

Τμήμα Τεχνών Ήχου και Εικόνας
Ιόνιο Πανεπιστήμιο

Μάθημα: Ψηφιακή Επεξεργασία Ήχου

Εργαστηριακή Άσκηση 5
«Διαδραστικός έλεγχος στερεοφωνικής εικόνας ήχου»

Διδάσκων: Φλώρος Ανδρέας
Δρ. Ηλ/γος Μηχ/κός & Τεχνολογίας Υπολογιστών

Κέρκυρα, 2013

Πίνακας περιεχομένων

1. Εισαγωγή.....	3
2. Το λογισμικό Processing.....	3
2.1. Δομή εργασίας.....	3
2.2. Γλώσσα προγραμματισμού.....	4
2.3. Δημιουργία κώδικα.....	5
3. Αναπαραγωγή και επεξεργασία δεδομένων ήχου με το λογισμικό Processing	5
3.1. Αναπαραγωγή δεδομένων ήχου.....	5
3.1.1. Οι βασικές κλάσεις για την αναπαραγωγή ήχου.....	6
3.1.2. Οι βασικές μέθοδοι για την αναπαραγωγή ήχου.....	6
3.2. Επεξεργασία δεδομένων ήχου	7
3.2.1. Οι βασικές κλάσεις για την επεξεργασία της στερεοφωνικής εικόνας του ήχου	7
3.2.2. Οι βασικές μέθοδοι για την επεξεργασία της στερεοφωνικής εικόνας του ήχου	8
4. Διαδραστική χρήση του λογισμικού Processing	9
4.1. Λήψη δεδομένων από την θέση του δείκτη του ποντικιού.....	9
5. Υλοποίηση της άσκησης.....	9
5.1. Βήματα υλοποίησης της άσκησης	9
6. Παράδοση της εργασίας	10
7. Σημαντικές παρατηρήσεις.....	10

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5

«Διαδραστικός έλεγχος στερεοφωνικής εικόνας ήχου»

1. Εισαγωγή

Με την ολοένα αυξανόμενη διείσδυση των ηλεκτρικών/ηλεκτρονικών συσκευών, με δυνατότητες αναπαραγωγής ήχου, στην καθημερινή ζωή του ανθρώπου, αυξήθηκαν και οι απαιτήσεις για την ηχητική αναπαράσταση, των ηχητικών γεγονότων, από τις συσκευές αυτές. Έτσι, η απαίτηση για την απλή στερεοφωνική αναπαράσταση ενός ηχητικού γεγονότος, την θέση του δηλαδή στον πλάγιο άξονα του χώρου, εξελίχθηκε. Άμεση συνέπεια αυτής της εξέλιξης είναι η αναπαράσταση κινούμενων πηγών στο χώρο. Δηλαδή, ηχητικών σημάτων που δίνουν την αίσθηση, ή αναπαριστούν, ηχητικές πηγές οι οποίες μετακινούνται στον πλάγιο άξονα του χώρου.

Οι δυνατότητες αυτής της τεχνικής έδωσαν πνοή στην αναπαράσταση, π.χ., αμαξιών που περνάνε, ανθρώπων που κινούνται και άλλων πολλών, κινούμενων, ηχητικών πηγών. Παρ' όλο που η επεξεργασία του ηχητικού σήματος για την αναπαράσταση μίας κινούμενης ηχητικής πηγής απαιτεί δεξιότητα, ο διαδραστικός έλεγχος της θέσης της πηγής στον πλάγιο άξονα του χώρου απαιτεί περισσότερη.

Ένα αρκετά χρήσιμο εργαλείο – βοήθημα για τον διαδραστικό έλεγχο της θέσης της πηγής στον πλάγιο άξονα του χώρου, δηλαδή της στερεοφωνικής εικόνας, είναι και τα κατάλληλα λογισμικά των ηλεκτρονικών υπολογιστών. Τα λογισμικά αυτά οποία παρέχουν, σε πραγματικό χρόνο, την αλλαγή τιμών κάποιων μεταβλητών, ορισμένων από τον χρήστη, σε συνάρτηση με δεδομένα που δύναται να αλλαχθούν από την θέση του χρήστη, την συναισθηματική του κατάσταση, την προτίμησή του κ.α.

