ΔΕΥΤΕΡΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΟΛΥΜΕΣΩΝ

Ιανουάριος 2021

Αναστάσιος Αρτέμιος Γιαννακούλιας 3150024
Ιάσων Χατζόπουλος 3150197

ΘΕΜΑ:

Θα πρέπει να υλοποιήσετε ένα πρόγραμμα μέσω του οποίου ο χρήστης θα μπορεί να δημιουργεί ήχους από ταλαντωτές και να εφαρμόζει φίλτρα σε αυτούς(αφαιρετική σύνθεση ήχου).

Η αλληλεπίδραση θα γίνεται μέσω διεπαφής που θα περιέχει τις κατάλληλες λειτουργίες ελέγχου των φίλτρων. Προαιρετικά, μπορείτε να φτιάξετε και διεπαφή για εκτέλεση μουσικής (piano keyboard) ή έλεγχο μέσω MIDI. Συγκεκριμένα, θα πρέπει να υλοποιηθούν:

•**Electronic Oscillator:** ταλαντωτής που παράγει συνθετικές κυματομορφές.

•**Voltage-controlled filter:** διαμορφώνει τον συνθετικό ήχο στον χώρο των συχνοτήτων.

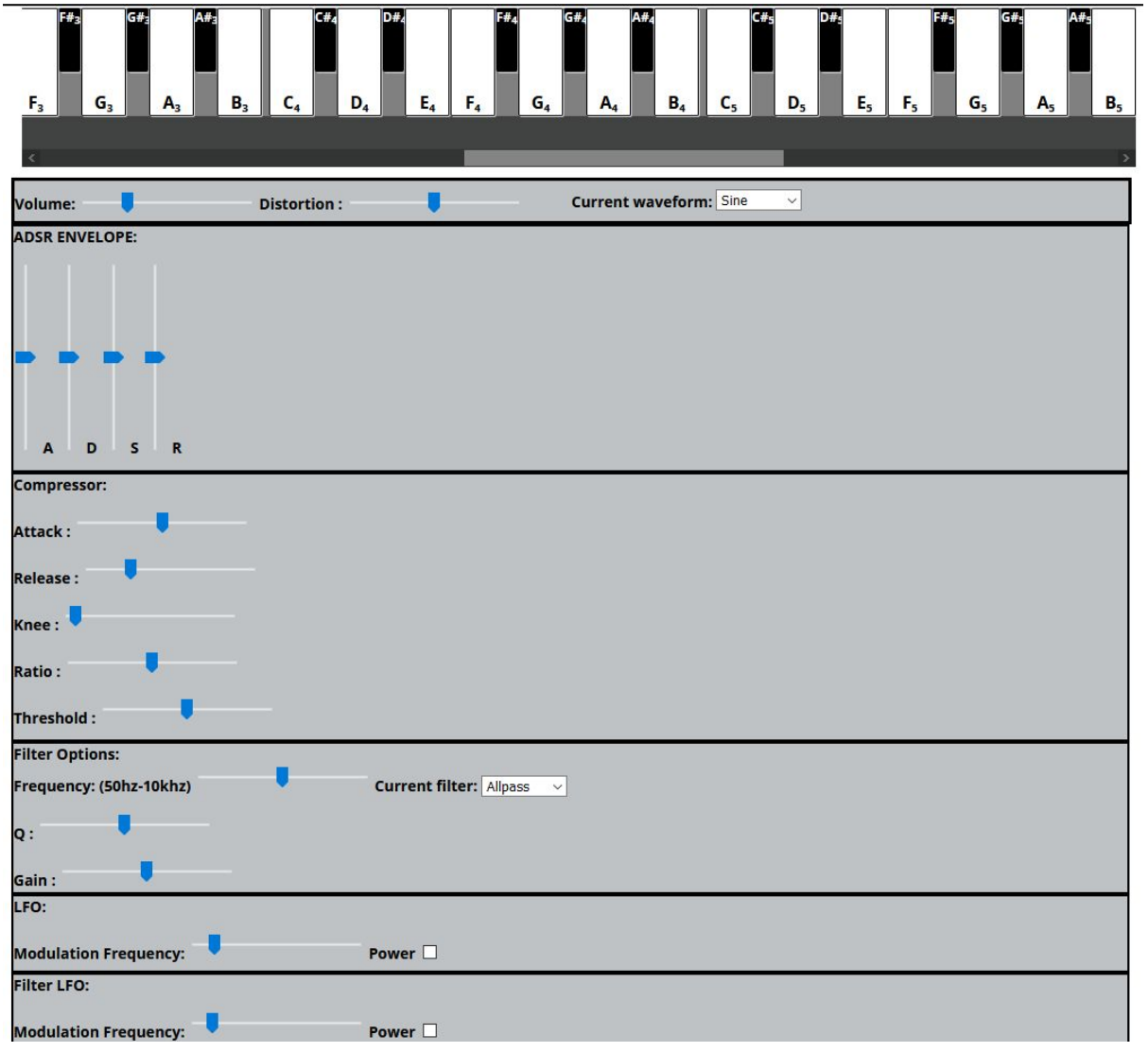
•**Voltage-controlled amplifier:** ενισχύει την ένταση (πλάτος κύματος) του ήχου που προκύπτει μετά την εφαρμογή των φίλτρων.

