



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

**Ανάπτυξη ρομποτικού εκπαιδευτικού περιβάλλοντος για
παιδιά**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΑΡΑΜΠΑΣΙΑΔΗ Ν. ΗΛΙΑΝΑ

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2012



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

Ανάπτυξη ρομποτικού εκπαιδευτικού περιβάλλοντος για παιδιά

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΑΡΑΜΠΑΣΙΑΔΗ ΗΛΙΑΝΑ

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29^η Μαρτίου 2012.

.....
Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

.....
Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Κωνσταντίνα Νικήτα
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2012

(Υπογραφή)

.....

ΚΑΡΑΜΠΑΣΙΑΔΗ ΗΛΙΑΝΑ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ηλιάνα Ν. Καραμπασιάδη, 2012

Με επιφύλαξη παντός δικαιώματος. All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η ραγδαία πρόοδος και εξέλιξη σε όλους τους τομείς της τεχνολογίας στις μέρες μας, κάνει πιο προσιτή την επαφή των ανθρώπων με τους ηλεκτρονικούς υπολογιστές σε όλο και νεαρότερες ηλικίες. Παράλληλα, η πρόοδος της ρομποτικής και γενικότερα των ρομποτικών συστημάτων διευκολύνει πολλές πτυχές της καθημερινότητάς μας και βελτιώνει τον τρόπο και την ποιότητα της ζωής μας. Τα ρομπότ, και δη τα ανθρωπόμορφα, έχουν αποδειχθεί ιδιαίτερα ευεργετικά για παιδιά με αυτισμό ή μαθησιακές δυσκολίες. Η παρούσα διπλωματική εργασία είναι η υλοποίηση ενός σεναρίου για την εκμάθηση απλών αριθμητικών πράξεων σε παιδιά προσχολικής και πρώτης σχολικής ηλικίας, με τη βοήθεια του Nao robot της Aldebaran, με τον πιο ευχάριστο και δημιουργικό τρόπο. Το ρομπότ αυτό είναι κατάλληλο για τη συγκεκριμένη εφαρμογή, διότι είναι ανθρωπόμορφο και εύκολο να αλληλεπιδράσει με παιδιά μικρής ηλικίας. Το παιδί λοιπόν, θα πρέπει να φέρει σε πέρας ένα σύνολο δοκιμασιών με τη βοήθεια του ρομπότ και κάποιων επιπλέον εξαρτημάτων που θα χρησιμοποιήσουμε, κερδίζοντας κάθε φορά ένα «έπαθλο» μέχρι να ολοκληρώσει την αποστολή. Έτσι, αλληλεπιδρώντας το παιδί με το ρομπότ, εξοικειώνεται με τη διαδικασία επίλυσης προβλημάτων ενώ παράλληλα γνωρίζει τα βασικά στοιχεία ενός ρομποτικού συστήματος.

Λέξεις Κλειδιά: Ρομπότ, Εκπαιδευτική ρομποτική, Nao robot , Αλληλεπίδραση ανθρώπου-μηχανής, Μαθησιακές δυσκολίες, Arduino, Εκμάθηση αριθμητικής σε παιδιά

Abstract

Nowadays, the rapid progress and development in all areas of technology, makes more accessible to people the contact with computers at ever younger ages. Furthermore, the progress of robotics and robotic systems in general facilitates many aspects of your daily life and improves the way and quality of life. Also, anthropomorphic robots have proved to be highly beneficial for children with autism or learning difficulties. This project accomplishes a scenario for learning simple arithmetic in preschool and primary school children, with the help of the Nao robot of Aldebaran, in the most pleasant and creative way. This robot is suitable for this application because it is anthropomorphic and easy to interact with young children. Therefore, the child should carry out some tests with the help of robot and some extra accessories we use, winning for every single test a “prize” to complete the mission. So, the child both familiarizes with the procedure of solving problems and at the same time knows about the basic elements of a robotic system.

Keywords: Robot, Educational robots, Nao robot, Human-machine interaction, Learning difficulties, Arduino, Learning arithmetic to children

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω ιδιαίτερα τον κ.Καρπούζη για την ανάθεση της παρούσας διπλωματικής εργασίας και για τη βοήθειά και την καθοδήγησή του καθ' όλη τη διάρκεια της εκπόνησής της.

Ευχαριστώ επίσης τον Σταύρο Αποστόλου που ήταν δίπλα μου σε κάθε τροχοπέδη κατά την εκπόνηση αυτής της διπλωματικής εργασίας και για τις εναλλακτικές ιδέες του ως προς την υλοποίησή της, καθώς και τα μέλη του εργαστηρίου Ψηφιακής Επεξεργασίας Σημάτων και Εικόνων που με βοηθούσαν σε κάθε απορία και δυσκολία που αντιμετώπιζα. Τέλος, ευχαριστώ πολύ την οικογένειά μου για τη στήριξη καθ' όλη τη διάρκεια των σπουδών μου στο Εθνικό Μετσόβιο Πολυτεχνείο.

Πίνακας περιεχομένων

1	Εισαγωγή	14
1.1	Γενική περιγραφή εργασίας	14
1.2	Ποιος ο σκοπός της παρούσας διπλωματικής εργασίας.....	15
1.3	Οργάνωση κειμένου	16
2	Σχετικές εργασίες	18
2.1	Ρομπότ Topo της εταιρείας Abdrobot.....	18
2.2	Lego Mindstorms Robotic System	19
2.3	P.BOT 100.....	19
2.4	Kaspar.....	20
2.5	Robotic blocks	20
3	Θεωρητικό υπόβαθρο.....	21
3.1	Ρομποτική-Ρομπότ	21
3.2	Εκπαιδευτική ρομποτική-Edycational robotics	22
3.3	Ρομποτικά συστήματα	26
3.4	Serious games	26
3.5	Θεωρία του νου (Theory of mind).....	27
3.6	Θεωρία ροής (Flow theory).....	28
3.7	Nao Robot. Γιατί προτιμήσαμε αυτό το ρομπότ;.....	29
3.8	Arduino.....	30
3.9	Gostai Suite.....	31
3.10	Choregraphe.....	32
4	Αναλυτική περιγραφή σεναρίου	33
4.1	Σενάριο	33
4.2	Αποστολές	34
4.3	Διάγραμμα καταστάσεων.....	36
5	Υλοποίηση σεναρίου	37
5.1	Σύνδεση του Gostai με το Nao.....	37
5.2	Σύνδεση με το Choregraphe.....	38

5.3	Υλοποίηση αποστολών.....	38
5.3.1	Δημιουργία <i>Urbigraph</i>	39
5.3.2	Αναλυτική παρουσίαση υλοποίησης.....	40
5.3.3	Το πρόγραμμα <i>Monitor</i>	70
6	Πειράματα - Δοκιμές	73
7	Προβλήματα	74
7.1	Προβλήματα γλώσσας.....	74
7.2	Προβλήματα "όρασης"	75
7.3	Πρόβλημα προσομοίωσης	76
7.4	Άλλα προβλήματα.....	76
8	Επίλογος.....	78
8.1	Σύνοψη, συμπεράσματα	78
8.2	Αποτελέσματα.....	80
8.3	Μελλοντικές επεκτάσεις, βελτιώσεις και προτάσεις	81
9	Βιβλιογραφία.....	83
	Ευρετήριο εικόνων	87

1 Εισαγωγή

1.1 Γενική περιγραφή εργασίας

Η χρήση ρομποτικών συστημάτων στην εκπαίδευση τα τελευταία χρόνια (εκπαιδευτική ρομποτική) έχει αποδείξει ότι έχει αξιοθαύμαστα και γρήγορα αποτελέσματα σε σύγκριση με τις κλασικές μεθόδους μάθησης. Παράλληλα, τα παιδιά μαθαίνουν από μικρή ηλικία να αλληλεπιδρούν με τους υπολογιστές και την ρομποτική τεχνολογία, καθώς αυτή η μέθοδος έχει αποδειχθεί ιδιαίτερα αποτελεσματική όταν εφαρμόζεται σε παιδιά με μαθησιακές δυσκολίες και αυτισμό([Ξ12]). Άλλωστε, η πρόοδος της τεχνολογίας τις τελευταίες δεκαετίες έχει φέρει τους ανθρώπους σε επαφή σε όλο και μικρότερη ηλικία.

Η εκπαιδευτική ρομποτική είναι ένας ραγδαία αναπτυσσόμενος κλάδος σε όλες τις βαθμίδες της εκπαίδευσης παγκοσμίως. Σε επόμενο κεφάλαιο παρατίθενται εργασίες που έχουν ήδη πραγματοποιηθεί σε Ελλάδα και εξωτερικό με σκοπό πάντα την εκπαίδευση με τη βοήθεια ρομπότ.

Το ερευνητικό ενδιαφέρον για εισαγωγή των ρομποτικών συστημάτων σε εκπαιδευτικές εφαρμογές είναι σε έξαρση, κυρίως λόγω της ωρίμανσης των χαμηλού κόστους ρομποτικών συστημάτων που διατίθενται, οι οποίες όμως διαθέτουν μεγάλη επεξεργαστική ισχύ σε σχέση με το πρόσφατο παρελθόν, αλλά και αισθητήρες (κάμερες, μικρόφωνο, αισθητήρες αφής) μέσω των οποίων αλληλεπιδρούν με το χρήστη με φυσικό τρόπο. Παράλληλα, η οικεία μορφή ενός ανθρωπόμορφου ρομπότ ([Ξ14]) το κάνει ιδανικό για την αλληλεπίδραση με παιδιά και έτσι, την ανάπτυξη σύγχρονου εκπαιδευτικού υλικού, το οποίο θα εκμεταλλεύεται έννοιες από την ψυχολογία, όπως η Θεωρία του Νου (Theory of Mind) για την εκτίμηση του ενδιαφέροντος και της προσήλωσης με βάση το βλέμμα του χρήστη και τη θεωρία ροής (flow) για τα παιχνίδια σοβαρού σκοπού (serious games).

Στο πλαίσιο της προτεινόμενης διπλωματικής εργασίας, θα υλοποιηθεί ένα σενάριο εκμάθησης βασικών γνώσεων αριθμητικής για παιδιά προσχολικής και πρώτης σχολικής ηλικίας. Το σενάριο αυτό εμπλέκει τη ρομποτική πλατφόρμα Nao της Aldebaran, ένα ανθρωποειδές ρομπότ που χρησιμοποιείται κατά κόρον και από το εργαστήριο σε μελέτες σε περιβάλλον παιδιών. Χρησιμοποιώντας το περιβάλλον ανάπτυξης Gostai Studio, το σενάριο θα πρέπει να εμπλέκει τη φυσική σημασία των αριθμητικών πράξεων με την εξέλιξη στο παιχνίδι: για παράδειγμα, ο παίκτης θα διαθέτει ειδικά κατασκευασμένα “νομίσματα” που θα φέρουν πάνω τους ένα λογότυπο, έτσι ώστε να είναι εφικτή η οπτική αναγνώριση και καταμέτρησή τους από το Nao - για να μπορέσει ο παίκτης να περάσει στο επόμενο “επίπεδο” θα πρέπει να συμπληρώσει συγκεκριμένο αριθμό νομισμάτων, εκτελώντας “αποστολές” που θα του επιφέρουν πρόσθετα νομίσματα και η εκπλήρωσή τους θα πρέπει να μπορεί να επιβεβαιωθεί μέσω των αισθητήρων του ρομπότ ή πρόσθετων (π.χ. καμερών, τοποθετημένων εκεί που πρέπει να πάει ο παίκτης).

1.2 Ποιος ο σκοπός της παρούσας διπλωματικής εργασίας

Στη σύγχρονη εποχή, η ρομποτική έχει εισβάλει στην παραγωγική διαδικασία των αγαθών (αυτοματοποιημένη παραγωγή) αλλά και στην καθημερινότητά μας (ρομποτικές ηλεκτρικές σκούπες, ρομποτικοί οικιακοί βοηθοί). Δικός μας σκοπός είναι να ενισχύσουμε την εισαγωγή των ρομποτικών συστημάτων και στην εκπαίδευση, γεγονός που ήδη συμβαίνει σε σχολεία του εξωτερικού με αξιοσημείωτα αποτελέσματα.

Τα σημερινά ρομποτικά συστήματα είναι προσιτά προς τον άνθρωπο και συνήθως είναι ανθρωπόμορφα, γεγονός που τα κάνει ακόμη πιο οικεία. Διαθέτουν αισθητήρες, κάμερες, μπορούν να κινούνται, να ακούν τον άνθρωπο, να του απαντούν και να τον αναγνωρίζουν.

Δική μας αποστολή δεν είναι να φτιάξουμε μόνο ένα παιχνίδι εκμάθησης αριθμητικής, αλλά να φτιάξουμε ένα παιχνίδι σοβαρού σκοπού που θα μπορεί να μαθαίνει σε παιδιά προσχολικής και πρώτης σχολικής ηλικίας βασικές αριθμητικές έννοιες με έναν ευχάριστο και συνάμα δημιουργικό τρόπο όπως θα δούμε παρακάτω. Έτσι, τα παιδιά θα εξοικειώνονται με τη διαδικασία επίλυσης προβλημάτων, ενώ παράλληλα θα εξασκούνται και θα γνωρίζουν τα βασικά στοιχεία ενός ρομποτικού συστήματος. Θα κατανοήσουν ότι για ένα ρομπότ δεν είναι τίποτα αυτονόητο γιατί είναι μία μηχανή προγραμματιζόμενη από τον άνθρωπο. Αξίζει να σημειώσουμε ότι ένα ανθρωπόμορφο ρομπότ βοηθά σημαντικά παιδιά με μαθησιακές δυσκολίες και προβλήματα κοινωνικοποίησης, σύμφωνα με έρευνες που έχουν γίνει από το

πανεπιστήμιο της Καλιφόρνια. Παιδιά με αυτισμό μαθαίνουν να λειτουργούν σωστά στο κοινωνικοσυναισθηματικό περιβάλλον τους¹.

Πώς θα το κατορθώσουμε; Με πολύτιμα εργαλεία υλικού και λογισμικού. Για την ακρίβεια, θα χρησιμοποιήσουμε τη ρομποτική πλατφόρμα Nao της εταιρείας Aldebaran. Ένα ανθρωπόμορφο ρομπότ με πολλές δυνατότητες και καλά χαρακτηριστικά. Ένα ρομπότ που διαθέτει γρήγορο επεξεργαστή, 25 βαθμούς ελευθερίας, δική του μνήμη, κάμερες, αισθητήρες αφής, , μπορεί να κινείται όπως ένας άνθρωπος, να ακούν τον άνθρωπο, να του απαντά και να τον αναγνωρίζει και πολλές άλλες δυνατότητες που θα δούμε στην πορεία. Επιπλέον, θα χρησιμοποιήσουμε το προγραμματιστικό περιβάλλον Gostai Suite για να βάλουμε τις αποστολές μας σε σειρά και να τις προγραμματίσουμε όπως επιθυμούμε και τέλος την πλατφόρμα Arduino η οποία σε συνεργασία με το Nao θα μας βοηθήσει να εκπληρώσουμε μία από τις αποστολές μας.

Ας ελπίσουμε αυτή η διπλωματική εργασία να είναι ένα έναυσμα για να φέρουμε την εκπαιδευτική ρομποτική στα ελληνικά σχολεία και για να εισάγουμε τα ρομπότ σε εκπαιδευτικές εφαρμογές που είναι τόσο ευεργετικές για τα παιδιά.

1.3 Οργάνωση κειμένου

Τα κεφάλαια της διπλωματικής εργασίας έχουν την εξής διαρρύθμιση:

Στο *πρώτο κεφάλαιο* κάνουμε μία εισαγωγή του θέματος εργασίας και μία γενική περιγραφή. Αναφέρουμε το γενικό σκοπό της, αλλά και τι μπορούμε να κατορθώσουμε μέσα από αυτή, όπως για παράδειγμα η εκμάθηση βασικών αρχών αριθμητικής σε παιδιά.

Στο *δεύτερο κεφάλαιο* παραθέτουμε παρεμφερείς εργασίες που έχουν γίνει στο παρελθόν, με θέμα την εκπαιδευτική ρομποτική, τα παιχνίδια σοβαρού σκοπού και την ενίσχυση παιδιών με μαθησιακές δυσκολίες.

Στο *τρίτο κεφάλαιο* αναλύουμε κάποιες θεωρητικές έννοιες πάνω στις οποίες βασίζεται η εργασία μας, όπως είναι τα ρομποτικά συστήματα και τα παιχνίδια σοβαρού σκοπού. Παράλληλα, απαριθμούμε τους λόγους για τους οποίους επιλέξαμε το Nao robot και παραθέτουμε πληροφορίες σχετικά με το λογισμικό και το υλικό που χρησιμοποιήσαμε.

Στο *τέταρτο κεφάλαιο* ξεκινά το πρακτικό μέρος της εργασίας, και ξεκινάμε με την αναλυτική περιγραφή του σεναρίου. Εδώ εξηγούμε αναλυτικά το σενάριο του εκπαιδευτικού παιχνιδιού,

¹ <http://articles.latimes.com/2011/oct/17/health/la-he-autism-robots-20111017>

αναλύουμε τις αποστολές μία προς μία και τέλος παίρνουμε μία πρώτη γεύση από το διάγραμμα καταστάσεων του σεναρίου.

Στο *πέμπτο κεφάλαιο* έχουμε την αναλυτική παρουσίαση της υλοποίησης του σεναρίου. Εξηγούμε αναλυτικά τη διαδικασία σύνδεσης με το ρομπότ, το διάγραμμα καταστάσεων όπως απεικονίζεται στο Gostai Studio καθώς και τον κώδικα κάθε κόμβου. Αξίζει να επισημάνουμε ότι αποτελεί το πιο σημαντικό κεφάλαιο της εργασίας, καθώς εδώ λύνουμε κάθε σκοτεινό και δυσνόητο σημείο υλοποίησης των αποστολών.

Ακολουθεί το *έκτο κεφάλαιο* το οποίο εξηγεί σε ποιες ομάδες πληθυσμού γίνεται το πείραμα και ποια στοιχεία παρατηρήσαμε και καταγράψαμε.

Στο *έβδομο κεφάλαιο* δίνουμε μία εικόνα των προβλημάτων που αντιμετωπίσαμε σε όλη τη διάρκεια της υλοποίησης των αποστολών.

Στο *όγδοο κεφάλαιο* κάνουμε μία σύνοψη των αποτελεσμάτων και καταγράφουμε τα συμπεράσματά μας. Επίσης, προτείνουμε μελλοντικές βελτιώσεις κάποιων τμημάτων της υλοποίησης.

Στο *ένατο και τελευταίο κεφάλαιο* παραθέτουμε τη βιβλιογραφία που χρησιμοποιήσαμε για να συντάξουμε την εργασία αλλά και για την υλοποίησή της. Επίσης, θεωρήσαμε χρήσιμο να παραθέσουμε και κάποιες ηλεκτρονικές διευθύνσεις που μας έδωσαν χρήσιμες θεωρητικές και πρακτικές κατευθύνσεις.

2 Σχετικές εργασίες

Στην προετοιμασία αυτής της διπλωματικής και αναζητώντας το ακριβές της θέμα, εντοπίσαμε άρθρα, εργασίες και εφαρμογές που έχουν ως σκοπό την εκπαίδευση παιδιών με τη βοήθεια ρομποτικών συστημάτων, την βοήθεια παιδιών με προβλήματα υγείας ή μαθησιακές δυσκολίες. Δηλαδή, εφαρμογές της εκπαιδευτικής ρομποτικής:

2.1 *Kaspar*²

Τα ρομπότ δεν είναι γνωστά για την ευαισθησία τους. Αντιθέτως, δεν έχουν σχέση με συναισθήματα. Όμως, μία νέα ομάδα ερευνητών δουλεύει για να κατασκευάσει ρομπότ που θα συνδέονται συναισθηματικά με αυτιστικά παιδιά και θα τα βοηθούν να αλληλεπιδράσουν με τους ανθρώπους. Το ρομπότ που ονομάζεται Κάσπαρ είναι προγραμματισμένο να χαμογελάει, να γελάει, να κλείνει το μάτι και να κουνάει τα χέρια του. Έχει το ύψος ενός μικρού παιδιού και είναι κατευθυνόμενο από καθηγητές μέσω υπολογιστή.



Εικ 1: Kaspar robot

² <http://www.physorg.com/news/2011-03-kaspar-friendly-robot-autistic-kids.html>

2.2 *Lego Mindstorms Robotic System [Ξ8]*

Ένα ρομπότ που έγινε γνωστό το 1998 από την εταιρεία Lego. Αμέσως κυκλοφόρησε στα σχολεία του εξωτερικού με σκοπό να μάθει στους μαθητές βασικές έννοιες της τεχνολογίας, της μηχανικής και των μαθηματικών, καθώς ήταν ο κώδικάς του Open Source και μπορούσαν οι μαθητές να το προγραμματίσουν και να παίξουν μαζί του αποκτώντας πολύτιμες γνώσεις. Το ρομπότ αυτό περιελάμβανε δύο κινητήρες, δύο αισθητήρες αφής και έναν αισθητήρα φωτός, που σημαίνει ότι είναι κατάλληλο για αλληλεπίδραση με μικρά παιδιά.

2.3 *P.BOT 100³*

Μία επαναστατική εφαρμογή αποτελεί το συγκεκριμένο εκπαιδευτικό ρομπότ. Συγκεκριμένα, βοηθάει παιδιά που πάσχουν από σοβαρές ασθένειες να είναι «παρόντα» στην τάξη τους. Είναι ένα ρομπότ με κάμερα, μικρόφωνο και ακουστικό το οποίο πηγαίνει στο σχολείο αντί του παιδιού και το φέρνει σε διαρκή επικοινωνία μέσω διαδικτύου με την τάξη.



Εικόνα.2: P.BOT 100

³ http://news.cnet.com/8301-17938_105-20030156-1.html

2.4 Ρομπότ Τορο της εταιρείας Abdrobot⁴

Ένα εκπαιδευτικό ρομπότ το οποίο βγήκε στην αγορά το 1980 για καταναλωτές και για εκπαιδευτικούς σκοπούς. Η γλώσσα προγραμματισμού του επέτρεπε στο ρομπότ να κάνει γεωμετρικές κινήσεις, να κινείται μέσα σε ένα δωμάτιο σε τροχιές γεωμετρικών σχημάτων, να κάνει διάφορες εργασίες και να μιλάει με τους ανθρώπους γύρω του.

2.5 Robotic blocks⁵

Το ρομποτικό παιχνίδι που σχεδίασαν οι **Ethan Huber** και **Haggai Levi**, αποτελείται από διάφορα τμήματα και συνεργαζόμενες μονάδες που δίνει στα παιδιά την δυνατότητα να δημιουργήσουν απλές εντολές και να προγραμματίσουν την κίνηση του ρομποτικού παιχνιδιού τους.