Σκοπός της παρούσας εργαστηριακής άσκησης είναι η ανάπτυξη απλού κώδικα διαδραστικού ελέγχου της στερεοφωνικής εικόνας ενός μονοφωνικού ήχου, με βάση την θέση του δείκτη του ποντικιού ενός ηλεκτρονικού υπολογιστή. Για τις ανάγκες της εργαστηριακής άσκησης θα χρησιμοποιηθούν οι ηλεκτρονικοί υπολογιστές του τμήματος «Τεχνών Ήχου & Εικόνας», καθώς και το, εγκατεστημένο σε αυτούς, λογισμικό – πλατφόρμα προγραμματισμού Processing.

2. Το λογισμικό Processing

Δεδομένου της χρήσης του λογισμικού Processing σε μαθήματα προηγούμενων εξαμήνων, σε αυτή την ενότητα θα παρουσιαστούν μερικά βασικά στοιχεία του λογισμικού. Έτσι, το λογισμικό Processing είναι ένα λογισμικό ανοιχτού κώδικα, αρχικά σχεδιασμένο για την επεξεργασία εικόνας. Αναπτύχθηκε και υλοποιήθηκε από τους Casey Reas και Ben Fry, του τεχνολογικού ινστιτούτου της Μασαχουσέτης (M.I.T., Massachusetts' Institute of Technology). Ως λογισμικό ανοιχτού κώδικα, υπάρχει η δυνατότητα για την δωρεάν προμήθεια του λογισμικού όπως επίσης και πρόσβαση του χρήστη στον κώδικα από βασικές βιβλιοθήκες του λογισμικού.

2.1. Δομή εργασίας

Το κύριο περιβάλλον του Processing αποτελείται από την μπάρα εργασίας, στο πάνω μέρος της οθόνης και με τις επιλογές «File», «Edit» κ.τ.λ., τις επιλογές του λογισμικού, όπως «Start», «Stop», που βρίσκονται κάτω από την μπάρα εργασίας και πάνω από το παράθυρο συγγραφείς κώδικα, το παράθυρο συγγραφής κώδικα, όπου καλύπτει το μεγαλύτερο μέρος της οθόνης του λογισμικού, και το παράθυρο εξόδου μηνυμάτων προς τον χρήστη. Εδώ θα πρέπει να σημειωθεί πως το Processing έχει και ένα ακόμη, «κρυφό», παράθυρο, όπου εμφανίζεται όταν «τρέξει» ο

κώδικας και χρησιμοποιείται για την δημιουργία εικόνων – γραφικών (όπως κινούμενα σχήματα, εικόνες, αναπαραγωγή βίντεο ή, ακόμη, και για την εμφάνιση κινούμενων, ή μη, κυματομορφών. Η οθόνη του λογισμικού φαίνεται στην Εικόνα 1.



Εικόνα 1. Η επιφάνεια εργασίας του λογισμικού Processing

Ακόμα, κάθε, νέος ή μη, κώδικας που γράφετε με το Processing αποθηκεύεται ως «Sketch», π.χ. σε αντίθεση με τα αρχεία source, .cpp, της γλώσσας C++ ή τα αρχεία .java της γλώσσας Java. Τα αρχεία αποθηκεύονται ως αρχεία τύπου .pde, όπου σημαίνει Processing Developing Environment.

Κάθε «sketch» του Processing έχει δύο «σκέλη». Το πρώτο είναι η συνάρτηση `setup()`, και το δεύτερο η συνάρτηση `draw()`. Αμφότερες έχουν τύπο επιστροφής `void`. Η πρώτη χρησιμοποιείται για την δήλωση μεταβλητών ή/και αντικειμένων που θα χρησιμοποιηθούν από τον κώδικα, και η δεύτερη για την δημιουργία κώδικα που θα εκτελείται σε επανάληψη μέχρι τον τερματισμό της επανάληψης μέσω του κώδικα ή της επιλογής «Stop» από τις επιλογές του λογισμικού.

2.2. Γλώσσα προγραμματισμού

Η γλώσσα προγραμματισμού του Processing βασίζεται στην γλώσσα προγραμματισμού Java, και είναι αντικειμενοστραφής. Χρησιμοποιεί όλες τις γνωστές μεθόδους/δομές αποφάσεων και επανάληψης όπως τους βρόγχους `for` και `while` και την δομή απόφασης `if`.