•**Attack Decay Sustain Release (ADSR) envelope:** διαμορφώνει χρονικά την ένταση του ενισχυμένου ήχου.

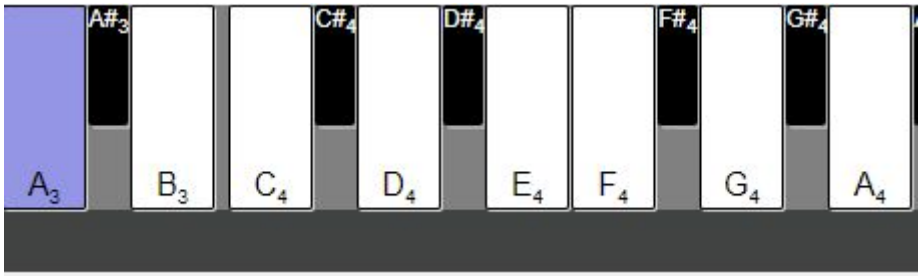
•**Low-Frequency Oscillator (LFO):** συνδυάζεται με τον κύριο ήχο για να δημιουργήσει εφέ όπως vibrato, tremolo και wah-wah.

• Προαιρετικά, μπορείτε να υλοποιήσετε και αλλαγή της διάταξης των στοιχείων (modules).

Για την εργασία επιλέξαμε να δημιουργήσουμε ένα Synthesizer χρησιμοποιώντας τους ταλαντωτές απο το Web Audio API μαζί όλα τα κλασσικά φίλτρα(all pass, highpass, lowpass, bandpass, low shelf, high shelf, peaking και notch filters) με τους κατάλληλους ελέγχους για τα χαρακτηριστικά τους (Q, Gain, Frequency), τον έλεγχο έντασης , ένα Dynamics Compressor και ένα φίλτρο LFO ακολουθώντας τεχνικές του design pattern model view controller(MVC) .



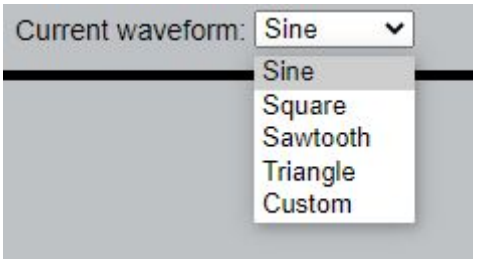
Οδηγίες χρήσης:



Πιέστε αριστερό κλικ πάνω σε κάποια νότα. Αν συρθεί το ποντίκι σε άλλη νότα ενώ είναι πατημένο το αριστερό κλικ θα αλλάξει και η νότα (Portamento).

Επιλογές:

- Volume: αλλάζει την τελική ένταση του οργάνου.
- Distortion: αλλαγή του gain του αρχικού σήματος.



