



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

Ρομποτικά Εκπαιδευτικά Περιβάλλοντα για παιδιά

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης Α. Χατζηχριστοδούλου

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ
ΣΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

Ρομποτικά Εκπαιδευτικά Περιβάλλοντα για παιδιά

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης Α. Χατζηχριστοδούλου

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Ιουλίου 2011.

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

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

.....
Γιώργιος Στάμου
Λέκτορας Ε.Μ.Π.

Αθήνα, Ιούλιος 2011.

.....
Παναγιώτης Α. Χατζηχριστοδούλου
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

Copyright © Παναγιώτης Χατζηχριστοδούλου, Εθνικό Μετσόβιο Πολυτεχνείο, 2011.
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

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 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 Suite, 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 επιβεβαιώνεται μέσω των αισθητήρων του ρομπότ ή πρόσθετων αισθητήρων.

Λέξεις Κλειδιά

Εκπαιδευτική Ρομποτική, Θεωρία του Νου, Θεωρία ροής παιχνιδιών σοβαρού σκοπού, Ρομποτικά συστήματα, Σενάριο εκμάθησης, Nao, Gostai Studio, Arduino, Αλληλεπίδραση Ανθρώπου-Μηχανής

Abstract

The research interest for introducing the robotic systems in educational applications is on the rise, mainly due to the maturation of low cost robotic systems available, which have great processing power in comparison with the recent past, as they are also equipped with many sensors (camera, microphone, touch sensors) through which they interact with the user in a natural way. Furthermore, the relevant form of an anthropomorphic robot makes it ideal for interacting with children and thus the development of modern teaching materials which exploits concepts from psychology, such as Theory of Mind to assess the interest and commitment to under the gaze of the user and the Theory of Flow for serious games. Under the proposed thesis, we implemented a scenario learning literacy numeracy for preschool and primary school children. This scenario involves the robotic platform of Nao Aldebaran, a humanoid robot used predominantly from the laboratory in studies in a children environment. Using the development environment Gostai Studio Suite, the script will must involve the physical meaning of arithmetic operations with developments in the game: for example, the player will have specially designed "Coins" that carry a logo on them so that they can be visually identified and counted by the Nao - to allow the player to reach the next "level" a certain number of coins should be completed, by running "tasks" that will bring additional currencies and their performance will be confirmed through the robot's sensors or additional sensors.

Key Words

Educational Robotics, Theory of Mind, Flow Theory of serious games, Robotic systems, Learning Scenario, Nao, Gostai Studio, Arduino, Human - Computer Interaction

Ευχαριστίες

Ευχαριστώ την οικογένεια, και κυρίως τους γονείς μου, που βρίσκονταν πάντα στο πλάι μου σε όλη τη διάρκεια της ακαδημαϊκής μου πορείας στο Εθνικό Μετσόβιο Πολυτεχνείο, τον κ. Καρπούζη, που ήταν η βασική πηγή βοήθειας στην εκπόνηση αυτής της διπλωματικής αλλά και μια συνεχής πηγή στήριξης και έμπνευσης για μένα, και όλα τα μέλη του Εργαστηρίου Ψηφιακής Επεξεργασίας Σημάτων και Εικόνων του Ε.Μ.Π. που πάντα έλυναν κάθε είδους απορία που εμφανιζόταν.

Περιεχόμενα

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Περιεχόμενα	12
1 Εισαγωγή	13
1.1 Σκοπός	13
1.2 Διαρρύθμιση Κεφαλαίων	14
2 Γενικά	17
2.1 Εκπαιδευτική Τεχνολογία	17
2.2 Ρομποτικά Συστήματα	18
2.3 Εκπαιδευτική Ρομποτική	18
2.4 Θεωρία του Νου (Theory of Mind)	21
2.5 Παρακολούθηση του Βλέμματος	22
2.6 Θεωρία ροής παιχνιδιών σοβαρού σκοπού	23
2.7 Ανάπτυξη Σεναρίων με Ρομπότ	23
2.8 Gostai Suite	24
2.9 Ρομποτική Πλατφόρμα Nao	25
2.10 Arduino	26
3 Σενάριο	27
3.1 Η πρώτη ιδέα	27
3.2 Αναλυτική Περιγραφή του Σεναρίου	27
3.2.1 Περιγραφή αποστολών	28
3.3 Διαγραμματική παρουσίαση του τελικού σεναρίου	30
3.4 Παραδοχές	32
4 Υλοποίηση σεναρίου στο Gostai Studio	35
4.1 Ανάλυση λειτουργιών προγράμματος	35
4.2 Ανάλυση εντολών	36
4.2.1 Ανάλυση βασικών εντολών	36
4.2.2 Ανάλυση εντολών ρομποτικού πακέτου	38
4.2.3 Ανάλυση οπτικής αναγνώρισης Naomark	41
4.3 Παρουσίαση της υλοποίησης	43
4.4 Παραδείγματα χρήσης	56

5	Αποστολές με Arduino	59
5.1	Ανάλυση των Αποστολών	59
5.2	Υλοποίηση Αποστολών	60
5.2.1	Κύκλωμα παραγωγής φωνής	60
5.2.2	3η Αποστολή	63
5.2.3	4η Αποστολή	65
6	Υλοποίηση του σεναρίου στη ρομποτική πλατφόρμα Nao	69
6.1	Προσομοίωση στο Choregraphe	69
6.2	Διασύνδεση με το Nao	70
6.3	Ειδικά Νομίσματα	71
6.3.1	Κατασκευή Ειδικών Νομισμάτων	71
6.3.2	Προσομοίωση οπτικής αναγνώρισης στο Telepathe	73
6.4	Πειράματα	73
6.5	Αποτελέσματα	74
7	Προβλήματα - Βελτιώσεις	77
7.1	Προβλήματα	77
7.2	Βελτιώσεις	78
8	Αξιολόγηση - Συμπεράσματα	81
8.1	Γενική Αξιολόγηση - Συμπεράσματα	81
8.2	Η δική μας συνεισφορά	83
	Βιβλιογραφία	85

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός

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 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 Suite, 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 επιβεβαιώνεται μέσω των αισθητήρων του ρομπότ ή πρόσθετων (σε συνεργασία με την πλακέτα μικροελεγκτή ανοιχτού κώδικα Arduino).

Επομένως, θα υλοποιήσουμε ένα σενάριο με τη χρήση του Gostai Studio το οποίο θα εφαρμόσουμε στη ρομποτική πλατφόρμα Nao, με την παράλληλη βοήθεια της πλακέτας Arduino, και μέσα από αυτό το σενάριο, θα προσπαθήσουμε να αναδείξουμε τα ακόλουθα:

- Εισαγωγή των ρομποτικών συστημάτων σε εκπαιδευτικές εφαρμογές: μέσα από την υλοποίηση αυτού του σεναρίου, θα δείξουμε πώς μπορούν να χρησιμοποιηθούν ρομποτικά συστήματα στην διαδικασία εκπαιδευτικής μάθησης παιδιών νεαρής ηλικίας. Με τον τρόπο αυτό, η διπλωματική αυτή εργασία μπορεί να χρησιμοποιηθεί ως εφαλτήριο για την ανάπτυξη και άλλων παρομοίων εφαρμογών αλληλεπίδρασης των παιδιών με τα ρομποτικά συστήματα, που θα βοηθήσει την εκπαιδευτική τους μάθηση, αλλά παράλληλα θα τους εντάξει από μικρή ηλικία στις νέες πραγματικότητες των ρομποτικών συστημάτων που αποτελούν το μέλλον της τεχνολογίας.
- Χρήση της νέας κονσόλας Gostai Studio: για την υλοποίηση αυτού του σεναρίου, όπως προαναφέραμε, θα χρησιμοποιηθεί η κονσόλα Gostai Studio. Η κονσόλα αυτή αποτελεί ένα εντελώς νέο και πρωτοποριακό τρόπο προγραμματισμού ρομποτικών πλατφόρμων με τη χρήση της πλατ-

φόρμας ανοιχτού κώδικα URBI. Ο προγραμματισμός, όπως θα δούμε στη συνέχεια, θα γίνει με τη χρήση διαγραμμάτων κατάστασης. Κάτι εντελώς καινούριο στον τομέα της ρομποτικής. Η διπλωματική εργασία που θα παρουσιάσουμε, είναι η πρώτη στο Ε.Μ.Π. που ασχολείται με τη συγκεκριμένη πλατφόρμα, οπότε μπορεί να χρησιμοποιηθεί στις μεταγενέστερες έρευνες για την εισαγωγή της κονσόλας αυτής στις διαδικασίες προγραμματισμού των διαφόρων ρομποτικών συστημάτων, κάτι που θα διευκολύνει κατά πολύ τις διαδικασίες αυτές.

- Χρήση της πλατφόρμας ανοιχτού κώδικα Arduino: με τη βοήθεια της συγκεκριμένης πλακέτας, θα υλοποιήσουμε κάποιες από τις αποστολές που πρέπει το παιδί να κάνει για να πάρει τα νομίσματα που χρειάζεται. Επομένως, θα δείξουμε πώς μπορεί να εισαχθεί το Arduino σε διάφορες ερευνητικές εργασίες.

1.2 Διαρρύθμιση Κεφαλαίων

Τα κεφάλαια της διπλωματικής εργασίας, θα έχουν την ακόλουθη διαρρύθμιση:

Κεφάλαιο 2: Γίνεται μια γενικότερη ανάλυση και περιγραφή των εννοιών που θα χρησιμοποιήσουμε ως βάση της διπλωματικής εργασίας. Αναλύονται εκτενώς οι έννοιες της *Εκπαιδευτικής τεχνολογία*, των *Ρομποτικών Συστημάτων*, της *Εκπαιδευτικής Ρομποτικής*, της *Ψυχολογίας - Θεωρίας του Νου*, της *Θεωρίας ροής παιχνιδιών σοβαρού σκοπού*, της *Παρακολούθησης του Βλέμματος*, της *Ανάπτυξης Σεναρίων με τη χρήση ρομπότ*, του *Gostai Console*, της *Ρομποτικής Πλατφόρμας Nao* και του *Arduino*.

Κεφάλαιο 3: Γίνεται παρουσίαση του σεναρίου που θα υλοποιήσουμε. Αρχικά, παρουσιάζεται η πρώτη ιδέα που δημιουργήσαμε και έπειτα, μετά την αντιμετώπιση των διαφόρων προβλημάτων που προέκυψαν στην αρχική ιδέα, γίνεται εκτενής περιγραφή του τελικού σεναρίου το οποίο και θα υλοποιηθεί.

Κεφάλαιο 4: Αναλύεται εκτενώς η λειτουργία της κονσόλας Gostai Studio. Περιγράφονται οι λειτουργίες που θα χρησιμοποιήσουμε και οι εντολές της συγκεκριμένης κονσόλας όπως και το πώς αντιμετωπίζει τα διάφορα ρομπότ. Τέλος, παρουσιάζεται η υλοποίηση που κάναμε του σεναρίου μας στο Gostai Studio, όπως αυτό είχε γίνει πριν τη δοκιμή σε πραγματικό χρόνο μέσω της ρομποτικής πλατφόρμας του Nao.

Κεφάλαιο 5: Περιγράφονται οι αποστολές που πρέπει να εκτελέσει το παιδί ούτως ώστε να μπορέσει να συγκεντρώσει τον απαραίτητο αριθμό νομισμάτων για να περάσει στο επόμενο “επίπεδο” και αφορούν τη συνεργασία με το Arduino. Αναλύεται η χρήση της κονσόλας ελεύθερου λογισμικού Arduino. Περιγράφονται τα ηλεκτρονικά κυκλώματα που χρησιμοποιήσαμε όπως και ο κώδικας που γράψαμε στο Arduino αλλά και ο τρόπος που επιλέξαμε να γίνεται η επικοινωνία μεταξύ του Arduino και το Gostai Studio και άρα κατ’ επέκταση και του Nao.