Χρησιμοποιεί βιβλιοθήκες κώδικα για την χρήση συναρτήσεων, όπως πολλές άλλες γλώσσες προγραμματισμού, και όλους τους γνωστούς τύπους μεταβλητών (`integer`, `character`, `double`, `float`, `Boolean`, `char`, `string` και `void`).

Επίσης, λόγω του αντικειμενοστραφούς χαρακτήρα της γλώσσας προγραμματισμού, χρησιμοποιούνται οι αρχές του αντικειμενοστραφούς προγραμματισμού, όπως κλάσεις και αντικείμενα, μέθοδοι, ιδιωτικότητα και κληρονομικότητα.

2.3. Δημιουργία κώδικα

Ο κώδικας γράφεται στην περιοχή συγγραφής του κώδικα, η οποία καλύπτει και το μεγαλύτερο μέρος της οθόνης του λογισμικού και μπορεί να φανεί στην Εικόνα 1. Οι κανόνες για τον ορισμό των μεταβλητών που ισχύουν στις περισσότερες γλώσσες ισχύουν και στο Processing, δηλαδή μία μεταβλητή υπάρχει μέσα στο block όπου ορίστηκε εκτός και εάν δηλωθεί διαφορετικά. Αυτό σημαίνει ότι οι μεταβλητές που δηλώνονται έξω από οποιοδήποτε block, π.χ. στην αρχή του κώδικα και πριν την συνάρτηση `setup()` έχουν καθολική διάρκεια, `global` μεταβλητές. σ

3. Αναπαραγωγή και επεξεργασία δεδομένων ήχου με το λογισμικό Processing

Παρ' όλο που το Processing σχεδιάστηκε αρχικά για την δημιουργία, επεξεργασία και αναπαραγωγή στατικής/κινούμενης εικόνας, προσφέρεται και ως εργαλείο δημιουργίας, επεξεργασίας και αναπαραγωγής ήχου. Η συνηθέστερη βιβλιοθήκη που προσφέρει συναρτήσεις σχετικά με την επεξεργασία και αναπαραγωγή του ήχου είναι η βιβλιοθήκη `minim`.

3.1. Αναπαραγωγή δεδομένων ήχου

Για την αναπαραγωγή των δεδομένων ήχου από το Processing θα πρέπει να χρησιμοποιηθεί η βιβλιοθήκη `Minim` και οι κατάλληλες μέθοδοι της. Για την λειτουργία της βιβλιοθήκης αυτής, θα αναλυθεί ένα παράδειγμα που παρέχεται από την ίδια την βιβλιοθήκη. Έτσι, στην επιλογή του καταλόγου «File» «Examples», υπάρχει η επιλογή «Libraries» και μετά από αυτή, η επιλογή «Minim (Sound)». Επιλέγοντας το «GetLineOut», εμφανίζεται ο Κώδικας 1.

```
import ddf.minim.*;
import ddf.minim.signals.*;

Minim minim;
AudioOutput out;
SineWave sine;

void setup()
{
  size(512, 200, P2D);

  minim = new Minim(this);

  // get a line out from Minim, default sample rate is 44100, default bit depth is 16
  out = minim.getLineOut(Minim.STEREO, 2048);

  // create a sine wave Oscillator, set to 440 Hz, at 0.5 amplitude, sample rate 44100 to match the line out
  sine = new SineWave(440, 0.5, out.sampleRate());
  // add the oscillator to the line out
  out.addSignal(sine);
}

void draw()
{
  background(0);
  stroke(255);
  // draw the waveforms
  for(int i = 0; i < out.bufferSize() - 1; i++)
  {
    line(i, 50 + out.left.get(i)*50, i+1, 50 + out.left.get(i+1)*50);
    line(i, 150 + out.right.get(i)*50, i+1, 150 + out.right.get(i+1)*50);
  }
}
```

```

void stop()
{
    // always close Minim audio classes when you are done with them
    out.close();
    minim.stop();

    super.stop();
}

```

Κώδικας 1. Κώδικας παραδείγματος χρήσης των κλάσεων και των μεθόδων αναπαραγωγής ήχου

Εκτός από την, γνωστή, μορφή εισαγωγής σχολίων, με τους τελεστές «//», για σχόλια μίας γραμμής, ή «/* */», για σχόλια πολλών γραμμών, μπορούν να φανούν τα κύρια αντικείμενα καθώς και οι κλάσεις τους. Έτσι, υπάρχουν τα αντικείμενα:

- «minim», της κλάσης «Minim»,
- «out», της κλάσης «AudioOutput», και
- «sine», της κλάσης «SineWave»

Επίσης, μπορούν να φανούν οι κύριες μέθοδοι για την αναπαραγωγή ήχου, που είναι:

- «getLineOut», του αντικειμένου «minim», και
- «addSignal», του αντικειμένου «out».