⁴ <http://www.robots-and-androids.com/educational-robots.html>

⁵ <http://www.designhome.gr/reportaz/nea-proionta/358-2012-01-19-09-35-26.html>

3 Θεωρητικό υπόβαθρο

3.1 Ρομποτική⁶ - Ρομπότ

Ρομποτική είναι η επιστήμη και η τεχνολογία των ρομπότ, που περιλαμβάνει τον σχεδιασμό, την κατασκευή και τον προγραμματισμό τους. Η λέξη ρομπότ εισήχθη από τον Τσέχο συγγραφέα Karel Capek και στα τσέχικα «roboty» σημαίνει «εργασία». Ρομπότ είναι μία ηλεκτρομηχανική ή βιομηχανική συσκευή που μπορεί να εκτελέσει αυτόνομα ή προκαθορισμένα καθήκοντα. Συνήθως τα ρομπότ χρησιμοποιούνται για να κάνουν εργασίες που είναι δύσκολες και επικίνδυνες για τον άνθρωπο ή στην αυτοματοποιημένη παραγωγή. Τα ρομπότ μπορούν να έχουν διαφορετικές μορφές και κειμούνται από ανθρωποειδή (μιμούνται την ανθρώπινη φωνή και τον τρόπο κίνησης) έως βιοχημικά των οποίων η μορφή είναι ανάλογη της αποστολής τους. Ένα ρομπότ διαθέτει:

- ✓ Αισθητήρες για να μπορεί να κυκλοφορήσει στο περιβάλλον
- ✓ Δυνατότητες επεξεργασίας (αντίληψη, συλλογισμός)
- ✓ Επενεργητές (actuators) για την εκτέλεση κάποιας εργασίας στο περιβάλλον

Επιπλέον, τα χαρακτηριστικά που το κάνουν να διαφέρει από τα υπόλοιπα υπολογιστικά συστήματα είναι τα εξής:

- ✓ Δυνατότητα επαναπρογραμματισμού
- ✓ Δυνατότητα μηχανικής δράσης
- ✓ Προσαρμοστικότητα και ευελιξία

⁶ <http://en.wikipedia.org/wiki/Portal:Robotics>

Κάποιες κατηγορίες ρομπότ είναι:

- ✓ Κινητά ρομπότ, π.χ. αυτόνομα οχήματα.
- ✓ Ρομπότ-χειριστές, π.χ. βιομηχανικά ρομπότ.
- ✓ Αυτο-αναδιαρθρώσιμα ρομπότ που προσαρμόζονται ανάλογα με την αποστολή τους.
- ✓ Ρομπότ ελεγχόμενα από τον άνθρωπο, π.χ. εξουδετέρωση εκρηκτικών.
- ✓ Ρομπότ που αποφασίζουν σύμφωνα με την τεχνητή νοημοσύνη τους.

Το 1950 ο ρωσικής καταγωγής Αμερικανός καθηγητής και πολυγραφότατος συγγραφέας Ισαάκ Ασίμοφ έγραψε το ιστορικό πιά βιβλίο επιστημονικής φαντασίας «Εγώ, το ρομπότ».⁷

Οι τρεις θρυλικοί νόμοι της ρομποτικής που περιέγραφε σ' αυτό (και αποτέλεσαν το υπόβαθρο, πάνω στο οποίο βασίστηκαν στο εξής τα βιβλία του) προκάλεσαν μεγάλη αίσθηση ακόμα και στους επιστημονικούς κύκλους, όντας το αντίστοιχο των «Δέκα Εντολών» στο χώρο των ρομπότ.

Αυτές ήταν:

- ✓ Ένα ρομπότ δεν μπορεί να βλάψει έναν άνθρωπο ούτε να τον αφήσει να πάθει κακό
- ✓ Ένα ρομπότ πρέπει να υπακούει στις διαταγές που του δίνει ο άνθρωπος, εκτός κι αν αυτό που του ζητάει έρχεται σε αντίθεση με τον πρώτο νόμο, και
- ✓ Ένα ρομπότ πρέπει να προφυλάσσει τον εαυτό του, εκτός κι αν αυτό αντιβαίνει στον πρώτο και το δεύτερο νόμο...

3.2 Εκπαιδευτική ρομποτική – *Educational robotics*⁸

Η εκπαιδευτική ρομποτική⁹ είναι ένας ραγδαία αναπτυσσόμενος κλάδος σε όλες τις βαθμίδες της εκπαίδευσης παγκοσμίως. Τα εκπαιδευτικά ρομπότ έχουν τα ίδια χαρακτηριστικά με τα βιομηχανικά ρομπότ, είναι χαμηλότερου κόστους και είναι ασφαλέστερο να χρησιμοποιηθούν από μαθητές και φοιτητές.

Εκπαιδευτική ρομποτική ονομάζεται το υπολογιστικό περιβάλλον που εποτελείται από ένα ή περισσότερα ρομπότ (είτε αυτόνομα είτε συνοδευόμενα από υπολογιστή) το οποίο ενθαρρύνει τους μαθητές να σκεφτούν καλύτερα ένα πρόβλημα, να συνεργαστούν, βοηθά τους εκπαιδευόμενους να αποκτήσουν γνώσεις, κριτική σκέψη, εξοικείωση με τους

⁷ <http://technologein.pathfinder.gr/nao-robots/>

⁸ http://en.wikipedia.org/wiki/Category:Educational_robots

⁹ http://earthlab.uoi.gr/earthlab_files/articles/Educational_Robotics_as_mindtools.pdf

υπολογιστές. Επιπλέον, τα ρομπότ πάνε ένα βήμα παραπέρα καθώς βγάζουν τον μαθητή από τα στενά όρια της οθόνης του υπολογιστή στον πραγματικό κόσμο. Αυτό βοηθά τους μαθητές να ξεπερνούν ουσιαστικές δυσκολίες που θα είχαν αν δούλευαν μόνο με έναν υπολογιστή. Πρακτικές εφαρμογές της εκπαιδευτικής ρομποτικής είδαμε σε προηγούμενη παράγραφο.

Η ρομποτική¹⁰ αφενός, είναι μία διασκεδαστική και ενδιαφέρουσα δραστηριότητα που δίνει τη δυνατότητα στο μαθητή να εμπλακεί με τη δράση, αφετέρου μπορεί να χρησιμοποιηθεί σε όλες τις βαθμίδες εκπαίδευσης για τη **διδασκαλία διαφόρων εννοιών**, κυρίως, από τις Φυσικές Επιστήμες και άλλα γνωστικά αντικείμενα.([H8])

- ✓ Φυσική (μελέτη της κίνησης, μελέτη της επίδρασης της τριβής, μελέτη της σχέσης των δυνάμεων, μεταφορά ενέργειας κ.α)
- ✓ Μαθηματικά και Γεωμετρία (αναλογίες, μέτρηση αποστάσεων, κατανόηση βασικών γεωμετρικών ιδιοτήτων όπως η περίμετρος κ.α)
- ✓ Μηχανική (κατασκευή, έλεγχος και αξιολόγηση μηχανικών λύσεων κ.α)
- ✓ Τεχνολογία (τεχνολογικός αλφαριθμητισμός κ.α)

Η εκπαιδευτική Ρομποτική ([E1], [E2]) έχει **θετικές επιπτώσεις** εκτός από το γνωστικό τομέα και **στο συναισθηματικό** (αυτοεκτίμηση,αυτοπεποίθηση) και **κοινωνικό** (κοινωνικοποίηση, απομυθοποίηση). Επιπλέον, με τη βοήθεια της ρομποτικής στη διδασκαλία του ο εκπαιδευτικός μπορεί να επικεντρωθεί στην **ανάπτυξη κρίσιμων δεξιοτήτων** του 21ου αιώνα ([E3]):

- ✓ ομαδική εργασία
 - ✓ επίλυση προβλημάτων (ανάλυση, σχεδίαση, υλοποίηση, δοκιμή και πειραματισμός, αξιολόγηση)
 - ✓ καινοτομία
 - ✓ διαχείριση έργου (διαχείριση χρόνου, κατανομή έργου και πόρων κ.α)
 - ✓ προγραμματισμός
 - ✓ δεξιότητες επικοινωνίας
 - ✓ πολύτιμες νοητικές δεξιότητες (αναλυτική και συνθετική σκέψη, δημιουργικότητα, κριτικήσκέψη)
- κ.λπ.

¹⁰<http://edurobotics.weebly.com/epsilonkappapialphaiotadeltaepsilonupsilontauioetakappa942-rhoomicronmupiomicrontauioetakappa942.html>

αλλά και προσωπικών ικανοτήτων ([Ξ10]):

- ✓ Συνεργασία με άλλα παιδιά
- ✓ Γνωστικές δεξιότητες
- ✓ Αυτοεκτίμηση
- ✓ Πνευματική αντίληψη
- ✓ Χωρική αντίληψη

Όπως συμπεραίνουμε από όλα τα παραπάνω, η ρομποτική φέρνει επαναστατικές αλλαγές στην εκπαίδευση και αλλάζει ριζικά τον παραδοσιακό χαρακτήρα της διδασκαλίας:

- ✓ Η εκπαιδευτική ρομποτική συνδυάζει τη μάθηση με το παιχνίδι και έτσι μετατρέπει την εκπαίδευση σε μία διασκεδαστική δραστηριότητα
- ✓ Εμπλέκει ενεργά τους μαθητές στη μάθησή τους με την επίλυση αυθεντικών προβλημάτων συνεργαζόμενα και με άλλα παιδιά ([Ξ6])
- ✓ Η Εκπαιδευτική ρομποτική παρέχει στους μαθητές ευκαιρίες επίλυσης προβλημάτων με προσωπικό νόημα για τους ίδιους μέσω χειρισμού και κατασκευής πραγματικών ή ιδεατών αντικειμένων.
- ✓ Μπορεί ακόμα να αξιοποιηθεί σαν «σκαλωσιά» για την κατάκτηση συνθετότερων εννοιών
- ✓ ([Ε6],[Ξ3]) Ως παιδαγωγική προσέγγιση εγγράφεται στο πλαίσιο του κλασικού εποικοδομιστικού (constructivism) και ειδικότερα του κατασκευαστικού εποικοδομισμού (constructionism). Η εποικοδομητική (constructivist) αντίληψη για το φαινόμενο της μάθησης υποστηρίζει ότι το μαθησιακό περιβάλλον θα πρέπει να παρέχει αυθεντικές δραστηριότητες ενταγμένες σε διαδικασίες επίλυσης ανοιχτών προβλημάτων από τον πραγματικό κόσμο, να ενθαρρύνει την έκφραση και την προσωπική εμπλοκή στη μαθησιακή διαδικασία και να υποστηρίζει την κοινωνική αλληλεπίδραση. Επιπλέον ο “κατασκευαστικός” εποικοδομισμός (constructionism) υποστηρίζει ότι οι μαθητευόμενοι οικοδομούν πιο αποτελεσματικά τη γνώση όταν εμπλέκονται ενεργά στη σχεδίαση και κατασκευή (χειρωνακτική και ψηφιακή) πραγματικών αντικειμένων που έχουν νόημα για τους ίδιους είτε αυτά είναι κάστρα από άμμο, είτε κατασκευές LEGO και προγράμματα υπολογιστών (Papert, 1991).

Τέλος, ([Ξ1]) οι δραστηριότητες της εκπαιδευτικής ρομποτικής μπορούν να αναλυθούν και να χαρακτηριστούν από τα ακόλουθα δύο μοντέλα:

- ✓ Αρχιτεκτονική ικανοτήτων τεσσάρων επιπέδων με τα εξής επίπεδα:
 - **Δυναμικές ικανότητες:** οι δυναμικές ικανότητες σχετίζονται άμεσα με το κίνητρο. Οι ικανότητες αυτές ανταποκρίνονται στην πρωτοβουλία, τη θέληση, την ευχαρίστηση και το κίνητρο του εκπαιδευόμενου. Άρα, η εκπαιδευτική ρομποτική μέσα από αυτές στοχεύει στη δημιουργία μιας εργασίας με νόημα.
 - **Στρατηγικές ικανότητες:** έχουν άμεση σχέση με τη γνώση των δυνατοτήτων μας. Μέσα από την εκπαιδευτική ρομποτική στόχος αυτών των ικανοτήτων είναι η κοινωνικοποίηση και η επίλυση προβλημάτων.
 - **Αποπολλαπλασιαστικές ικανότητες:** αυτές οι ικανότητες επιτρέπουν στο χρήστη να βρει πληροφορίες από μόνος του και να αναπτύξει συγκεκριμένες ικανότητες (διάβασμα, ακοή, επικοινωνία)
 - **Συγκεκριμένες ικανότητες:** αυτές οι ικανότητες αφορούν συγκεκριμένα περιεχόμενα (ιστορία, φυσική κλπ.). Μέσα από την εκπαιδευτική ρομποτική, οι ικανότητες αυτές στοχεύουν στο να αναπτύξουν τις ικανότητες του ατόμου όσον αφορά τον προγραμματισμό και την τεχνολογία.

- ✓ Τα έξι μαθησιακά και διδακτικά παραδείγματα. Αυτά μας δείχνουν διαφορετικά είδη δραστηριοτήτων μεταξύ εκπαιδευτή-εκπαιδευόμενου:
 - **Δημιουργία:** κατασκευαστικό και προγραμματιστικό κομμάτι του ρομπότ
 - **Εξερεύνηση:** ανακάλυψη κάποιων εννοιών από τον εκπαιδευόμενο, όπως η διαδικτυακή βοήθεια, οι οδηγοί αναφορών και άλλα
 - **Πειραματισμός του μαθητή:** ελευθερία του μαθητή να πειραματιστεί με το ρομπότ τόσο στο κατασκευαστικό κομμάτι όσο και στο προγραμματιστικό
 - **Μίμηση** ενεργειών και πραγμάτων που έχει ήδη δει ο μαθητής αλλού.
 - **Υποδοχή:** ικανότητα να υποδέχεται καλοπροαίρετα νέες ιδέες
 - **Εξάσκηση:** μέσα από την εκπαιδευτική ρομποτική και την ικανότητα των εκπαιδευομένων να επαναλάβουν διαδικασίες, εξασκούνται σε αυτό που τους ενδιαφέρει και γίνονται όλο και καλύτεροι.

3.3 Ρομποτικά συστήματα¹¹

Ρομποτικό σύστημα ονομάζεται το μηχανικό σύστημα το οποίο αισθάνεται, σκέφτεται και επενεργεί, διότι έχει αισθητήρες με τη βοήθεια των οποίων έρχεται σε επαφή με το περιβάλλον και με την εσωτερική του κατάσταση. Παράλληλα, διαθέτει υπολογιστικές ικανότητες με τη βοήθεια των οποίων αποφασίζει για διάφορες καταστάσεις. Οι πιο διαδεδομένες κατηγορίες ρομποτικών συστημάτων είναι οι παρακάτω:

- ✓ Βιομηχανικοί (κλασσικοί) ρομποτικοί χειριστές (industrial robot manipulators)¹²
- ✓ Επιδέξιοι ρομποτικοί χειριστές (dextrous robots)
- ✓ Αυτοκινούμενα ρομπότ – ρομπότ προσφοράς υπηρεσιών (mobile/service robotics)
- ✓ Μικρο-ρομποτική (micro-robotics)

3.4 *Serious games* [E5]

Ένα σοβαρό παιχνίδι ¹³ (**serious game**) είναι ένα παιχνίδι σχεδιασμένο για έναν συγκεκριμένο σκοπό, και όχι για πλήρη ψυχαγωγία. Το επίθετο «σοβαρό» αναφέρεται σε προϊόντα που χρησιμοποιούνται από βιομηχανίες όπως η άμυνα, η εκπαίδευση, η επιστημονική έρευνα, η υγειονομική περίθαλψη, η μηχανική.

Τα σοβαρά παιχνίδια είναι σχεδιασμένα για να λύνουν ένα πρόβλημα. Παρ' όλο που είναι διασκεδαστικά, ο βασικός σκοπός τους είναι να εκπαιδεύσουν, να ερευνήσουν ή να διαφημίσουν. Κάποιες φορές ένα παιχνίδι μπορεί να προσφέρει τέρψη και διασκέδαση με σκοπό να κατορθώσει την επιθυμητή πρόοδο από τον παίκτη. Τα σοβαρά παιχνίδια είναι μία κατηγορία παιχνιδιών με διαφορετικούς σκοπούς. Αυτή η κατηγορία περιλαμβάνει εκπαιδευτικά παιχνίδια, πολιτικά παιχνίδια κλπ. Αυτή η κατηγορία σοβαρών παιχνιδιών που έχει σκοπό την εξάσκηση είναι γνωστή ως «εκπαιδευτική-μάθηση».

Τα σοβαρά παιχνίδια αποτελούν χρήσιμα εργαλεία για:

- ✓ Εταιρική κατάρτιση,
- ✓ Εκπαιδευτικές δράσεις και εκστρατείες ευαισθητοποίησης,
- ✓ Μάρκετινγκ,
- ✓ Δραστηριότητες προώθησης,
- ✓ Κοινωνικές εκστρατείες.

¹¹ <http://users.softlab.ece.ntua.gr/~ktzaf/Courses/robotics-II-1-1-intro.pdf>

¹² <http://www.plant-management.gr/index.php?id=14943>

¹³ http://en.wikipedia.org/wiki/Serious_game

Οι ρομποτικές πλατφόρμες και τα παιχνίδια σοβαρού σκοπού βρίσκονται στο επίκεντρο της εκπαιδευτικής τεχνολογίας, καθώς διαθέτουν χαρακτηριστικά που τα κάνουν οικεία και ενδιαφέροντα, ειδικά για παιδιά προσχολικής ηλικίας. Η ανθρωπόμορφη εμφάνιση ενός ρομπότ σαν το Nao και η ικανότητά του να αλληλεπιδρά με το παιδί, κάνει τα παιδιά να αναπτύσσουν πιο στενούς δεσμούς οικειότητας με το ρομπότ απότι με άλλα παιχνίδια. Παράλληλα, τα ηλεκτρονικά παιχνίδια βρίσκονται στην καθημερινότητα των παιδιών και μπορούν να βοηθήσουν σε μαθήματα γενικής υποδομής, όπως η φυσική ή τα μαθηματικά έως κοινωνικά θέματα. Αυτό που κάνει τα παιχνίδια ελκυστικά είναι η κατάσταση της ροής, στην οποία ο μαθητής κυνηγά τους εκπαιδευτικούς σκοπούς χωρίς να τους αντιλαμβάνεται σαν ξεχωριστή διαδικασία.

3.5 Θεωρία του νου (*Theory of mind*)¹⁴

([Ξ23]) Θεωρία του νου είναι η ικανότητα να αποδίδει κάποιος πνευματικές καταστάσεις (πεποιθήσεις, προθέσεις, επιθυμίες, γνώσεις κλπ) για τον εαυτό του και τους άλλους και να κατανοεί ότι οι άλλοι έχουν πεποιθήσεις, επιθυμίες και προθέσεις που διαφέρουν από τις δικές του. Αν και υπάρχουν φιλοσοφικές προσεγγίσεις για θέματα που τέθηκαν σε τέτοιες συζητήσεις, η θεωρία του νου είναι διαφορετική από τη φιλοσοφία του νου.

Η υπόθεση ότι οι άλλοι έχουν μυαλό, ονομάζεται θεωρία του νου, διότι κάθε άνθρωπος μπορεί να δεισθανθεί την ύπαρξη του νου του μέσω της εσωτερικής αναζήτησης και κανείς δεν έχει άμεση πρόσβαση στο μυαλό του άλλου. Τυπικά, θεωρείται ότι όλοι έχουν μυαλό με βάση τον αμοιβαίο χαρακτήρα της κοινωνικής αλληλεπίδρασης, τη λειτουργική χρήση της γλώσσας και την κατανόηση των συναισθημάτων και των κινήσεων των άλλων ανθρώπων. Η θεωρία του νου μας βοηθά να αποδώσουμε τις σκέψεις, τις επιθυμίες και τις προθέσεις μας σε σχέση με τους άλλους, να προβλέψουμε και να εξηγήσουμε τις πράξεις τους και να συζητάμε για τις προθέσεις τους. Δηλαδή, βοηθά κάποιον να κατανοήσει ότι οι νοητικές καταστάσεις μπορεί να είναι η αιτία άλλων συμπεριφορών.

¹⁴ http://en.wikipedia.org/wiki/Theory_of_mind

3.6 Θεωρία ροής (*Flow theory*)¹⁵

([Ξ22]) Όπως αναλύσαμε παραπάνω, τα παιχνίδια σοβαρού σκοπού έχουν άμεσο σκοπό να δώσουν γνώσεις στον παίκτη μέσα από την επίλυση προβλημάτων και όχι τόσο να τέρψουν τον χρήστη. Εδώ λοιπόν υπεισέρχεται η έννοια της ροής.

Ο ψυχολόγος Mihaly Csikszentmihalyi ονομάζει ροή το συναίσθημα της πλήρους και ενεργοποιημένης εστίασης σε μια δραστηριότητα, με ένα υψηλό επίπεδο απόλαυσης και εκπλήρωσης. Η θεωρία αυτή μπορεί να εφαρμοστεί σε παιχνίδια σοβαρού σκοπού, σε βιντεοπαιχνίδια, στο διάβασμα αλλά και σε κάθε δραστηριότητα της καθημερινής ζωής.

Τα συστατικά μιας ροής που παράγει τη δραστηριότητα είναι:

- ✓ Είμαστε σε θέση να επικεντρωθούμε στη δραστηριότητα
- ✓ Η δραστηριότητα έχει σαφείς στόχους
- ✓ Θεωρούμε ότι ελέγχουμε τη δραστηριότητα
- ✓ Απώλεια αυτοσυνείδησης
- ✓ Οι ανησυχίες μας εξαφανίζονται
- ✓ Είμαστε συγκεντρωμένοι και προσηλωμένοι
- ✓ Ισορροπούμε μεταξύ ικανότητας και πρόκλησης
- ✓ Λανθασμένη αίσθηση του χρόνου

Δηλαδή, οι προκλήσεις που παρουσιάζονται και η ικανότητα του ατόμου για την επίλυσή τους, σχεδόν ταυτίζονται και πολλές φορές το άτομο κατορθώνει πράγματα για τα οποία δε γνώριζε ότι είναι ικανό. Εκεί ακριβώς βασίζονται τα παιχνίδια σοβαρού σκοπού. Αν το παιχνίδι είναι πολύ εύκολο, καταλήγει να χάνει το ενδιαφέρον του ο παίκτης. Αν είναι δύσκολο, ο παίκτης απογοητεύεται. Ένα καλό παιχνίδι πρέπει να κρατά το ενδιαφέρον του παίκτη και να τον διατηρεί σε κατάσταση ροής.

¹⁵ [http://wikipedia.qwika.com/en2el/Flow_\(psychology\)](http://wikipedia.qwika.com/en2el/Flow_(psychology))

3.7 Nao Robot. Γιατί προτιμήσαμε αυτό το ρομπότ; ([H6],[H7])

Το ρομπότ Nao¹⁶ είναι ένα αυτόνομο, προγραμματιζόμενο, μεσαίου μεγέθους ανθρωπόμορφο ρομπότ που δημιούργησε η Aldebaran το 2004. Είναι διαθέσιμο σε πανεπιστήμια και εργαστήρια για έρευνα και εκπαιδευτικούς σκοπούς καθώς είναι κατάλληλο για να αλληλεπιδράσει με τον άνθρωπο και με παιδιά

Τεχνικά χαρακτηριστικά:¹⁷

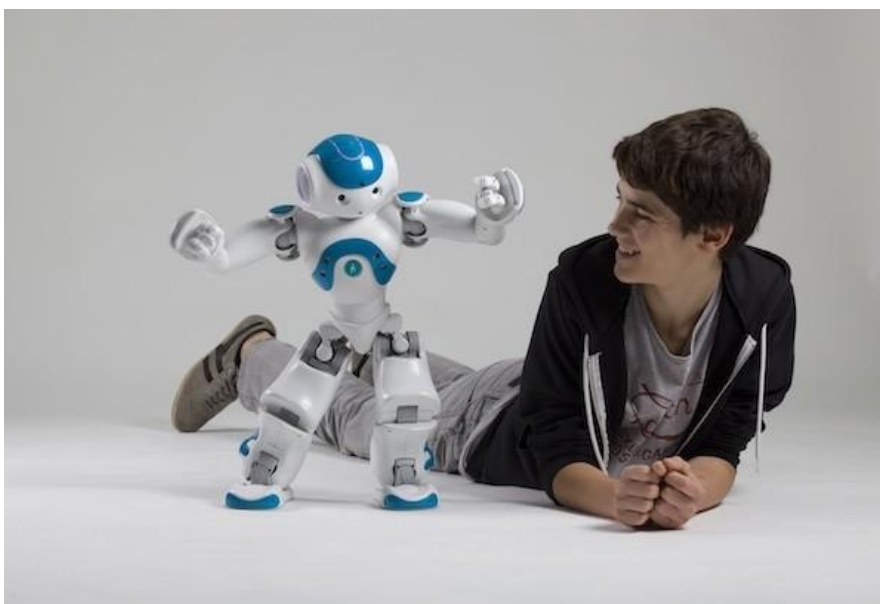
- ✓ Ύψος: 58cm
- ✓ Βάρος: 4.3 kg
- ✓ Αυτονομία μπαταρίας: 90 λεπτά
- ✓ Βαθμοί ελευθερίας :25
- ✓ CPU: Intel Atom 1,6 GHz με πυρήνα που τρέχει linux
- ✓ Δίκτυο αισθητήρων, δύο κάμερες 920p στο μέτωπο και στο στόμα , 4 μικρόφωνα, 2 εκπομπούς υπεριώθρων, 8 αισθητήρες πίεσης
- ✓ Πολλές συσκευές επικοινωνίας, όπως σύνθεση φωνής, Led φωτάκια, και 2 μεγάφωνα υψηλής ποιότητας.
- ✓ Ικανότητα να στέκεται, να κινείται, να περπατάει και να αναγνωρίζει πρόσωπα, φωνές και αντικείμενα
- ✓ Αναγνώριση ειδικών συμβόλων (Naomarks) για τα οποία θα μιλήσουμε στη συνέχεια
- ✓ Προγραμματίζεται σε C++ και σε Python
- ✓ Συνδεσιμότητα: υποστηρίζει wifi και Ethernet, και είναι συμβατό με το πρωτόκολλο IEEE 802.11g Wi-Fi standard. Επιπλέον, επικοινωνεί και με άλλα Nao μέσω υπεριώθρων.
- ✓ Ο κώδικάς του είναι ανοιχτός για όλους (Open Source).