- Current Waveform: αλλαγή κυματομορφής,
  - Sine: Ημιτονοειδής συνάρτηση κυματομορφής,
  - Square: Τετραγωνική κυματομορφή,
  - Sawtooth: Πριονωτή κυματομορφή,
  - Triangle: Τριγωνική κυματομορφή και
  - Custom: όπου αλλάζει μέσα από τον κώδικα του script (γρ. 213 script.js)

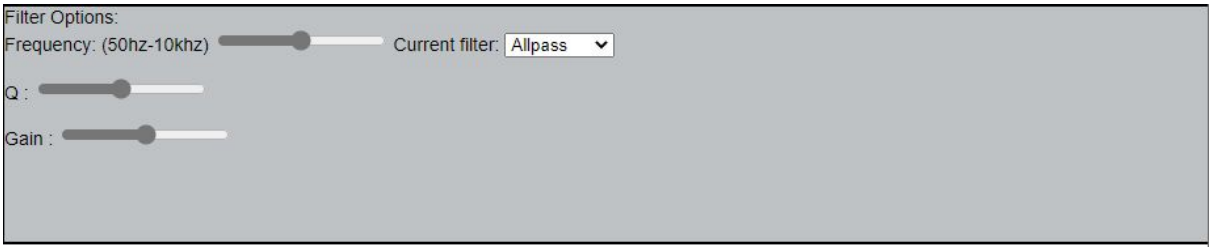
- ADSR Envelope:
  - Attack: Πόσο απότομα θα γίνει η αύξηση του σήματος του ταλαντωτή,
  - Decay: πόσο απότομα θα μειωθεί το peak της νότας,
  - Sustain: η ένταση της νότας μετά το decay και
  - release: πόσο απότομα θα φτάσει η ένταση στο 0 με το άφημα της νοτας.

- LFO amplitude modulation: περιέχει ένα slider με τη συχνότητα ταλάντωσης της έντασης του σήματος και ένα on/off μέσω του checkbox.

- LFO Filter Modulation: περιέχει ένα slider με τη συχνότητα ταλάντωσης της συχνότητας που περνάει από το φίλτρο και ένα on/off μέσω του checkbox.

Dynamics Compressor: αυξομειώνει το σήμα που δέχεται με βάση τα:

- Attack: πόσο σύντομα θα αρχίσει τη λειτουργία,
- Release: μετά από πόσο χρόνο θα σταματήσει η συμπίεση του σήματος,
- Knee: ρυθμίζει πόσο απότομη θα είναι η μετάβαση από το ασυμπίεστο σήμα στο συμπιεσμένο.
- Ratio: μείωση του σήματος για κάθε db που ξεπερνάει το threshold και
- Threshold: τα db όπου ξεκινάει ο κομπρεσορας.



- Filters: Επιλογή είδους φίλτρου με το Current Filter:
  - All-pass: επιτρέπει σε όλες τις συχνότητες να περάσουν, αλλά αλλάζει τη σχέση φάσης μεταξύ των συχνοτήτων.
  - Low-pass: επιτρέπει τη διέλευση συχνοτήτων κάτω από τη συχνότητα αποκοπής και εξασθενίζει τις συχνότητες πάνω από το όριο.
  - High-pass: Οι συχνότητες πάνω από τη συχνότητα αποκοπής περνούν, αλλά οι συχνότητες κάτω από το όριο εξασθενίζονται
  - Band-pass: επιτρέπει σε ένα εύρος συχνοτήτων να διέρχεται και να εξασθενεί τις συχνότητες κάτω και πάνω από αυτό το εύρος συχνοτήτων.
  - Low-shelf: περνάνε όλες οι συχνότητες, αλλά προσθέτει μια ώθηση (ή εξασθένηση) στις χαμηλότερες συχνότητες
  - High-Shelf: περνάνε όλες τις συχνότητες, αλλά προσθέτει μια ώθηση στις υψηλότερες συχνότητες.

- Peaking: περνάνε όλες οι συχνότητες, αλλά προσθέτει μια ώθηση (ή εξασθένηση) σε ένα εύρος συχνοτήτων.
- Notch: περνάνε όλες οι συχνότητες, εκτός από ένα σύνολο συχνοτήτων.

Υλοποίηση:

Το πρόγραμμα καλείτε αφού φορτωθεί το html αρχείο στο browser. Αρχικά θα κληθεί η μέθοδος setup(). Με το κάλεσμα της δημιουργείται ένα array που περιέχει τις συχνότητες για κάθε πλήκτρο του Synthesizer, το όνομά της νότας και τον αριθμό της οκτάβας της. Σε κάθε νότα προστίθενται ένα HTML element με event listener ώστε σε κάθε κλικ πάνω στις νότες να κληθεί η μέθοδος notePressed και noteReleased όταν αφήσουμε τη νοτα. Για τις νότες A, B, C, D, E , F, G δημιουργείται ένα css στοιχείο που έχει άσπρο χρώμα ενώ για διέσεις δημιουργείται ένα μαύρο.

