

Report 2^{ης} εργασίας Τεχνολογίας Πολυμέσων:

Αλέξανδρος Βεντούρας AM: 3160013 Email: alvent98@gmail.com

Δικαία Σωτηροπούλου AM: 3160172 Email: dikeasot@hotmail.com

Το συγκεκριμένο έγγραφο έχει ως στόχο να δώσει ορισμένες χρήσιμες πληροφορίες αναφορικά με την δεύτερη εργασία που εκπονήθηκε στα πλαίσια του μαθήματος «Τεχνολογία Πολυμέσων» κατά το Χειμερινό εξάμηνο του 2020.

A. Οδηγίες εγκατάστασης :

Προκειμένου να εκτελεστεί το λογισμικό, απαιτείται η εγκατάσταση της Python 3, διότι με τη βοήθειά της θα δημιουργηθεί ο απαραίτητος εξυπηρετητής. Έπειτα, αλλάζουμε το path της εκτέλεσης, στο path που βρίσκονται όλα τα χρήσιμα αρχεία της εργασίας, τα οποία έχουμε ήδη κάνει unzip (με την εντολή `cd path/to/softwarefiles`). Στη συνέχεια, αφού έχουμε εντοπίσει το path στο οποίο βρίσκεται το `python.exe` αρχείο, εκτελούμε την παρακάτω εντολή: `path/to/python.exe -m http.server 8000`, και ο εξυπηρετητής που θα φιλοξενήσει την εφαρμογή μας είναι πλέον έτοιμος προς χρήση.

Ακολούθως, ανοίγοντας έναν οποιοδήποτε φυλλομετρητή (αλλά κατά προτίμηση έναν από τους chrome ή firefox, οι οποίοι επιβεβαιωμένα εκτελούν την εν λόγω εφαρμογή), πληκτρολογούμε “ `http://localhost:8000/cdplayer.html`”, και εάν έχουμε ολοκληρώσει τις διαδικασίες που αναφέρονται στα παραπάνω βήματα, η ιστοσελίδα αναμένεται να φορτωθεί επιτυχώς.

Εναλλακτικά, εάν δεν υπάρχει η Python εγκατεστημένη στον υπολογιστή, είναι δυνατή η εκτέλεση του προγράμματος με τη βοήθεια του extension Live Server που παρέχεται δωρεάν από το πολυεργαλείο επεξεργασίας κώδικα Visual Studio Code. Για να σηκωθεί ο server, απαιτείται να επιλέξουμε το “open with Live Server”, μέσα από το συγκεκριμένο code editor.

B. Οδηγίες χρήσης:

Η εν λόγω εργασία προσομοιώνει ένα αναλογικό synthesizer. Ο χρήστης μπορεί να πατήσει ένα πλήκτρο από τις παρεχόμενες οκτάβες, να επιλέξει να αλλάξει οκτάβα (παρέχονται τόσο οι χαμηλότερες όσο και οι υψηλότερες συχνότητες), και να τροποποιήσει την ένταση του ήχου, ή την ισορροπία μετάδοσης του σε δύο ακουστικά ή ηχεία.

Επιπλέον, μπορεί να αλλάξει τα features του ADSR envelope, μέσα από τους εννιά παρεχόμενους range sliders (για το attack, το decay, και το release), οι οποίοι ρυθμίζουν την ένταση καθώς και το χρόνο έναρξης και λήξης κάθε χρονικής συνισταμένης της εν λόγω νότας. Να σημειωθεί ότι τα χαρακτηριστικά του sustain δεν είναι δυνατό να αλλαχτούν άμεσα από το χρήστη, αλλά αυτό γίνεται έμμεσα μέσα από τα άλλα τρία features, μιας και το sustain δεν προκαλεί κάποια μεταβολή στον ήχο.

Επιπροσθέτως, μπορεί να επιλέξει το είδος της κυματομορφής του κυρίως oscillator, και του LFO, όπως και να ρυθμίσει το detune του πρώτου και τη συχνότητα του δεύτερου. Παράλληλα, μπορεί μέσα από δύο *gainNode* να ρυθμίσει το εάν και κατά πόσο επιδρούν στο τελικό ηχητικό αποτέλεσμα τα δύο effects, το tremolo και το vibrato.

Τέλος, δίνεται η ευκαιρία στο χρήστη να αλλοιώσει περεταίρω τον ήχο, επηρεάζοντας κατάλληλα τις ιδιότητες ενός *BiquadFilterNode* (συγκεκριμένα, την έντασή του, τη συχνότητά του, καθώς και το quality).