Κεφάλαιο 6: Σε αυτό το κεφάλαιο αναφερόμαστε στην πρακτική υλοποίηση του σεναρίου στη ρομποτική πλατφόρμα του Nao. Κάνουμε μια σύντομη αναφορά στην αρχή σε μια προσομοίωση που χρησιμοποιήσαμε και έπειτα αναφέρουμε τη διαδικασία σύνδεσης με το Nao. Έπειτα, γίνεται εκτενής παρουσίαση των ειδικών νομισμάτων που κατασκευάσαμε και χρησιμοποιήσαμε. Στη συνέχεια, αναφέρουμε τα πειράματα που μπορούν να γίνουν με την τελική υλοποίηση του σεναρίου μας ενώ τέλος γίνεται μια λεπτομερής αναφορά των αποτελεσμάτων μας από την πρακτική εφαρμογή του σεναρίου μας σε πραγματικό χρόνο στο Nao.

Κεφάλαιο 7: Γίνεται μια εκτενής αναφορά στα προβλήματα που υπήρξαν κατά την υλοποίηση του σεναρίου στην πράξη τα οποία κληθήκαμε να αντιμετωπίσουμε. Αναφέρονται επίσης οι λύσεις που δώσαμε στα προβλήματα αυτά, ενώ στο τέλος γίνεται μια αναφορά στις βελτιώσεις που μπορούσαν

να γίνουν ώστε να τελειοποιηθεί το σενάριό μας.

Κεφάλαιο 8: Στο τελευταίο κεφάλαιο γίνεται μια γενικότερη αξιολόγηση της διπλωματικής εργασίας. Αναλύουμε εκτενώς τα αποτελέσματα που πήραμε από τα πειράματά μας και αξιολογούμε τα αποτελέσματα αυτά σε σχέση με τον αρχικό μας στόχο. Αξιολογείται, επίσης, και η χρήση της νέας πλατφόρμας Gostai Studio όπως και του Arduino. Τέλος, γίνεται ιδιαίτερη μνεία στα καινούρια πράγματα που έχουν αναδειχθεί μέσα από τη διπλωματική μας εργασία.

Κεφάλαιο 2

Γενικά

2.1 Εκπαιδευτική Τεχνολογία

Η Εκπαιδευτική Τεχνολογία, είναι η επιστήμη που, εφαρμόζοντας την τεχνολογική γνώση και μια ποικιλία από τεχνολογικά προϊόντα, μελετά και συμβάλλει στη συστηματική επίλυση προβλημάτων που αφορούν τη διδασκαλία και τη μάθηση με στόχο τη βελτίωσή τους. Ο αντίστοιχος αγγλικός όρος, είναι το "Instructional Technology", που με βάση την Επιτροπή Ορισμών και Ορολογίας του Συλλόγου Εκπαιδευτικών Επικοινωνιών και Τεχνολογίας (AECT), αναφέρεται στη θεωρία και εφαρμογή του σχεδιασμού, της ανάπτυξης, της χρησιμοποίησης, της διαχείρισης και αξιολόγησης όλων των διαθέσιμων διεργασιών και πόρων που αφορούν τη μάθηση¹. Στο ανώτερο επίπεδο της εκπαιδευτικής τεχνολογίας, περιλαμβάνονται οι όροι οπτικοακουστική διδασκαλία, η εκπαιδευτική ανάπτυξη/σχεδιασμός, τα επιμορφωτικά υλικά και η εξατομικευμένη μάθηση.

Η Εκπαιδευτική Τεχνολογία δεν είναι μια έννοια εντελώς καινούρια στον τομέα της εκπαίδευσης. Από ανέκαθεν, οι διδάσκοντες χρησιμοποιούσαν διάφορα διαθέσιμα οπτικοακουστικά μέσα στη διδασκαλία τους με σκοπό να διευκολύνουν το έργο τους. Η εξέλιξη της τεχνολογίας τις τελευταίες δεκαετίες όμως, έφερε την εδραίωση της έννοιας αυτής και την ανάπτυξή της ολοένα και περισσότερο. Τα εξελιγμένα υπολογιστικά συστήματα, το διαδίκτυο, τα μεγάλα αποθηκευτικά μέσα, τα ρομποτικά συστήματα και πολλά άλλα επιτεύγματα της τεχνολογίας, χρησιμοποιούνται πλέον κατά κόρον στην εκπαιδευτική μάθηση. Οι διδάσκοντες, εμπιστεύονται όλο και περισσότερο τα νέα τεχνολογικά επιτεύγματα, ενώ στον αντίποδα τα ίδια τα παιδιά αναζητούν το συνδυασμό της εκπαίδευσης με την τεχνολογία.

Παρόλα αυτά, επικρατεί η άποψη ότι η ένταξη της τεχνολογίας στην εκπαιδευτική μάθηση δεν αποτελεί αυτοσκοπό ούτε πανάκεια για την επίλυση όλων των προβλημάτων που ταλανίζουν την εκπαίδευση σήμερα. Υπάρχουν, όμως, κάποιες προϋποθέσεις, οι οποίες αν τηρηθούν στο έπακρο, τότε θα συντελέσουν την ομαλή προσαρμογή της τεχνολογίας στη διαδικασία της εκπαιδευτικής μάθησης, ενώ παράλληλα θα αποκομίσουμε τα μέγιστα από αυτό το συνδυασμό. Οι προϋποθέσεις αυτές είναι:

- Τα οποιαδήποτε ευφυή συστήματα που χρησιμοποιούνται στη διαδικασία της μάθησης, πρέπει να χρησιμοποιούνται ως εργαλεία σκέψης και αντίληψης.
- Η συνεχής κατάρτιση των διδασκόντων είναι απαραίτητη για να συμβαδίζουν με την πραγματικότητα
- Συνεχής αξιολόγηση και βελτίωση των εκπαιδευτικών τεχνολογιών που χρησιμοποιούνται

Όπως έχουμε αντιληφθεί, η εκπαιδευτική τεχνολογία έχει γίνει πλέον σύγχρονη πραγματικότητα. Τα οπτικοακουστικά μέσα υποβοηθούν το έργο του διδάσκοντα, διευκολύνουν την διεκπεραίωση

¹http://en.wikipedia.org/wiki/Instructional_technology

του μαθήματος και μεταβάλλουν τη διδασκαλία σε εργασία ευχάριστη και συνάμα αποδοτική. Βέβαια, για να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα, αναμένεται η ευρύτερη χρήση τους και η ενσωμάτωσή τους, σε καθημερινή βάση, στο σχολικό πρόγραμμα. Παράλληλα, απαιτείται και η επιμελής επιμόρφωση των διδασκόντων σε αυτά, τόσο σε θεωρητική όσο και σε πρακτική βάση. Επομένως, όπως μας έχει δείξει το πρόσφατο παρελθόν, το μέλλον είναι άμεσα συνυφασμένο με την Εκπαιδευτική Τεχνολογία αφού κάθε νέα τεχνολογία που θα αναδεικνύεται θα παίρνει τη θέση της σιγά-σιγά μέσα στο εκπαιδευτικό μας σύστημα.

2.2 Ρομποτικά Συστήματα

Για να κατανοήσουμε την έννοια του Ρομπότ, ας κοιτάξουμε την ετυμολογία της λέξης σε διάφορες γλώσσες: στα Τσέχικα, "robota" σημαίνει αμισθί - εξαναγκασμένη εργασία, στα Σλάβικα, "rabu" σημαίνει σκλάβος ενώ στα Γερμανικά, "arbeit" σημαίνει εργασία ². Επομένως, η ετυμολογία αυτή, μας δείχνει ότι το ρομπότ είναι μια μηχανή με σκοπό να εξυπηρετεί τον άνθρωπο. Γι' αυτό, διέπεται από τους 3 νόμους της Ρομποτικής, οι οποίοι θεσπίστηκαν από τον Isaac Asimov: ³

1. Ένα ρομπότ δεν μπορεί να τραυματίσει ένα ανθρώπινο ον, ή λόγω αδράνειας να επιτρέψει σε ένα ανθρώπινο ον να πάθει κακό.
2. Ένα ρομπότ πρέπει να υπακούει οποιεσδήποτε εντολές του δίνονται από ανθρώπινα όντα, εκτός και αν τέτοιες εντολές θα έρθουν σε σύγκρουση με τον πρώτο νόμο.
3. Ένα ρομπότ πρέπει να προστατεύει τη δική του ύπαρξη, όσο όμως αυτή η προστασία δεν έρχεται σε σύγκρουση με τους 2 πρώτους νόμους.

Ρομποτικό σύστημα, λοιπόν, είναι ένα μηχανικό σύστημα, το οποίο αισθάνεται σκέφτεται και επενεργεί. Και αυτά τα κάνει γιατί διαθέτει αισθητήρες, μέσω των οποίων αποκτά πληροφορία για το εξωτερικό περιβάλλον αλλά και σε σχέση με την εσωτερική του κατάσταση, δυνατότητες υπολογιστικής επεξεργασίας, που του επιτρέπουν να σκέφτεται και να λαμβάνει αποφάσεις, και επενεργητές (actuators), μέσω των οποίων επενεργεί με το εξωτερικό του περιβάλλον.

Με τη ραγδαία ανάπτυξη του κλάδου της Ρομποτικής, έχουν δημιουργηθεί πολλές κατηγορίες Ρομποτικών Συστημάτων. Μερικές από αυτές είναι οι ακόλουθες:

- Βιομηχανικοί ρομποτικοί χειριστές που χρησιμοποιούνται σε βαριές βιομηχανίες για την εκτέλεση διαφόρων εργασιών
- Αυτόνομα αυτοκινούμενα ρομπότ τα οποία χρησιμοποιούνται για παροχή υπηρεσιών αλλά και για ερευνητικούς σκοπούς
- Μικρο-ρομποτικά συστήματα για πιο επιδέξιες και μικρές εργασίες
- Ανθρωποειδή-ανθρωπόμορφα ρομπότ τα οποία έχουν εμφάνιση βασισμένη στο ανθρώπινο σώμα

Εμείς θα ασχοληθούμε με τα ανθρωπόμορφα και αυτόνομα ρομποτικά συστήματα, τα οποία έχουν αναπτυχθεί πάρα πολύ τα τελευταία χρόνια. Αυτό το γεγονός είναι κάτι στο οποίο θα βασιστούμε, αφού είναι πολύ σημαντικό το παιδί να αλληλεπιδρά με ένα ρομπότ που να του θυμίζει άνθρωπο, οπότε να του είναι πιο οικείο και πιο εύκολο στη μάθηση.

2.3 Εκπαιδευτική Ρομποτική

Τα Ρομποτικά Συστήματα, έχουν εισαχθεί στην Τριτοβάθμια εκπαίδευση ως μαθήματα διδασκαλίας εδώ και αρκετές δεκαετίες. Η ένταξή τους όμως στην πρωτοβάθμια και δευτεροβάθμια εκπαίδευση,

²<http://users.softlab.ece.ntua.gr/~ktzaf/Courses/robotics-I-shmmy-kinematics-1.pdf>

³<http://en.wikipedia.org/wiki/Robot>

άρχισε να γίνεται δειλά-δειλά τα τελευταία χρόνια, ενώ πλέον στις μέρες μας έχει εδραιωθεί για τα καλά, με τον όρο Εκπαιδευτική Ρομποτική, όπου δηλαδή η χρήση των ρομποτικών συστημάτων, υποβοηθά την εκπαιδευτική μάθηση των παιδιών. Οι λόγοι που συντέλεσαν στην ένταξη της ρομποτικής στα παιδιά του δημοτικού αλλά και του γυμνασίου είναι οι ακόλουθοι:

- Το κόστος ενός ρομποτικού συστήματος έχει μειωθεί πάρα πολύ σε τέτοιο βαθμό που να είναι προσιτό για τον καθένα από εμάς. Πλέον, έτοιμα κατασκευαστικά και προγραμματιστικά ρομποτικά πακέτα διατίθενται στην αγορά. Ένα από αυτά είναι και το LEGO MINDSTORMS™, το οποίο πλέον χρησιμοποιείται σε περισσότερα από 25 000 ιδρύματα ανά το παγκόσμιο, από Δημοτικά μέχρι και Πανεπιστήμια ⁴.
- Η επεξεργαστική ισχύς των ρομποτικών συστημάτων έχει αυξηθεί σε εξαιρετικά μεγάλο βαθμό, παρά τη μείωση του κόστους, ενώ η ύπαρξη πολλών και διαφόρων αισθητήρων επιτρέπει την αλληλεπίδραση με τον άνθρωπο με τρόπο πιο φυσικό από ότι τα παλαιότερα χρόνια.
- Η ανάπτυξη νέων εργαλείων λογισμικού, πιο απλών και πιο προσιτών στον απλό χρήστη, ακόμη και σε ένα μικρό παιδί, βοήθησαν στο να εισαχθεί η ρομποτική στα σχολεία.
- Η ανάγκη και η απαίτηση των ίδιων των παιδιών, μικρών και μεγάλων, να ασχοληθούν με την τεχνολογία που βλέπουν γύρω τους, οδήγησαν τους διδάσκοντες να εισάξουν τα ρομποτικά συστήματα στη διδασκαλία τους ως μέσο για να κεντρίσει το ενδιαφέρον του παιδιού.
- Πολλοί διεθνείς οργανισμοί, όπως η Ευρωπαϊκή Ένωση και η UNESCO, ενθαρρύνουν την ένταξη της τεχνολογίας στην εκπαίδευση επιχορηγώντας μάλιστα σχετικά προγράμματα, ενώ παράλληλα θεωρούν ότι ο Τεχνολογικός Αλφαριθμητισμός, είναι "δικαίωμα στη δημοκρατία" ⁵.
- Πολλά από τα νέα ρομποτικά συστήματα, είναι πλέον ανθρωπόμορφα. Αυτό επιτρέπει, κυρίως στα παιδιά προσχολικής και πρώτης σχολικής ηλικίας, να εξοικειωθούν με το ρομπότ και να δεχθούν ευκολότερα την αλληλεπίδραση με το ρομποτικό σύστημα και κατ' επέκταση τη μάθηση μέσω αυτού.

