Granulies

SOFTWARE FOR GRANULAR SYNTHESIS

MADE WITH PYTHON

Μιχάλης Ξηράκης | Προγραμματισμός Εφαρμογών Ήχου (εργασία εξαμήνου-συνοπτική αναφορά) | 2022

Τι είναι το Granulies ;;;

Λογισμικό το οποίο διαβάζει ένα οποιοδήποτε αρχείο ήχου, που φορτώνει ο χρήστης και πραγματοποιεί «κοκκώδη σύνθεση», παίρνοντας τους «κόκκους» μέσα από το αρχείο αυτό.

- Δίνει τη δυνατότητα στο χρήστη να επιλέξει κάποιες παραμέτρους της κοκκώδους σύνθεσης, όπως το μέγεθος των κόκκων, την παραθυρική συνάρτηση (περιβάλλουσα), βάσει της οποίας οι κόκκοι μορφοποιούνται, τη συνέλιξή τους με κάποιον ηλεκτρικό παλμό, όπως το πριονωτό σήμα, για τη δημιουργία ηχητικών εφέ, καθώς και το να μπορεί ο χρήστης να ρυθμίζει απλά τις παραμέτρους σε τυχαία (random) λειτουργία, ώστε το αποτέλεσμα να είναι πιο «αυθόρμητο».
- Η κοκκώδης σύνθεση, που πραγματοποιείται κάθε φορά μπορεί αυτόματα να καταγραφεί και να αποθηκευτεί με διαφορετικό κάθε φορά όνομα, χωρίς να υπάρξει διαγραφή κι αντικατάσταση (replace) του προηγούμενου αρχείου.
- Το πρόγραμμα έχει επαναληπτική λειτουργία και κάθε φορά η ταχύτητα δημιουργίας και καταγραφής της κοκκώδους σύνθεσης μπορεί να διαφέρει, ανάλογα το μέγεθος του αρχείου από το οποίο λαμβάνονται οι κόκκοι και της παραμέτρους που εισάγει ο χρήστης. Για παράδειγμα ένα αρχείο κοκκώδους σύνθεσης θα είναι πολύ μικρότερο κι άρα λιγότερος ο χρόνος δημιουργίας του για μέγεθος κόκκου 45 δειγμάτων και χωρίς συνέλιξη με κάποιο ηλεκτρικό σήμα, απ' ότι για μέγεθος κόκκου 4410 δειγμάτων με συνέλιξη.

ΔΟΜΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Εικόνα 1.Το Loading Screen του προγράμματος

Με βάση τον γενικό ορισμό της κοκκώδους σύνθεσης, ως μια τέτοια σύνθεση μπορεί να χαρακτηριστεί ένα ηχητικό αρχείο, του οποίου τα στοιχεία (κόκκοι) λαμβάνονται από ένα ή περισσότερα βασικά αρχεία ήχου, αφού αυτά «τεμαχιστούν» σε μικρά κομμάτια (κόκκους) των οποίων η διάρκεια κυμαίνεται από 1 έως 100 ms και στο τέλος έχοντας αυτά υποστεί κάποια επεξεργασία, ενώνονται τυχαία, δίνοντάς μας νέα σύνθεση. Στον πηγαίο κώδικα υπάρχει η συνάρτηση

get_random_part(), η οποία παίρνει ως ορίσματα ένα ηχητικό σήμα, το μήκος των κόκκων, την παραθυρική συνάρτηση (περιβάλλουσα) που θέλουμε να έχουν, τον παλμό με τον οποίο αυτοί συνελίσσονται, καθώς κι το αν αυτοί συνελίσσονται και τέλος την τυχαιότητα για όλες τις παραπάνω παραμέτρους. Ως μέγεθος του κόκκου ορίζουμε ένα κομμάτι του βασικού αρχείου από 45 έως 4410 δείγματα, καθώς το βασικό αρχείο είναι γραμμένο με ρυθμό δειγματοληψίας στα 44100 Hz κι άρα έχουμε: 45/44100 = 0.001 s = 1 ms, 4410/44100= 0.1 s = 100 ms, που είναι αντίστοιχα η ελάχιστη και η μέγιστη διάρκεια του κόκκου, βάσει του ορισμού. Σε κάποιες ελάχιστες περιπτώσεις το μέγεθος του κόκκου ενδεχομένως να είναι μικρότερο, αφού οι κόκκοι λαμβάνονται τυχαία από οποιοδήποτε σημείο του σήματος κι αν ληφθούν πολύ κοντά από το τέλος του, το μέγεθός τους μπορεί να είναι μικρότερο των 45 samples έως και 0 σε μια σπάνια περίπτωση.

Εικόνα 2. Η κονσόλα του προγράμματος με χρήση της βιβλιοθήκης γραφικών Tkinter

Στο πρόγραμμα έχει χρησιμοποιηθεί μέρος του κώδικα του μαθήματος 7 (class07), λαμβάνοντας έτοιμες συναρτήσεις, όπως η συνάτηση make_sine_with_adsr(), η οποία δημιουργεί ένα ημιτονοειδές σήμα τυχαίας συχνότητας, πλάτους και διάρκειας, με μια τυχαία περιβάλλουσα τύπου ADSR, ώστε το σήμα αυτό στη συνέχεια να συνελιχθεί με τους κόκκους, αλλοιώνοντας κατά κάποιο τρόπο το αρχικό περιεχόμενο τους και προσδίδοντας στην τελική σύνθεση κάποιο εφέ, όπως ο ήχος από κάποιο lazer, σταγόνες νερού, κ.λ.π. Τέλος ένας τρόπος για να πάρουμε από τη μια μεριά κόκκους από τυχαία σημεία του αρχικού σήματος, αλλά από την άλλη να μπορούμε να καθορίσουμε, ότι οι κόκκοι θα προέρχονται απ’ όσο το δυνατόν πιο πολλές διαφορετικές χρονικές στιγμές του βασικού αρχείου, είναι να χωρίσουμε το αρχικό μας σήμα σε blocks των N=2048 δειγμάτων και με ένα ρυθμό επικάλυψης hop=N/2, να το «διαπεράσουμε» λαμβάνοντας 2 κόκκους από κάθε block, αντί απ’ όλο το αρχείο, έτσι ώστε να εξασφαλίσουμε όσο το δυνατόν πιο διαφορετικούς κόκκους από διαφορετικές χρονικές στιγμές του αρχικού σήματος για ένα πιο ενδιαφέρον και ίσως «καθορισμένο» αποτέλεσμα. Το μέγεθος τόσο για το block, όσο και για την επικάλυψη (hop) είναι τυχαίο και καθορίζεται πιο πολύ στοχαστικά, βάσει του μεγέθους του αρχικού σήματος από το οποίο λαμβάνουμε τους κόκκους, αφού μικρότερο block size, αποφέρει περισσότερους κόκκους, κάνοντας το τελικό αρχείο μεγαλύτερο σε όγκο και απαιτώντας από το σύστημα περισσότερο χρόνο μέχρι να ολοκληρωθεί η σύνθεση.