Γ. Επιμέρους συνεισφορά:

Η παρούσα ομαδική εργασία υλοποιήθηκε βάσει των αρχών ευέλικτων διαδικασιών ανάπτυξης κώδικα (agile processes), και συγκεκριμένα της διαδικασίας extreme programming, βάσει της οποίας ο κώδικας συγγράφεται κατά ζεύγη (pair programming). Με άλλα λόγια, εναλλάξ τότε το ένα και τότε το άλλο μέλος της ομάδας γράφει τον κώδικα και το άλλο μέλος επιβλέπει προσεκτικά προκειμένου να αποφεύγονται τυχόν λάθη που μπορεί να οδηγήσουν αργότερα σε αδιέξοδο. Κατ' αυτόν τον τρόπο, δεν χωρίστηκε η εργασία σε τμήματα, αλλά συγγράφτηκε με ομαδικό πνεύμα και αυτό έχει ως αποτέλεσμα να είναι γνωστός και οικείος ο κώδικας στο σύνολό του και από τα δύο μέλη της ομάδας.

Δ. Χρησιμοποιούμενες τεχνολογίες:

Για την εκπόνηση της δεύτερης εργασίας χρησιμοποιήθηκε το Web Audio API, το οποίο επιτρέπει με σχετικά μεγάλη ευκολία τη διαχείριση του ήχου. Αναφορικά με τα χρησιμοποιούμενα φίλτρα, αυτά ανήκουν σε διάφορες κλάσεις:

Πρώτον, στην κλάση *GainNode*, η οποία δίνει τη δυνατότητα αυξομείωσης της έντασης ενός ήχου, καθώς και τη διαχείριση των effects του LFO (tremolo και vibrato) και των εντάσεων του ADSR envelope. Δεύτερον, στην κλάση *StereoPannerNode*, που επιτρέπει την ελεγχόμενη εκτροπή διαφορετικών ποσοστών της έντασης του εκπεμπόμενου ήχου σε δύο ηχεία ή δύο ακουστικά. Τρίτον, στην κλάση *Oscillator*, η οποία παράγει συνθετικό ήχο και για αυτό κατέχει κομβικό ρόλο στην όλη λειτουργία της εφαρμογής (τόσο τον κύριο ήχο, όσο και τον ήχο που προκαλείται από τον LFO). Μία ελάχιστονα κατηγορία αποτελεί αυτή του *WaveShaper*, η οποία κανονικοποιεί την ένταση του ήχου στο tremolo effect.

Τέλος, χρησιμοποιήθηκαν και φίλτρα της κλάσης *BiquadFilterNode*, τα στιγμιότυπα της οποίας μπορούν να απομονώσουν ήχους ενός συγκεκριμένου εύρους συχνοτήτων, και μέσω κατάλληλα ρυθμιζόμενων παραμέτρων να αλλοιώσουν αυτές τις συχνότητες ή και την έντασή τους.

Ε. Χρησιμοποιούμενο λογισμικό:

Προκειμένου να συνταχθεί η συγκεκριμένη εργασία, όπως ειπώθηκε και στην προηγούμενη ενότητα, έγινε χρήση του Web Audio API. Επικουρικά, αξιοποιήθηκαν σε μεγάλο βαθμό οι δυνατότητες που προσφέρει η λειτουργικότητα της JavaScript, η οποία σε συνδυασμό με τη CSS έδωσαν μια ιδιαίτερη μορφή στη δομή του κώδικα, γραμμένου σε HTML. Η όλη εργασία σχεδιάστηκε αποκλειστικά με το πρόγραμμα Visual Studio Code, σε λειτουργικό

σύστημα Windows, και σαν κύριος φυλλομετρητής χρησιμοποιήθηκε ο Chrome, αν και έγιναν ορισμένοι έλεγχοι διαλειτουργικότητας στο Firefox.

ΣΤ. Πηγές:

Για τη συγγραφή της εργασίας αυτής, χρησιμοποιήθηκαν διάφορες πηγές, όπως ένα παράδειγμα προσομοιωτή αναλογικού synthesizer¹, το οποίο χρησίμευσε στη γενικότερη κατανόηση θεωρητικών εννοιών που έχουν να κάνουν με την παραγωγή συνθετικού ήχου καθώς και των διάφορων εφέ που μπορούν να τον αλλοιώσουν. Σχετικά με το σχεδιασμό της διεπαφής (το σχεδιαστικό κομμάτι των πλήκτρων), χρήσιμες πληροφορίες πάρθηκαν από ένα παράδειγμα το οποίο δεν έκανε χρήση της Javascript, παρά μόνο της HTML και της CSS². Αναφορικά με τον LFO, καλή πηγή υπήρξε και ένα παράδειγμα από τον ιστότοπο jsfiddle.net³. Γενικότερα, όποτε οι ευρύτερες θεωρητικές μας γνώσεις πάνω στο αντικείμενο δεν επαρκούσαν, προστρέχαμε για βοήθεια στην επίσημη τεκμηρίωση του Web Audio⁴.