Η ρομποτική στην εκπαίδευση και κυρίως στα μικρά παιδιά, βοηθάει στη διδασκαλία διαφόρων εννοιών, όπως είναι τα Μαθηματικά, η Φυσική, ο Προγραμματισμός και η χρήση Η/Υ, ο Σχεδιασμός, η Τεχνολογία αλλά ακόμη και η Ιστορία, αφού για την δημιουργία μιας ρομποτικής κατασκευής, μπορεί να τους δοθεί το ιστορικό υπόβαθρο εκείνης της μηχανικής κατασκευής. Πέρα από τη γνώση των μαθημάτων, η Εκπαιδευτική Ρομποτική, αποσκοπεί και στην ανάπτυξη άλλων δεξιοτήτων των παιδιών, όπως η ομαδικότητα, η καινοτομία, η διαχείριση ενός έργου και άλλα πολλά.

Η συμβολή της ρομποτικής στην εκπαίδευση στις μέρες μας, στηρίζεται σε μεγάλο βαθμό στο γεγονός ότι τα ανθρωπόμορφα ρομπότ γίνονται όλο και περισσότερο μέρος της ζωής μας. Τα παιδιά, συγκεκριμένα, όπως έχει αποδειχθεί ⁶, τη σήμερον ημέρα με την επαφή που έχουν με τα ανθρωπόμορφα ρομπότ, τα κατανοούν πολύ περισσότερο από παλαιότερες γενιές. Μάλιστα, τα θεωρούν ως κοινωνικά όντα τα οποία μοιάζουν πάρα πολύ στο ανθρώπινο είδος, αφού πλέον τα παιδιά μέσα από τη φαντασία τους δημιουργούν εικόνες για τις σχέσεις και τα συναισθήματα, τις επιθυμίες τους στόχους και τα κίνητρα των ρομπότ. Επιπλέον, τα παιδιά επειδή έχουν την προδιάθεση και την περιέργεια να εξερευνούν και να αλληλεπιδρούν με νέα πράγματα, όπως τα ρομπότ, αναμένουν από το ρομπότ να παίξει μαζί τους με διάφορους τρόπους. Γι' αυτό και πλέον τα σύγχρονα ρομπότ, οφείλουν να είναι πολυλειτουργικά και ευέλικτα.

Οι δραστηριότητες της Εκπαιδευτικής Ρομποτικής, μπορούν να αναλυθούν και να χαρακτηρισθούν από τα ακόλουθα 2 μοντέλα:⁷

⁴<http://www.active-robots.com/products/mindstorms4schools/nxt-education/active-lme.pdf>

⁵http://www.ecedu.upatras.gr/didinfo/eishghseis_DIDINF008/DIDINF008_243_252.pdf

⁶R. Severson, S. Carlson, *Behaving as or behaving as if? Children's conceptions of personified robots and the emergence of a new ontological category*

⁷B. Denis, S. Hubert, *Collaborative learning in an educational robotics environment*

1. Η αρχιτεκτονική ικανοτήτων 4 επιπέδων. Σε αυτή την αρχιτεκτονική διακρίνονται 4 επίπεδα ικανοτήτων:

- *Δυναμικές ικανότητες:* Οι δυναμικές ικανότητες συσχετίζονται άμεσα με το κίνητρο. Αυτό το επίπεδο είναι το πιο ευαίσθητο αφού μπορεί πιο εύκολα να επηρεαστεί. Οι ικανότητες αυτές ανταποκρίνονται στην πρωτοβουλία, τη θέληση, την ευχαρίστηση και το κίνητρο του εκπαιδευόμενου. Επομένως η εκπαιδευτική ρομποτική μέσα από αυτές στοχεύει στη δημιουργία μιας εργασίας με νόημα.
- *Στρατηγικές ικανότητες:* Οι ικανότητες αυτές έχουν άμεσο σχέση με τη μετα-γνώση, τη γνώση δηλαδή του εαυτού μας, των δυνατοτήτων μας, των αδυναμιών μας. Μέσα από την εκπαιδευτική ρομποτική, στόχος αυτών των ικανοτήτων είναι η κοινωνικοποίηση και η επίλυση προβλημάτων.
- *Αποπολλαπλασιαστικές ικανότητες:* Οι ικανότητες αυτές, επιτρέπουν στο χρήστη να βρει πληροφορίες από μόνος του και να αναπτύξει συγκεκριμένες ικανότητες, όπως το διάβασμα, την ακοή, την επικοινωνία, το να κρατά σημειώσεις και άλλες παρόμοιες ικανότητες.
- *Συγκεκριμένες ικανότητες:* Οι συγκεκριμένες ικανότητες αφορούν συγκεκριμένα περιεχόμενα (ιστορία, γεωγραφία, φυσική, λεξιλόγιο). Μέσα από την εκπαιδευτική ρομποτική, οι ικανότητες αυτές στοχεύουν στο να αναπτύξουν τις ικανότητες του ατόμου όσον αφορά τους υπολογιστές, τον προγραμματισμό, τα ηλεκτρονικά, τα ηλεκτρικά κυκλώματα και άλλα.

2. Τα 6 μαθησιακά/διδασκτικά παραδείγματα. Αυτά τα παραδείγματα μας δείχνουν διαφορετικά είδη δραστηριοτήτων μεταξύ του εκπαιδευτή και του εκπαιδευόμενου:

- *Δημιουργία:* Στη δημιουργία ανήκουν η κατασκευή ενός ρομπότ και ο προγραμματισμός του μέσω μιας γλώσσας προγραμματισμού.
- *Εξερεύνηση:* Στην κατηγορία της εξερεύνησης ανήκει η ανακάλυψη κάποιων εννοιών από τον εκπαιδευόμενο, όπως η διαδικτυακή βοήθεια, οι οδηγοί αναφορών και άλλα.
- *Πειραματισμός:* Ο πειραματισμός του μαθητή, έγκειται στην ελευθερία του και την ευελιξία που έχει στην κατασκευή ενός ρομποτικού συστήματος αλλά και στη διαχείριση των ενεργειών του.
- *Μίμηση:* Ο μαθητής, μπορεί να μιμηθεί πράγματα που μπορεί να διαβάσει ή να δει και να τα αντιγράψει στο δικό του έργο.
- *Υποδοχή:* Ο εκπαιδευόμενος μαθαίνει να δέχεται νέα πράγματα και να μαθαίνει καινούριες ιδέες.
- *Εξάσκηση:* Μέσα από την εκπαιδευτική ρομποτική και την ικανότητα των εκπαιδευομένων να επαναλάβουν διαδικασίες, εξασκούνται σε αυτό που τους ενδιαφέρει με αποτέλεσμα να γίνονται ολοένα και καλύτεροι.

Υπάρχουν διάφορα παραδείγματα χρήσης της ρομποτικής στην εκπαίδευση. Μερικά παραδείγματα είναι τα ακόλουθα:⁸

- Σε μια εκπαιδευτική εφαρμογή, ένας μικρός ρομποτικός βραχίονας χρησιμοποιήθηκε για την προώθηση της αλληλεπίδρασης των μικρών παιδιών με το περιβάλλον μέσω της αύξησης του ελέγχου που είχαν παιδιά με ειδικές ικανότητες πάνω σε αντικείμενα και κοινωνικές εκδηλώσεις.
- Μια άλλη μελέτη εξέτασε τις δυνατότητες των ρομπότ ως προσθήκη στη γνωστική μάθηση στους τομείς των χωρικών σχέσεων και του περιβαλλοντικού ελέγχου.

⁸S.Napper, R.Seaman, *Applications of Robots in Rehabilitation*

- Σε μια άλλη εφαρμογή, χρησιμοποιήθηκε μια διαδραστική ρομποτική συσκευή για την κλινική αξιολόγηση παιδιών προσχολικής ηλικίας. Το σύστημα αυτό χρησιμοποιήθηκε για την παροχή απόκρισης και ενδεχόμενης διέγερσης για την ενίσχυση της ενεργητικότητας, της εξερεύνησης και της χειραγώγησης των παιδιών.
- Ένα ακόμη παράδειγμα είναι η χρήση ενός ρομποτικού συστήματος για τη διερεύνηση των ρομπότ στην ειδική εκπαίδευση μέσα από την αξιολόγηση των αισθητήρων και του συστήματος όρασης που διαθέτει, με στόχο τη μείωση της γνωστικής επιβάρυνσης του χρήστη του ρομπότ.
- Μια ακόμη πολύ σημαντική χρήση της ρομποτικής στην εκπαίδευση, είναι και η συμβολή της ρομποτικής στην αντιμετώπιση προβλημάτων αυτισμού σε παιδιά νεαρής ηλικίας. Διάφορες έρευνες και μελέτες έγιναν σε αυτό το θέμα με κατάλληλα ρομπότ για τη μελέτη της συμπεριφορά αυτιστικών παιδιών⁹. Οι έρευνες αυτές, έδειξαν ότι τα αυτιστικά παιδιά τα οποία αντιμετωπίζουν δυσκολίες στην προσέγγιση και στη διαπροσωπική επικοινωνία, δημιουργήσαν σταδιακά φυσική και ψυχολογική σχέση με το ρομπότ και μοιράστηκαν τα συναισθήματά τους. Αυτό χρησιμοποιήθηκε από τους γιατρούς ως ένα μέτρο θεραπευτικής αγωγής για τα παιδιά αυτά.

Συνεπώς, η Εκπαιδευτική Ρομποτική, θεωρείται ότι αλλάζει την παραδοσιακή μάθηση προς το καλύτερο. Τα παιδιά, συνδυάζουν τη μάθηση με το παιχνίδι, οπότε η μάθηση επιτυγχάνεται ευκολότερα, ουσιαστικότερα και γρηγορότερα. Δίνονται παράλληλα κίνητρα στους μαθητές για να μελετήσουν και να ασχοληθούν με την επιστήμη και την τεχνολογία. Γι' αυτό και τη σήμερον ημέρα, βλέπουμε ολοένα και περισσότερους καθηγητές να εντάσσουν στο πρόγραμμά διδασκαλία τους τη χρήση κάποιου ρομποτικού συστήματος για να εξελίσσουν τη διαδικασία της μάθησης και να την κάνουν συνάμα πιο ευχάριστη.