Με το πάτημα οποιουδήποτε πλήκτρου θα κληθεί η μέθοδος playTone μέσω της notePressed, με όρισμα την μεταβλητή frequency, δηλαδή την συχνότητα που θέλουμε να παίξει ο ταλαντωτής.

Με την κλήση της playTone με όρισμα την παράμετρο frequency θα δημιουργηθούν οι κόμβοι επεξεργασίας του σήματος καθώς και η ψηφιακή συνδεσμολογία τους.

Πρώτα δημιουργείται ο dynamics compressor . Στην υλοποίηση αυτή, αλλάζουμε τις τιμές του attack, δηλαδή του πόσο σύντομα θα ξεκινήσει η συμπίεση της έντασης με ελάχιστη τιμή 0 και μέγιστη τιμή 1.Το knee, το οποίο ελέγχει τη μετάβαση από το ασυμπίεστο σήμα στο συμπιεσμένο, με ελάχιστη τιμή 0 και μέγιστη 40. Το ratio δηλώνει πόση μείωση θα λάβει το τελικό σήμα όποτε περνάει το threshold. Το threshold δηλώνει την ένταση στην οποία ξεκινάει να δουλεύει ο compressor.

Στην συνέχεια δημιουργείται ένα biquadFilter. Το είδος του φίλτρου ελέγχεται μέσα από την τάξη html και μέσα στο script χρησιμοποιούμε τη δομή switch case με τα σωστά φίλτρα για την αποφυγή σφαλμάτων. Τα φίλτρα που χρησιμοποιούμε είναι τα Highpass, lowpass, bandpass, all pass, notch που χρησιμοποιούν μόνο το Q και τις συχνότητες που πιάνουν, τα φίλτρα low shelf και high shelf δεν χρησιμοποιούν το Q αλλά gain για τις συχνότητες που πιάνει ενώ το peaking χρησιμοποιεί και το q και το gain.

Έπειτα θα δημιουργηθεί ένας δεύτερος ταλαντωτής για το LFO. Με το range απο το HTML αρχείο ελέγχουμε τη συχνότητα με την οποία θα ταλαντώνει ένα gain module το οποίο συνδέεται με το oscillator μας ώστε να έχουμε ταλάντωση στην ένταση με βάση τη συχνότητα και το σχήμα που θα διαλέξουμε στην τελική νότα.

Θα δημιουργηθεί επίσης ένας κόμβος για το distortion. Στο distortion πειράζουμε την καμπύλη του σήματος μετατρέποντας την τιμή του input σε ένα 32-bit float πίνακα. Κανονικοποιούμε την τιμή πριν την κλήση της μεθόδου makeDistortionCurve, διαιρώντας δια 1000 καθώς γινόταν clipping πολύ γρήγορα.

Με το άφημα του πλήκτρου, σταματάει και το oscillator τη λειτουργία του μέσω της μεθόδου noteReleased και του πίνακα oscList.