Πρόσφατα¹⁸ η Aldebaran αναβάθμισε το Nao και έτσι τώρα πλέον δημιούργησε το λεγόμενο "Nao Next Generation" , τον νέο διάδοχο του Nao. Έχει βελτιωμένες ικανότητες όπως καλύτερη κάμερα και άρα καλύτερη αναγνώριση, καλύτερη άρθρωση, καλύτερο περπάτημα, περισσότερες γλώσσες.

¹⁶ [http://en.wikipedia.org/wiki/Nao_\(robot\)](http://en.wikipedia.org/wiki/Nao_(robot))

¹⁷ <http://www.aldebaran-robotics.com/en/Discover-NAO/Key-Features/hardware-platform.html>

¹⁸ <http://www.engadget.com/2011/12/10/aldebaran-robotics-announces-nao-next-gen-humanoid-robot-video/>



Εικ.3: Nao Robot

Οι λόγοι¹⁹ που μας ώθησαν να δουλέψουμε με το Nao είναι ουσιαστικοί. Το Nao επιτρέπει στα παιδιά μικρών ηλικιών και στους χρήστες του γενικότερα να εξοικειωθούν με τα ρομπότ και τις λειτουργίες τους καθώς είναι ικανό να αλληλεπιδράσει πολύ εύκολα με τους χρήστες και είναι φιλικό λόγω της ανθρώπινης μορφής του. Παράλληλα, η εκμάθηση γίνεται πιο ζωντανή και διασκεδαστική για όλες τις ηλικίες αφού μπορούν οι ίδιοι οι μαθητές να πειραματιστούν μόνοι μαζί του. Επιπλέον, οι τρόποι αλληλεπίδρασης του ανθρώπου με το ρομπότ είναι αρκετά απλοί και κατανοητοί και δεν απαιτούν γνώσεις προγραμματισμού ή τεχνολογίας όπως ο αισθητήρας αφής του κεφαλιού, η αναγνώριση φωνής και η αναγνώριση των Naomarks όπως θα δούμε παρακάτω.

3.8 *Arduino*^{20 21}

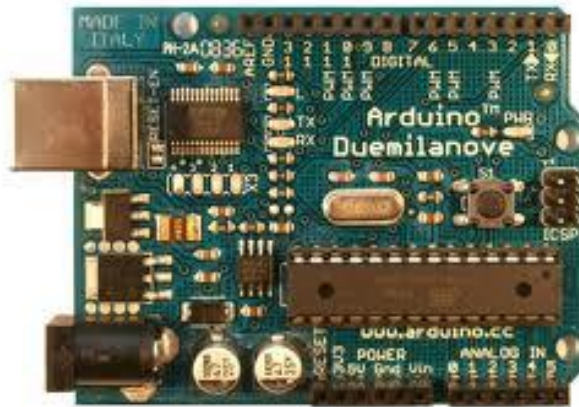
Το **Arduino** είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων/διαδραστικών αντικειμένων αλλά

¹⁹ <http://www.aldebaran-robotics.com/en/Solutions/For-Education/introduction.html>

²⁰ <http://arduino.cc/>

²¹ <http://el.wikipedia.org/wiki/Arduino>

και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους. Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Εμείς θα χρησιμοποιήσουμε το Arduino Duemilanove σε μία από τις αποστολές μας.



Εικ.4: Arduino Duemilanove

3.9 Gostai Suite²²

Το Gostai Suite είναι μία πολύ χρήσιμη σουίτα για να προγραμματίσουμε ένα ρομπότ σχεδιάζοντας βασικά ένα διάγραμμα καταστάσεων και γράφοντας λιγότερο κώδικα. Αυτό καθιστά το Gostai ιδιαίτερα φιλικό στον προγραμματιστή καθώς τον διευκολύνει σε ότι αφορά τη ροή του σεναρίου. Το Gostai Studio περιλαμβάνει έναν editor που θα χρησιμοποιήσουμε για να προγραμματίσουμε το ρομπότ μας και να αλληλεπιδράσουμε με το Nao. Συγκεκριμένα, μέσω αυτού του προγράμματος μπορούμε να δημιουργήσουμε εμείς τα διαγράμματα κατάστασεων ανάλογα με το τι περιμένουμε να κάνει το ρομπότ μας και το Gostai να παράγει τον κώδικα που θα «διαβάσει» το Nao. Το πρόγραμμα αυτό είναι σχεδιασμένο πάνω στην πλατφόρμα Urbi με τη βοήθεια της οποίας ελέγχουμε ρομποτικά

²² <http://www.gostai.com/products/studio/>

συστήματα. Η γλώσσα προγραμματισμού που χρησιμοποιούμε εμείς για να ορίσουμε τι θα κάνει το ρομπότ σε κάθε κατάσταση ονομάζεται Urbiscript και έχει πολλές ομοιότητες με τη C.

3.10 *Choregraphe*²³

Το Choregraphe είναι το λογισμικό που επιτρέπει στους χρήστες του Nao να δώσουν στο ρομπότ εντολές κίνησης και να αλληλεπιδράσουν με αυτό εύκολα. Χρησιμοποιώντας τις δυνατότητές του μπορούμε να προγραμματίσουμε το Nao χωρίς να μπούμε στη διαδικασία σύνταξης κώδικα. Είναι διαθέσιμο και μπορούμε να το «κατεβάσουμε» από την ιστοσελίδα της κατασκευάστριας εταιρείας Aldebaran. Η έκδοση που χρησιμοποιήσαμε εμείς είναι η 1.12.0.62.

²³ <http://www.aldebaran-robotics.com/en/Discover-NAO/Software/choregraphe.html>

4 Αναλυτική περιγραφή σεναρίου

4.1 Σενάριο

Όπως αναφέραμε και παραπάνω σκοπός της εργασίας είναι η εκμάθηση απλών αριθμητικών εννοιών σε παιδιά προσχολικής και πρώτης σχολικής ηλικίας με τη βοήθεια της ρομποτικής πλατφόρμας Nao και φυσικά με την παρουσία δασκάλου.

Η ιδέα του σεναρίου είναι η εξής:

Ο χρήστης βρίσκεται αντιμέτωπος με τέσσερις δοκιμασίες τις οποίες και πρέπει να φέρει εις πέρας επιτυχώς. Ξεκινά έχοντας στα χέρια του 2 ευρώ και πρέπει να αγοράσει μπιγιές που κοστίζουν 10 ευρώ. Σε κάθε αποστολή θα πρέπει να υπολογίζει πόσα χρήματα χρειάζεται ακόμη για να αποκτήσει τις μπιγιές και να απαντήσει ανάλογα με τον τρόπο που του υποδεικνύει το ρομπότ. Να διευκρινίσουμε ότι το ρομπότ θα μιλάει ελληνικά, ώστε να είναι κατανοητό από χρήστες τόσο μικρής ηλικίας. Σε περίπτωση που απαντήσει σωστά, κερδίζει κάποια ευρώ (ανάλογα την αποστολή) και μπορεί να περάσει στο επόμενο επίπεδο. Σε περίπτωση όμως που απαντήσει λάθος, πρέπει να εκτελέσει ξανά τη δοκιμασία μέχρι να δώσει τη σωστή απάντηση. Στο τέλος των δοκιμασιών το παιδί θα έχει αποκτήσει τα 10 ευρώ, και το Nao θα του ζητάει να τα δείξει ένα προς ένα για να ολοκληρωθεί η διαδικασία.

4.2 Αποστολές

Επεξηγήσεις - Παραδοχές:

- ✓ Σε κάθε επιτυχία το Nao λέει «Μπράβο κέρδισες «χ» ευρώ» και αναβοσβήνουν τα Led στα μάτια του. Σε κάθε αποτυχία λέει «Λάθος απάντηση, χρειάζεσαι περισσότερα/λιγότερα χρήματα» και κοκκινίζουν τα μάτια του.
- ✓ Τα νομίσματα έχουν αξία 1,2 και 3 ευρώ και έχουν διάμετρο 8-9 εκατοστά
- ✓ Επίσης, στη φωνητική αναγνώριση το ρομπότ κατανοεί και αναγνωρίζει μόνο τα Αγγλικά.
- ✓ Τα μπαλάκια πρέπει να τα δίνει το παιδί ένα-ένα, όπως και τα Naomarks.
- ✓ Όσα πρέπει να δείξει το παιδί στο ρομπότ πρέπει να είναι σε σημείο που να τα βλέπει το ρομπότ. Αυτό το εξασφαλίζουμε με τη βοήθεια του προγράμματος Monitor που κατεβάσαμε μαζί με το Choregraphe από το site της Aldebaran.

1^η αποστολή

Το παιδί έχει στα χέρια του ήδη 2 ευρώ. Στην πρώτη δοκιμασία ερωτάται από το Nao πόσα χρήματα χρειάζεται για να αγοράσει τα μολύβια. Αυτό που πρέπει να κάνει είναι να υπολογίσει τη διαφορά $10-2=8$ και να κτυπήσει τον μεσαίο αισθητήρα που έχει στο κεφάλι του το Nao 8 φορές. Σε κάθε επανάληψη το ρομπότ καθοδηγεί το παιδί λέγοντάς του πότε να κτυπήσει ώστε να μπορεί να καταμετρήσει σωστά τα χτυπήματα. Αν απαντήσει σωστά το παιδί τότε επιβραβεύεται παίρνοντας 2 ευρώ, αν απαντήσει λάθος επαναλαμβάνει την αποστολή.

2^η αποστολή

Τώρα το παιδί έχει στα χέρια του έχει 4 ευρώ. Το Nao του ζητά να πει κοντά στο ακουστικό πόσα ευρώ του λείπουν. Ο υπολογισμός που πρέπει να κάνει είναι $10-4=6$ Αν το παιδί απαντήσει «six» τότε κερδίζει 3 ευρώ αλλιώς επαναλαμβάνει την αποστολή.

3^η αποστολή

Τώρα το παιδί έχει στα χέρια του έχει 7 ευρώ. Το Nao ζητά να του δείξει τόσα κόκκινα μπαλάκια όσα τα ευρώ που του λείπουν. Η πράξη που πρέπει να κάνει στο μυαλό το παιδί

είναι $10-7=3$. Αν του δείξει 3 μπαλάκια κερδίζει 2 ευρώ και συνεχίζει, αλλιώς επαναλαμβάνει την αποστολή. Εδώ πρέπει να τονίσουμε ότι τα μπαλάκια πρέπει να τα δείξει ένα προς ένα στην κάμερα, καθώς το ρομπότ όταν βλέπει περισσότερες από μία κόκκινες μπαλες συγκρατεί μόνο τη μεγαλύτερη.

4^η αποστολή

Η αποστολή αυτή περιλαμβάνει την πλατφόρμα Arduino καθώς και τρία κουτάκια που περιέχουν και γράφουν απέξω 1,2 και 3 ευρώ αντίστοιχα. Τώρα το παιδί έχει στα χέρια του έχει 9 ευρώ. Κάνοντας την αφαίρεση $10-9=1$ πρέπει να διαλέξει το σωστό κουτί που αναγράφει 1 ευρώ. Αν ανοίξει το κουτί με το 1 ευρώ, ένα πράσινο Led θα ανάψει στο Arduino και το ρομπότ θα δώσει στο παιδί το τελευταίο ευρώ. Αν όχι, επαναλαμβάνει την αποστολή.

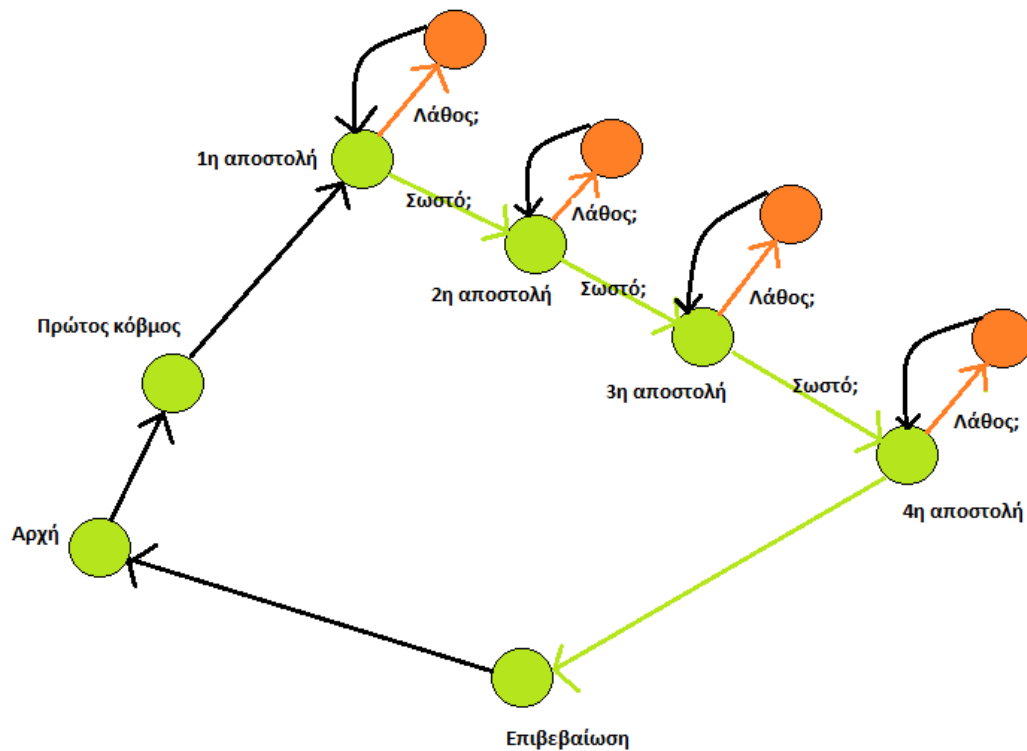
Στάδιο επιβεβαίωσης

Εδώ πλέον το παιδί έχει όλα τα νομίσματα. Το μόνο που μένει είναι να τα δείξει στο Nao ένα προς ένα για να πάρει τις μπογιές. Μόλις το Nao μετρήσει την αξία των νομισμάτων σηκώνει τα χέρια και του δίνει τις μπογιές.

Όλες οι αποστολές θα περιγραφούν παρακάτω αναλυτικά με τον κώδικα που χρησιμοποιήσαμε καθώς και όλες οι λεπτομέρεις που αφορούν τα νομίσματα, τα κόκκινα μπαλάκια και το Arduino.

4.3 Διάγραμμα καταστάσεων

Ακολουθεί το διάγραμμα καταστάσεων για τις αποστολές που αναφέραμε παραπάνω.



Εικ.5: Διάγραμμα καταστάσεων όλων των αποστολών

Ανάλυση των κόμβων

- ✓ Αρχή: είναι ο αρχικός κόμβος, δεν έχει κάποια λειτουργία
- ✓ Πρώτος κόμβος: ανακοινώνει πόσα ευρώ κοστίζουν οι μπογιές
- ✓ 1^η αποστολή: η αποστολή με τον αισθητήρα
- ✓ 2^η αποστολή: η αποστολή με την αναγνώριση φωνής
- ✓ 3^η αποστολή: η αποστολή με τα κόκκινα μπαλάκια
- ✓ 4^η αποστολή: η αποστολή με το Arduino
- ✓ Επιβεβαίωση: η επίδειξη όλων των νομισμάτων και η τελική επιβράβευση
- ✓ Οι κόκκινοι κόμβοι είναι βοηθητικοί και μεταβαίνουμε εκεί μόνο σε περίπτωση λάθους. Δηλαδή, δεν εκτελείται κάποιο κομμάτι κώδικα σε αυτούς, απλώς είναι ένας τρόπος για να επιστρέψουμε και πάλι στον κόμβο που περιέχει την αποστολή στην οποία αποτύχαμε.

5

Υλοποίηση σεναρίου

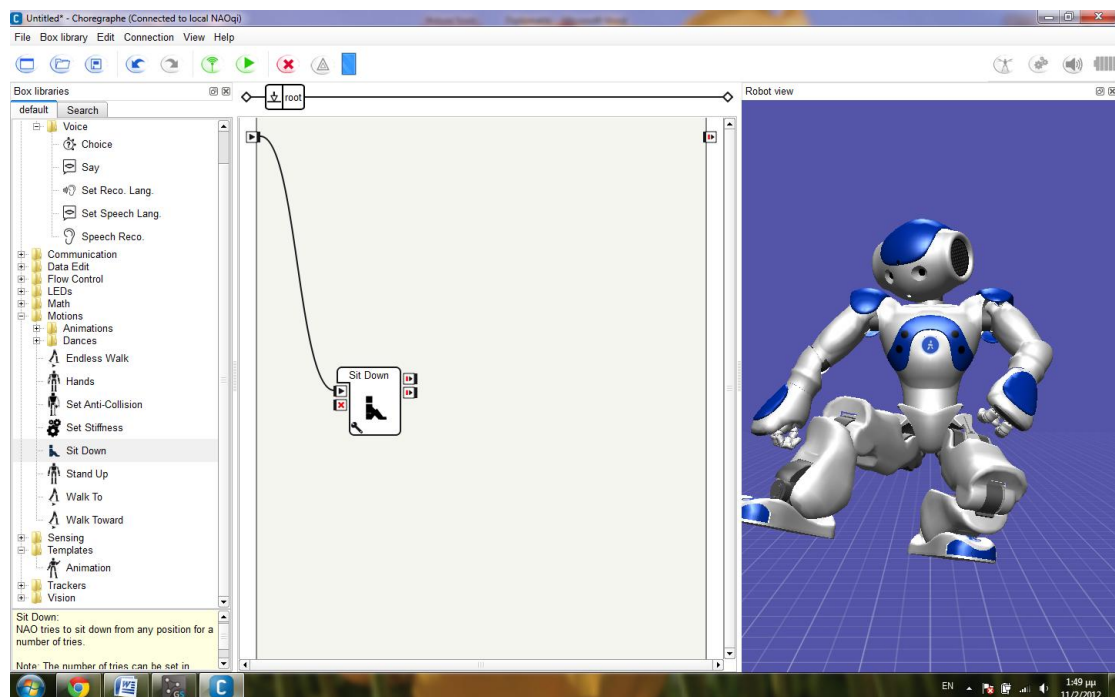
Σε αυτό το κεφάλαιο θα παρουσιάσουμε τη διαδικασία υλοποίησης του σεναρίου με τη βοήθεια του Gostai Studio. Είναι το πιο σημαντικό κομμάτι της διπλωματικής εργασίας καθώς θα παραθέσουμε το διάγραμμα καταστάσεων όπως το φτιάξαμε εκεί, τον κώδικα που συνοδεύει τον κάθε κόμβο καθώς και τον τρόπο που συνδεθήκαμε με το ρομπότ για να εκτελέσουμε το σενάριο.

5.1 Συνδεση του Gostai με το Nao

Αφού έχουμε ορίσει σαφώς το σενάριο και το τι πρέπει να κάνει το ρομπότ μας σε κάθε αποστολή, είναι ώρα να συνδέσουμε το Nao με την κοσνόλα του Gostai όπου γράφουμε τον κώδικά μας και φτιάχνουμε το διάγραμμα καταστάσεων. Βασική προϋπόθεση είναι να «κατεβάσουμε» από την ιστοσελίδα του Gostai το Gostai Suite και να το εγκαταστήσουμε στον υπολογιστή μας. Επίσης, πρέπει να έχουμε εγκαταστήσει την Urbi όπως προαναφέραμε και να συνδεθούμε ενσύρματα (κατά προτίμηση) με τον εξυπηρετητή (server) του δικτύου μας. Στη συνέχεια, θέτουμε σε λειτουργία το Nao. Μόλις ολοκληρωθεί η εκκίνηση (boot), με ένα καλώδιο Ethernet συνδέουμε το ρομπότ (έχει ειδική υποδοχή στο πίσω μέρος του κεφαλιού του) με τον εξυπηρετητή επίσης. Έτσι, θα «επικοινωνούν» μέσω του τοπικού δικτύου. Τέλος, στο Gostai Studio δηλώνουμε την IP του Nao και την θύρα μέσω της οποίας θα επικοινωνούμε (π.χ. 147.102.11.192:54000) και συνδεόμαστε με το ρομπότ. Τώρα, είμαστε έτοιμοι να στείλουμε τον κώδικά μας στο Nao.

5.2 Σύνδεση με το Choregraphe

Για να συνδεθούμε με το ρομπότ μέσω του Choregraphe ακολουθούμε παρόμοια διαδικασία. Αφού έχουμε συνδέσει και το ρομπότ και τον υπολογιστή με τον εξυπηρετητή, επιλέγουμε από τη λίστα συνδέσεων του Choregraphe το Nao που έχουμε συνδέσει (όχι κάποιο εικονικό Nao). Όταν συνδεθεί, θα εμφανιστεί στα δεξιά η μορφή του Nao και μπορούμε πλέον να ελέγξουμε το ρομπότ σε ότι αφορά τις κινήσεις του, τη στάση του, καθώς επίσης και να το διδάξουμε να αναγνωρίζει φωνές, αντικείμενα και πρόσωπα. Το Choregraphe μας φάνηκε ιδιαίτερα χρήσιμο σε σημεία του κώδικα που θέλαμε να ελέγξουμε αν κινούμε τα σωστά μέλη του σώματος του ρομπότ και να ορίσουμε σε τι θέση θέλουμε να είναι. Ακολουθεί ένα στιγμιότυπο αλλαγής θέσης του Nao μέσω του Choregraphe.