2.4 Θεωρία του Νου (Theory of Mind)

Η Θεωρία του Νου, είναι η ικανότητα κάποιου να αποδίδει πνευματικές καταστάσεις - πεποιθήσεις, προθέσεις, επιθυμίες, προσποίηση, γνώση - για τον εαυτό του και τους άλλους, και να κατανοεί ότι οι άλλοι έχουν πεποιθήσεις, προθέσεις και επιθυμίες διαφορετικές από τη δική του.¹⁰ Για να το καταλάβουμε καλύτερα, θα δώσουμε ένα πείραμα της θεωρίας του Νου στα παιδιά:

Ένα παιδί κάθεται στο σαλόνι του σπιτιού του και λέει ιστορίες στη μικρότερη του αδελφή. Της λέει μια ιστορία, όπου η Sally έχει μια μπάλα, η οποία της αρέσει πολύ, και θα την κρύψει στο καλάθι που βρίσκεται μπροστά της. Μαζί με τη Sally, βρίσκεται η φίλη της η Ann με την οποία παίζουν μαζί. Σε μια στιγμή, η Sally φεύγει από το δωμάτιο, οπότε η Ann παίρνει τη μπάλα της Sally από το καλάθι και την βάζει σε ένα πράσινο κοντί που βρίσκεται δίπλα της. Η Sally επιστρέφει στο δωμάτιο και θέλει να παίξει με τη μπάλα. Τότε η αφηγήτρια, ρωτάει τη μικρή της αδερφή, πού θα βάξει η Sally για να βρει την κρυμμένη της μπάλα. Αν η αδερφή της αφηγήτριας είναι ένα παιδί τριών χρονών, τότε θα απαντήσει "στο πράσινο κοντί" γιατί γνωρίζει ότι εκεί βρίσκεται η μπάλα, αλλά δεν μπορεί να φανταστεί ότι υπάρχουν άτομα με διαφορετική γνώση από τη δική της. Αν η αδερφή της αφηγήτριας είναι 4-5 χρονών, θα απαντήσει "θα κοιτάξει στο καλάθι γιατί εκεί νομίζει ότι είναι". Ενώ ένα παιδί 5-6 χρονών θα απαντήσει "θα κοιτάξει στο καλάθι γιατί λανθασμένα νομίζει ότι η μπάλα βρίσκεται στο καλάθι αλλά εγώ ξέρω ότι έχει μετακινηθεί στο πράσινο κοντί"¹¹.

Επομένως, βλέπουμε ότι μέχρι τα 6 χρόνια τους τα παιδιά αναπτύσσουν αυτό που λέμε "Θεωρία του Νου", δηλαδή αναπτύσσουν την ικανότητα να αποδίδουν πνευματικές καταστάσεις σε άλλους.

⁹H. Kozima, C. Nakagawa, Y. Yasuda, *Children-robot interaction: a pilot study in autism therapy*

¹⁰http://en.wikipedia.org/wiki/Theory_of_mind

¹¹Robert Seyfarth: Theory of Mind, <http://c0116791.cdn.cloudfiles.rackspacecloud.com/Theory-of-Mind-Robert-Seyfarth-RDFTV.mov>

Την ονομάζουμε "θεωρία" γιατί δεν υπάρχει κάτι στο άλλο άτομο που να μας δείχνει ότι σκέφτονται. Οπότε ο μόνος τρόπος να θεωρήσουμε ότι υπάρχουν οι σκέψεις στους ανθρώπους οι οποίες προκαλούν ενέργειες, είναι να υπάρχει μια θεωρία για το πώς ο κόσμος δουλεύει, που λέει ότι οι άνθρωποι έχουν σκέψεις οι οποίες τους καθοδηγούν να κάνουν αυτό που κάνουν.

Μια νύξη που μπορεί να υποδείξει στο άτομο τι πρόκειται το άτομο που βρίσκεται μπροστά του να κάνει, είναι η προσήλωση με βάση το βλέμμα του. Τα παιδιά μάλιστα, είναι πολύ ευαίσθητα στην ανίχνευση του βλέμματος του ομιλητή τους, κυρίως στο να ξεχωρίζουν την προσήλωση του άλλου με βάση το εάν ο ομιλητής τους κοιτά στα μάτια ή όχι. Αυτό είναι κάτι που μπορεί να χρησιμοποιηθεί στην πρόβλεψη της επόμενης κίνησης ενός ρομπότ. Αν κοιτάζουμε τις κάμερες ενός ανθρωποειδούς ρομπότ, μπορούμε να κάνουμε διάφορες παρατηρήσεις για το που αυτές εστιάζουν. Με βάση αυτές τις παρατηρήσεις, μπορούμε να κάνουμε προβλέψεις για την επόμενη του κίνηση .

Όπως, λοιπόν, γίνεται εύκολα αντιληπτό, ένα παιδί προσχολικής και πρώτης σχολικής ηλικίας που βρίσκεται στα πρώιμα στάδια ανάπτυξης της Θεωρίας του Νου, μπορεί πολύ εύκολα να λάβει πληροφορίες και να μάθει από ένα ανθρωπόμορφο ρομπότ αφού με βάση την προσήλωση με βάση το βλέμμα και τις κάμερες του ρομπότ, θα αρχίσει να αναπτύσσει αυτήν την ικανότητα να αποδίδει πνευματικές καταστάσεις στο ρομπότ. Επομένως, είναι ιδανική η ηλικία αυτή για να γίνει χρήση της εκπαιδευτικής ρομποτικής στην ανάπτυξη των γνώσεων του παιδιού.

2.5 Παρακολούθηση του Βλέμματος

Η παρακολούθηση του βλέμματος (Gaze Following) είναι μια πολύ σημαντική συνιστώσα της ανθρώπινης κοινωνικής νόησης. Οι ενήλικες δίνουν έμφαση στο βλέμμα των συνανθρώπων τους ενώ παρακινούνται να δουν τι ακριβώς βλέπει ο άνθρωπος που βρίσκεται δίπλα τους. Αυτό είναι κάτι που χαρακτηρίζει το ανθρώπινο είδος, αφού είναι ένας από τους μηχανισμούς που έχει αναπτύξει ο άνθρωπος για να αποκτήσει πληροφορίες από άλλους ανθρώπους και πράγματα, μέσα από την παρατήρησή τους. Η παρακολούθηση του βλέμματος ενός ανθρώπου, βοηθάει στην αντίληψη των συναισθημάτων του ατόμου που παρακολουθούμε ενώ παράλληλα μαθαίνουμε τα ενδιαφέροντά του απλά και μόνο παρακολουθώντας τι βλέπει. Παράλληλα, η παρακολούθηση του βλέμματος βοηθάει στη μιμητική μάθηση όπου παρακολουθώντας το βλέμμα του διδάσκοντα γίνεται ευκολότερη η αντίληψη των προθέσεων του και των στόχων του.

Η κατανόηση της σημαντικότητας της παρακολούθησης του βλέμματος στις ανθρώπινες κοινωνικές σχέσεις, πυροδότησαν το ενδιαφέρον της προικοδότησης των ρομπότ με την ικανότητα αυτή, της παρακολούθησης δηλαδή το βλέμματος του ατόμου που βρίσκεται απέναντί τους. Εδώ όμως γεννιέται ένα ερώτημα: ποιού ατόμου το βλέμμα θα πρέπει να ακολουθεί το ρομπότ και κάτω από ποιες συνθήκες; Η απάντηση αυτού του ερωτήματος, θα δώσει τη λύση στο σχεδιασμό ενός ρομπότ το οποίο ένα μικρό παιδί θα μπορεί να αντιληφθεί ως ψυχολογικός πράκτορας που βλέπει τον έξω κόσμο. Μια σχετική έρευνα για αυτό το θέμα ¹² έχει δείξει ότι η ανθρωπόμορφη εμφάνιση ενός ρομπότ δεν είναι ο μόνος καθοριστικός παράγοντας για να προκαλέσουν στο παιδί την παρακολούθηση του βλέμματος του παιδιού. Χρειάζεται επιπλέον την ικανότητα του ίδιου του ρομπότ να ακολουθεί το βλέμμα του παιδιού αλλά παράλληλα να μπορεί να προκαλέσει κίνητρο στο παιδί ούτως ώστε να παρακολουθήσει το βλέμμα του ρομπότ.

Επομένως, με την εξέλιξη των ανθρωπόμορφων ρομποτικών συστημάτων και των τεχνολογιών όπως την παρακολούθηση του βλέμματος, τα ρομπότ πλέον έχουν καταφέρει να βλέπονται στα μάτια των παιδιών ως ψυχολογικές οντότητες, εφάμιλλες του ανθρώπου. Κάτι που παίζει σημαντικό

¹²A. Meltzoff, R. Brooks, A. Shonb, R. Raob, "Social" robots are psychological agents for infants: A test of gaze following, 2010

ρόλο στην εκπαιδευτική ρομποτική, αφού πλέον ένα ανθρωπόμορφο ρομπότ μπορεί επάξια να παίζει το ρόλο του διδάσκοντα στα μάτια ενός παιδιού.

2.6 Θεωρία ροής παιχνιδιών σοβαρού σκοπού

Τα παιχνίδια σοβαρού σκοπού (serious games) είναι παιχνίδια σχεδιασμένα με πρώτιστο στόχο κάτι άλλο πέρα από τη διασκέδαση. Τα σοβαρά παιχνίδια σχεδιάζονται με στόχο την επίλυση ενός προβλήματος. Μπορεί, βέβαια, να είναι διασκεδαστικά, αλλά τα παιχνίδια σοβαρού σκοπού έχουν ως κύριο τους στόχο να εκπαιδεύουν, να διερευνούν ή να διαφημίζουν. Κάποιες φορές μάλιστα, τα παιχνίδια αυτά θυσιάζουν εσκεμμένα τη διασκέδαση και την ευχαρίστηση με στόχο να επιτύχουν ένα σοβαρό σκοπό.

Η θεωρία ροής για τα παιχνίδια σοβαρού σκοπού, απευθύνεται στο ζήτημα της εστίασης και της προσοχής κατά τη διάρκεια μιας ενέργειας και εξετάζει το θέμα της βύθισης. Η θεωρία αυτή μπορεί να εφαρμοστεί και στα βιντεοπαιχνίδια αλλά κάλλιστα και στις εκπαιδευτικές εφαρμογές. Μάλιστα, ταιριάζει απόλυτα με τις προϋποθέσεις που θέτουν τα παιχνίδια σοβαρού σκοπού. Η έννοια αυτή είχε εισαχθεί από τον Mihaly Csikszentmihalyi. Ο Csikszentmihalyi αναφέρει τα ακόλουθα ως απαραίτητες προϋποθέσεις για τη διατήρηση του ατόμου σε κατάσταση ροής:

1. Ξεκάθαροι στόχοι
2. Συγκέντρωση και προσήλωση
3. Απώλεια αίσθησης της αυτο-συνείδησης
4. Λανθασμένη αίσθηση του χρόνου
5. Άμεση και απευθείας ανάδραση
6. Ισορροπία μεταξύ ικανότητας και πρόκλησης
7. Αίσθηση προσωπικού έλεγχου
8. Ύπαρξη δραστηριότητα που ανταμείβει εκ φύσεως της
9. Έλλειψη εγρήγορσης

Εν ολίγοις, στην κατάσταση της Θεωρίας Ροής, οι προκλήσεις που παρουσιάζονται και η ικανότητα του ατόμου για την επίλυσή τους, ταυτίζονται σχεδόν πλήρως και πολλές φορές το άτομο καταφέρνει πράγματα που νόμιζε ότι δεν μπορεί να επιτύχει. Σε αυτό ακριβώς είναι βασισμένα και τα παιχνίδια σοβαρού σκοπού. Σκοπεύουν, με τις πιο πάνω προαναφερθέντες μεθόδους, στη διατήρηση του ατόμου στην κατάσταση ροής. Αν κάνουν το παιχνίδι πολύ εύκολο για τους παίχτες, τότε αυτό θα επιφέρει την πλήξη του παίχτη και θα σταματήσει να παίζει. Αν από την άλλη το παιχνίδι είναι πολύ δύσκολο, τότε ο παίχτης απογοητεύεται, εκνευρίζεται και σταματά να παίζει. Μια πολύ καλή σχεδίαση παιχνιδιών σοβαρού σκοπού, περιλαμβάνει τη διατήρηση των παιχτών σε αυτή την κατάσταση ροής¹³.