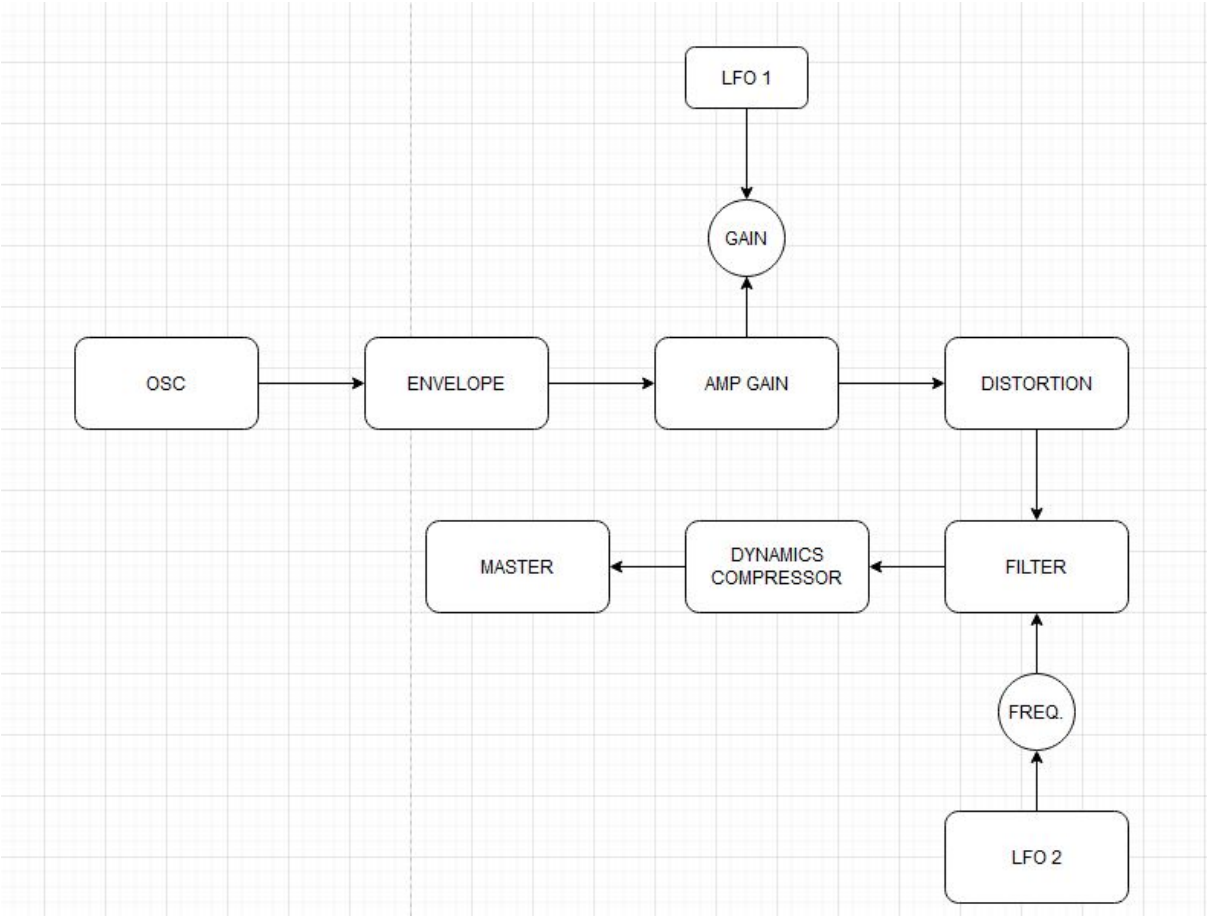
Για τα LFO χρησιμοποιήσαμε oscillators. Σαν συχνότητα παίρνουν την παράμετρο απο το UI και ανάλογα με τα state των checkbox δημιουργείται και η κατάλληλη συνδεσμολογία. Το Amplitude LFO συνδέεται απευθείας στο ταλαντωτή που παράγει το ηχητικό σήμα ενω ο αλλος συνδέεται στη συχνότητα του φίλτρου που λειτουργεί. Ανάλογα την είσοδο απο το html αρχείο αλλάζει η συνδεσμολογία με βάση ποιό LFO ενεργοποιείται ή απενεργοποιείται.

Η υλοποίηση του Envelope έγινε χρησιμοποιώντας τις μεθόδους από την βιβλιοθήκη και αυξάνουμε την ένταση του σήματος για ένα χρονικό διάστημα και μόλις αφήσει ο χρήστης το πλήκτρο η ένταση μειώνεται