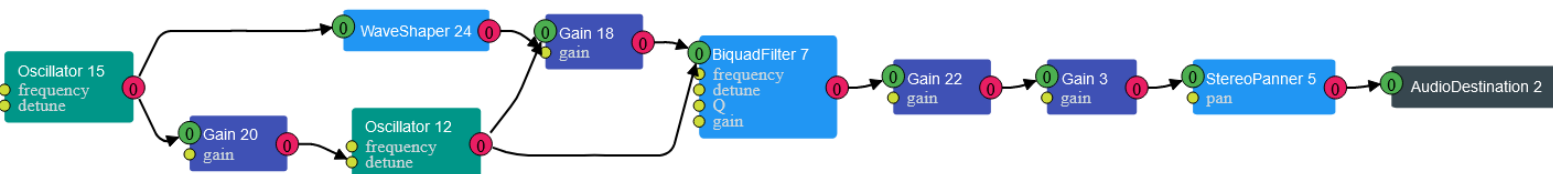
Επίσης, αναφορικά με τη μορφοποίηση της εφαρμογής (grid, radio buttons, κλπ), αναζητήθηκαν ορισμένες πληροφορίες σχετικά με τη CSS και αρκετές ζητούμενες λύσεις δόθηκαν μέσα από τον ιστότοπο w3schools.com.

Τέλος, για τις τιμές των συχνοτήτων των πλήκτρων του synthesizer, ως οδηγός χρησιμοποιήθηκε μία σελίδα της Βικιπαίδειας⁵.

Ζ. Προβλήματα που προέκυψαν:

Κατά την εκπόνηση της εργασίας, προέκυψαν διάφορα προβλήματα τα οποία εν τέλει, ευτυχώς για εμάς, επιλύθηκαν. Γενικότερο εμπόδιο στάθηκε η μη ενασχόλησή μας με το αντικείμενο της μουσικής, κάτι που είχε ως αποτέλεσμα να δυσκολευόμαστε να κατανοήσουμε βασικές μουσικές έννοιες.

Το κυριότερο πρόβλημα, ήταν η δυσλειτουργική σύνδεση των διάφορων components, η οποία δεν δημιουργούσε το αναμενόμενο τελικό ηχητικό αποτέλεσμα. Τελικά, μετά από πολλούς πειραματισμούς, καταλήξαμε στην παρακάτω συνδεσμολογία, η οποία επέφερε το βέλτιστο κατ' εμάς αποτέλεσμα. Το παρακάτω σχήμα δημιουργήθηκε με τη βοήθεια του web audio inspector, ένα σχετικό extension του Firefox.



¹ <https://www.davidmorrin.com/webaudiosynth/>

² <https://codepen.io/zastrow/pen/oDBki>

³ <http://jsfiddle.net/TurplePurtle/rsv1guvy/>

⁴ https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

⁵ https://en.wikipedia.org/wiki/Scientific_pitch_notation

Στο παραπάνω σχήμα, από αριστερά προς τα δεξιά, έχουμε τον low frequency oscillator (LFO – Oscillator 15), ο οποίος δίνει σήμα στο *GainNode* του vibrato (Gain 20), και στον *WaveShaper*, που με τη σειρά του δίνει σήμα στο *GainNode* του tremolo (Gain 18). Ο *GainNode* του vibrato, δίνει σήμα στο βασικό oscillator (Oscillator 12), ο οποίος μεταφέρει το σήμα του τόσο στο *GainNode* του tremolo (Gain 18), όσο και στον *BiquadFilter* του VCF, στον οποίο καταλήγει και το σήμα του tremolo. Στη συνέχεια, έχουμε δημιουργήσει μία σειριακή σύνδεση, η οποία από το VCF πηγαίνει στο *GainNode* του ADSR (Gain 22), και στη συνέχεια στο *GainNode* του master volume (Gain 3). Έπειτα, το τελικό ηχητικό αποτέλεσμα περνάει από τον *PannerNode*, προκειμένου να επιτύχουμε την επιθυμητή μεταφορά του ήχου σε δύο ηχεία ή ακουστικά, και τέλος διοχετεύεται στην έξοδο.