2.7 Ανάπτυξη Σεναρίων με Ρομπότ

Τα τελευταία χρόνια, έχει εισαχθεί μια διαφορετική μέθοδος που αφορά τον τρόπο διεξαγωγής ερευνών και μελετών αλλά και της ίδιας της εκπαιδευτικής μάθησης. Η μέθοδος αυτή είναι η μέθοδος

¹³Marc Prensky, *Digital game-based learning*, 2001

Σχεδιασμού βασισμένη σε Σενάρια. Η πιο ευρέως διαδεδομένη μέθοδος και αυτή που χρησιμοποιείται κατά κόρον τόσα χρόνια, είναι η μέθοδος σχεδιασμού με κέντρο το χρήστη. Η διαδικασία αυτή, βάζει το χρήστη στο κέντρο του σχεδιασμού, οπότε στόχος είναι να δημιουργηθεί κάτι το οποίο πληροί τις προδιαγραφές του χρήστη. Αυτή η διαδικασία σχεδιασμού, όμως, έχει ένα μεγάλο ελάττωμα: αναφέρεται σε μια διαδικασία που υλοποιείται με βάση την ήδη υπάρχουσα τεχνολογία που έχει κυκλοφορήσει στην αγορά. Το πρόβλημα αυτό, οδήγησε στην εισαγωγή της νέας μεθόδου σχεδιασμού με βάση τα σενάρια. Η μέθοδος εισάγει τη χρήση των σεναρίων στη σχεδίαση. Τα σενάρια χρησιμοποιούνται ως κεντρικές αναπαραστάσεις κατά τη διάρκεια των κύκλων σχεδίασης περιγράφοντας πρώτα το στόχο και τις ανησυχίες της τρέχουσας χρήσης, ενώ έπειτα μεταμορφώνονται επιτυχώς μέσα από επαναληπτικές διαδικασίες αλλά και διαδικασιών αξιολόγησης. Τα σενάρια βοηθούν στον οραματισμό του αποτελέσματος, παρέχουν μια βάση για δοκιμές διεγείρουν τη σκέψη για πιθανές εκβάσεις της διαδικασίας. Όσον αφορά τώρα την εισαγωγή των σεναρίων στις ρομποτικές εφαρμογές, τα σενάρια χρησιμοποιούνται όχι μόνο ως ενδιάμεσα βήματα για το σχεδιασμό και την ανάπτυξη της διαδικασίας χρήσης του ρομπότ, αλλά και ως πλαίσια παιχνιδιού που επιτρέπουν στους χρήστες να αξιολογούν ειδικά εφαρμοσμένες λειτουργίες του τελικού αποτελέσματος του έργου. Η πιο σημαντική όμως πτυχή της χρησιμοποίησης των σεναρίων σε ένα έργο, είναι το γεγονός ότι δεν απευθύνονται στην ήδη υπάρχουσα τεχνολογία. Τα σενάρια αναφέρονται σε γενικότερες ιδέες και επομένως μπορούν να εφαρμοστούν από την εκάστοτε υπάρχουσα τεχνολογία και όχι μόνο από την τεχνολογία που υπήρχε τη στιγμή που δημιουργήθηκε το έργο. Αυτό δίνει μια ευελιξία και μια εξελισσιμότητα στο έργο αφού με την πρόοδο της τεχνολογίας το έργο μπορεί να επαναληφθεί με ακριβώς την ίδια διαδικασία απλά χρησιμοποιώντας διαφορετικά τεχνολογικά μέσα για βελτιώσουμε τα αποτελέσματά μας.

2.8 Gostai Suite

Το Gostai Suite είναι μια σουίτα που αποτελείται από το Gostai Studio και το Gostai Lab και έχει ως σκοπό τον έλεγχο πολλών ρομπότ με μια πιο απλοποιημένη και εύκολη διαδικασία. Το πρώτο πρόγραμμα, το Gostai Studio, είναι ένα γραφικό περιβάλλον συμπεριφοράς και διαγραμμάτων καταστάσεων, ένα ενσωματωμένο λογισμικό ανάπτυξης (IDE) το οποίο απευθύνεται σε επαγγελματίες αλλά και πρωτάρηδες στο χώρο της ρομποτικής, μέσω του οποίου μπορούν να χειριστούν οποιοδήποτε ρομποτικό σύστημα με μια νέα πρωτοποριακή μέθοδο της σχεδίασης διαγραμμάτων κατάστασης. Το Gostai Lab, είναι ένα πολύ δυνατό εργαλείο για τη σχεδίαση ενός γραφικού περιβάλλοντος διασύνδεσης με το χρήστη (GUI) με απώτερο στόχο τη δημιουργία γραφικών στοιχείων (widgets) μέσω των οποίων θα γίνεται ασύρματος χειρισμός του ρομπότ αλλά και των μερών που το αποτελούν .

Το Gostai Suite είναι κτισμένο πάνω σε μια πλατφόρμα ανοιχτού λογισμικού γραμμένη σε C/C++, που ονομάζεται URBI (Universal Robot Body Interface). Σκοπός της πλατφόρμας αυτής, είναι ο έλεγχος ρομπότ αλλά και γενικότερα πολύπλοκων συστημάτων. Περιλαμβάνει μια συνιστώσα βιβλιοθήκη σε C++ που αποτελείται από μια πλήρη ρομποτική διασύνδεση προγραμματισμού εφαρμογών (API). Όλα αυτά υλοποιούνται σε μια γλώσσα προγραμματισμού, την Urbiscript, που μοιάζει πολύ με τη γλώσσα Python και συνδυάζει όλα αυτά για να περιγράψει υψηλού επιπέδου συμπεριφορές. Ο στόχος του URBI, είναι να κάνει τα ρομπότ συμβατά και να απλοποιήσει τη διαδικασία συγγραφής προγραμμάτων προγραμματισμού των ρομπότ. Όπως έχει αναφερθεί, το URBI είναι ανοιχτού λογισμικού κώδικας, κάτι που σημαίνει ότι έχει τεράστια περιθώρια εξέλιξης .

Εμείς στη διπλωματική αυτή εργασία, θα χρησιμοποιήσουμε το Gostai Studio αφού πρώτα έχουμε εγκαταστήσει το URBI. Ο προγραμματισμός ενός ρομπότ μέσω του Gostai Studio, είναι μια εντελώς καινούρια ιδέα, τουλάχιστον για τα πλαίσια του Ε.Μ.Π. Θα ασχοληθούμε με το νέο στυλ προγραμματισμού που αναφέρεται στη δημιουργία διαγραμμάτων κατάστασης για κάθε μια από τις ενέργειες που θέλουμε να κάνει το ρομπότ. Είναι ένας εύκολος και πιο πρακτικός τρόπος

αλληλεπίδρασης με το ρομποτικό σύστημα, ενώ μάλιστα έχει απεριόριστες δυνατότητες, αφού λόγω του ότι αναφερόμαστε σε λογισμικό ανοιχτού κώδικα, μας δίνει τη δυνατότητα να δημιουργήσουμε οποιοδήποτε αντικείμενο εμείς επιθυμούμε, πέρα από τα διαθέσιμα που υπάρχουν στη βιβλιοθήκη UObject του URBI.

2.9 Ρομποτική Πλατφόρμα Nao

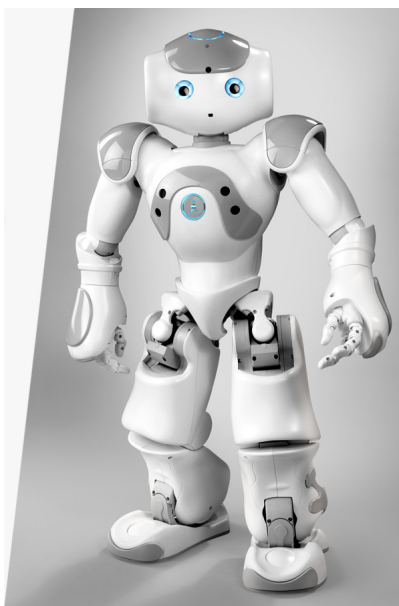
Το Nao είναι ένα αυτόνομο, προγραμματιζόμενο, ανθρωπόμορφο ρομπότ μεσαίου μεγέθους το οποίο έχει σχεδιαστεί από τη Γαλλική εταιρία Aldebaran. Το Nao είναι το πιο ευρέως χρησιμοποιούμενο ανθρωπόμορφο ρομπότ για ακαδημαϊκούς και εκπαιδευτικούς σκοπούς παγκοσμίως. Η εταιρία κατασκευής του, έχει επιλέξει να κάνει την τεχνολογία του Nao διαθέσιμη σε οποιοδήποτε πρόγραμμα τριτοβάθμιας εκπαίδευσης. Είναι πλήρως διαδραστικό και διασκεδαστικό, ενώ εξελίσσεται συνέχεια, στοιχεία που το καθιστούν το καταλληλότερο ρομποτικό σύστημα για εκπαίδευση μαθητών σε όλα τα επίπεδα. Η ευελιξία του Nao, επιτρέπει στους χρήστες μια τεράστια γκάμα δυνατοτήτων, από απλό οπτικό προγραμματισμό μέχρι και πολύπλοκα σενάρια εκμάθησης. Το Nao είναι επίσης και το επίσημο ρομπότ του Παγκοσμίου Πρωταθλήματος Ρομποτικού Ποδοσφαίρου (Robot Soccer World Cup - Robocup).

Το Nao διαθέτει 25 βαθμούς ελευθερίας. Αυτό είναι κάτι που το καθιστά εξαιρετικά ευκίνητο και επιτρέπει τον προγραμματισμό πολύπλοκων κινήσεων αλλά και διαφόρων χορογραφιών. Τα χέρια του αποτελούνται από 3 αλληλοεξαρτώμενα δάχτυλα, τα οποία μπορούν να κρατήσουν μικρά αντικείμενα.

Το Nao έχει διάφορες δυνατότητες μέσω των διαφόρων αισθητήρων του. Αυτές είναι οι ακόλουθες:

- Το Nao ακούει. Διαθέτει 4 μικρόφωνα τα οποία βρίσκονται στο κεφάλι του, ενώ παράλληλα διαθέτει ένα σύστημα αναγνώρισης και ανάλυσης της φωνής. Μπορεί να ανιχνεύει λέξεις που ακούει, μέχρι τώρα μόνο στα Αγγλικά και στα Γαλλικά. Μπορεί, επίσης, να ανιχνεύσει την πηγή ενός ήχου ή μιας φωνής και να αρχίσει να αλληλεπιδρά.
- Το Nao μιλά. Μπορεί να εκφραστεί μέσω φωνής από τα δυο μεγάφωνα που υπάρχουν στις δυο πλαϊνές μεριές του κεφαλιού του. Έχει τη δυνατότητα σύνθεσης φωνής, οπότε μπορεί να διαβάσει κάτι και να το μετατρέψει σε φωνή και πάλι όμως μέχρι τώρα μόνο στα Αγγλικά και στα Γαλλικά.
- Το Nao βλέπει. Μέσω των 2 καμερών που διαθέτει και είναι τοποθετημένες στο μέτωπό του και στο στόμα του, που μπορούν να αποθηκεύσουν μέχρι και 30 εικόνες το δευτερόλεπτο. Επίσης διαθέτει ειδικούς αλγόριθμους αναγνώρισης προσώπων και σχημάτων.
- Το Nao αισθάνεται. Εφοδιασμένο με αισθητήρες αφής στο κεφάλι αλλά και στα πόδια του, μπορεί να αισθανθεί το άγγιγμα του χεριού και να αλληλεπιδράσει. Επίσης, διαθέτει αισθητήρες πίεσης στις πατούσες των ποδιών του, κάτι που του επιτρέπει να ρυθμίσει τη στάση του σώματός του.
- Το Nao στέχεται από μόνο του. Διαθέτει ένα αξελερόμετρο και ένα γυρόμετρο τα οποία του δίνουν μια αίσθηση ισορροπίας και του δίνουν τη δυνατότητα να σταθεί όταν έχει πέσει.
- Το Nao αποφεύγει εμπόδια. Μέσω των δυο ζευγαριών υπερηχητικών αισθητήρων, λαμβάνει ανάδραση από διάφορες αντηχήσεις. Αυτό έχει ως αποτέλεσμα να γνωρίζει που υπάρχουν εμπόδια και να προσπαθεί να τα αποφύγει.