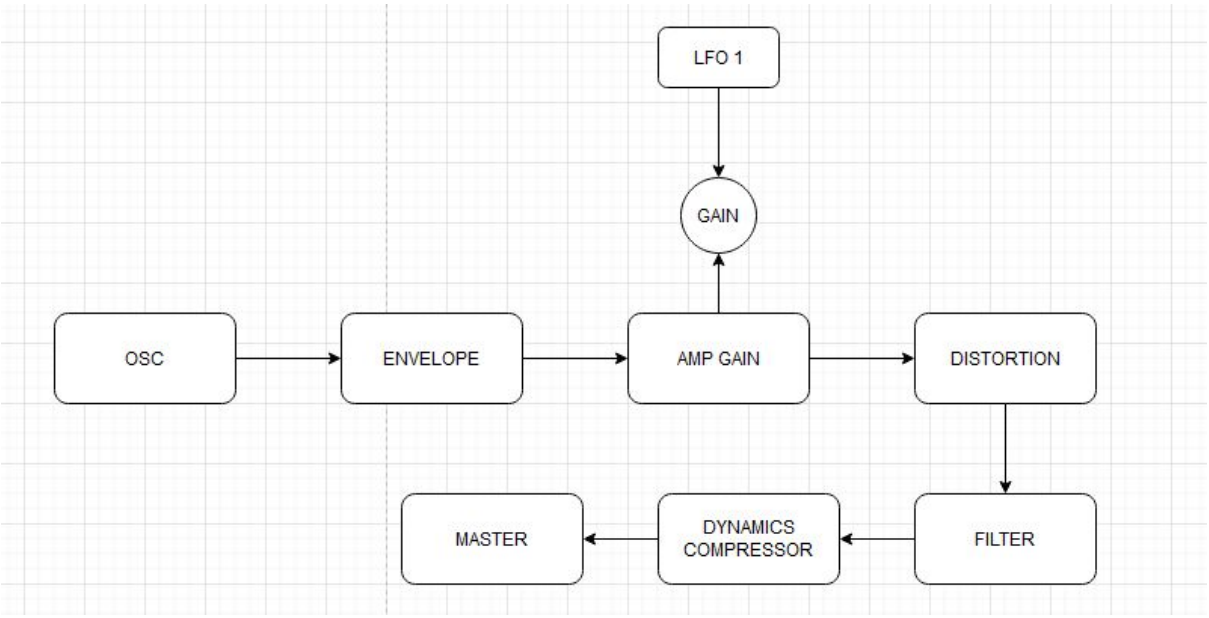
ΣΥΝΔΕΣΜΟΛΟΓΙΑ:

Έχουμε 4 περιπτώσεις συνδεσμολογίας.

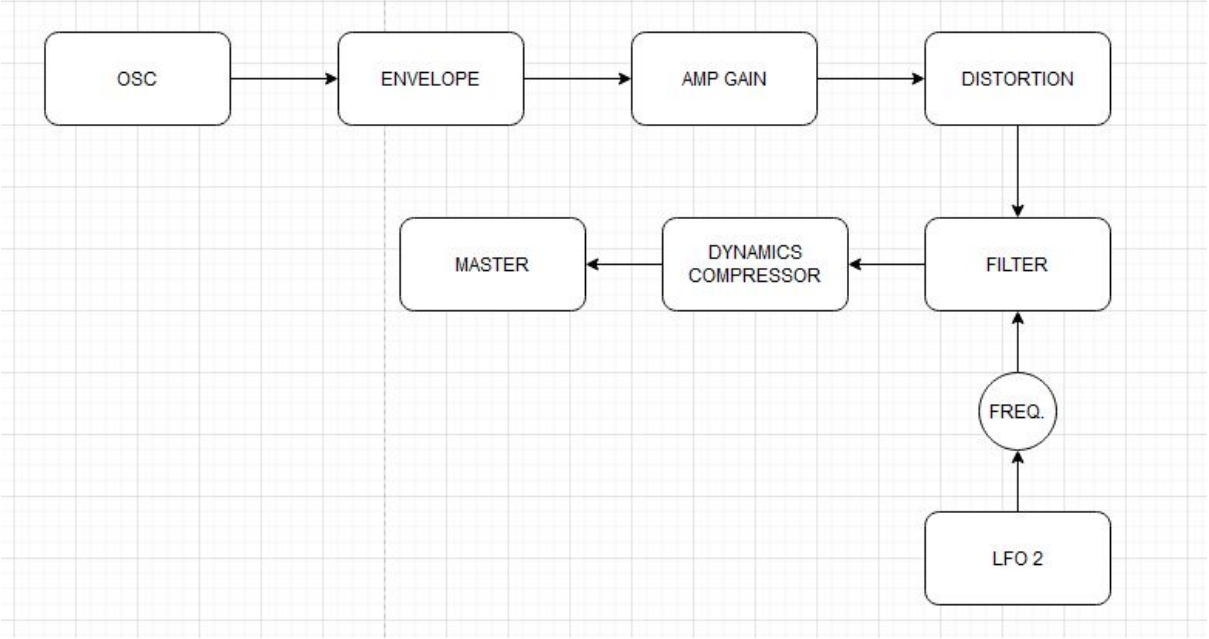
Amplitude LFO ON && Filter LFO ON:



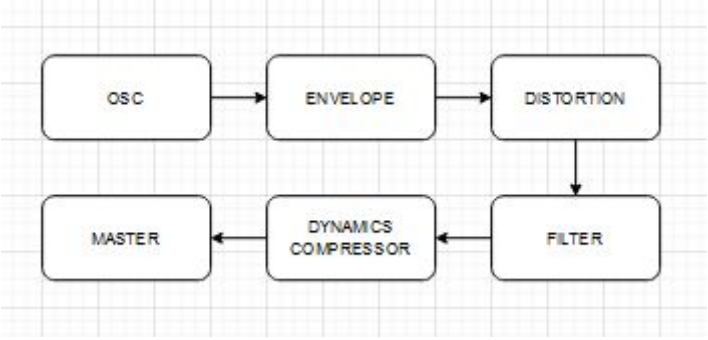
Amplitude LFO ON && Filter LFO OFF:



Amplitude LFO OFF && Filter LFO ON:

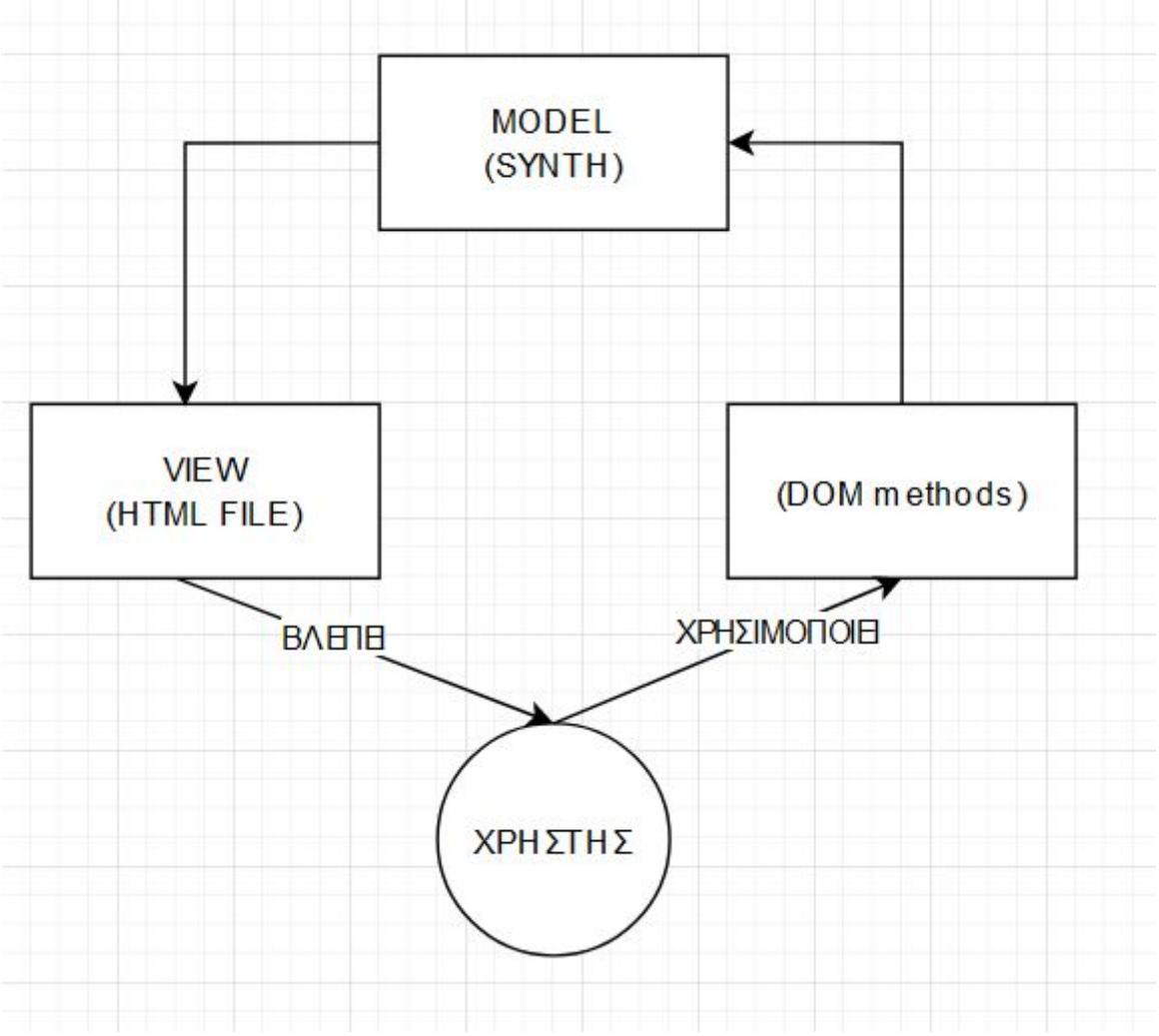


Amplitude LFO OFF && Filter LFO OFF:



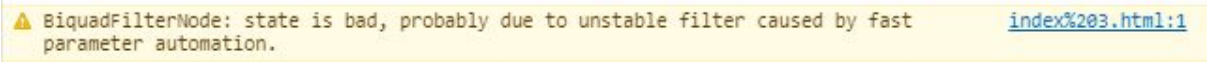
Λίγα λόγια για την υλοποίηση MVC:

Το προτυπο σχεδίασης MVC επιτρέπει στον χρήστη να αλληλεπιδρά εύκολα με την εφαρμογή μέσα από το script (Controller) το οποίο τροποποιεί και ελέγχει το μοντέλο, δηλαδή, το synthesizer και ενημερώνεται η σελίδα με την δημιουργια των πλήκτρων που βλέπει ο χρήστης. Αυτή η υλοποίηση λόγω της απλότητας της κάνει πολύ εύκολη την δημιουργία ενός τέτοιου project εφόσον έχουμε τις βιβλιοθήκες του DOM. Χρησιμοποιώντας τις μεθόδους addEventListener(συνθηκη, method_to_activate, true/false) , QuerySelector(name=...) και document.createElement().



Προβλήματα που συναντήσαμε κατά την υλοποίηση:

- Bugs με το dynamics compressor, έκανε τρομερή συμπίεση του σήματος επειδή δεν είχαμε βάλει event listeners στα στοιχεία ελέγχου του.
- Το Distortion έκανε πολύ έντονη παραμόρφωση στό σήμα με πολύ μικρές τιμές στο slider, λύθηκε κανονικοποιώντας την τιμή που περνούσε ως όρισμα.
- Το Peaking με τη default τιμή στο gain από το web audio api documentation είχε πολλά bugs και δεν σταμάταγε να παίζει η νότα αφού αφήναμε το αριστερό κλικ και έκανε πολύ δυνατό clipping το οποιο το λύσαμε με χαμηλότερη τιμή στο gain.