3.1.1. Οι βασικές κλάσεις για την αναπαραγωγή ήχου

Για να καταστεί δυνατή η χρήση αρχείων ήχου, προσπέλαση και αναπαραγωγή, θα πρέπει να οριστεί ένα αντικείμενο της κλάσης Minim. Μέσω αυτού, μπορούν να χρησιμοποιηθούν οι μέθοδοι getLineIn() και getLineOut(), όπου προσφέρουν την δυνατότητα για εισαγωγή/εγγραφή δεδομένων ήχου και την δυνατότητα για εξαγωγή/αναπαραγωγή δεδομένων ήχου, αντιστοίχως. Ο ορισμός του αντικειμένου κλάσης Minim, γίνεται μέσω του τελεστή new.

Επιπλέον, μέσω της βιβλιοθήκης Minim, μπορούν να οριστούν, ενδεικτικά, αντικείμενα των κλάσεων:

- AudioInput
- AudioOutput

και άλλα. Τα αντικείμενα AudioInput και AudioOutput χρησιμοποιούν την κάρτα ήχου του υπολογιστή για την εισαγωγή/εγγραφή και έξοδο/αναπαραγωγή σημάτων ήχου, αντίστοιχα. Χρησιμοποιούν τις παραμέτρους για την έξοδο ή την είσοδο που ορίζονται από τις μεθόδους, του αντικειμένου κλάσης Minim, getLineIn() και getLineOut().

Επίσης, για την δημιουργία σημάτων ήχου μπορούν να χρησιμοποιηθούν υλοποιήσεις της αφηρημένης (abstract) κλάσης Oscillator, που είναι υλοποίηση της αφηρημένης (abstract) κλάσης/διεπαφής (interface) AudioSignal, SineWave και SawWave. Εδώ αξίζει να σημειωθεί πως οι ήχοι που δημιουργούνται με αντικείμενα κλάσης SineWave ή/και SawWave είναι μονοφωνικά.

3.1.2. Οι βασικές μέθοδοι για την αναπαραγωγή ήχου

Για την αναπαραγωγή ήχου θα πρέπει, εν αρχή, να δηλωθεί εάν θα είναι στερεοφωνική ή μονοφωνική αναπαραγωγή και έπειτα να ορισθεί το μέγεθος του buffer, σε δείγματα, ο οποίος θα περιέχει τα δείγματα του ήχου, η συχνότητα της δειγματοληψίας του ήχου και το μέγεθος της

λέξης/δείγματος, σε bits, του ήχου. Αυτές οι παράμετροι, για την αναπαραγωγή του ήχου, ορίζονται από την μέθοδο `getLineOut()` της κλάσης `Minim`, ή οποία μπορεί να χρησιμοποιηθεί από το/τα αντικείμενο/αντικείμενα της κλάσης `AudioOutput`. Εδώ θα πρέπει να σημειωθεί πως τα ορίσματα της μεθόδου `getLineOut()` έχουν αρχικές τιμές, και η σειρά δήλωσής τους είναι α) Τύπος(μονοφωνική ή στερεοφωνική έξοδος), β) μέγεθος `buffer`, γ) ρυθμός δειγματοληψίας και δ) μέγεθος λέξης/δείγματος, και έτσι δεν απαιτείται η απόδοση τιμών σε όλα, απαιτείται όμως η απόδοση τιμών με την σειρά ύπαρξης, δηλαδή δεν μπορεί να μην δηλωθεί ο τύπος της εξόδου εάν δηλωθεί το μέγεθος του `buffer` ενώ μπορεί να μην δηλωθεί το μέγεθος του `buffer` εάν δηλωθεί ο τύπος της εξόδου, που είναι «πιο πριν» στην σειρά των ορισμάτων από το όρισμα του μεγέθους του `buffer`. Οι τιμές μπορούν να χρησιμοποιηθούν για:

- τον τύπο της εξόδου είναι α) MONO ή β) STEREO (τύπου `int`, όπου MONO και STEREO είναι προσδιορισμοί για τις τιμές του ορίσματος, όπως γίνεται με τον τελεστή `#define` στην C),
- για το μέγεθος του `buffer` είναι δυνάμεις του 2 (τύπου `int`, μέγεθος μέτρησης `samples`),
- για τον ρυθμό δειγματοληψίας, όλοι ευρέως χρησιμοποιούμενοι ρυθμοί (τύπου `float`, μέγεθος μέτρησης `Hz`), και
- για το μήκος της λέξης όλα τα ευρέως χρησιμοποιούμενα μήκη (τύπου `int`, μέγεθος μέτρησης `bits`)

Οι αρχικές τιμές των παραπάνω ορισμάτων είναι STEREO, 1024, 44100 και 16, αντίστοιχα.

Με την δήλωση των παραπάνω παραμέτρων, και την χρήση της μεθόδου `getLineOut()`, δηλώνεται, επίσης, και μία έξοδος ήχου η οποία μπορεί να αντιστοιχιστεί στην έξοδο ήχου του υπολογιστή μέσω της κλάσης `AudioOutput`.

Μέσω της κλάσης `AudioOutput`, μπορούν να χρησιμοποιηθούν μέθοδοι για την προσθήκη σημάτων ήχου στην έξοδο ήχου, να «ζητηθεί» ο ρυθμός δειγματοληψίας που έχει προσδιοριστεί, να ζητηθεί το μέγεθος του `buffer` που έχει προσδιοριστεί καθώς και να ζητηθούν οι τιμές κάθε δείγματος που είναι στον `buffer`. Οι μέθοδοι για την υλοποίηση των προαναφερθέντων διεργασιών είναι:

- `addSignal(Oscillator)`,
- `sampleRate(void)`,
- `bufferSize(void)`, και
- `left.get(int)` και `right.get(int)`

αντιστοίχως. Η χρήση των παραπάνω μεθόδων μπορεί να φανεί στον Κώδικα 1.

3.2. Επεξεργασία δεδομένων ήχου

Η βιβλιοθήκη `Minim` παρέχει κλάσεις και μεθόδους για την επεξεργασία των δεδομένων ήχου, όπως, ενδεικτικά, ανάλυση `Fourier`, μεταβολή του κέρδους κ.α. Για τις ανάγκες της συγκεκριμένης εργαστηριακής άσκησης θα αναλυθούν οι συσχετιζόμενες κλάσεις και μέθοδοι με την αλλαγή της στερεοφωνικής εικόνας του ήχου.

3.2.1. Οι βασικές κλάσεις για την επεξεργασία της στερεοφωνικής εικόνας του ήχου

Η βασική κλάση που σχετίζεται με τον έλεγχο παραμέτρων, όπως στερεοφωνική εικόνα, κέρδος και στάθμη ήχου, είναι η κλάση `Controller`. Δεν υλοποιείται από τον χρήστη, παρ' όλο που δεν είναι αφηρημένη. Παρέχει όμως μία συνάρτηση/μέθοδο, την `printControls()`, μέσω της οποίας

εμφανίζονται οι διαθέσιμες παράμετροι που μπορούν να μεταβληθούν από τον χρήστη στο συγκεκριμένο/εκάστοτε σύστημα.

Επίσης, οι μέθοδοί της μπορούν να χρησιμοποιηθούν από κλάσεις τύπου `AudioInput` και `AudioOutput`.

3.2.2. Οι βασικές μέθοδοι για την επεξεργασία της στερεοφωνικής εικόνας του ήχου

Οι μέθοδοι που παρέχονται από την κλάση `Controller`, μέσω των οποίων μπορούν να μεταβληθούν οι προαναφερθείσες παράμετροι είναι:

- `setBalance(float value)`, όπου για στερεοφωνικά αρχεία ήχου ρυθμίζεται το κέρδος αναπαραγωγής ανά κανάλι με εύρος τιμών -1 έως 1,
- `setGain(float value)`, όπου ρυθμίζεται το κέρδος με εύρος τιμών -80 έως 6,
- `setPan(float value)`, όπου ρυθμίζεται η στερεοφωνική εικόνα για μονοφωνικά αρχεία με εύρος τιμών -1 έως 1, και
- `setVolume(float value)`, όπου ρυθμίζεται η στάθμη αναπαραγωγής με απροσδιόριστο εύρος τιμών.