Εικ.6: Το Choregraphe όταν έχει συνδεθεί με το Nao

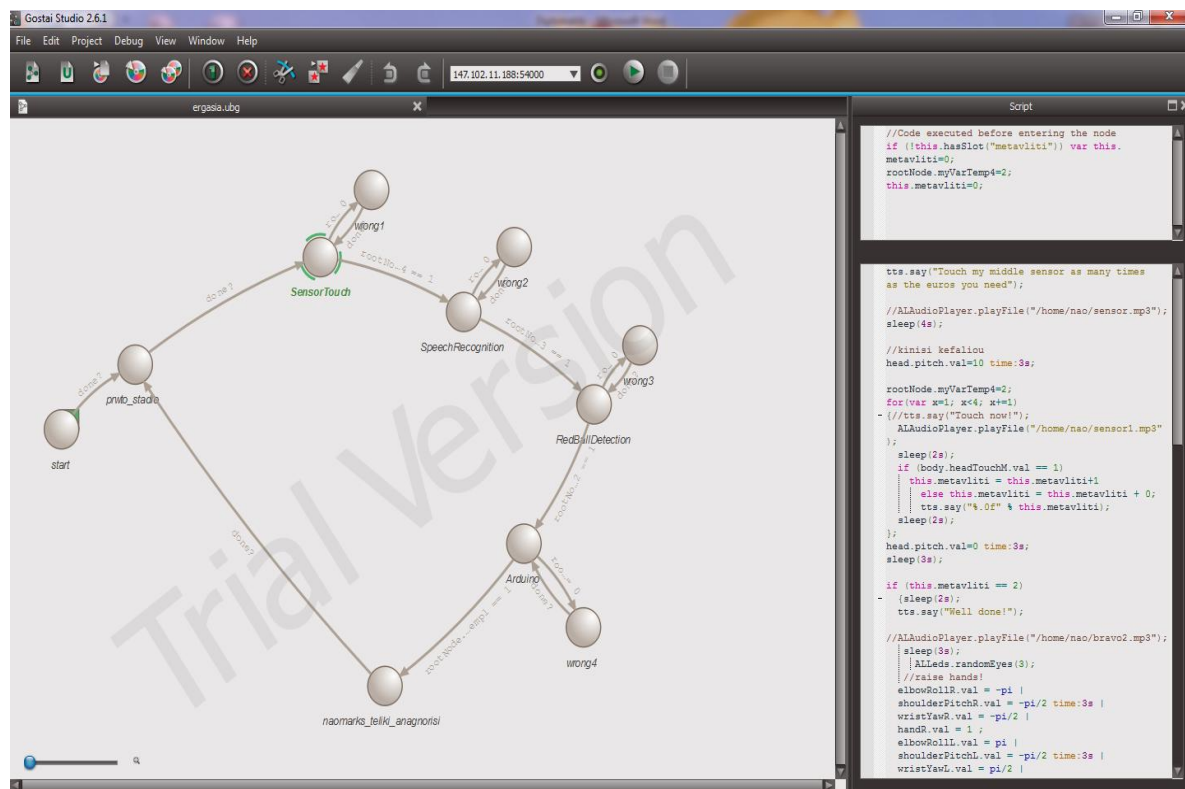
5.3 Υλοποίηση αποστολών

Είναι ώρα λοιπόν να γίνουμε ακριβείς στις περιγραφές μας. Δηλαδή, να παραθέσουμε το τελικό διάγραμμα καταστάσεων όπως αυτό απεικονίζεται στο Gostai Studio και τον κώδικα που γράψαμε σε κάθε κόμβο. Παράλληλα, θα εξηγήουμε αναλυτικά τι κάνουμε σε κάθε βήμα

και μέσα στον κώδικα θα προσθέσουμε επεξηγηματικά σχόλια. Επίσης, θα εξηγήσουμε πώς λειτουργεί η αναγνώριση των Naomarks, των κόκκινων σφαιρών, του Arduino, και το πώς αναφερόμαστε στα διάφορα μέλη του ρομπότ. Γενικά, θα ξεκαθαρίσουμε κάθε δυσνόητο σημείο της διαδικασίας υλοποίησης.

5.3.1 Δημιουργία Urbigraph

Το διάγραμμα καταστάσεων στο Gostai ονομάζεται Urbigraph και έχει την παρακάτω μορφή:



Εικ. 7: Το διάγραμμα καταστάσεων όπως απεικονίζεται στο Gostai Studio

Παρατηρούμε ότι στο πάνω κεντρικό μέρος της μπάρας έχουμε γράψει την IP του Nao καθώς και τη θύρα επικοινωνίας. Στο αριστερό μεγάλο παράθυρο έχουμε το διάγραμμα καταστάσεων το οποίο έχει την ίδια μορφή με αυτό που σχεδιάσαμε σε θεωρητικό επίπεδο πριν την υλοποίηση του σεναρίου. Στη δεξιά στήλη γράφουμε τον κώδικα που θα εκτελεστεί στον κάθε κόμβο. Το πάνω μέρος είναι ο κώδικας που εκτελείται πριν μπούμε στον κόμβο και συνήθως εκεί δηλώνουμε τοπικές και καθολικές μεταβλητές. Στο υπόλοιπο κομμάτι ακολουθεί ο υπόλοιπος κώδικας που θα εκτελεστεί μόλις μπει στον κόμβο.

5.3.2 Αναλυτική παρουσίαση υλοποίησης

Κόμβος «Start»

Είναι ο πρώτος κόμβος του διαγράμματος καταστάσεων. Σε αυτόν δεν εκτελείται κάποια διεργασία και δεν γράφουμε κώδικα καθώς είναι ο πρώτος κόμβος που σηματοδοτεί την εκκίνηση του σεναρίου. Αμέσως μεταβαίνουμε στον επόμενο.

Κόμβος «Πρώτο στάδιο»

Είναι ουσιαστικά ο πρώτος κόμβος που ξεκινά να «τρέχει» το σενάριο. Εδώ δηλώνουμε τις καθολικές μεταβλητές στο σημείο πριν μπούμε στον κόμβο και στο κυρίως μέρος το ρομπότ ανακοινώνει στο παιδί πόσο στοιχίζουν οι μπογιές που πρέπει να αγοράσει. Η ανακοίνωση γίνεται με χρήση της σύνθεσης φωνής του Ναο όπως θα εξηγήσουμε παρακάτω. Ακολουθεί ο κώδικας.

Κώδικας κόμβου

```
//Code executed before entering the node
//dilosi global metavliton
if (!rootNode.hasLocalSlot("myVar1")) var rootNode.myVar1=0;
if (!rootNode.hasLocalSlot("myVar2")) var rootNode.myVar2=0;
if (!rootNode.hasLocalSlot("myVar3")) var rootNode.myVar3=0;
if (!rootNode.hasLocalSlot("myPrice")) var rootNode.myPrice=0;
if (!rootNode.hasLocalSlot("myName")) var rootNode.myName=0;
if (!rootNode.hasSlot("myVarTemp4")) var rootNode.myVarTemp4=2;
if (!rootNode.hasSlot("myVarTemp3")) var rootNode.myVarTemp3=2;
if (!rootNode.hasSlot("myVarTemp2")) var rootNode.myVarTemp2=2;
if (!rootNode.hasSlot("myVarTemp1")) var rootNode.myVarTemp1=2;

//Main code
//Anakoinwsi tis aksias
```



```
ALLeds.fadeRGB("FaceLeds", 0x0000FF, 3); //eye colour=blue  
tts.say("ta molivia kostyzun, deka euro ");  
sleep(3s);
```

Πώς παράγουμε ελληνικές λέξεις;

Εδώ θα ήταν χρήσιμο να εξηγήσουμε πώς παράγουμε προτάσεις στα ελληνικά.

Όπως γνωρίζουμε από το API του Nao, οι μόνες γλώσσες που μπορεί να «μιλήσει» το ρομπότ μας είναι αγγλικά και γαλλικά. Αυτό που κάναμε εμείς είναι να χρησιμοποιήσουμε το λατινικό αλφάβητο και να «κατασκευάσουμε» τις ελληνικές λέξεις που θέλουμε, όσο είναι δυνατόν.

Η εντολή της Urbi “tts.say(“ ”)” χρησιμοποιείται για να μετατρέψει το κείμενο που γράφουμε μέσα στα εισαγωγικά σε φωνή (TextToSpeech). Εμείς λοιπόν που θέλουμε να παράγουμε ελληνικές συλλαβές, λέξεις και προτάσεις, χρησιμοποιούμε αυτήν την εντολή σε συνδυασμό με το λατινικό αλφάβητο για να συνθέσουμε συλλαβές και στη συνέχεια λέξεις.

Παράδειγμα:

Αν θέλουμε να πει «Γεια σου είμαι ο Nao» πρέπει να γράψουμε

```
tts.say("Gia sou, ime o Nao");
```

Αυτό βέβαια δεν είχε άρτιο αποτέλεσμα καθώς κάποιες λέξεις δεν είναι εύηχες, αλλά ήταν ο μόνος τρόπος να κάνουμε το ρομπότ να μιλήσει την ελληνική γλώσσα. Η χρήση ηχογραφημένων αρχείων δυστυχώς δεν ήταν λειτουργική.

Κόμβος «Sensor Touch»

Μόλις φτάσαμε στην πρώτη μας αποστολή. Εδώ θα ζητήσει το ρομπότ από το παιδί να αγγίξει τον μεσαίο αισθητήρα του κεφαλιού του τόσες φορές όσα τα ευρώ που του λείπουν για να αγοράσει τιςμπογιές. Αν αγγίξει 8 φορές η απάντηση επιβραβεύεται με 2 ευρώ και μεταβαίνει στον επόμενο κόμβο μέσω του βέλους επιτυχίας. Αλλιώς μεταβαίνει στον κόμβο «wrong1» (βλ.παρακάτω) και επανέρχεται στον ίδιο κόμβο για να εκτελέσει ξανά την αποστολή. Σε κάθε επιτυχία το Nao σηκώνει τα χέρια ψηλά, επιβραβεύει τον παίκτη και αναβοσβήνουν τα Led στα μάτια του. Σε κάθε αποτυχία κοκκινίζουν τα μάτια του και λέει στο παιδί να προπαθήσει ξανά. Ακολουθεί ο κώδικας

```
//Code executed before entering the node
//local variable counting the "hits" of the sensor
if (!this.hasSlot("metavliti")) var this.metavliti=0;
//transfer variable (μεταβλητη μεταβασεων)
rootNode.myVarTemp4=2;
this.metavliti=0;

//Main code
tts.say("Aggyksei toe meseio eisthitira toseis for s, osa tah euro
pou sou lee poon");
//head move
head.pitch.val=10 time:3s;
rootNode.myVarTemp4=2;
for(var x=1; x<10; x+=1)
{tts.say("aggyksei!");
  if (body.headTouchM.val == 1)
    this.metavliti = this.metavliti+1
    else this.metavliti = this.metavliti + 0;
  tts.say("%.0f" % this.metavliti); };
head.pitch.val=0 time:3s;
//right answer
if (this.metavliti == 8)
  {sleep(2s);
  tts.say("Bravo, keerdysess dio euro!");
  //Leds
  ALLeds.randomEyes(3);
  //raise hands!
  elbowRollR.val = -pi |
  shoulderPitchR.val = -pi/2 time:3s |
```

```

wristYawR.val = -pi/2 |
handR.val = 1 ;
elbowRollL.val = pi |
shoulderPitchL.val = -pi/2 time:3s |
wristYawL.val = pi/2 |
handL.val = 1 ;
sleep(2s);
//and back
elbowRollR.val = 0 |
shoulderPitchR.val = pi/2 time:3s |
wristYawR.val = pi/2 |
handR.val = 0 ;
elbowRollL.val = 0 |
shoulderPitchL.val = pi/2 time:3s |
wristYawL.val = -pi/2 |
handL.val = 0 ;
ALLeds.fadeRGB("FaceLeds", 0x0000FF, 3);
sleep(5s);
rootNode.myVarTemp4 = 1; };
//if hit more than 8 times. Red Leds lighting
if (this.metavliti > 8)
{ tts.say("laath oss, sou lee poon, lygo terra, euro.
Prospathyssei, xanai. ");
sleep(2s);
ALLeds.fadeRGB("FaceLeds", 0xCD00CD, 3); //eye colour
sleep(3s);
rootNode.myVarTemp4 = 0; };
if (this.metavliti < 8)
{ tts.say("laath oss, sou lee poon, perisso terra, euro.
Prospathyssei, xanai. ");
sleep(2s);

```

```
ALLeds.fadeRGB("FaceLeds", 0xFFFFFF, 3);  
sleep(3s);  
rootNode.myVarTemp4 = 0; };
```

Κώδικας μεταβάσεων

```
rootNode.myVarTemp4 == 1 //metavasi stin epomeni dokimasia  
rootNode.myVarTemp4 == 0 //epanalipsi dokimasia
```



Εικ. 8 : Το Nao όταν μας επιβραβεύει

Πώς αναβοσβήνουν τα LED στα μάτια του Nao;

Ανοίγοντας το API Guide του Nao μπορούμε να βρούμε 36 Modules. Δηλαδή λειτουργίες που έχει εγκατεστημένες στη μνήμη του το ρομπότ μας. Μερικές από αυτές είναι η σύνθεση φωνής, η αναγνώριση φωνής, η αναγνώριση των Naomarks, η κίνηση του ρομπότ, η επικοινωνία με υπερύθρες κλπ. Εκεί βρίσκεται και το Module ALLeds. Μόλις το καλέσουμε μαζί με την κατάλληλη συνάρτηση δίνει στα Led το χρώμα και τη φωτεινότητα που θέλουμε. Για παράδειγμα, αν θέλουμε να κάνουν blink τα μάτια του ρομπότ για 3 δευτερόλεπτα (κάτι που γίνεται σε κάθε νίκη του παιδιού) χρησιμοποιούμε την εξής εντολή:

ALLeds.randomEyes(3)

Δηλαδή, ενεργοποιούμε το module ALLeds και χρησιμοποιούμε τη συνάρτηση randomEyes με όρισμα 3 διότι θέλουμε να ανάψουν τα μάτια για 3 δευτερόλεπτα σε τυχαίους χρωματισμούς.

Επιπλέον, αν θέλουμε να δώσουμε έναν συγκεκριμένο χρωματισμό στα led του ρομπότ για συγκεκριμένο χρόνο χρησιμοποιούμε την εντολή

ALLeds.fadeRGB("AllLeds", 0xFF0000, 3)

Η συνάρτηση fadeRGB δίνει το χρώμα που θέλουμε στα Led που θέλουμε εμείς για όσο χρόνο επιθυμούμε εμείς.

Το όρισμα «AllLeds» αναφέρεται σε όλα τα Led του ρομπότ.

Το όρισμα «0xFF0000» είναι ο δεκαεξαδικός κωδικός του κόκκινου χρώματος. Μπορούμε στη θέση αυτή να βάλουμε όποιο χρώμα επιθυμούμε.

Τέλος, το όρισμα «3» καθορίζει τη διάρκεια που θα μείνουν αναμμένα σε δευτερόλεπτα.

Ένας δεύτερος τρόπος είναι να ορίσουμε για κάθε μάτι το χρώμα που θέλουμε και μετά να το επαναφέρουμε στο αρχικό του χρώμα.

Παράδειγμα: Αν θέλω το δεξί μάτι να έχει κόκκινο χρώμα πρέπει να δώσω την εξής εντολή:

```
robot.body.head.eye[right].r=0 & robot.body.head.eye[right].g=255 &  
robot.body.head.eye[right].b =0;
```

Όμως, αυτή η μέθοδος δεν είναι η πιο λειτουργική, καθώς μετά από μερικές αλλαγές τα leds δεν ακολουθούν τις εντολές μας. Άρα, προτιμήσαμε τον πρώτο τρόπο. Συγκεκριμένα:

Αν το παιδί δώσει σωστή απάντηση, τα μάτια του Nao αναβοσβήνουν σε τυχαίους χρωματισμούς με την εντολή που είδαμε παραπάνω. Αν απαντήσει με αριθμό μεγαλύτερο από τον αναμενόμενο, τα μάτια αποκτούν χρώμα μωβ, με την εκτέλεση της εντολής

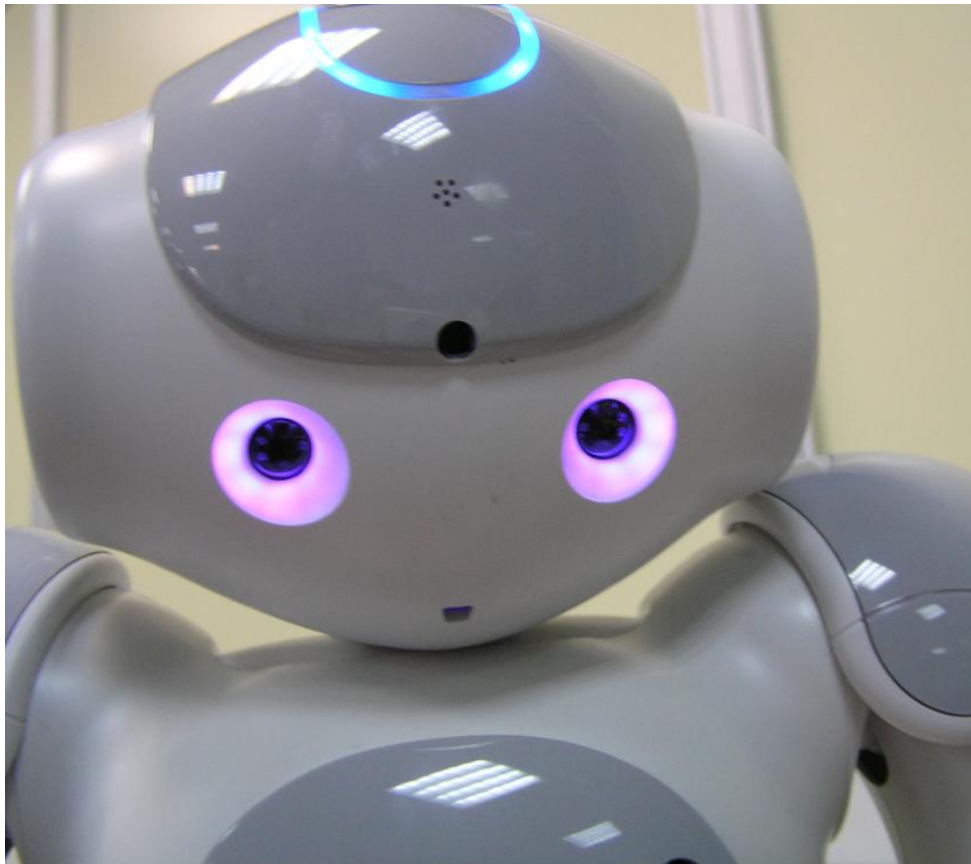
```
ALLeds.fadeRGB("FaceLeds", 0xCD00CD, 3);
```

Και τα επαναφέρουμε μετά στο μπλε.

Αν η απάντηση είναι μικρότερη από την αναμενόμενη, τότε τα μάτια αποκτούν χρώμα λευκό με την εντολή

```
ALLeds.fadeRGB("FaceLeds", 0xFFFFFFFF, 3);
```

Και τα επαναφέρουμε και πάλι στο αρχικό μπλε τους χρώμα



Εικ. 9 : Τα μάτια του Nao όταν τα Led είναι μωβ

Πώς μεταβαίνω στον επιθυμητό κόμβο;

Παρατηρήστε ότι στο τέλος του κυρίως κώδικα έχουμε παραθέσει και τον κώδικα μεταβάσεων. Αυτές οι εντολές γράφονται πάνω στα βέλη των μεταβάσεων. Η μεταβλητή `rootNode.myVarTemp4` (και η αντίστοιχη σε κάθε κόμβο) μηδενίζεται αν δώσουμε λάθος απάντηση (οπότε και μεταβαίνουμε στον κόμβο `wrong`) και γίνεται μονάδα αν απαντήσουμε σωστά (οπότε και μεταβαίνουμε στην επόμενη αποστολή). Παράλληλα, στον κώδικα που εκτελείται πριν εισέλθουμε στον κόμβο, φροντίζουμε να της δώσουμε μία τρίτη τιμή (π.χ. 2) ώστε να αποφύγουμε άσκοπες και λανθασμένες μεταβάσεις. Αξίζει να δούμε ένα παράδειγμα από τον τρέχοντα κόμβο.

Όταν λοιπόν η απάντηση είναι σωστή δίνουμε τιμή 1 στη μεταβλητή μετάβασης

```
if (this.metavliti == 8)
{
.....
rootNode.myVarTemp4 = 1;  };
```

και μεταβαίνουμε στην επόμενη αποστολή. Αντίθετα, αν κάνουμε λάθος η μεταβλητή μετάβασης παίρνει τιμή 0 και μεταβαίνουμε στον κόμβο «wrong»

```
if (this.metavliti < 8)
{
.....
rootNode.myVarTemp4 = 0;  };
```

Κόμβος «Speech Recognition»

Και φτάνουμε στη δεύτερη αποστολή. Εδώ θα ζητήσει το ρομπότ από το παιδί να του πει πόσα ευρώ του λείπουν για να αγοράσει τις μπογιές. Αν απαντήσει σωστά, κερδίζει 3 ευρώ και μεταβαίνει στον επόμενο κόμβο μέσω του βέλους επιτυχίας. Αλλιώς μεταβαίνει στον κόμβο «wrong2» και επανέρχεται στον ίδιο κόμβο για να εκτελέσει ξανά την αποστολή. Σε κάθε επιτυχία το Nao σηκώνει τα χέρια ψηλά, επιβραβεύει τον παίκτη και αναβοσβήνουν τα Led στα μάτια του. Σε κάθε αποτυχία κοκκινίζουν τα μάτια του και λέει στο παιδί να προπαθήσει ξανά. Ακολουθεί ο κώδικας.

```
//Code executed before entering the node
if (!this.hasSlot("data1")) var this.data1=0;
rootNode.myVarTemp3=2;
//create proxy for SpeechRecognition
if (!this.hasSlot("speechProxy"))
var this.speechProxy=ALProxy("SpeechRecognition", [], ["setLanguage"
, "subscribe" , "unsubscribe"]);

//Main code
tts.say("Pess moo, posah euro sou lee poon?");
rootNode.myVarTemp3=2;
//voice recognition. Set the Language, the word to be recognized
this.speechProxy.setLanguage("English");
var wordlist=["six"];
speechProxy.setWordListAsVocabulary(wordlist);
//enable recognition
speechProxy.subscribe("MyModule");
tts.say("Pess mu, torah!");
//save the word Nao heard to a variable
this.data1 = robot.ALMemory.getData_string("WordRecognized");
//repeat until it hears a word
var j=1;
while ((this.data1[0] == "") && (j<4))
{ tts.say("Pess mu, torah!");
this.data1 = robot.ALMemory.getData_string("Word");
j+=1; };
sleep(3s);
//disable recognition
speechProxy.unsubscribe("MyModule");
```



```

if (this.data1[0] == "six")
    {tts.say("Bravo, keerdysseess tree ah euro!");
    ALLeds.randomEyes(3);
//raise hands!
    elbowRollR.val = -pi | shoulderPitchR.val = -pi/2 time:3s |
    wristYawR.val = -pi/2 | handR.val = 1 ;
    elbowRollL.val = pi | shoulderPitchL.val = -pi/2 time:3s |
    wristYawL.val = pi/2 | handL.val = 1 ;
    sleep(2s);
//return
    elbowRollR.val = 0 | shoulderPitchR.val = pi/2 time:3s |
    wristYawR.val = pi/2 | handR.val = 0 ;
    elbowRollL.val = 0 | shoulderPitchL.val = pi/2 time:3s |
    wristYawL.val = -pi/2 | handL.val = 0 ;
    ALLeds.fadeRGB("FaceLeds", 0x0000FF, 3);
    sleep(2s);
//set the transfer variable
rootNode.myVarTemp3 = 1; };
else {
tts.say("Laath oss, apantysee. Prospathyssei, xanai.");
ALLeds.fadeRGB("FaceLeds", 0xCD00CD, 3);
sleep(3s);
//zero the transfer variable
rootNode.myVarTemp3 = 0; };

```

Κώδικας μεταβάσεων

```

rootNode.myVarTemp3 == 1
rootNode.myVarTemp3 == 0

```

Πώς αναγνωρίζει τη φωνή του παιδιού;