- Δυσκολία στη διασύνδεση του LFO καθώς δεν ημασταν σιγουροι σε ποιά audioparams έπρεπε να συνδέσουμε το LFO. Λύθηκε φτιάχνοντας ένα Gain Node (amp) και το συνδέσαμε στο βασικο Osc.
- Bug σε MS EDGE ενώ στο Mozilla δεν υπάρχει αντίστοιχο σφάλμα:



Να σχολιαστεί πως α) με refresh (f5) φτιάχνει αλλά ξαναεμφανίζεται.

Υποστήριξη BROWSER:

Δοκιμάστηκε σε Firefox 84.0.2 (64-bit), Windows 10, Microsoft Edge Version 87.0.664.75 (Official build) (64-bit)/ (Chromium based)

ΠΗΓΕΣ:

1. [setup] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Simple_synth
2. [distortion curve] https://stackoverflow.com/questions/22312841/waveshaper-node-in-webaudio-how-to-emulate-distortion
3. [wave shaping]
   i. https://developer.mozilla.org/en-US/docs/Web/API/WaveShaperNode
   ii. https://www.w3.org/TR/webaudio/#waveshapernode
4. [filters] https://www.w3.org/TR/webaudio/#biquadfilternode
5. [ADSR] https://www.redblobgames.com/x/1618-webaudio/ https://github.com/mohayonao/adsr-envelope
6. [filter modulation] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Advanced_techniques https://github.com/mdn/webaudio-examples/blob/master/step-sequencer/index.html