Επομένως, γίνεται εύκολα αντιληπτό γιατί έχουμε επιλέξει το συγκεκριμένο ρομπότ ως το ρομποτικό σύστημα με το οποίο θα αλληλεπιδράσει το παιδί. Είναι ανθρωπόμορφο, στοχεύει σε εκπαιδευτικούς σκοπούς, έχει πάρα πολλές δυνατότητες, είναι πλήρως προγραμματιζόμενο ενώ διατίθεται στο εργαστήριό μας, όπως και σε κάθε άλλο μεγάλο εργαστήριο αλληλεπίδρασης ανθρώπων με μηχανές.

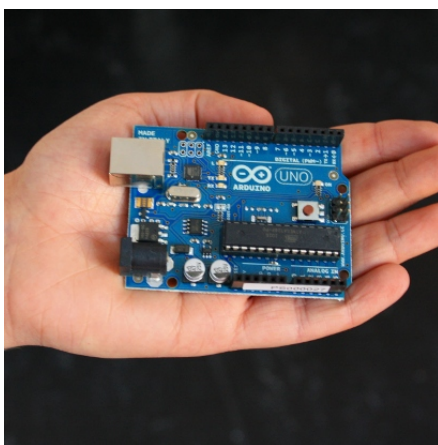


Σχήμα 2.1: Το ρομπότ Nao

2.10 Arduino

Το Arduino είναι μια ανοιχτού κώδικα ηλεκτρονική προτυποποιημένη πλατφόρμα βασισμένη σε ευέλικτα, εύκολα στη χρήση υλικά και προγράμματα. Απευθίνεται σε όλες τις κατηγορίες ανθρώπων, είτε αυτοί είναι επιστήμονες, είτε είναι εκπαιδευτικοί, είτε άνθρωποι που ασχολούνται με το συγκεκριμένο σαν χόμπυ, είτε και σε μικρά παιδιά. Το Arduino αλληλεπιδρά με το περιβάλλον μέσω των διαφόρων αισθητήρων του και μπορεί να επηρεάσει το περιβάλλον μέσω των διαφόρων επενεργητών που μπορούν να συνδεθούν πάνω του. Η πλακέτα διαθέτει ένα μικροελεγκτή ο οποίος προγραμματίζεται με βάση την γλώσσα προγραμματισμού Arduino και με το περιβάλλον προγραμματισμού Arduino .

Εμείς θα χρησιμοποιήσουμε το Arduino ως βοήθημα για τις αποστολές που πρέπει να εκτελέσει το παιδί, ούτως ώστε μέσω αυτού να λαμβάνει το ρομπότ μας πληροφορία για το αν η αποστολή που έπρεπε να εκτελέσει το παιδί έχει στεφθεί με επιτυχία.



Σχήμα 2.2: Το Arduino

Κεφάλαιο 3

Σενάριο

3.1 Η πρώτη ιδέα

Λαμβάνοντας υπόψη τα όσα αναφέραμε στο προηγούμενο κεφάλαιο και γνωρίζοντας τις δυνατότητες του ανθρωπόμορφου ρομπότ Nao που διαθέτει το εργαστήριό μας, σχηματίσαμε το ακόλουθο σενάριο εκμάθησης βασικών γνώσεων αριθμητικής για παιδιά προσχολικής και πρώτης σχολικής ηλικίας.

Το σενάριο που έχουμε κατασκευάσει, έχει ως πρωταγωνιστές ένα παιδί προσχολικής ή πρώτης σχολικής ηλικίας και τη ρομποτική πλατφόρμα Nao που διαθέτουμε στο εργαστήριό. Το παιδί, παίζει ένα παιχνίδι σοβαρού σκοπού, το οποίο χωρίζεται σε διάφορα επίπεδα δυσκολίας. Το παιχνίδι θα χωρίζεται σε δυο μέρη:

- Το πρώτο μέρος, θα αποτελείται από μια συνεργασία του παιδιού με το διδάσκοντά του ή το γονέα του, όπου θα γίνονται κάποια παιχνίδια μεταξύ των δυο και το παιδί θα ανταμείβεται με ένα αριθμό νομισμάτων. Αυτό το κομμάτι του παιχνιδιού δεν θα μας απασχολήσει εμάς.
- Το δεύτερο μέρος, το οποίο και θα μελετήσουμε εμείς, αφορά τη μετάβαση του παιδιού στο επόμενο επίπεδο δυσκολίας. Μετά την ολοκλήρωση κάποιων παιχνιδιών-δοκιμασιών με τον επιβλέποντά του, το παιδί θα έχει τη δυνατότητα να μεταβεί στο επόμενο επίπεδο δυσκολίας όπου πλέον τα παιχνίδια και οι δοκιμασίες θα είναι δυσκολότερες. Για να το επιτύχει όμως αυτό, το παιδί καλείται να αλληλεπιδράσει κατάλληλα με το ρομπότ. Ζητείται, δηλαδή, από το παιδί, να αγοράσει χρησιμοποιώντας τα νομίσματά που διαθέτει, ένα συγκεκριμένο αντικείμενο, το οποίο και θα παραδώσει στον επιβλέποντά του για να επιτευχθεί η μετάβαση στο επόμενο επίπεδο. Αν τα νομίσματα που διαθέτει είναι επαρκή, τότε θα τα δώσει στο ρομπότ και θα πάρει το ζητούμενο αντικείμενο. Αν δεν επαρκούν, τότε και υπό τις οδηγίες του ρομπότ, θα εκτελέσει κάποιες αποστολές με τις οποίες θα αποκτήσει τα νομίσματα που απαιτούνται για να αγοράσει το αντικείμενο και να μεταβεί στο επόμενο επίπεδο δυσκολίας. Αυτό θα είναι στην ουσία το σενάριο που θα υλοποιήσουμε.

3.2 Αναλυτική Περιγραφή του Σεναρίου

Θα αναλύσουμε τώρα λεπτομερώς, το σενάριο που αναφέραμε πιο πάνω.

- Το παιδί κρατάει στα χέρια του ένα αριθμό νομισμάτων που έχει αποκτήσει είτε κατά τη διάρκεια της πρώτης φάσης του παιχνιδιού, είτε από αποστολές που έχει εκτελέσει κατά τη δεύτερη φάση του παιχνιδιού για να αποκτήσει ένα επιθυμητό αντικείμενο. Εδώ πρέπει να σημειώσουμε, ότι τα νομίσματα αυτά που θα χρησιμοποιήσει το παιδί, είναι ειδικά κατασκευασμένα νομίσματα από την εταιρία κατασκευής του ρομπότ Nao, την Aldebaran, τα οποία φέρουν ένα ειδικό

λογότυπο επάνω ¹. Υπάρχουν 10 νομίσματα με διαφορετικό λογότυπο για αναγνώριση από το Nao. Αυτό, διευκολύνει μετέπειτα το στάδιο της οπτικής αναγνώρισης αυτών αφού με αυτά τα ειδικά νομίσματα είναι πλέον εφικτή η οπτική αναγνώρισή τους αλλά συνάμα και η καταμέτρησή τους από το Nao. Εμείς θα χρησιμοποιήσουμε 4 διαφορετικά είδη νομισμάτων από τα διαθέσιμα τα οποία θα αναλυθούν σε επόμενο κεφάλαιο. Συνεπώς, κάθε ένα από τα 4 διαφορετικά νομίσματα, αντιστοιχεί και σε μια τιμή. Έχουμε επιλέξει τις διαβαθμίσεις των 1 ευρώ, 2 ευρώ, 4 ευρώ και 5 ευρώ. Επομένως, το παιδί κρατάει, πιθανώς, ένα συνδυασμό νομισμάτων. Ο επιβλέπωντας του παιδιού, μετά το πέρας της πρώτης φάσης του παιχνιδιού, έχει εισχωρήσει στο ρομπότ μέσω του προγράμματος Gostai Studio, το αντικείμενο που πρέπει να αγοράσει το παιδί. Τότε το ρομπότ, ανακοινώνει στο παιδί το όνομα του αντικειμένου που πρέπει να αγοράσει αλλά και την αγοραστική του αξία.

- Το παιδί, αφού έχει ακούσει το αντικείμενο και την τιμή του, καλείται να δει πόσα ευρώ διαθέτει σε νομίσματα και να δει αν είναι αρκετά ή όχι για να αγοράσει το επιθυμητό αντικείμενο. Είτε το παιδί αποφανθεί ότι έχει τον απαιτούμενο αριθμό νομισμάτων είτε όχι, θα δώσει στο ρομπότ κάποια νομίσματα. Το ρομπότ τότε, μέσω της διαδικασίας οπτικής αναγνώρισης και καταμέτρησης των νομισμάτων που θα του δώσει το παιδί, θα ελέγξει αν αυτά τα νομίσματα είναι τα ζητούμενα σε σχέση με το αντικείμενο που έχει εισαχθεί από τον επιβλέποντα, οπότε σε αυτή τη σύγκριση διακρίνουμε τρεις περιπτώσεις:
 - Εάν η αξία του αντικειμένου είναι ίση με την αξία των νομισμάτων που έχει πάρει το ρομπότ, τότε στην ουσία θα έχουμε επιτυχία. Το ρομπότ θα ενημερώνει το παιδί ότι έχει βάλει το σωστό αριθμό νομισμάτων και επομένως θα του επιτρέψει να πάρει το επιθυμητό αντικείμενο.
 - Εάν η αξία του αντικειμένου είναι μικρότερη από την αξία των νομισμάτων που έδωσε το παιδί, τότε το ρομπότ θα ενημερώσει το παιδί ότι έχει δώσει περισσότερα νομίσματα από αυτά που στην πραγματικότητα στοιχίζει το αντικείμενο και θα το αναγκάσει να ξαναδεί τα νομίσματά του, να δει τι έκανε λάθος και να ξαναδώσει στο ρομπότ το σωστό αριθμό, πλέον, νομισμάτων. Οπότε η διαδικασία σύγκρισης των νομισμάτων επαναλαμβάνεται έως ότου το παιδί να δώσει το σωστό αριθμό νομισμάτων, τον οποίο προφανώς και κατέχει απλά κάνει λάθος στις αριθμητικές του πράξεις.
 - Εάν η αξία του αντικειμένου είναι μεγαλύτερη από την αξία των νομισμάτων που έδωσε το παιδί στο ρομπότ, τότε το ρομπότ θα ενημερώσει το παιδί ότι τα νομίσματα που διαθέτει δεν είναι επαρκή, οπότε για να συμπληρώσει τον απαιτούμενο αριθμό νομισμάτων, χρειάζεται να εκτελέσει κάποιες αποστολές. Οι αποστολές αυτές θα γίνουν είτε αποκλειστικά με τη χρήση του Nao είτε με τη συνεργασία και τη συμβολή της πλατφόρμας Arduino. Μετά το πέρας αυτών των αποστολών, το Nao θα κρίνει αν οι αποστολές έχουν εκτελεστεί σωστά ή όχι. Αν έχουν εκτελεστεί σωστά, τότε θα επιτρέψει στο παιδί να πάρει από ένα κουτί που θα βρίσκεται δίπλα του, τον υπολειπόμενο αριθμό νομισμάτων που χρειάζεται για να αποκτήσει στο αντικείμενο και να επαναλάβει τη διαδικασία παράδοσης και σύγκρισης των νομισμάτων από το ρομπότ. Αν, πάλι, η αποστολή δεν έχει εκτελεστεί σωστά, το ρομπότ θα ενημερώσει το παιδί ότι πρέπει να εκτελέσει ξανά την αποστολή γιατί στην προηγούμενή του προσπάθεια απέτυχε.