Εδώ θα χρειαστούμε ένα Module από το API Guide. Είναι το ALSpeechRecognition. Η διαδικασία είναι η ίδια με του ALLeds. Δηλαδή, δημιουργούμε στην αρχή του κώδικα του κόμβου ένα ALProxy για το συγκεκριμένο Module:

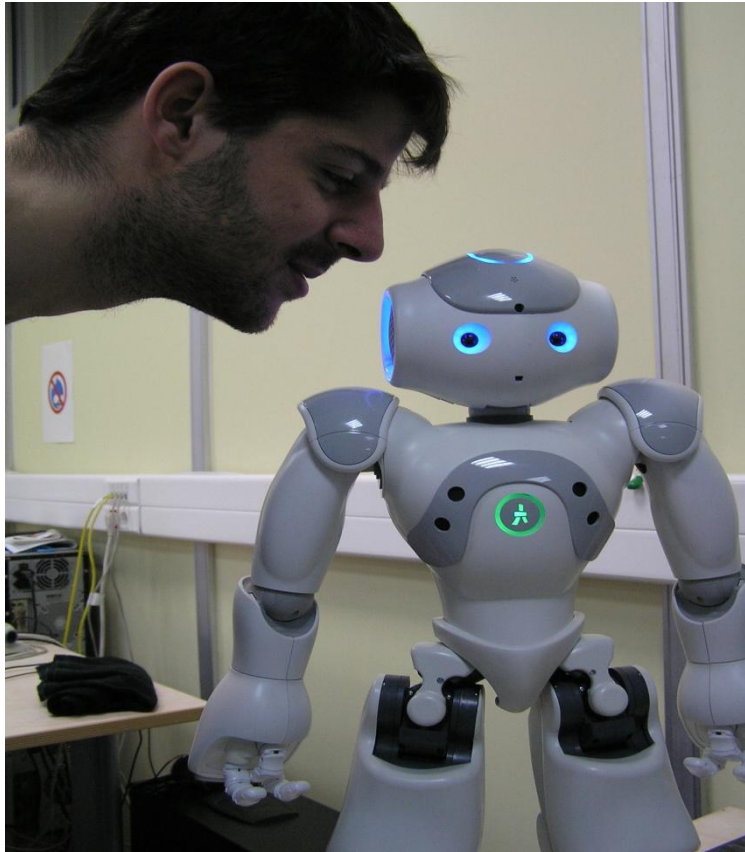
```
this.speechProxy=ALProxy("SpeechRecognition", [], ["setLanguage" ,  
"subscribe" , "unsubscribe"]);
```

Στη συνέχεια ορίζουμε ως γλώσσα τα Αγγλικά και θέτουμε ως αναμενόμενη λέξη το “six”:

```
this.speechProxy.setLanguage("English");  
var wordlist=["six"];  
speechProxy.setWordListAsVocabulary(wordlist);
```

Όταν έρθει η στιγμή να «ακούσει» την απάντηση του παιδιού, αποθηκεύει στο ALMemory στη μεταβλητή getData_string την απάντηση που άκουσε, την διασταυρώνει με αυτήν που περιμένει και αποφασίζει ανάλογα:

```
this.data1 = robot.ALMemory.getData_string("WordRecognized");  
if (this.data1[0] == "six")  
    {tts.say("Well Done!");
```



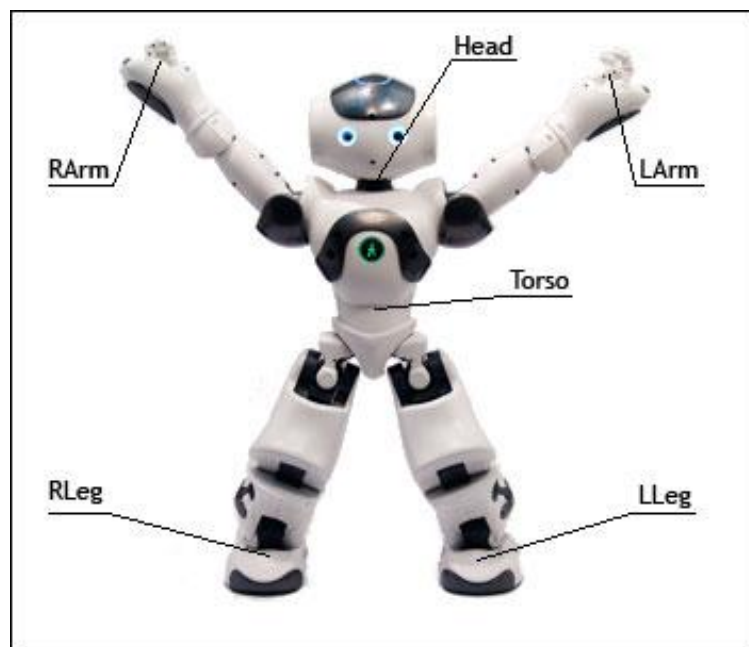
Εικ. 10 :Αναγνώριση ομιλίας

✚ Πώς κινώ τα μελη του ρομπότ;

Αν κοιτάξουμε προσεκτικά τον κώδικα, θα δούμε ότι σε κάποια σημεία κουνάμε κάποια μέλη του ρομπότ όπως το κεφάλι και τα χέρια ανάλογα με την εξέλιξη της αποστολής. Για να το κατορθώσουμε αυτό χρησιμοποιούμε τις εντολές αναφοράς στα μέλη του ρομπότ. Τα μέλη του ρομπότ που χρειαστήκαμε στις αποστολές που υλοποιήσαμε ιεραρχούνται όπως φαίνεται στον παρακάτω πίνακα και όπως απεικονίζονται στη εικόνα που ακολουθεί:

Head	LArm	LLeg	RLeg	RArm
HeadYaw	LShoulderPitch	LHipYawPitch	RHipYawPitch	RShoulderPitch
HeadPitch	LShoulderRoll	LHipRoll	RHipRoll	RShoulderRoll
HeadCamera	LElbowYaw	LHipPitch	RHipPitch	RElbowYaw
	LElbowRoll	LKneePitch	RKneePitch	RElbowRoll
	LWristYaw	LAnklePitch	RAnklePitch	RWristYaw
	LHand	LAnkleRoll	RAnkleRoll	RHand

Εικ.11: Πίνακας αναφοράς στα μέλη του Nao



Εικ.12: Τα μέλη του ρομπότ μας

Συνδυάζοντας λοιπόν αυτές τις εντολές μπορούμε να κινήσουμε οποιοδήποτε μέλος του σώματος και του κεφαλιού του Nao robot. Παράδειγμα:

Έστω ότι θέλουμε να σηκώσουμε τα χέρια του ρομπότ ψηλά σε περίπτωση νίκης και να ανοίξουμε τις παλάμες. Θα χρησιμοποιήσουμε τις εντολές:

```
RElbowRoll.val = -pi | //Δεξιά αγκώνας σε γωνία -π
RShoulderPitch.val = -pi/2 | //Δεξιά ώμος σε γωνία -π/2
RWristYaw.val = -pi/2 | //Δεξιά καρπός σε γωνία -π/2
RHand.val = 1 ; //Δεξιά παλάμη
LElbowRoll.val = pi | //Αντίστοιχα για το αριστερό
LShoulderPitch.val = -pi/2 |
LWristYaw.val = pi/2 |
LHand.val = 1 ;
```

Ο κατάλληλος συνδυασμός εντολών λοιπόν μας επιτρέπει να δώσουμε στο ρομπότ μας όποια στάση επιθυμούμε και να κάνει όποια κίνηση θέλουμε, αφού, ας μην ξεχνούμε ότι έχει 25 βαθμούς ελευθερίας.

Κόμβος «Red Ball Detection»

Έχουμε επιτυχώς φτάσει ως την τρίτη αποστολή. Το παιδί έχει πλέον στα χέρια του 7 ευρώ, άρα του λείπουν 3 ευρώ για να ολοκληρώσει το παιχνίδι. Στην αποστολή αυτή καλείται να κάνει και πάλι την αφαίρεση $10-7=3$ για να υπολογίσει πόσα ευρώ του λείπουν αυτή τη φορά. Για να δώσει στο ρομπότ τη σωστή απάντηση πρέπει να του δείξει τρία κόκκινα μπαλάκια. Αυτό που πρέπει να επισημάνουμε εδώ είναι ότι τα μπαλάκια πρέπει να τα δείξει ένα-ένα, διότι το Nao αναγνωρίζει μόνο ένα μπαλάκι κάθε φορά. Ακόμη και αν του δείξει περισσότερα, αυτό συγκρατεί το μεγαλύτερο ή αυτό που είναι πιο κοντά του. Αν απαντήσει σωστά, κερδίζει 2 ευρώ. Αν απαντήσει λάθος, επαναλαμβάνει την αποστολή. Ακολουθεί ο κώδικας.

```
//Code executed before entering the node

rootNode.myVarTemp2=2;
if (!this.hasSlot("mydata")) var this.mydata=1;
if (!this.hasSlot("ballnumber")) var this.ballnumber=0;
if (!this.hasSlot("timestamp")) var this.timestamp=0;
if (!this.hasSlot("ballProxy")) var
this.ballProxy=ALProxy("RedBallDetection", [] ,
["subscribe","unsubscribe"]);

//Main code

tts.say("dikseh mou toseis bah less, osa tah euro pou sou lee
poon");
rootNode.myVarTemp2=2;
this.ballnumber=0;
//head move
head.pitch.val=10 time:2s;
var period = 500;
//enable detection
this.ballProxy.subscribe("TestBall" , period, 0.0 );
//start counting the ball one by one
var j=0;
while (j<4)
{ this.mydata = robot.ALMemory.getData("redBallDetected");
this.timestamp = this.mydata[0][0];
tts.say("dikseh mou mia bah la,");
//get data from detection
this.mydata = robot.ALMemory.getData("redBallDetected");
this.mydata = robot.ALMemory.getData("redBallDetected");
tts.say("OK");
//check if the robot saw a ball
if (this.mydata[0][0] != this.timestamp) {
```

```

        this.ballnumber = this.ballnumber + 1;
        this.timestamp = this.mydata[0][0]; };
if (this.mydata[0][0] == this.timestamp)
    this.ballnumber = this.ballnumber + 0;
    j+=1;  };
head.pitch.val=0 time:3s;
if (this.ballnumber == 3) {
tts.say("Bravo, keerdysse dio euro!");
ALLeds.randomEyes(3);
//raise hands!
    elbowRollR.val = -pi |  shoulderPitchR.val = -pi/2 time:3s |
    wristYawR.val = -pi/2 | handR.val = 1 ;
    elbowRollL.val = pi |  shoulderPitchL.val = -pi/2 time:3s |
    wristYawL.val = pi/2 | handL.val = 1 ;
    sleep(2s);
//return
    elbowRollR.val = 0 |  shoulderPitchR.val = pi/2 time:3s |
    wristYawR.val = pi/2 | handR.val = 0 ;
    elbowRollL.val = 0 |  shoulderPitchL.val = pi/2 time:3s |
    wristYawL.val = -pi/2 | handL.val = 0 ;
    ALLeds.fadeRGB("FaceLeds", 0x0000FF, 3);
    rootNode.myVarTemp2 = 1;  };
if (this.ballnumber < 3) {
tts.say("laath oss, sou lee poon, perisso terra, euro.
Prospathyssei, xanai.");
//Eye LEDS
ALLeds.fadeRGB("FaceLeds", 0xFFFFFFFF, 3);
sleep(3s);
rootNode.myVarTemp2=0;  };
if (this.ballnumber > 3) {

```

```
tts.say("laath oss, sou lee poon, lygo terra, euro. Prospathyssei,  
xanai.");  
//eye LEDS  
ALLeds.fadeRGB("FaceLeds", 0xCD00CD, 3);  
sleep(3s);  
rootNode.myVarTemp2=0; };  
//disable detection  
ballProxy.unsubscribe("TestBall", period, 0.0);
```

Κώδικας μεταβάσεων

```
rootNode.myVarTemp2 == 1  
rootNode.myVarTemp2 == 0
```

Πώς αναγνωρίζει τις κόκκινες μπάλες;

Για να μπορέσουμε να μετρήσουμε λοιπόν τις μπάλες που δείχνει το παιδί στην κάμερα του ρομπότ, πρέπει να μπορεί το Nao να αναγνωρίζει τις κόκκινες μπάλες. Για αυτήν ακριβώς τη λειτουργία θα χρησιμοποιήσουμε και πάλι ένα Module του Nao. Αυτό το Module λέγεται ALRedBallDetection και λειτουργεί ως εξής:

Κάθε φορά που θα ανιχνεύσει μία κόκκινη μπάλα, γράφει τα «στοιχεία» της στο ALMemory στο microEvent που λέγεται redBallDetected. Λέγοντας «στοιχεία» εννοούμε τη χρονική στιγμή που την είδε, τα γνωρίσματα της μπάλας και τις πληροφορίες της κάμερας. Δηλαδή, η μεταβλητή redBallDetected έχει τη μορφή:

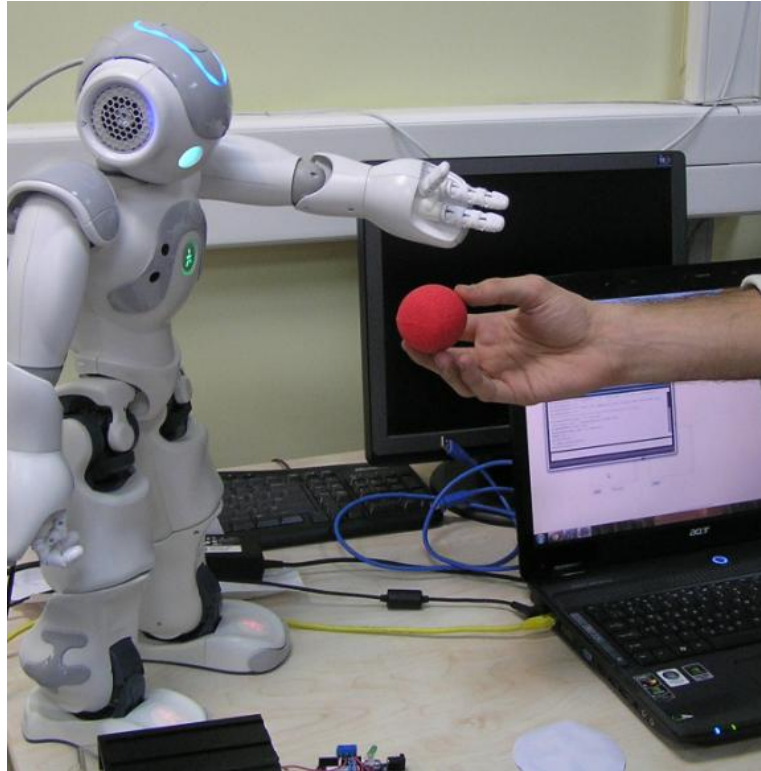
[[time_info], [ball_info], [camera_info]]

Το Tag time_info έχει τα δεδομένα [timestamp_seconds, timestamp_microseconds], δηλαδή τα λεπτά και τα δευτερόλεπτα που είδα την κόκκινη μπάλα.

Το Tag ball_info έχει τα δεδομένα [u, v, centerX, centerY]. Τα u και v είναι το χρώμα της μπάλας στο χρωματικό φάσμα. Τα centerX and centerY είναι οι γωνιακές συντεταγμένες

Το Tag camera_info έχει τα δεδομένα [x, y, z, wx, wy, wz].

Εμείς θα ασχοληθούμε μόνο με το Tag `time_info` και συγκεκριμένα με τα δευτερόλεπτα. Η θέση του πίνακα στην οποία αποθηκεύονται τα δευτερόλεπτα είναι η `data[0][0]` διότι είναι το πρώτο στοιχείο του πρώτου tag. Όταν λοιπόν η τιμή αυτής της θέσης αλλάξει σημαίνει πως έχει δει κόκκινη μπάλα και αυξάνομε τον μετρητή κατά 1.



Εικ.13: Αναγνώριση της κόκκινης μπάλας από την κάμερα του Nao

Κόμβος «Arduino»

Είμαστε στην τελευταία αποστολή. Μένει το 1 ευρώ που πρέπει να αποκτήσει ο χρήστης. Στην αποστολή αυτή πρέπει το παιδί να υπολογίσει πόσα ευρώ του λείπουν ακόμη και να ανοίξει το σωστό κουτί. Έχει μπροστά του λοιπόν τρία κουτιά που γράφουν 1,2,3 ευρώ. Αν διαλέξει το κουτάκι με το 1 ευρώ, κερδίζει. Αλλιώς επαναλαμβάνει την αποστολή. Ακολουθεί ο κώδικας και αναλυτική επεξήγηση για το πώς επιβεβαιώνεται το αποτέλεσμα.

Κώδικας κόμβου

```
//Code executed before entering the node  
if (!this.hasSlot("ballnumber")) var this.ballnumber=0;
```

```

if (!this.hasSlot("mydata")) var this.mydata=1;
if (!this.hasSlot("timestamp")) var this.timestamp=0;
if (!this.hasSlot("ballProxy")) var this.ballProxy =
ALProxy("RedBallDetection", [] , ["subscribe","unsubscribe"]);
rootNode.myVarTemp1=2;

//Main code
tts.say("posah euro sou lee poon? vress , toe ,sauce toe, koo
tea");
//head move
head.pitch.val=10 time:2s;
this.ballnumber = 0;
var period = 500;
//enable detection
ballProxy.subscribe("TestBall" , period, 0.0 );
sleep(3s);
//start
this.mydata = robot.ALMemory.getData("redBallDetected");
this.timestamp = this.mydata[0][0];
sleep(3s);
//get data from detection
this.mydata = robot.ALMemory.getData("redBallDetected");
this.mydata = robot.ALMemory.getData("redBallDetected");
sleep(3s);
//elegxos
if (this.mydata[0][0] != this.timestamp) {
    this.ballnumber = this.ballnumber + 1;
    this.timestamp = this.mydata[0][0]; };
if (this.mydata[0][0] == this.timestamp)
    this.ballnumber = this.ballnumber + 0;
//disable

```

```

ballProxy.unsubscribe("TestBall", period, 0.0);

if (this.ballnumber < 1)
{ tts.say("Bravo, keerdysess enah euro!");
ALLeds.randomEyes(3);
sleep(3s);
    //raise hands!
    elbowRollR.val = -pi | shoulderPitchR.val = -pi/2 time:3s |
    wristYawR.val = -pi/2 | handR.val = 1 ;
    elbowRollL.val = pi | shoulderPitchL.val = -pi/2 time:3s |
    wristYawL.val = pi/2 | handL.val = 1 ;
    sleep(2s);
    //return
    elbowRollR.val = 0 | shoulderPitchR.val = pi/2 time:3s |
    wristYawR.val = pi/2 | handR.val = 0 ;
    elbowRollL.val = 0 | shoulderPitchL.val = pi/2 time:3s |
    wristYawL.val = -pi/2 | handL.val = 0 ;
    ALLeds.fadeRGB("FaceLeds", 0x0000FF, 3);
    sleep(3s);
    rootNode.myVarTemp1 = 1;  };

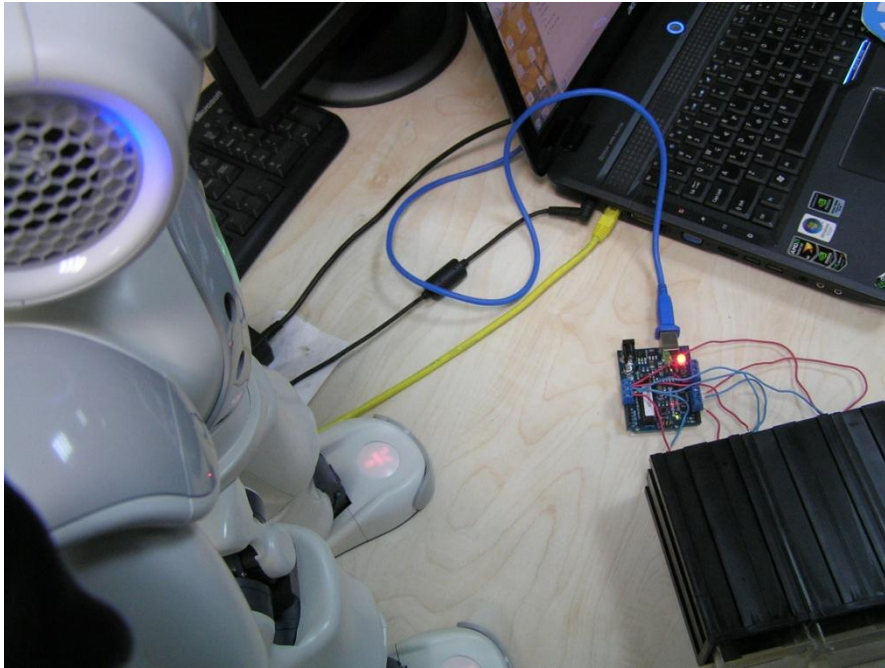
if (this.ballnumber == 1)
{ tts.say("laath oss, sou lee poon, lygo terra, euro.
Prospathyssei, xanai.");
//Eye LEDS
ALLeds.fadeRGB("FaceLeds", 0xCD00CD, 3);
sleep(3s);
rootNode.myVarTemp1=0; };

```

Κώδικας μεταβάσεων

```
rootNode.myVarTemp1 == 0
```

```
rootNode.myVarTemp1 == 1
```



Εικ. 14 : Το Nao «βλέπει» το κόκκινο Led

✚ Πώς προγραμματίζουμε το Arduino; Ποια είναι η συνδεσμολογία;

Ας πάρουμε τα πράγματα με τη σειρά. Το σημείο αυτό περιλαμβάνει προγραμματιστικό ([Ξ18]), κατασκευαστικό αλλά και ηλεκτρονικό κομμάτι. Πρώτο μας μέλημα είναι να προγραμματίσουμε σωστά το Arduino μας. Δηλαδή, όπως θα διαπιστώσετε και από τον κώδικα που ακολουθεί, έχουμε χρησιμοποιήσει για εισόδους 3 ψηφιακές εισόδους της πλατφόρμας (στις οποίες θα συνδεθούν οι διακόπτες μας όπως θα δούμε παρακάτω). Αν ανοίξει το σωστό κουτί (και άρα έχουμε τον σωστό συνδυασμό ψηφιακών εισόδων), τότε ανάβει το πράσινο Led. Αλλιώς, για κάθε άλλη επιλογή ανάβει το κόκκινο Led. Ο κώδικας που στέλουμε στο Arduino είναι ο εξής:

Κώδικας Arduino

```
#include <SPI.h> //Δήλωση βιβλιοθηκών
#include <SoftwareSerial.h>
int inputPin1 = 2; //Δήλωση μεταβλητών
int inputPin2 = 4;
int inputPin3 = 6;

void setup() { //Δήλωση εισόδων και εξόδων
```

```

pinMode(inputPin1, INPUT);
pinMode(inputPin2, INPUT);
pinMode(inputPin3, INPUT);
pinMode(12, OUTPUT);
pinMode(10, OUTPUT);
delay(100); }

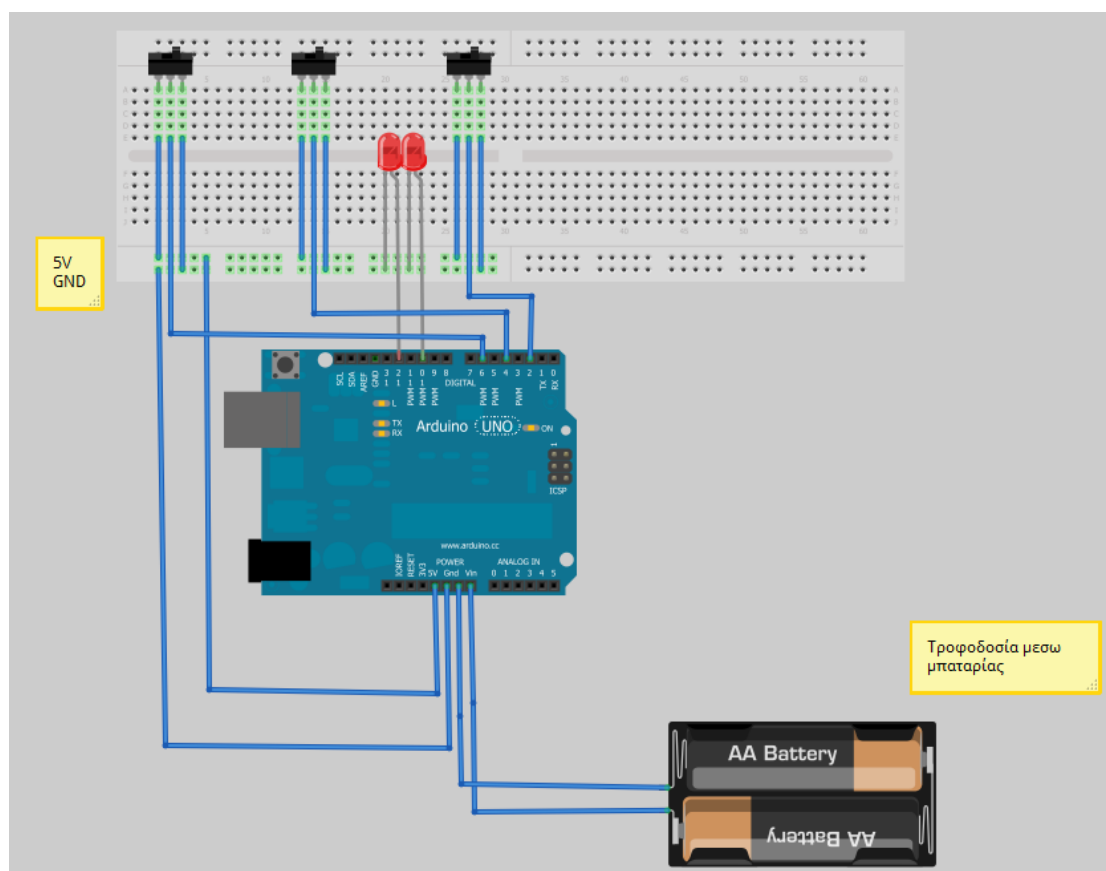
void loop() {                                     //Βρόχος ελέγχου εισόδων και καθορισμού εξόδου
if (digitalRead(inputPin1)==LOW    &&    digitalRead(inputPin2)==LOW    &&
digitalRead(inputPin3)==LOW) {
    digitalWrite(10, LOW);
    digitalWrite(12, LOW);
    delay(1); }
    if (digitalRead(inputPin1)==HIGH  &&    digitalRead(inputPin2)==LOW  &&
digitalRead(inputPin3)==LOW) {
    digitalWrite(10, HIGH);
    digitalWrite(12, LOW);
    delay(1); }
    if (digitalRead(inputPin1)==LOW   &&    digitalRead(inputPin2)==HIGH   &&
digitalRead(inputPin3)==LOW) {
    digitalWrite(10, LOW);
    digitalWrite(12, HIGH);
    delay(1); }
    if (digitalRead(inputPin1)==LOW   &&    digitalRead(inputPin2)==LOW   &&
digitalRead(inputPin3)==HIGH) {
    digitalWrite(10, LOW);
    digitalWrite(12, HIGH);
    delay(1); }
delay(1); }

```

Στη συνέχεια, χρειαζόμαστε τρία συρταράκια τα οποία θα περιέχουν από ένα νόμισμα 1, 2 και 3 ευρώ αντίστοιχα. Στο βάθος κάθε συρταριού έχουμε κολλήσει ένα διακοπτάκι (Switch) το οποίο στον ένα ακροδέκτη του έχει τη γείωση και στο άλλο την κατάσταση HIGH. Το ίδιο ισχύει και για τους υπόλοιπους δύο. Έτσι, όταν θέλουμε να ανοίξουμε ένα συρταράκι, το πιέζουμε προς τα μέσα ώστε να επανέλθει στην κανονική κατάσταση ο διακόπτης και να στείλει το σήμα «HIGH» στο Arduino. Αν είναι το σωστό κουτί ανάβει το πράσινο Led, αλλιώς φωτοβολεί το κόκκινο. Στην ουδέτερη κατάσταση όπου όλα τα κουμπάκια είναι πατημένα και αναμένουν μία απάντηση, όλα τα Led είναι σβηστά.

Αξίζει να σημειωθεί ότι αφού φορτώσουμε τον κώδικα στο Arduino, αποσυνδέουμε το Usb καλώδιο τροφοδοσίας, καθώς με μία αυτοσχέδια συνδεσμολογία τροφοδοτήσαμε την πλακέτα μας με μπαταρίες. Για την ακρίβεια, συνδέσαμε δύο μπαταρίες εν σειρά οι οποίες μας δίνουν 5V τα οποία τα συνδέουμε στα Pins: Vin και GND. Έτσι, αποφεύγουμε τη χρήση πολλών καλωδίων και η πλατφόρμα είναι πλέον αυτόνομη και ανεξαρτητή από τον υπολογιστή.

Ακολουθεί το ηλεκτρονικό κύκλωμα καθώς και μία εικόνα της τελικής υλοποίησης ώστε να γίνει πλήρως κατανοητή η λειτουργία.

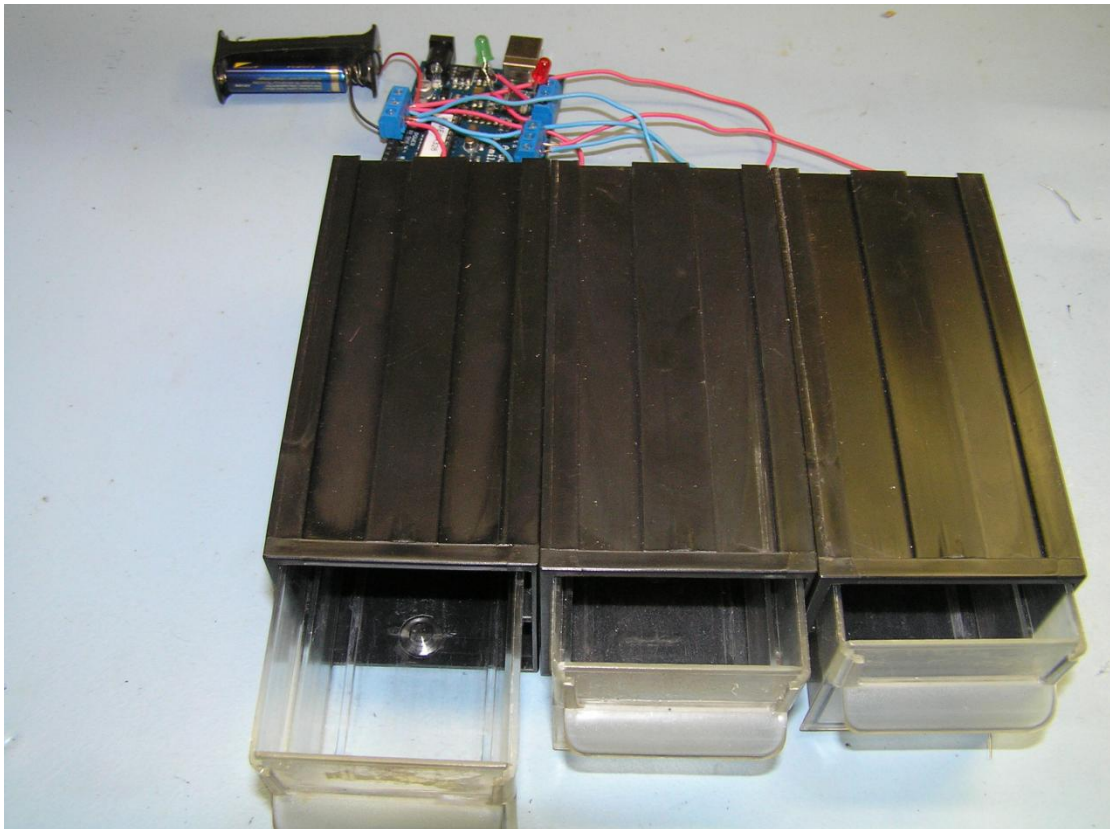


Εικ.15: Κυκλωματικό διάγραμμα της συνδεσμολογίας στην αποστολή με το Arduino με τη βοήθεια του προγράμματος Fritzing

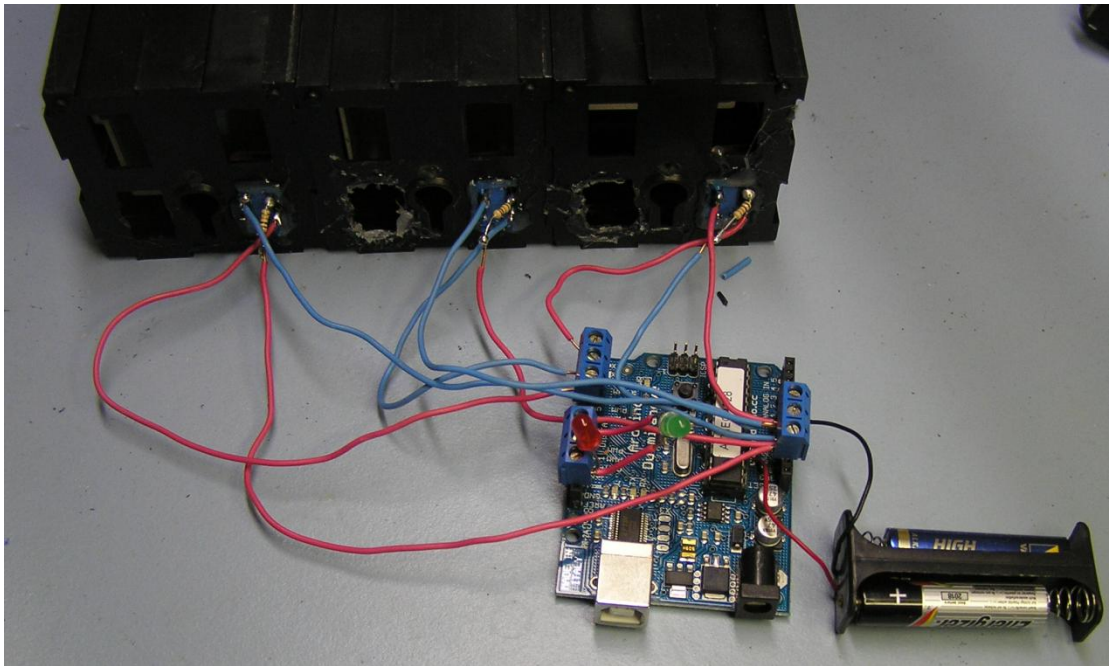
✚ Πώς ενημερώνεται το Nao για το τελικό αποτέλεσμα;

Έχουμε μέχρι στιγμής ολοκληρώσει το πιο σημαντικό κομμάτι αυτής της αποστολής. Αυτό που μένει είναι να διασαφηνίσουμε πώς το Arduino «στελνει» τις πληροφορίες των ψηφιακών εξόδων του στο Nao. Η απάντηση λοιπόν είναι η εξής. Εκμεταλλευόμαστε την ικανότητα του Nao να αναγνωρίζει κόκκινα μπαλάκια! Χρησιμοποιώντας το Module RedBallDetection, το Nao είναι σε θέση να αναγνωρίσει οτιδήποτε έχει στρογγυλό σχήμα, και κόκκινο χρώμα.

Όταν εισερχόμαστε στην αποστολή αυτή, το Nao θέτει στο παιδί την ερώτηση και αμέσως μετά περιμένει (μερικά δευτερόλεπτα) να δει ποιο Led θα ανάψει. Αν δώσουμε λάθος απάντηση, το Led κοκκινίζει, το Nao «νομίζει» ότι έχει εντοπίσει κόκκινη μπάλα και απαντά «Λάθος απάντηση». Αν όμως η απάντηση είναι σωστή, ανάβει το πράσινο Led. Το Nao περιμένει μερικά δευτερόλεπτα και εφόσον δεν εντοπίζει κόκκινη μπάλα (Led), θεωρεί την απάντηση σωστή και απαντά «Σωστή απάντηση!».



Εικ. 16 : Τα συρταράκια που περιέχουν 1,2,3 ευρώ και στην πίσω πλευρά το Arduino



Εικ.17 : Σύνδεση Arduino με τους διακόπτες στο εσωτερικό των συρταριών όπως δείχνει και το διάγραμμα

Κόμβος «Naomarks»

Έχουμε φτάσει στο σημείο που έχουν ολοκληρωθεί επιτυχώς όλες οι αποστολές από το παιδί. Το μόνο που μένει λοιπόν είναι να ελέγξουμε συνολικά τα νομίσματα που έχει συλλέξει το παιδί μέσα από το παιχνίδι αυτό για να ολοκληρωθεί η διαδικασία. Αυτό θα γίνει με την οπτική αναγνώριση των νομισμάτων.

Όπως έχουμε αναφέρει και παραπάνω, τα νομίσματα που χρησιμοποιήσαμε έχουν πάνω τους από τη μία πλευρά τον αριθμό 1, 2 ή 3 ανάλογα με την αξία τους και από την άλλη το Naomark που αντιστοιχεί στην αξία αυτή. Το παιδί λοιπόν θα δείχνει ένα προς ένα τα νομίσματα (όταν του λέει το ρομπότ) και σε κάθε νόμισμα θα προσθέτει την αξία του τρέχοντος νομίσματος σε μία μεταβλητή. Ακολουθεί ο κώδικας.

```
//Code executed before entering the node
if (!this.hasSlot("data")) var this.data=1;
if (!rootNode.hasLocalSlot("myVar1")) var rootNode.myVar1=0;
if (!rootNode.hasLocalSlot("myVar2")) var rootNode.myVar2=0;
if (!rootNode.hasLocalSlot("myVar3")) var rootNode.myVar3=0;
if (!rootNode.hasLocalSlot("myPrice")) var rootNode.myPrice=0;
if (!this.hasSlot("myExtra1")) var this.myExtra1=1;
if (!this.hasSlot("myExtra2")) var this.myExtra2=2;
if (!this.hasSlot("myExtra3")) var this.myExtra3=3;

//Main code
tts.say("Dikse dikseh mou ta no miss hmata, enah enah.");
//head move
head.pitch.val=10 time:2s;
//create a proxy to ALMarkDetection
var markProxy = ALProxy("LandMarkDetection", [] ,
["subscribe","unsubscribe"]);
var period = 500;
//enable detection
markProxy.subscribe("TestMark", period, 0.0 );

//loop mexri na deiksei ola ta marks
while (rootNode.myPrice < 10)
{tts.say("dikseh mou enah no miss hma.");

//get data from detection
sleep(3s);
this.data = robot.ALMemory.getData("LandmarkDetected");
sleep(3s);
}
//do it 4 times for sure
```

```

var i=1;
while ((this.data == []) && (i<5))
{
    sleep(3s);
    this.data = robot.ALMemory.getData("LandmarkDetected");
    i+=1;
    sleep(3s); };
if (this.data != []) {
    switch(this.data[1][0][1][0])
    {
        case 64: rootNode.myPrice = rootNode.myPrice + this.myExtra1;
        case 85: rootNode.myPrice = rootNode.myPrice + this.myExtra2;
        case 108: rootNode.myPrice = rootNode.myPrice + this.myExtra3;
    };};};
//disable
markProxy.unsubscribe("TestMark", period, 0.0);

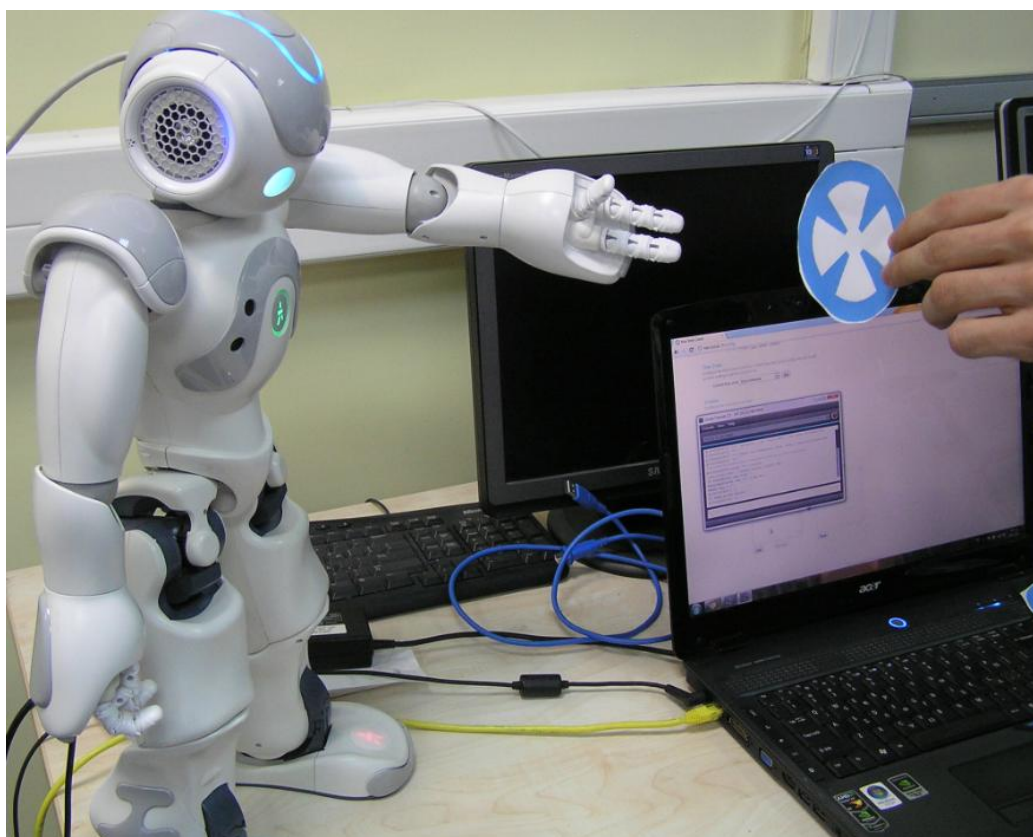
if (rootNode.myPrice == 10) {
tts.say("Bravo! Ker dyss es,!");
elbowRollR.val = -pi | shoulderPitchR.val = 0 time:3s
|handR.val = 1 ;elbowRollL.val = pi |
shoulderPitchL.val = 0 time:3s | handL.val = 1 ;
sleep(2s);
elbowRollR.val = -pi | shoulderPitchR.val = pi/2 time:3s |
handR.val = 0 ;elbowRollL.val = pi |shoulderPitchL.val = pi/2
time:3s | handL.val = 0 ;
sleep(2s); };

if (rootNode.myPrice < 10)
tts.say("laath oss, sou lee poon, perisso terra, euro.
Prospathyssei, xanai.");
if (rootNode.myPrice > 10)

```

```
tts.say("laath oss, sou lee poon,lygo terra,euro.Prospathyssei,  
xanai.");
```

Σε αυτόν τον κόμβο δεν έχουμε κώδικα μεταβάσεων, καθώς γίνεται η τελική επιβεβαίωση του σεναρίου. Μετά από τον κόμβο αυτό, ο έλεγχος περνά και πάλι στον πρώτο κόμβο και ένα νέο παιχνίδι ξεκινά!

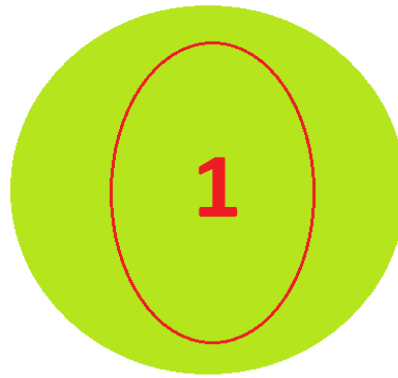
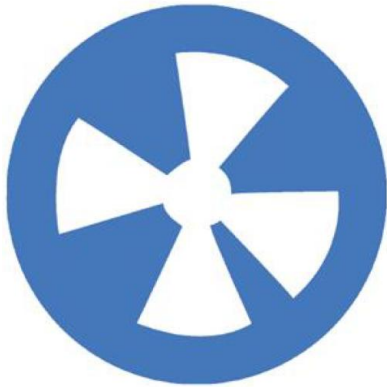


Εικ. 18 : Το Nao αναγνωρίζει το Naomark

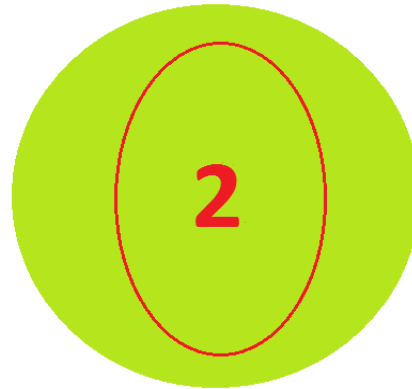
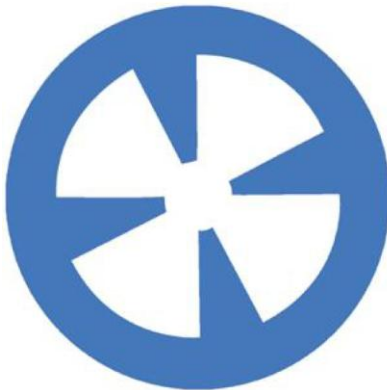
Τι είναι τα Naomarks;

Τα Naomarks είναι συγκεκριμένα σύμβολα-εικονίδια τα οποία έχει σχεδιάσει ίδια η εταιρεία Aldebaran ώστε να μπορεί να τα ανιχνεύει και να τα αναγνωρίζει η κάμερα του Nao. Συνολικά, υπάρχουν 10 διαφορετικά Naomarks, αλλά εμείς χρειαστήκαμε μόνο τρία από αυτά για να συμβολίσουμε τα 1,2 και 3 ευρώ αντίστοιχα.