Υπάρχουν και οι ανάλογες «get» μέθοδοι. Εδώ θα πρέπει να αναφερθεί πως υπάρχει η μέθοδος `hasControl()`, όπου παίρνει ως όρισμα ένα από τα:

- `Controller.PAN`
- `Controller.VOLUME`
- `Controller.SAMPLE_RATE`
- `Controller.BALANCE`
- `Controller.MUTE`
- `Controller.GAIN`

και επιστρέφει `true` εάν το εκάστοτε σύστημα/υπολογιστής επιτρέπει τον έλεγχο μίας εκ των παραμέτρων (`pan`, `volume`, `sample rate`, `gain` κ.τ.λ.) ή `false` εάν το σύστημα δεν το επιτρέπει.

Όπως αναφέρθηκε, οι παραπάνω μέθοδοι μπορούν να χρησιμοποιηθούν με αντικείμενα της κλάσης `AudioOutput` και ένα παράδειγμα φαίνεται στον Κώδικα 2.

```
if ( out.hasControl(Controller.PAN) )
{
    // map the mouse position to the range of the pan
    float val = map(mouseX, 0, width, -1, 1);
    // if a pan control is not available, this will do nothing
    out.setPan(val);
    // if a pan control is not available this will report zero
    text("The current pan is " + out.getPan() + ".", 5, 20);
}

else
{
    text("The output doesn't have a pan control.", 5, 20);
}
```



```
}  
}
```

Κώδικας 2. Κώδικας παραδείγματος χρήσης των μεθόδων για την μεταβολή της στερεοφωνικής εικόνας

Όπου η μεταβλητή `out` είναι αντικείμενο της κλάσης `AudioOutput`. Η μέθοδος `map`, σταθμίζει το εύρος των τιμών, 0 έως `width`, μίας μεταβλητής, `mouseX`, σε ένα άλλο εύρος τιμών, -1 έως 1.

4. Διαδραστική χρήση του λογισμικού Processing

Ένα από τα εξαιρετικά θετικά στοιχεία του λογισμικού Processing, είναι η απλή υλοποίηση διαδραστικού κώδικα. Μία από τις ενσωματωμένες λειτουργίες του λογισμικού αυτού που μπορούν να προσφέρουν διάδραση, είναι η αναγνώριση της θέσης του δείκτη του ποντικιού.

4.1. Λήψη δεδομένων από την θέση του δείκτη του ποντικιού

Η λήψη δεδομένων από την θέση του δείκτη του ποντικιού πραγματοποιείται με την χρήση δύο ενσωματωμένων μεταβλητών του λογισμικού τις `mouseX` και `mouseY`. Αυτές οι δύο μεταβλητές περιέχουν την θέση του δείκτη, στον άξονα `x'` και `y'` αντιστοίχως, και μπορούν να χρησιμοποιηθούν ως όρισμα σε οποιαδήποτε μέθοδο οποιουδήποτε αντικειμένου.

Ένα παράδειγμα της χρήσης των παραπάνω μεταβλητών, σε συνάρτηση με την μεταβολή της στερεοφωνικής εικόνας του ήχου, μπορεί να φανεί στον Κώδικα 2.

5. Υλοποίηση της άσκησης

Για την υλοποίηση της άσκησης θα πρέπει να υλοποιηθεί κώδικας ο οποίος να δημιουργεί ημιτονικό σήμα, συχνότητας 1 kHz, και να μεταβάλει την στερεοφωνική του εικόνα σε σχέση με την θέση του ποντικού στον άξονα `x'`. Επίσης, θα πρέπει να μεταβάλλεται το κέρδος του σήματος σε σχέση με την θέση του δείκτη του ποντικιού στον άξονα `y'`. Το μέγεθος του παραθύρου στο οποίο θα ανιχνεύεται η θέση του δείκτη του ποντικιού (και θα εμφανίσει το Processing) θα πρέπει να είναι 800 x 600 pixels.