3.2.1 Περιγραφή αποστολών

Οι αποστολές, θα ποικίλουν ανάλογα με την υπολειπόμενη αξία σε ευρώ που χρειάζεται το παιδί να αποκτήσει. Ανάλογα, δηλαδή, με τη διαφορά των ευρώ που έχει το παιδί από τα ευρώ που χρειάζεται πραγματικά για να αγοράσει το αντικείμενο, θα υπάρχει και μια αποστολή. Θα φτιάξουμε συνολικά

¹Nao References CD

4 αποστολές όπου κάθε μια θα δίνει στο παιδί 1, 2, 4, ή 5 ευρώ. Οι 4 αυτές αποστολές είναι οι ακόλουθες:

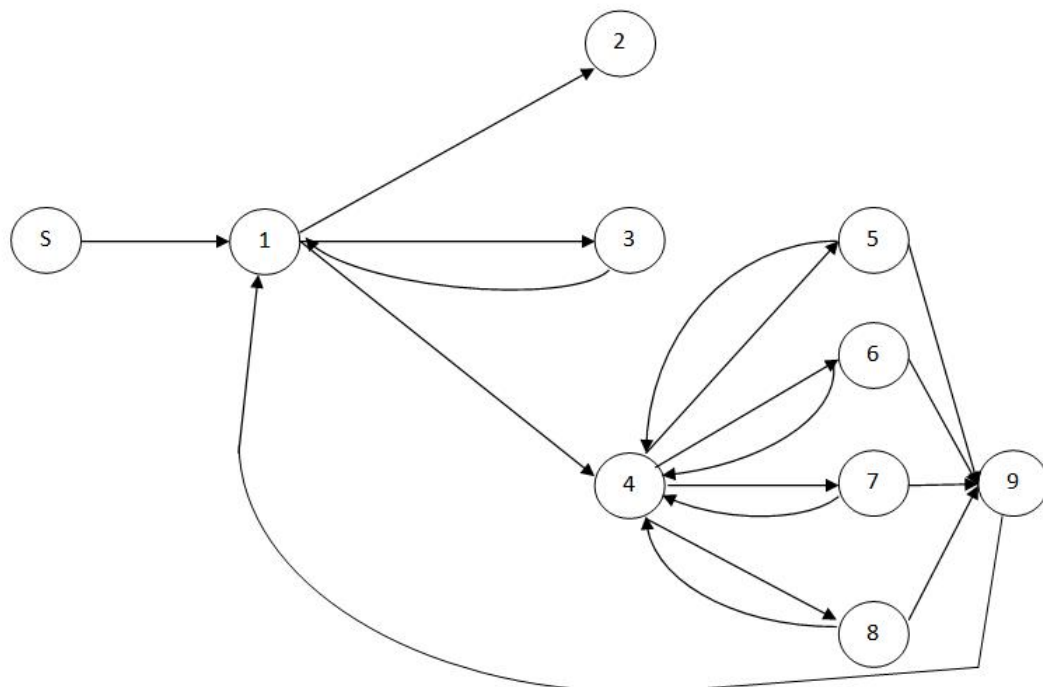
1. Το ρομπότ θα ζητήσει από το παιδί να του πει στα αγγλικά πόσα ευρώ χρειάζεται ακόμη για να πάρει το αντικείμενο που θέλει. Το παιδί θα κάνει την αριθμητική πράξη και θα πει μπροστά από το πρόσωπο του ρομπότ τα ευρώ που χρειάζεται ακόμη. Τότε το Nao θα επεξεργαστεί αυτήν την απάντηση μέσω του προγράμματος αναγνώρισης φωνής που διαθέτει (voice recognition) και αν διαπιστώσει ότι είναι σωστή η απάντησή του, θα του αναφέρει την επιτυχία και θα του επιτρέψει να πάρει τα λεφτά που χρειάζεται. Για αυτή τη δοκιμασία το παιδί θα πάρει 1 νόμισμα του 1 ευρώ αν επιτύχει. Αν αποτύχει, τότε το ρομπότ θα του αναφέρει ότι έδωσε λανθασμένο αριθμό και θα το παρακαλέσει να επαναλάβει τη διαδικασία.
2. Το ρομπότ θα ζητήσει από το παιδί να κτυπήσει τον μπροστινό αισθητήρα αφής (touch sensor) του κεφαλιού του Nao, όσες φορές είναι και τα νομίσματα που χρειάζεται το παιδί να πάρει για να μπορέσει να αγοράσει το αντικείμενο που χρειάζεται. Οπότε το παιδί, θα πρέπει να κάνει την αριθμητική πράξη της αφαίρεσης, να βρει πόσα ακόμα ευρώ χρειάζεται και να κτυπήσει τόσες φορές τον συγκεκριμένο αισθητήρα του Nao. Αν βρει τη σωστή απάντηση, τότε το Nao θα του αναφέρει την επιτυχία και θα του επιτρέψει να πάρει τα νομίσματα που θέλει από ένα κουτί που υπάρχει δίπλα του. Αν το παιδί δεν εκτελέσει την αποστολή σωστά, τότε το Nao θα το ενημερώσει για το λάθος του και θα του ζητήσει να επαναλάβει την αποστολή. Η επιτυχία του παιδιού σε αυτήν την αποστολή θα του δώσει 1 νόμισμα των 2 ευρώ.
3. Στην αποστολή αυτή θα γίνεται συνεργασία με την πλακέτα Arduino. Το ρομπότ, θα ζητάει από το παιδί να διαλέξει από 3 διαθέσιμα κουτιά που αναγράφουν πάνω τους 3 διαφορετικούς αριθμούς, ποιο από τα τρία αντιπροσωπεύει τον αριθμό των χρημάτων που χρειάζεται επιπλέον το παιδί για να αποκτήσει το αντικείμενο που θέλει. Τα τρία αυτά κουτιά, θα είναι συνδεδεμένα με μικρούς διακόπτες (pushbuttons) οι οποίοι θα είναι συνδεδεμένοι με το Arduino. Μόλις ανοίξει ένα κουτί το παιδί, στην ουσία θα πατιέται ένας από τους διακόπτες και θα στέλνεται ένα σήμα στο Arduino. Το σωστό κουτί θα στέλνει λογικό 1 ενώ τα άλλα 2 κουτιά θα στέλνουν λογικό 0. Όταν το Arduino λάβει το λογικό 1, τότε θα ενεργοποιείται ένα άλλο κύκλωμα που θα κατασκευάσουμε μέσω του Arduino, παραγωγής φωνής. Μέσω αυτού του κυκλώματος θα παράγεται ένας ήχος τον οποίο και θα ανιχνεύει το Nao για να ξέρει ότι η αποστολή έχει ολοκληρωθεί με επιτυχία και να επιτρέψει στο παιδί να πάρει τα νομίσματα που θέλει. Αν το παιδί ανοίξει το λάθος κουτί, τότε θα σταλεί σήμα λογικού 0 στο Arduino και πάλι με το κύκλωμα παραγωγής φωνής, θα παραχθεί μια διαφορετική φωνή που θα δείχνει στο Nao ότι η αποστολή ήταν ανεπιτυχής και έτσι πρέπει το παιδί να την επαναλάβει. Από αυτήν την αποστολή, το παιδί θα πάρει αν την ολοκληρώσει με επιτυχία, 1 νόμισμα των 4 ευρώ.
4. Τέλος, και σε αυτήν την αποστολή θα γίνει συνεργασία με το Arduino. Το ρομπότ, θα ζητήσει από το παιδί να πατήσει σε ένα ειδικά κατασκευασμένο πληκτρολόγιο, τον αριθμό των χρημάτων που χρειάζεται για να αποκτήσει το αντικείμενο που θέλει. Το ειδικά κατασκευασμένο αυτό πληκτρολόγιο, θα το κατασκευάσουμε εμείς. Θα αποτελείται στην ουσία από 4 διακόπτες πίεσης (pushbuttons) που θα φέρουν πάνω τους ένα αυτοκόλλητο με ένα αριθμό πάνω. Μόνο ο ένας από τους 4 αριθμούς θα είναι αυτός που θα αντιστοιχεί στα χρήματα που χρειάζεται επιπλέον το παιδί. Αν αυτός ο διακόπτης πατηθεί από το παιδί, τότε και πάλι θα ενεργοποιηθεί η πλακέτα παραγωγής φωνής και μέσω αυτής θα ειδοποιηθεί το Nao για την επιτυχή ολοκλήρωση της αποστολής. Αν το παιδί πατήσει κάποιον από τους άλλους διακόπτες τότε το Nao θα ειδοποιηθεί και πάλι μέσω της πλακέτας παραγωγής φωνής ότι το παιδί απέτυχε και θα το παρακαλέσει να επαναλάβει τη διαδικασία. Από αυτή τη διαδικασία το παιδί θα πάρει αν την ολοκληρώσει επιτυχώς, 1 νόμισμα των 5 ευρώ.

Σε όλες τις αποστολές, όπως έχουμε δει, το ρομπότ θα ελέγχει την ορθότητα της εκτέλεσης. Αν η αποστολή έχει εκτελεστεί σωστά, τότε θα ενημερώνει το παιδί ότι έχει ολοκληρώσει επιτυχώς την αποστολή και ότι δικαιούται να πάρει τα νομίσματα που αντιστοιχούν στην αποστολή τα οποία θα βρίσκονται δίπλα από το ρομπότ. Αν η αποστολή δεν έχει εκτελεστεί σωστά, τότε το ρομπότ θα ενημερώσει το παιδί ότι έχει κάνει λάθος και ό,τι πρέπει να επαναλάβει τη διαδικασία. Αυτό θα επαναληφθεί έως ότου να καταφέρει να ολοκληρώσει με επιτυχία την αποστολή.

Το παιδί, μέσα από αυτές τις αποστολές θα πάρει τα χρήματα που του απομένουν για να συμπληρώσει το ποσό που χρειάζεται για να αποκτήσει το ποθητό αντικείμενο. Αφού θα συγκεντρώσει τα καινούρια νομίσματα, θα ξανα-ακολουθηθεί η διαδικασία κατά την οποία το παιδί πρέπει να δώσει στο ρομπότ τα νομίσματα που απαιτούνται για να πάρει το αντικείμενο που του λείει το ρομπότ. Πλέον το παιδί διαθέτει τα χρήματα που απαιτούνται αλλά πρέπει να δώσει στο ρομπότ το σωστό αριθμό νομισμάτων. Επομένως από το σημείο αυτό επαναλαμβάνεται το πρώτο βήμα όπου το ρομπότ θα αναγνωρίσει και θα καταμετρήσει τα νομίσματα που θα του δώσει το παιδί και θα τα συγκρίνει με την τιμή του αντικειμένου που χρειάζεται το παιδί.

3.3 Διαγραμματική παρουσίαση του τελικού σεναρίου

Πιο κάτω παρουσιάζαμε το σενάριο που θα υλοποιήσουμε όπως αυτό φαίνεται μέσα από διαγράμματα καταστάσεων. Στη συνέχεια, εξηγούμε τι θα συμβαίνει σε κάθε κατάσταση.