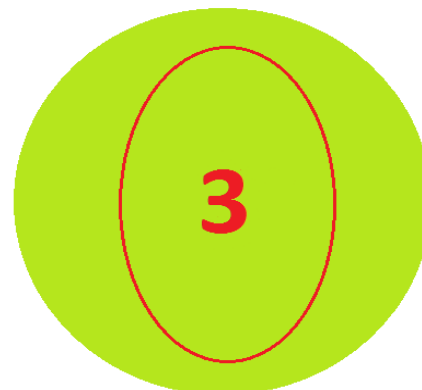
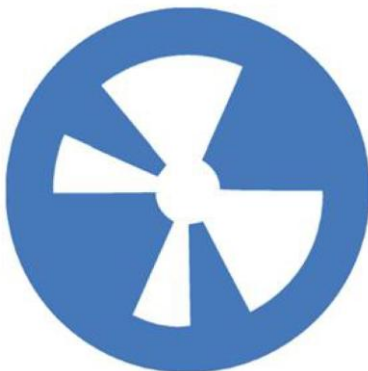
Για να μπορεί λοιπόν το παιδί να ξεχωρίσει τα νομίσματα μεταξύ τους, σε κάθε νόμισμα έχουμε από τη μία πλευρά τον αριθμό 1, 2 ή 3 ανάλογα με την αξία του και από την άλλη το Naomark που αναλογεί στην αξία αυτή. Τα Naomarks είναι το έπαθλο που κερδίζει το παιδί σε κάθε δοκιμασία του σεναρίου. Αυτά που θα χρησιμοποιήσουμε εμείς είναι τα ακόλουθα:



Ναοmark 64. Αντιστοιχεί σε 1 ευρώ



Ναοmark 85. Αντιστοιχεί σε 2 ευρώ



Ναοmark 108. Αντιστοιχεί σε 3 ευρώ

Εικ.19: Οι δύο όψεις των νομισμάτων

Πώς αναγνωρίζει το Nao τα Naomarks;

Όπως επισημάναμε, κάθε Naomark έχει έναν κωδικό αριθμό που το ξεχωρίζει από τα υπόλοιπα. Συνεπώς, σε κάθε οπτική αναγνώριση που γίνεται από την κάμερα του Nao, το νόμισμα αντιστοιχίζεται σε αυτόν τον κωδικό αριθμό και προστίθεται στο ποσό των ευρώ η αντίστοιχη αξία. Η αντιστοιχία μεταξύ κωδικού αριθμού και χρηματικού ποσού είναι η εξής:

Naomark 64. Αντιστοιχεί σε 1 ευρώ

Naomark 85. Αντιστοιχεί σε 2 ευρώ

Naomark 108. Αντιστοιχεί σε 3 ευρώ

Σε ότι αφορά τον κώδικα, χρησιμοποιούμε ένα Module το οποίο λέγεται ALLandMarkDetection και υπάρχει έτοιμο στο API Guide του Nao. Αυτό το Module καθιστά ικανό το ρομπότ μας να αναγνωρίζει τα Naomarks και να τα ξεχωρίζει μεταξύ τους. Μόλις λοιπόν η κάμερα του Nao εντοπίσει ένα Naomark αποθηκεύει τα «στοιχεία» του στο ALMemory, στο microEvent, στη μεταβλητή LandmarkDetected. Λέγοντας «στοιχεία» εννοούμε τη χρονική στιγμή που το είδε, και τα χαρακτηριστικά του Naomark. Δηλαδή, η μεταβλητή LandmarkDetected έχει τη μορφή:

***[[timestamp_seconds,timestamp_microseconds],
[mark_info_0,mark_info_1, ,mark_info_N]]***

με τόσα mark_info στοιχεία όσα έχουν ανιχνευθεί.

Αναλυτικά, τα tags είναι τα εξής:

mark_info_x = [[shape, alpha, beta, sizeX, sizeY, heading], [markID]].

- alpha και beta είναι η θέση του Naomark όπως το βλέπει η κάμερα
- sizeX και sizeY είναι το μέγεθος του mark όπως το βλέπει η κάμερα
- το heading περιγράφει πώς είναι προσανατολισμένο το Naomark σε σχέση με τον κατακόρυφο άξονα του κεφαλιού του Nao.
- Mark ID είναι ο αναγνωριστικός αριθμός του naomark

Μετά από μερικές δοκιμές με την κάμερα του ρομπότ (βλ. Monitor παρακάτω) διαπιστώσαμε ότι μία ικανοποιητική διάμετρος για το νόμισμα είναι τα 9 εκατοστά, ώστε να είναι ορατά από το ρομπότ σε απόσταση 30 εκατοστών από την κάμερα.

Επίσης, διαπιστώσαμε ότι τα νομίσματα πρέπει να τα βλέπει ένα προς ένα, καθώς μπορεί να αναγνωρίσει έως τρία νομίσματα μαζί η κάμερα. Εμείς όμως σε αυτό το στάδιο θα έχουμε περισσότερα και θα τα δείχνουμε ένα ένα προς αποφυγή προβλημάτων και λανθασμένων υπολογισμών.

Κόμβοι «wrong»

Παρατηρούμε ότι δίπλα σε κάθε αποστολή έχουμε και έναν κόμβο με το όνομα «wrong». Οι κόμβοι αυτοί έχουν όλοι την ίδια χρήση στο πρόγραμμά μας. Δηλαδή, τους χρησιμοποιούμε ως μεταβατικούς κόμβους στην περίπτωση που το παιδί απαντήσει λάθος στην ερώτηση της αποστολής. Οπότε, μεταβαίνει ο έλεγχος του προγράμματός μας εκεί και επιστρέφει εκ νέου στην αποστολή που απέτυχε ο παίκτης. Αυτό συμβαίνει όσες φορές και να πάρει λάθος απάντηση το ρομπότ, μέχρι να απαντήσει το παιδί σωστά στην ερώτηση και να πάει στην επόμενη δοκιμασία. Ο κώδικας είναι ο ίδιος και είναι ο εξής

Κώδικας κόμβου

```
sleep(2s);  
//επαναφορά του κεφαλιού γιατί έχει αλλάξει θέση στην αποστολή  
head.pitch.val=0 time:3s;  
ALLeds.fadeRGB("FaceLeds", 0x0000FF, 3); //επαναφορά ματιων στο μπλε  
sleep(2s);
```

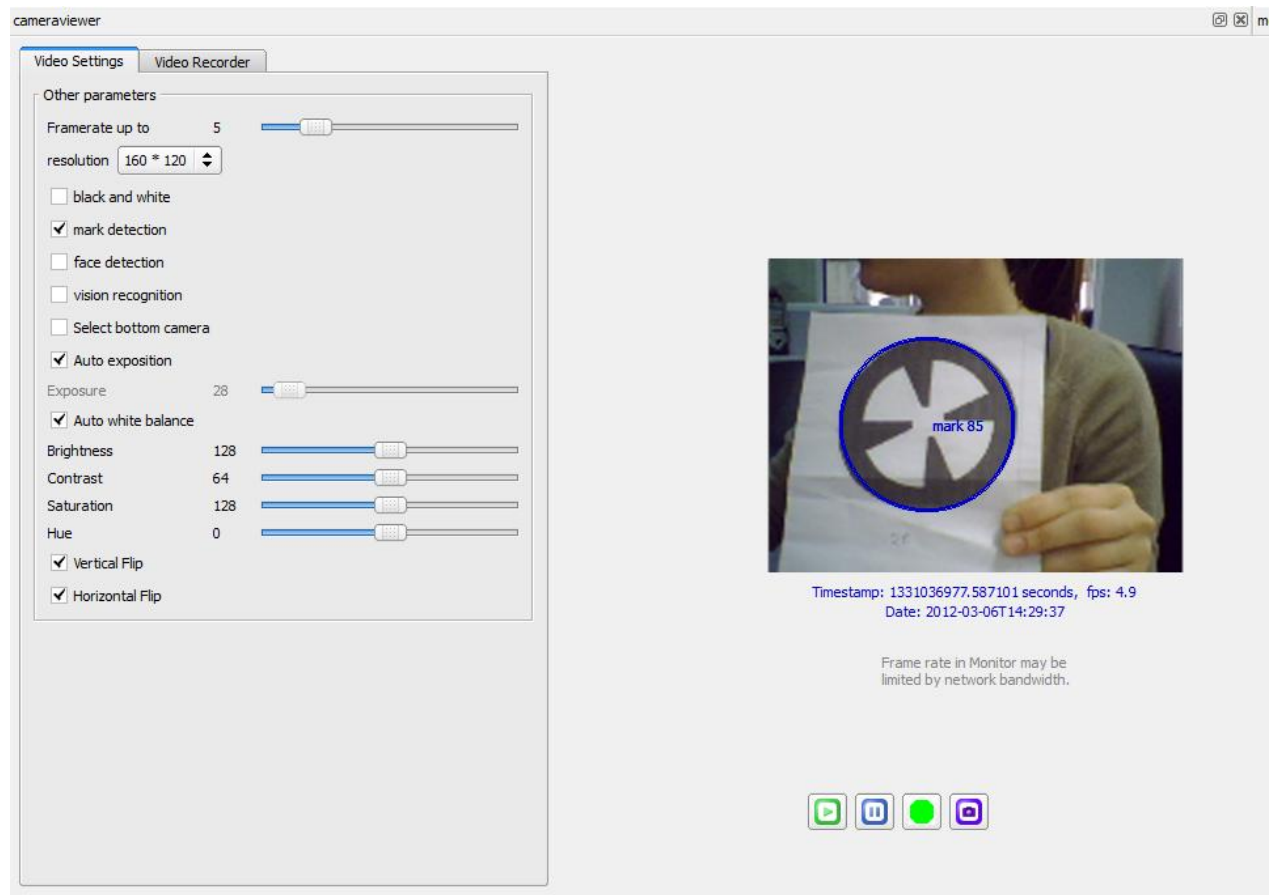
5.3.3 Το πρόγραμμα Monitor

✚ Πώς βλέπουμε μέσα από την κάμερα του Nao;

Όπως προαναφέραμε, για να αλληλεπιδράσουμε με το Nao «κατεβάσαμε» το Choregraphe από την επίσημη σελίδα της Aldebaran. Όταν εγκαθιστούμε το πρόγραμμα αυτό λοιπόν, εγκαθίσταται μαζί και το πρόγραμμα **Monitor**. Το πρόγραμμα αυτό το χρησιμοποιήσαμε όταν αρχίσαμε πλέον να κάνουμε τα πειράματά μας και θέλαμε να δούμε τι ακριβώς βλέπει το Nao.

Όταν θέλαμε να κάνουμε δοκιμές με τα νομίσματα για να καταλήξουμε στη διάμετρο του νομίσματος για να είναι ορατό από την κάμερα, χρησιμοποιήσαμε το Monitor. Όπως φαίνεται και στη φωτογραφία που ακολουθεί, όταν το Nao δει ένα Naomark, τότε εμφανίζει έναν κύκλο γύρω του και γράφει δίπλα το ID που αντιστοιχεί στο Naomark που μόλις είδε. Διαπιστώσαμε ότι μετά τα 8 εκατοστά αρχίζει να βλέπει σταθερά το νόμισμα και γιατί καταλήξαμε στη διάμετρο που επιλέξαμε.

Την ίδια διαδικασία ακολουθήσαμε και για να δούμε αν ανιχνεύει κόκκινες μπάλες. Στη μία πλευρά της οθόνης μας είχαμε την κάμερα για δούμε αν είναι στο οπτικό του πεδίο και στην άλλη την κατάσταση της μεταβλητής redballdetected στην ALMemory. Παρατηρούσαμε ότι μόλις ανίχνευε κόκκινη μπάλα, άλλαζαν τα δεδομένα της μεταβλητής και έτσι «καταλαβαίνει» το πρόγραμμά μας ότι είδε το ρομπότ κόκκινη μπάλα (αναλυτικά στην παράγραφο «Πώς αναγνωρίζει τις κόκκινες μπάλες;»).

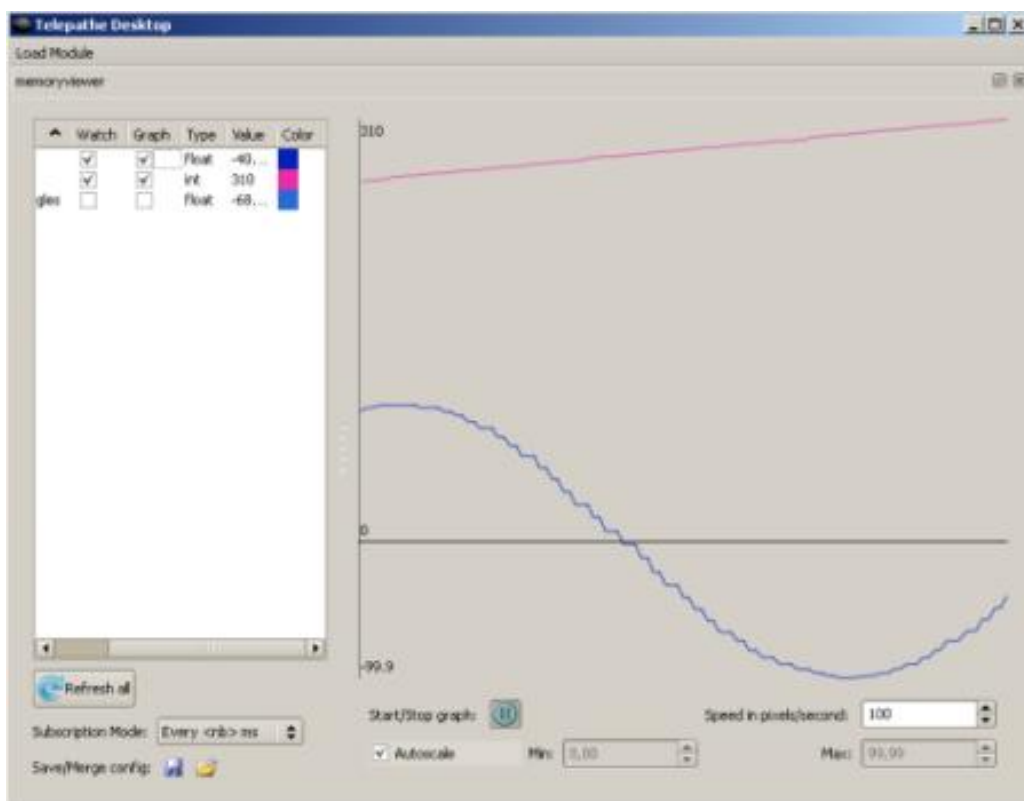


Εικ.20: Πώς βλέπουμε μέσα από την κάμερα του Nao

+ Πώς ελέγχουμε τις θέσεις της ALMemory που μας ενδιαφέρουν;

Δεν είναι όμως η μοναδική λειτουργία του Monitor αυτή που μόλις αναλύσαμε. Μπορούμε παράλληλα με την κάμερα να βλέπουμε και το πώς ενημερώνονται όλες οι μεταβλητές των microEvents που είναι αποθηκευμένες στο ALMemory π.χ. (redBallDetected) και έτσι να επιβεβαιώσουμε τη σωστή λειτουργία του προγράμματός μας.

Έτσι, μόλις συνδεθούμε με το ρομπότ μας, επιλέγουμε από τη λίστα όλων των μεταβλητών αυτές που θέλουμε να παρακολουθήσουμε την αλλαγή τους και αμέσως εμφανίζεται ο πίνακας με τις τιμές τους καθώς και ένα γραφικό σε συνάρτηση με το χρόνο (το οποίο δε χρειάστηκε στα πειράματά μας).



Εικ.21: Στιγμιότυπο του ALMemory όπως απεικονίζεται από το Monitor

6

Πειράματα - Δοκιμές

Σκοπός αυτής της διπλωματικής εργασίας όπως αναφέραμε είναι η εξοικείωση των παιδιών προσχολικής και πρώτης σχολικής ηλικίας με τα ρομποτικά συστήματα, αλλά βασικός στόχος είναι η εκμάθηση επίλυσης απλών αριθμητικών προβλημάτων με τη βοήθεια ενός ρομποτικού εκπαιδευτικού συστήματος. Πρέπει λοιπόν να κάνουμε δοκιμές σε παιδιά και να δούμε τις αντιδράσεις τους και τις επιδόσεις τους.

- ✓ Επιλέξαμε παιδιά προσχολικής και πρώτης σχολικής ηλικίας 4-7 ετών και των δύο φύλων ώστε να διαπιστώσουμε ποιες ηλικίες επιδέχονται εκμάθηση αριθμητικής με ρομπότ και ποιες όχι.
- ✓ Καταγράψαμε τις επιδόσεις των παιδιών αυτών. Έτσι θα συμπεράνουμε αν τα παιδιά μεγαλύτερης ηλικίας έχουν καλύτερες επιδόσεις από τα μικρότερα παιδιά στις απλές αριθμητικές πράξεις.
- ✓ Παράλληλα, δώσαμε ιδιαίτερη σημασία στις αντιδράσεις των παιδιών όση ώρα αλληλεπιδρούσαν με αυτό. Παρατηρήσαμε διαφορές ανάλογα με την ηλικία και το φύλο των παιδιών. Μας ενδιαφέρει να είναι ευχάριστη και προσιτή η όλη διαδικασία στο νεαρό χρήστη.
- ✓ Τέλος, καταφέραμε να κάνουμε αυτόνομη την τροφοδοσία του Arduino Duemillanove. Χρησιμοποιούμε δύο μπαταρίες εν σειρά οι οποίες μας δίνουν 5V και οδηγούμε την τάση αυτή στα Pins: Vin και GND. Έτσι, αφού «φορτώσουμε» τον κώδικα μέσω του καλωδίου USB, το αποσυνδέουμε και τροφοδοτείται πλέον μόνο από τις μπαταρίες. Έτσι, αποσυμφορούμε λίγο το κύκλωμά μας, και απομακρυνόμαστε από τον υπολογιστή.

7 Προβλήματα

Κατά τη διάρκεια της υλοποίησης του σεναρίου αλλά και κατά τη διάρκεια της πειραματικής διαδικασίας και του ελέγχου, βρεθήκαμε αντιμέτωποι με σημαντικά προβλήματα. Στόχος μας ήταν η εύρεση εναλλακτικών δρόμων επίλυσης των προβλημάτων μας αλλά και παράλληλα η διατήρηση του σεναρίου μας ως έχει στο μεγαλύτερο δυνατό βαθμό. Πράγματι, στα περισσότερα προβλήματα καταφέραμε να βρούμε εναλλακτικές λύσεις και να υλοποιήσουμε το σενάριο όπως το είχαμε αρχικά σχεδιάσει.

7.1 Προβλήματα γλώσσας

Το πρώτο εμπόδιο που συναντήσαμε ήταν η απουσία της ελληνικής γλώσσας από τη μνήμη του Nao. Έτσι, εμφανίστηκαν τα ακόλουθα προβλήματα:

- ✓ Σε όλη τη διάρκεια του σεναρίου και στα πλαίσια της αλληλεπίδρασης του ρομπότ με το παιδί, το Nao χρειάστηκε να «μιλήσει» ώστε να εξηγήσει σε αυτό τη διαδικασία του παιχνιδιού. Η απουσία της ελληνικής γλώσσας από τη μνήμη του ρομπότ, ήταν φυσικό να δημιουργήσει ένα πρόβλημα κατανόησης από τη πλευρά του παιδιού. Μία πρώτη ιδέα ήταν να ηχογραφήσουμε τις προτάσεις που έλεγε το ρομπότ στα ελληνικά. Το πρόβλημα όμως αμέσως εμφανίστηκε. Στις μεγάλες προτάσεις γινόταν αυτόματη αποσύνδεση από το ρομπότ με αποτέλεσμα να μην μπορούμε να προχωρήσουμε στο σενάριο. Καταλήξαμε στο συμπέρασμα ότι το Nao δεν μπορεί να εκτελέσει εντολές που έχουν διάρκεια μεγαλύτερη από 1-2 δευτερόλεπτα. Συνεπώς καταλήξαμε στη χρήση της εντολής “ `tts.say(“ ”)` ” και μέσα στα εισαγωγικά

χρησιμοποιήσαμε το λατινικό αλφάβητο συνθέτοντας συλλαβές που «ηχούν» ελληνικές.

- ✓ Το αποτέλεσμα αυτής της εντολής δεν είναι πάντα ικανοποιητικό, με αποτέλεσμα να μην είναι πάντα ξεκάθαρο και κατανοητό αυτό που λέει το ρομπότ.
- ✓ Στη δεύτερη αποστολή, το ζητούμενο ήταν να κάνει το παιδί την αφαίρεση $10-4=6$ και να δώσει την απάντηση «έξι» κοντά στο ακουστικό του ρομπότ. Όμως, τα παιδιά στα οποία απευθύνεται αυτό το σενάριο είναι 4-7 ετών τα οποία φυσικά δε γνωρίζουν την αγγλική αρίθμηση. Άρα, σε αυτό το στάδιο χρειάζεται η συμβολή κάποιου ενήλικα.

7.2 Προβλήματα «όρασης»

Στη συνέχεια, συναντήσαμε προβλήματα χρησιμοποιώντας την κάμερα του ρομπότ. Αυτά ήταν κυρίως προβλήματα αναγνώρισης αντικειμένων, τα οποία τελικά ξεπεράσαμε.

- ✓ Μία από τις αποστολές ήταν να δείξει το παιδί στο Nao τόσα κόκκινα μπαλάκια όσα τα ευρώ που του χρειάζονται. Παρατηρήσαμε ότι δείχνοντας στην κάμερα του ρομπότ περισσότερα από ένα κόκκινα μπαλάκια, το ρομπότ δεν μπορούσε να τα μετρήσει. Για την ακρίβεια έβλεπε μόνο το μεγαλύτερο, δηλαδή αυτό που ήταν πιο κοντά στην κάμερα. Συνεπώς, το παιδί θα πρέπει να δείξει τα μπαλάκια ένα προς ένα στην κάμερα.
- ✓ Επόμενη δυσκολία, η αναγνώριση των Naomarks. Τυπώσαμε διάφορα μεγέθη Naomarks και τα δοκιμάσαμε σε διάφορες αποστάσεις από το ρομπότ. Καταλήξαμε ότι πρέπει να έχουν διάμετρο περίπου 8-9 εκατοστά και να είναι σε απόσταση περίπου 20 εκατοστά από την κάμερα. Επίσης, θα πρέπει να είναι κάθετα στην ευθεία που ενώνει το σημείο που είναι η κάμερα με το κέντρο του Naomark. Επίσης, παρατηρήσαμε ότι στο οπτικό πεδίο της κάμερας χωράνε μόνο τρία Naomarks. Στο στάδιο που κάνουμε αναγνώριση όμως, έχουμε περισσότερα νομίσματα με αποτέλεσμα να μην τα βλέπει όλα το Nao. Η λύση που προτείναμε είναι να τα δείχνει το παιδί ένα προς ένα.

- ✓ Στην αποστολή που κάνουμε χρήση της πλατφόρμας Arduino, στηρίζομαστε στο γεγονός ότι σε περίπτωση λάθους θα ανάψει το κόκκινο λαμπάκι και θα το αναγνωρίσει το ρομπότ. Το πρόβλημά μας εδώ είναι ότι το Nao δε βλέπει πάντα το κόκκινο Led. Πρέπει το Arduino να είναι σε πολύ μικρή απόσταση από την κάμερα και κάθετο στην ευθεία που περνά από την κάμερα και από το λαμπάκι.

7.3 Πρόβλημα προσομοίωσης

Με την ολοκλήρωση κάθε αποστολής, έπρεπε να δοκιμάσουμε αν ο κώδικάς μας «έτρεχε» σωστά στο ρομπότ μας. Όμως, δεν ήταν πάντα εφικτό να έχουμε πρόσβαση σε αυτό, οπότε σαν εναλλακτική θα μπορούσαμε θεωρητικά να χρησιμοποιήσουμε το Choregraphe.

- ✓ Το Choregraphe παρ' όλο που η Aldebaran το δίνει ως ελεύθερο λογισμικό για τους χρήστες του Nao με σκοπό να έχουν στον υπολογιστή τους ένα «εικονικό ρομπότ», δεν ήταν εφικτό να «τρέξουμε» τον κώδικα που γράψαμε στο Gostai και να ελέγξουμε την ορθότητά του. Ήταν απαραίτητη κάθε φορά η σύνδεση με το ρομπότ του εργαστηρίου, αφού το Choregraphe δεν υποστήριζε όλες τις λειτουργίες που χρειαζόμασταν, όπως η σύνθεση φωνής, η φωνητική αναγνώριση ή το άγγιγμα του αισθητήρα.

7.4 Άλλα προβλήματα

- ✓ Στην αποστολή με χρήση του Arduino βρεθήκαμε μπροστά στο πρόβλημα επικοινωνίας Arduino – Nao. Έπρεπε να στείλουμε την απάντηση του παιδιού στο Nao, και αυτό να την ανακοινώσει. Μία ιδέα ήταν η σύνδεση με το Nao μέσω Ethernet. Χρησιμοποιήσαμε ένα Arduino Ethernet Shield ώστε να στέλνουμε τα δεδομένα μέσω του εξυπηρετητή, αλλά είχαμε πρόβλημα στη δημιουργία πελάτη-εξυπηρετητή μέσω του Gostai.
- ✓ Σε αρκετές δυσκολίες που αντιμετωπίσαμε, χρειάστηκε να καταφύγουμε στο φόρουμ της Urbi²⁴. Αρκετές από τις απορίες μας ήταν ήδη απαντημένες στα αντίστοιχα topics,

²⁴ <http://forum.urbiforge.org/viewforum.php?f=16>

αλλά υπήρχαν και προβλήματα που δεν ήταν απαντημένα πουθενά. Αφού θέταμε λοιπόν το ερώτημά μας, έπρεπε να περιμένουμε αρκετό καιρό για να λάβουμε απάντηση.