5.1. Βήματα υλοποίησης της άσκησης

Τα βήματα για την υλοποίηση της άσκησης είναι τα εξής:

1. Εισαγωγή της βιβλιοθήκης `Minim` και της βιβλιοθήκης `Minim.signals`, με βάση των Κώδικα 1.
2. Δήλωση των κατάλληλων αντικειμένων για την υλοποίηση των στόχων της άσκησης, ήτοι α) αντικείμενο της κλάσης `Minim`, αντικείμενο της κλάσης `AudioOutput`, αντικείμενο της κλάσης `Oscillator`.
3. Δήλωση της εξόδου ήχου, μέσω της μεθόδου `getLineOut()`.
4. Δημιουργία του ηχητικού σήματος, μέσω της αρχικοποίησης του αντικειμένου της κλάσης `SineWave`.
5. Προσθήκη στην έξοδο του δημιουργηθέντος σήματος, μέσω της μεθόδου `addSignal()`.
6. Έλεγχος εάν το σύστημα επιτρέπει τον έλεγχο της μεταβολής της στερεοφωνικής εικόνας ενός σήματος, μέσω της μεθόδου `hasControll` και με όρισμα `Controller.PAN`.
7. Έλεγχος εάν το σύστημα επιτρέπει τον έλεγχο της μεταβολής της στερεοφωνικής εικόνας ενός σήματος, μέσω της μεθόδου `hasControll` και με όρισμα `Controller.GAIN`.
8. Εάν οι έλεγχοι των βημάτων 6 και 7 είναι αληθείς, στάθμιση στο κατάλληλο εύρος των τιμών της θέσης του δείκτη του ποντικιού ανάλογα με την παράμετρο που θα ελέγχει κάθε τιμή
9. Λήξη των μεθόδων της βιβλιοθήκης `Minim`, με την χρήση της συνάρτησης `stop()`, όπως ακριβώς παρουσιάζεται στον Κώδικα 1.

6. Παράδοση της εργασίας

Παραδοτέο της συγκεκριμένης άσκησης αποτελεί ο πηγαίος κώδικας, πλήρως σχολιασμένος, καθώς και αναφορά σε ηλεκτρονική μορφή (τύπου .doc ή .pdf), στην οποία απαραίτητως θα περιλαμβάνονται οι παρακάτω ενότητες:

1. Εξώφυλλο με τα στοιχεία της εργασίας, τα προσωπικά σας στοιχεία (ονοματεπώνυμο, έτος σπουδών και αριθμός μητρώου) και η ομάδα / ημερομηνία εκτέλεσης της άσκησης.
2. Ενότητα «Εισαγωγή» στην οποία θα εξηγήσετε επιγραμματικά τους στόχους της άσκησης.
3. Ενότητα «Θεωρία» στην οποία θα αναπτύσσετε το θεωρητικό υπόβαθρο των αποτελεσμάτων της επεξεργασίας και της ανάλυσής σας.
4. Ενότητα «Υλοποίηση της άσκησης», στην οποία θα περιγράψετε τη διαδικασία εκτέλεσης της άσκησης που πραγματοποιήσατε.
5. Ενότητα «Αποτελέσματα» στην οποία θα δίνετε τα ζητούμενα από την άσκηση αποτελέσματα υπό μορφή πινάκων ή / και διαγραμμάτων συχνότητας ή / και διαγραμμάτων χρόνου ή / και πηγαίου κώδικα.
6. Ενότητα «Συμπεράσματα» στην οποία θα συνοψίζετε σε μία-δύο παραγράφους τις εργασίες που εκτελέσατε και θα ερμηνεύετε ποιοτικά και ποσοτικά τα αποτελέσματα που λάβατε.

Η κατάθεση της ηλεκτρονικής αναφοράς θα γίνει ηλεκτρονικά μέσω του e-class. Η προθεσμία υποβολής είναι αυστηρή και ορίζεται ως μία εβδομάδα μετά την εκτέλεση της άσκησης στο εργαστήριο. Υπενθυμίζεται ότι η εργασία είναι ατομική.

7. Σημαντικές παρατηρήσεις

1. Δεδομένου ότι τα βήματα που πρέπει να ακολουθηθούν για την εκτέλεση της εργαστηριακής άσκησης είναι καθορισμένα, παρακαλούνται οι συμμετέχοντες να προσέλθουν χωρίς καθυστέρηση την προβλεπόμενη ώρα, καθώς θα είναι αδύνατη η προς τα πίσω εξήγηση, η ερμηνεία επιμέρους λεπτομερειών και η παροχή πρόσθετων διευκρινίσεων.
2. Για την εκτέλεση της άσκησης στις συνθήκες του εργαστηρίου, απαραίτητη κρίνεται η χρήση ακουστικών. Για το λόγο αυτό παρακαλούνται όλοι οι φοιτητές να φέρουν μαζί τους το προσωπικό τους ζευγάρι ακουστικών.