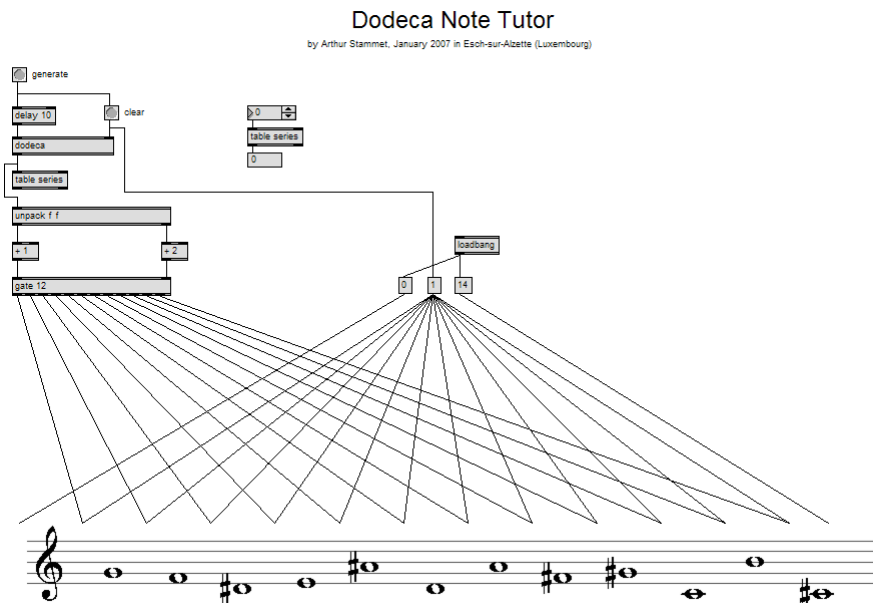


Figure 8 : Le patch « Dodeca Note Tutor » (janvier 2007) - vue lock

Un petit travail supplémentaire, regroupant le moteur de ce patch au sein d'un subpatcher me permit de construire en peu de temps une version très élaguée de ce même patch :

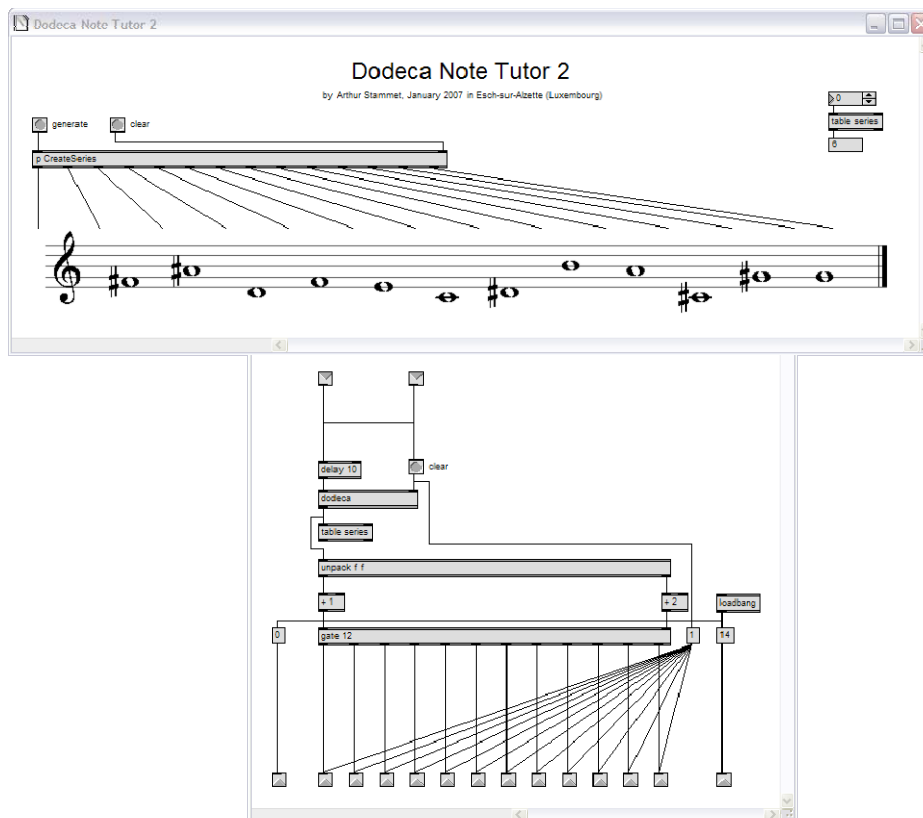


Figure 9 : Le patch « Dodeca Note Tutor 2 » avec le patcher CreateSeries (janvier 2007) - vues lock