- ✓ Το SDK της Urbi δεν ήταν επαρκές για αυτό που θέλαμε να κάνουμε. Υπήρχαν σημεία για τα οποία δεν βρίσκαμε επαρκείς πληροφορίες για το πώς θα τα υλοποιήσουμε. Τρανταχτό παράδειγμα ήταν η προσπάθειά μας να στείλουμε δεδομένα από το Arduino στο Nao δημιουργώντας έναν πελάτη και έναν εξυπηρετητή, αλλά δεν καταφέραμε να βρούμε κατάλληλη βοήθεια για να το υλοποιήσουμε.

8 Επίλογος

8.1 Σύνοψη, συμπεράσματα

Ας θυμηθούμε και πάλι τους στόχους που είχαμε θέσει ξεκινώντας αυτήν τη διπλωματική εργασία. Βασικός μας σκοπός ήταν να εξοικειώσουμε τα παιδιά προσχολικής και πρώτης σχολικής ηλικίας με τη διαδικασία επίλυσης προβλημάτων, να τους διδάξουμε με τον πιο ευχάριστο και δημιουργικό τρόπο βασικές έννοιες της αριθμητικής και να αποκτήσουν μία πρώτη επαφή με τα βασικά στοιχεία ενός ρομποτικού συστήματος. Ας μην ξεχνάμε βέβαια ότι για την υλοποίηση αυτού του σεναρίου χρησιμοποιήσαμε το ρομπότ Nao της Aldebaran καθώς και την πλατφόρμα Arduino. Τέλος, το Gostai Studio ήταν το ισχυρό μας εργαλείο μέσω του οποίου στέλναμε τις εντολές στο ρομπότ μας. Ακολουθούν κάποια συμπεράσματα στα οποία καταλήξαμε κατά την υλοποίηση του σεναρίου:

- ✓ Όπως είπαμε η εφαρμογή αυτή απευθύνεται σε παιδιά προσχολικής και πρώτης σχολικής ηλικίας. Από τα πειράματά μας διαπιστώσαμε ότι τα παιδιά έδειξαν ενθουσιασμό όταν αντίκρυσαν το Nao και ιδιαίτερο ενδιαφέρον να παίξουν μαζί του. Αυτό είναι ένας παράγοντας που συμβάλλει στην αποτελεσματικότητα της εκπαιδευτικής διαδικασίας, κάνοντας πιο ευχάριστη την εκμάθηση αριθμητικής. Πιστεύουμε ότι αν είχαμε στα πειράματά μας παιδιά με μαθησιακές δυσκολίες σίγουρα θα είχε ακόμη πιο θεαματικά αποτελέσματα αυτή η εφαρμογή.

- ✓ Παρατηρήσαμε ότι τα παιδιά μεγαλύτερων ηλικιών (6-7) είχαν καλύτερες επιδόσεις από τα μικρότερα και πιο γρήγορες αντιδράσεις. Το γεγονός αυτό οφείλεται στις βασικές γνώσεις αριθμητικής που έχουν ήδη τα παιδιά αυτής της ηλικίας είτε από το σχολείο είτε από τους γονείς τους. Επίσης, τα παιδιά αυτά ήταν πιο εξοικειωμένα με τη μορφή και την έννοια του ρομπότ, καθώς τα παιδιά πλέον έρχονται πολύ νεότερα σε επαφή με την τεχνολογία. Αντίθετα, με τα παιδιά που ανήκουν σε μικρότερες ηλικιακές ομάδες (4-5) είχαμε δυσκολία στην πρώτη επαφή με το ρομπότ, καθώς έχαναν την προσοχή τους λόγω εντυπωσιασμού και άγνοιας.



Εικ.22: Στιγμιότυπο πειραμάτων

- ✓ Όταν είχαμε σχεδιάσει στο χαρτί μας το σενάριο, έπρεπε να επιλέξουμε το πρόγραμμα που θα χρησιμοποιήσουμε για να το υλοποιήσουμε. Ξεκινήσαμε με το Microsoft Robotics Studio, αλλά παρατηρήσαμε ότι κάποιες λειτουργίες του Nao δεν υποστηρίζονταν από το πρόγραμμα αυτό. Έτσι, καταλήξαμε στο Gostai Suite. Το λογισμικό αυτό είναι πολύ εύχρηστο από τον προγραμματιστή, καθώς η συμπεριφορά του ρομπότ καθορίζεται από το διάγραμμα καταστάσεων που σχεδιάζουμε και όχι από αμιγή προγραμματισμό.
- ✓ Ξεκινώντας την υλοποίηση του σεναρίου δυσκολευτήκαμε να λύσουμε το πρόβλημα συμβατότητας μεταξύ Choregraphe, Gostai και Nao robot. Μετά από αρκετές δοκιμές και updates στο Nao καταλήξαμε να χρησιμοποιήσουμε το Gostai Studio 2.6.1 και το Choregraphe 1.12.0.62.
- ✓ Ας περάσουμε στην πλατφόρμα Arduino, μία πολύ εξελιγμένη ιδέα. Χρησιμοποιώντας το Arduino Duemillanove διαπιστώσαμε ότι αυτός ο

μικροελεγκτής είναι ιδιαίτερα φιλικός προς τον προγραμματιστή. Επιπλέον, έχει απεριόριστες δυνατότητες καθώς κάθε χρόνο η εταιρεία Arduino βγάζει στην αγορά νεότερα Arduino Shields τα οποία επεκτείνουν τις δυνατότητες της πλατφόρμας προς κάθε κατεύθυνση.

- ✓ Μετά την ολοκλήρωση της υλοποίησης και των πειραμάτων τολμούμε να πούμε ότι τα καταφέραμε! Έχουμε πλέον κατορθώσει να στήσουμε ένα παιχνίδι σοβαρού σκοπού (Serious Game) το οποίο εκπληρώνει απόλυτα τους στόχους μας. Όχι μόνο μαθαίνει βασικές αριθμητικές έννοιες στα παιδιά, αλλά το καταφέρνει με τον πιο ευχάριστο και δημιουργικό τρόπο.

8.2 Αποτελέσματα

Έχοντας ολοκληρώσει το προγραμματιστικό κομμάτι της εργασίας στο Gostai Studio ας δούμε πώς πήγε κάθε τμήμα της υλοποίησης.

- ✓ Το κομμάτι της ομιλίας του Nao πήγε σχετικά καλά σε ότι αφορά στη γλώσσα. Αρχικά είχαμε χρησιμοποιήσει τα αγγλικά, αλλά εφόσον απευθυνόμαστε σε μικρά παιδιά που δε γνωρίζουν αγγλικά, χρησιμοποιήσαμε το λατινικό αλφάβητο σχηματίζοντας ελληνικές λέξεις.
- ✓ Στα επαναλαμβανόμενα κομμάτια του κώδικα έπρεπε με κάποιον τρόπο να δώσουμε στο παιδί να καταλάβει ότι είναι η ώρα π.χ. να κτυπήσει τον αισθητήρα, ή να δείξει το επόμενο νόμισμα. Έτσι, όταν είναι έτοιμο το ρομπότ λέει στο παιδί να κάνει την επόμενη κίνηση.
- ✓ Τα Led στα μάτια δούλεψαν τέλεια χωρίς κανένα πρόβλημα
- ✓ Οι κινήσεις δοκιμάστηκαν αρχικά στο Choregraphe αλλά και στο ρομπότ τελικά. Όλα δούλεψαν χωρίς προβλήματα.
- ✓ Στην πρώτη αποστολή χρησιμοποιήσαμε τον ένα αισθητήρα του κεφαλιού. Αρχικά χρησιμοποιήσαμε τον μπροστινό αισθητήρα, αλλά μάλλον λόγω συχνής χρήσης δεν ανταποκρινόταν όπως θέλαμε. Έτσι, χρησιμοποιήσαμε τον μεσαίο.
- ✓ Η αναγνώριση της κόκκινης μπάλας (RedBallDetection) λειτούργησε άψογα στο ρομπότ, με τη μόνη παραδοχή ότι πρέπει να τις βλέπει μία προς μία.
- ✓ Η ηχητική αναγνώριση λειτούργησε καλά και αναγνωρίζει πάντα τη σωστή απάντηση.

- ✓ Η χρήση της πλατφόρμας Arduino δεν είχε προβλήματα. Κάποιες φορές όμως το Nao δεν εντοπίζει το κόκκινο λαμπάκι με αποτέλεσμα να μεταβαίνει ο έλεγχος στην αρχή της αποστολής.
- ✓ Η αναγνώριση των Naomarks δεν εμφάνισε προβλήματα. Δούλεψε εξ'αρχής καλά και αναγνώριζε σωστά όλα τα νομίσματα

8.3 *Μελλοντικές επεκτάσεις, βελτιώσεις και προτάσεις*

Φτάνοντας στο τέλος αυτής της εργασίας, αξίζει να παραθέσουμε μερικές ιδέες βελτίωσης αυτού του παιχνιδιού σοβαρού σκοπού με τη χρήση ρομποτικού συστήματος. Όπως είδαμε στο θεωρητικό υπόβαθρο, τέτοιου τύπου εφαρμογές φέρνουν τα παιδιά σε επαφή με την τεχνολογία, τους μαθαίνουν βασικές αριθμητικές έννοιες και βοηθούν την κοινωνικοποίηση παιδιών με μαθησιακές δυσκολίες και δυσλεξία ([Ξ13]).

Ας δούμε λοιπόν κάποιες ιδέες, βελτιώσεις και ενδεχόμενες μελλοντικές επεκτάσεις της εφαρμογής μας:

- ✓ Μία καλή πρόταση βελτίωσης του σεναρίου, είναι να αυξήσουμε την πολυπλοκότητά του και το επίπεδο δυσκολίας του. Αυτό μπορεί να γίνει με πολλούς τρόπους, όπως να εισάγουμε δυσκολότερες πράξεις. Πολλαπλά αθροίσματα, πολλαπλασιασμούς, αναγωγές στη μονάδα. Έτσι, θα είναι κατάλληλο και για μεγαλύτερες ηλικίες.
- ✓ Επίσης, απαραίτητη βελτίωση είναι να γίνει πιο κατανοητή η ομιλία του από τα παιδιά. Όπως έχουμε πει, τις ελληνικές λέξεις τις συνθέσαμε από το λατινικό αλφάβητο με αποτέλεσμα να μην είναι και πολύ κατανοητά τα ελληνικά. Το Nao Next Gen θα έχει τη δυνατότητα να μιλά 9 διαφορετικές γλώσσες, ενδεχομένως μία από αυτές να είναι και η ελληνική.
- ✓ Ας τολμήσουμε να προτείνουμε να τροφοδοτηθούν και τα σχολεία μας με τέτοια ρομποτικά συστήματα. Οι μεγαλύτερες ηλικίες (15-18) θα έχουν τη μοναδική ευκαιρία να γνωρίσουν, να έρθουν σε επαφή ακόμη και να προγραμματίσουν ένα ρομπότ, γεγονός που σίγουρα διευρύνει τις γνώσεις τους. Παράλληλα, οι μικρότερες ηλικίες θα μπουν στη διαδικασία της μάθησης με έναν τρόπο πιο ευχάριστο και

δημιουργικό. Ταυτόχρονα, παιδιά με μαθησιακές δυσκολίες, αυτισμό ([Ξ11]) και δυσκολίες κοινωνικοποίησης, σίγουρα θα σημειώσουν σημαντική βελτίωση.([Ε7])

- ✓ Ακόμη μία πρόταση ώστε να γίνει πιο ενδιαφέρουσα η αλληλεπίδραση με το ρομπότ είναι να το προγραμματίσουμε να κινείται περισσότερο. Μπορούμε για παράδειγμα σε μία αποστολή να ζητήσουμε από το παιδί να ακολουθήσει περπατώντας το Nao ώστε να βρουν μαζί το νόμισμα που χρειάζεται το παιδί.
- ✓ Το Nao έχει επίσης τη δυνατότητα να αναγνωρίσει πρόσωπα και αντικείμενα αρκεί να το «εκπαιδεύσουμε». Είναι ευκαιρία λοιπόν να μάθουμε στο παιδί τα σχήματα. Θα μπορούσαμε σε μία αποστολή να ζητήσουμε από ένα παιδί πολύ μικρής ηλικίας να επιλέξει π.χ. το τρίγωνο από ένα σύνολο σχημάτων, να το δείξει στο ρομπότ και να κρίνει το ρομπότ αν είναι το σωστό.
- ✓ Απ'ότι διαβάσαμε, το Arduino έχει πολλά Shields τα οποία του προσδίδουν ανέλπιστες δυνατότητες. Η αποστολή φωνητικής αναγνώρισης που απαιτεί να γνωρίζει το παιδί αγγλικά, μπορεί να αντικαταστασθεί με την παρακάτω. Το Arduino TouchShield Slide είναι ένα Touch Screen που συνδέεται με το Arduino. Εκεί μπορεί το παιδί να διαλέξει μία από τις τρεις απαντήσεις αγγίζοντας την οθόνη και να κερδίσει την αποστολή αν απαντήσει σωστά.
- ✓ Μία πρωτότυπη ιδέα είναι να χρησιμοποιήσουμε το Arduino Joystick Shield kit. Έστω ότι έχουμε τρία δωμάτια. Το παιδί πρέπει να καθοδηγήσει το Nao με το joystick στο σωστό δωμάτιο. Αν τα καταφέρει, κερδίζει την αποστολή.
- ✓ Ακόμη μία ιδέα που ξεφεύγει από τα πλαίσια της εκπαιδευτικής ρομποτικής είναι η χρήση του στην καθημερινότητα και συγκεκριμένα η χρήση του από άτομα τρίτης ηλικίας. Έστω ότι ένας ηλικιωμένος έχει τα χάπια του σε κάποιο συγκεκριμένο ντουλάπι και έχουμε συνδέσει στο ντουλάπι αυτό ένα Arduino με Arduino Ethernet Shield και αισθητήρες φωτός στα Digital inputs. Αν ο ηλικιωμένος ανοίξει το ντουλάπι την ώρα που πρέπει για να πάρει το χάπι του, ο αισθητήρας ανιχνεύει το φως και μέσω Ethernet ενημερώνει το Nao ότι ο άνθρωπος πήρε το χάπι του. Αλλιώς, πρέπει με τη βοήθεια της σύνθεσης φωνής να του υπενθυμίσει ότι είναι ώρα για το χάπι του.

9

Βιβλιογραφία

Ελληνική βιβλιογραφία

[E1] Tassos A. Mikropoulos and Joanna Bellou, Educational Robotics as Mindtools. The Educational Approaches to Virtual Reality Technologies Lab, University of Ioannina

[E2] Tassos A. Mikropoulos and Joanna Bellou, The unique features of educational virtual environments, University of Ioannina. International Conference e-Society 2006

[E3] V.Dagdilelis, M.Sartzemi, K.Kagani, Dept. of Educ. & Social Policy, Macedonia University, Thessaloniki, Greece: Teaching (with) Robots in Secondary Schools: Some new and not-so-new Pedagogical problems. Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference

[E4] Σολομωνίδου Χριστίνα: Εκπαιδευτική τεχνολογία:μέσα,υλικά,διδασκτική χρήση και αξιοποίηση, εκδόσεις Καστανιώτη 1999

[E5] Κώστας Καρπούζης, Ρομποτική και παιχνίδια σοβαρού σκοπού στην εκπαίδευση

[E6] Σολομωνίδου Χριστίνα: Νέες τάσεις στην εκπαιδευτική τεχνολογία: εποικοδομητισμός και σύγχρονα περιβάλλοντα μάθησης. Εκδόσεις Μεταίχμιο 2006

[E7] Σ.Αναγνωστάκης, Α.Μαργετουσάκη, Π.Γ.Μιχαηλίδης: Δυνατότητα εργαστηρίου εκπαιδευτικής ρομποτικής στα σχολεία. Εργασία από το Παιδαγωγικό τμήμα Δημοτικής Εκπαίδευσης Πανεπιστημίου Κρήτης

Ξένη βιβλιογραφία

[E1] Brigitte Denis and Sylviane Hubert, Collaborative learning in an educational robotics environment, Educational Technology Papers (2001)

[E2] David H.Jonassen: Computers as mindtools for schools:Engaging Critical Thinking. University of Michigan. Merrill 2000

- [E3] Marina U. Bers, Iris Ponte, Katherine Juelich, Alison Viera, Jonathan Schenker: Teachers as Designers: Integrating robotics in Early childhood education. Eliot-Pearson Department of Child Development Tufts University. Information Technology in Childhood Education Annual(2002)
- [E4] Ronald K.Thornton: Using the results of research in science education to improve science learning. Keynote address to the International Conference on Science Education, Nicosia, Cyprus (1999)
- [E5] Elizabeth Sklar(Columbia University), Amy Eguchi(University of Cambridge), Jeffrey Johnson(The Open University, Milton Keynes): RoboCupJunior:Learning with Educational Robotics, Berlin Heidelberg (2003)
- [E6] Brigitte Denis, Sylviane Hubert: Collaborative learning in an educational robotics environment. Elsevier Science Ltd 2001
- [E7] Hacker L.: Robotics in education: ROBOLAB and robotic technology as tools for learning science and engineering. Unpublished Senior Thesis, Department of Child Development, Tufts University (2003)
- [E8] Rachel Goldman (Columbia, New York), Amy Eguchi (University of Cambridge), Elizabeth Sklar (Columbia, New York): Using educational robotics to engage inner-city students with technology. ICLS 2004 Proceedings of the 6th international conference on Learning sciences
- [E9] Henrik Hautop Lund(University of Southern Denmark), Patrizia Marti, Valentina Palma(University of Siena): Educational Robotics: Manipulative Technologies for Cognitive Rehabilitation (2004). In Sugisaka and Tanaka(Eds.) Proceedings of 9th International Symposium on Artificial Life and Robotics.
- [E10] Karna-Lin E., Pihlainen-Bednarik K., Sutinen E., Virnes M.: Can robots teach? Preliminary Results on Educational Robotics in Special Education. Kerkrade. Advanced Learning Technologies, 2006. Sixth International Conference
- [E11] B.Robins, K.Dautenhamn, R. te Boekhorst, A.Billard: Effects of repeated exposure to a humanoid robot on children with autism. Cambridge Workshop Universal Access and Assistive Technology, Cambridge, UK, March 2004
- [E12] Werry I., Dautenhahn K.: Applying Mobile Robot Technology to the Rehabilitation of Autistic Children. 7th Symp on Intelligent Robotic Systems 1999
- [E13] B.Robins, K.Dautenhamn, R. te Boekhorst, A.Billard: Robotic assistants in therapy and education of children with autism:can a small humanoid robot help encourage social interaction skills? School of Computer Science, University of Hertfordshire(2005)

- [E14] Smaldino, Russel, Heinich, Molenda: Instructial Technology and Media for Learning. Pearson Education Company 2004
- [E15] Stan A.Napper, Romald L.Seaman: Applicatins of Robots in Rehabilitation. Department of Biomedical Engineering, Center for Rehabilitation Science and Biomedical Engineering Louisiana Tech University, Ruston, USA, (Feb 2003)
- [E16] Hideki Kozima (National Institute of Information and Communications Technology, Japan), Cocoro Nakagawa(National Institute of Information and Communications Technology, Japan), Yuriko Yasuda(Omihachiman City Day Care Center for Children with Special Needs, Japan): Children-robot interaction:a pilot study in autism therapy Clinical Trial Publications(Oct 2007)
- [E17] Andrew Meltzoff, Rechele Brooks, Aaron Shonb, Rajesh Raob:Social Robots are psychological agents for infants (Sept 2010)
- [E18] Michael McRoberts:Beginning Arduino. Apress publications.
- [E19] Jeonghye Han, Miheon Jo, Sungju Park, Sungho Kim: The educational use of home robots for children. Department of computer education. Robot-Human Interactive Communication, IEEE International Workshop,South Korea Aug 2005
- [E20] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, Hiroshi Ishigure: Interactive robots as social partners and peer tutors for children:a field trial. Journal Human-Computer Interaction NJ,USA 2004
- [E21] J.Gregory Trafton, Alan C.Schultz, Dennis Perznowski, Magdalena D. Bugajska, William Adams, Nicholas L.Cassimatis, Derek P. Brock: Children and robots learning to play hide and seek. HRI '06 Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, New York 2006
- [E22] Mihaly Csikszentmihalyi, Flow: The Psychology of Optimal Experience, London: Harper Perennial, 1990
- [E23] Baron-Cohen, Simon, Mindblindness: An Essay on Autism and Theory of Mind, A Bradford Book, 1995

Ηλεκτρονικές διευθύνσεις

- [H1] <http://www.gostai.com/>
- [H2] <http://arduino.cc/>
- [H3] <http://www.urbiforge.org/>
- [H4] Los Angeles Times:Robots Built to help autistic children. Chris Woolston (2011)
<http://articles.latimes.com/2011/oct/17/health/la-he-autism-robots-20111017>

[H5] Educational Robots That Teach You and Your Children Robotics

<http://www.robots-and-androids.com/educational-robots.html>

[H6] <http://www.aldebaran-robotics.com>

[H7] <http://users.aldebaran-robotics.com/>

[H8] Γιατί η ρομποτική στην εκπαίδευση;

<http://edurobotics.weebly.com/epsilonkappalphaiotaedeltaepsilonupsilontauiotakappa942-rhomicronmupiomicrontauiotakappa942.html>

[H9] <http://www.sparkfun.com>

Ευρετήριο εικόνων

Εικόνα 1	Kaspar robot	18
Εικόνα 2	P.BOT 100.....	19
Εικόνα 3	Nao Robot	30
Εικόνα 4	Arduino Duemilanove.....	31
Εικόνα 5	Διάγραμμα καταστάσεων όλων των αποστολών.....	36
Εικόνα 6	Το Choregraphe όταν έχει συνδεθεί με το Nao	38
Εικόνα 7	Το διάγραμμα καταστάσεων όπως απεικονίζεται στο Gostai Studio	39
Εικόνα 8	Το Nao όταν μας επιβραβεύει	44
Εικόνα 9	Τα μάτια του Nao όταν τα Led είναι μωβ	46
Εικόνα 10	Αναγνώριση ομιλίας	51
Εικόνα 11	Πίνακας αναφοράς στα μέλη του Nao	52
Εικόνα 12	Τα μέλη του ρομπότ μας	52
Εικόνα 13	Αναγνώριση της κόκκινης μπάλας από την κάμερα του Nao	57
Εικόνα 14	Το Nao «βλέπει» το κόκκινο Led	60
Εικόνα 15	Κυκλωματικό διάγραμμα της συνδεσμολογίας στην αποστολή με το Arduino με τη βοήθεια του προγράμματος Fritzing.....	62
Εικόνα 16	Τα συρταράκια που περιέχουν 1,2,3 ευρώ και στην πίσω πλευρά το Arduino	63
Εικόνα 17	Σύνδεση Arduino με τους διακόπτες στο εσωτερικό των συρταριών όπως δείχνει και το διάγραμμα	64
Εικόνα 18	Το Nao αναγνωρίζει το Naomark.....	67
Εικόνα 19	Οι δύο όψεις των νομισμάτων.....	68
Εικόνα 20	Πώς βλέπουμε μέσα από την κάμερα του Nao	71
Εικόνα 21	Στιγμιότυπο του ALMemory όπως απεικονίζεται από το Monitor	72
Εικόνα 22	Στιγμιότυπο πειραμάτων.....	79