Σχήμα 3.1: Το τελικό σενάριο μέσα από διαγράμματα καταστάσεων

Πίνακας 3.1: Επεξήγηση του διαγράμματος καταστάσεων

Κατάσταση	Σημασία
S	Αυτή είναι η αρχική κατάσταση. Δεν γίνεται οποιαδήποτε ενέργεια οπότε αμέσως γίνεται μετάβαση στην επόμενη κατάσταση.
1	Στην κατάσταση αυτή γίνεται μια σειρά από διεργασίες. Καταρχάς, το ρομπότ ανακοινώνει στο παιδί το όνομα του αντικειμένου που θέλει το παιδί αλλά και την τιμή του. Στη συνέχεια, το παιδί τοποθετεί τα νομίσματα που θεωρεί ότι αντιστοιχούν στην τιμή του αντικειμένου που θέλει, μπροστά ακριβώς από το κεφάλι του Ναο, πάνω στο τραπέζι που θα στέκεται το ρομπότ. Τότε το ρομπότ, θα σκύψει το κεφάλι του προς τα νομίσματα ούτως ώστε να εφαρμόσει οπτική αναγνώριση. Η διαδικασία αναγνώρισης αυτών των νομισμάτων, περιλαμβάνει την αναγνώριση πολλών νομισμάτων ταυτόχρονα. Επομένως, το παιδί, θα μπορεί να βάζει πολλά νομίσματα μαζί, το ένα δίπλα στο άλλο με την ειδική όψη του νομίσματος να φαίνεται προς τα πάνω ούτως ώστε το ρομπότ να μπορεί να εφαρμόσει οπτική αναγνώριση. Η αναγνώριση των νομισμάτων θα μετατρέψει τα νομίσματα σε χρηματική αξία και προσθέτοντας την αξία των νομισμάτων που έβαλε το παιδί, το Ναο πλέον γνωρίζει πόσα ευρώ του έχει δώσει το παιδί. Ανάλογα τώρα με αυτή τη χρηματική αξία, θα μεταβούμε σε μια από τις επόμενες καταστάσεις.
2	Στην κατάσταση αυτή θα μεταβούμε αν τα νομίσματα που έδωσε το παιδί στο ρομπότ αντιστοιχούν ακριβώς στην αξία του αντικειμένου που το παιδί θέλει, οπότε στην ουσία έχουμε επιτυχή ολοκλήρωση της διαδικασίας. Το ρομπότ θα σηκώσει τα χέρια του ψηλά ως ένδειξη επαίνου προς στο παιδί και θα φωνάξει στα Αγγλικά "Right". Αυτό θα δείξει στο παιδί ότι μπορεί πλέον να πάρει το αντικείμενο που θέλει από το κουτί που βρίσκεται δίπλα από το ρομπότ.
3	Στην κατάσταση αυτή θα μεταβούμε αν τα νομίσματα που έδωσε το παιδί στο ρομπότ έχουν μεγαλύτερη χρηματική αξία από αυτήν του αντικειμένου, οπότε το ρομπότ θα φωνάξει στα Αγγλικά "Try again" δείχνοντας στο παιδί ότι έβαλε έτσι περισσότερα νομίσματα από όσα έπρεπε και ότι πρέπει να δοκιμάσει ξανά, βάζοντας τα σωστά νομίσματα αυτή τη φορά. Μετά το πέρας αυτής της κατάστασης, θα μεταβούμε πίσω στην κατάσταση 1.
4	Στην κατάσταση αυτή θα μεταβούμε αν τα νομίσματα που έδωσε το παιδί στο ρομπότ έχουν μικρότερη αξία από τη χρηματική αξία του αντικειμένου που θέλει το παιδί, οπότε το ρομπότ θα φωνάξει στα αγγλικά την τιμή του αντικειμένου για ακόμα μια φορά, δείχνοντας στο παιδί ότι έχει δώσει λιγότερα νομίσματα από όσα έπρεπε και ότι χρειάζεται να εκτελέσει μια αποστολή για να πάρει τα υπόλοιπα νομίσματα που χρειάζεται. Ανάλογα, τώρα, από τα χρήματα που χρειάζεται ακόμη το παιδί για να αποκτήσει το αντικείμενο θα εκτελείται μια αποστολή, οπότε και θα μεταβαίνουμε στην ανάλογη επόμενη κατάσταση.
5	Στην κατάσταση αυτή θα μεταβούμε αν το παιδί χρειάζεται ακόμη 1 ευρώ για να αποκτήσει το αντικείμενο που θέλει. Στην περίπτωση αυτή, το ρομπότ θα ζητήσει από το παιδί να του πει πόσα χρήματα ακόμη χρειάζεται για να αποκτήσει το αντικείμενο που θέλει. Το ρομπότ, τότε, χρησιμοποιώντας την ικανότητα της αναγνώρισης ήχου που διαθέτει (voice recognition), θα αναγνωρίσει αν η απάντηση που έδωσε το παιδί είναι σωστή ή όχι. Αν η απάντηση είναι σωστή, τότε θα μεταβούμε στην κατάσταση 9. Αν είναι λάθος η απάντηση, θα επιστρέψουμε στην κατάσταση 4.

Κατάσταση	Σημασία
6	Στην κατάσταση αυτή θα μεταβούμε αν το παιδί χρειάζεται ακόμη 2 ευρώ για να αποκτήσει το αντικείμενο που θέλει. Στην περίπτωση αυτή, το ρομπότ θα ζητήσει από το παιδί να κτυπήσει τον μπροστινό αισθητήρα αφής του κεφαλιού του τόσες φορές όσα είναι και τα χρήματα που χρειάζεται ακόμη να πάρει. Αν η απάντηση είναι σωστή, θα μεταβούμε στην κατάσταση 9 ενώ αν είναι λάθος θα επιστρέψουμε στην κατάσταση 4.
7	Στην κατάσταση αυτή θα μεταβούμε αν το παιδί χρειάζεται ακόμη 4 ευρώ για να αποκτήσει το αντικείμενο που θέλει. Στην περίπτωση αυτή, η αποστολή που θα ζητηθεί να εκτελέσει το παιδί θα είναι σε συνεργασία με την πλακέτα Arduino. Το ρομπότ θα ζητάει από το παιδί να διαλέξει από 3 διαθέσιμα κουτιά που αναγράφουν πάνω 3 διαφορετικές τιμές, αυτό που αναγράφει την τιμή των χρημάτων που χρειάζεται το παιδί για να αγοράσει το αντικείμενο που θέλει. Αν το παιδί διαλέξει το σωστό κουτί, τότε μέσω ενός διακόπτη πίεσης που θα υπάρχει θα σταλεί λογικό 1 στο Arduino το οποίο στην ουσία θα ενημερώνει για τη σωστή εκτέλεση της αποστολής. Τότε, το Arduino μέσω ενός συστήματος παραγωγής φωνής που θα κατασκευάσουμε, θα παράγει τον ήχο "Yes" στα Αγγλικά, το οποίο θα αναγνωρίζει το Nao ως επιτυχή συμπλήρωση της αποστολής. Οπότε σε τέτοια περίπτωση θα μεταβούμε στην κατάσταση 9. Αν τώρα το παιδί διαλέξει λάθος κουτί, τότε θα σταλεί στο Arduino λογικό 0 οπότε μέσω του συστήματος παραγωγής φωνής θα παραχθεί ο ήχος "No" θα ενημερώνει το ρομπότ για αποτυχία της αποστολής και συνεπώς θα επιστρέψουμε στην κατάσταση 4.
8	Στην κατάσταση αυτή θα μεταβούμε αν το παιδί χρειάζεται ακόμη 5 ευρώ για να αποκτήσει το αντικείμενο που θέλει. Στην περίπτωση αυτή, η αποστολή που θα ζητηθεί να εκτελέσει το παιδί, θα είναι και πάλι σε συνεργασία με την πλακέτα Arduino. Το ρομπότ θα ζητάει από το παιδί να πατήσει από ένα πληκτρολόγιο που θα κατασκευάσουμε εμείς, το κουμπί στο οποίο αναγράφεται τα σωστά χρήματα που χρειάζεται ακόμη το παιδί για να πάρει το αντικείμενο που θέλει. Αν το παιδί πατήσει το σωστό κουμπί, τότε θα σταλεί λογικό 1 στο Arduino το οποίο στην ουσία θα ενημερώνει για τη σωστή εκτέλεση της αποστολής. Τότε, το Arduino μέσω ενός συστήματος παραγωγής φωνής που θα κατασκευάσουμε, θα παράγει τον ήχο "Yes" στα Αγγλικά, το οποίο θα αναγνωρίζει το Nao ως επιτυχή συμπλήρωση της αποστολής. Οπότε σε τέτοια περίπτωση θα μεταβούμε στην κατάσταση 9. Αν τώρα το παιδί πατήσει λάθος κουμπί, τότε θα σταλεί στο Arduino λογικό 0 οπότε μέσω του συστήματος παραγωγής φωνής θα παραχθεί ο ήχος "No" θα ενημερώνει το ρομπότ για αποτυχία της αποστολής και συνεπώς θα επιστρέψουμε στην κατάσταση 4.
9	Στην κατάσταση αυτή θα μεταβούμε μετά από επιτυχή ολοκλήρωση κάποιας από τις διαθέσιμες αποστολές. Όταν φθάσουμε στην κατάσταση αυτή, το ρομπότ θα σηκώσει και πάλι ψηλά τα χέρια του, θα φωνάξει στα αγγλικά "Right. Take (τον αριθμό των νομισμάτων που χρειάζεται να πάρει) coins" και θα δείξει έτσι στο παιδί ότι ολοκλήρωσε την αποστολή του σωστά και μπορεί να πάρει το ζητούμενο νόμισμα από ένα κουτί που θα βρίσκεται πίσω του. Μετά το πέρας της κατάστασης αυτής, θα μεταβούμε στην κατάσταση 1 για να επαναληφθεί η διαδικασία ελέγχου νομισμάτων, όπου τώρα το παιδί θα έχει τα σωστά νομίσματα και θα μπορεί να πάρει το επιθυμητό αντικείμενο.

3.4 Παραδοχές

Υπάρχουν κάποια προβλήματα τα οποία θα αναφέρουμε τώρα, τα οποία αντιμετωπίσαμε στο σενάριό μας με κάποιες παραδοχές. Αναφέρουμε, επίσης, κάποιες βελτιώσεις που μπορούσαν να γίνουν.

- Θεωρήσαμε ότι υπάρχουν διαθέσιμα μόνο τα νομίσματα των 1, 2, 4, 5 ευρώ. Με αυτά όμως τα νομίσματα αν η τιμή του αντικειμένου είναι αρκετά μεγάλη, προϋποθέτει ότι θα έχουμε πολλά από αυτά τα νομίσματα. Εμείς στο σενάριό μας έχουμε 1 από κάθε είδος. Συνεπώς, περιορίζουμε την αξία που θα έχουν τα αντικείμενα στα 11 ευρώ. Αν θέλαμε να εγγυηθούμε ότι θα καλύψουμε αντικείμενα με μεγαλύτερη αξία, έπρεπε να χρησιμοποιήσουμε και νομίσματα των 10 και 20 ευρώ.
- Όπως αναφέραμε πιο πάνω, το παιδί μπορεί να βάλει πολλά νομίσματα μπροστά στο ρομπότ για οπτική αναγνώριση. Αυτό είναι χαρακτηριστικό της διαδικασίας οπτικής αναγνώρισης που θα χρησιμοποιήσουμε. Παρόλα αυτά, για να αποφύγουμε οποιαδήποτε προβλήματα με πολλά νομίσματα μαζί, όπως δηλαδή να βγαίνουν από το πεδίο της κάμερας του ρομπότ και να μην μπορεί να τα αναγνωρίσει, θέτουμε ότι το παιδί θα βάλει το πολύ μέχρι 3 νομίσματα, το ένα δίπλα στο άλλο. Αυτό, φυσικά, περιορίζει το συνολικό αριθμό χρημάτων που μπορεί να διαθέσει το παιδί γιαυτό και αναγκαστήκαμε, όπως θα δούμε πιο κάτω, να περιορίσουμε τις τιμές των αντικειμένων σε σχετικά χαμηλά επίπεδα. Ίσως θα ήταν καλύτερο να μπορούμε να επαναλάβουμε τη διαδικασία οπτικής αναγνώρισης αρκετές φορές, μέχρις ότου να αναγνωριστούν όλα τα νομίσματα που θα έχει βάλει το παιδί. Αυτό όμως έχει το ελάττωμα του ότι δεν ξέρουμε από πριν πόσες ακριβώς φορές χρειάζεται να επαναλάβουμε τη διαδικασία οπτικής αναγνώρισης για να καλυφθούν όλα τα νομίσματα. Γι' αυτό προτιμήσαμε να περιορίσουμε τα νομίσματα που θα βάλει το παιδί σε 3 κάθε φορά.
- Θεωρήσαμε ότι κάθε αποστολή θα καθορίζεται από πριν από το πόσα χρήματα χρειάζεται ακόμη το παιδί για να αγοράσει το αντικείμενό του. Αφού χρησιμοποιήσαμε 4 αποστολές, είπαμε ότι για να κάνει την πρώτη αποστολή το παιδί σημαίνει χρειάζεται 1 ευρώ, για τη δεύτερη χρειάζεται 2 ευρώ, για την τρίτη 4 ευρώ και για την 4η αποστολή χρειάζεται 5 ευρώ. Αυτό όμως, δεν μας δίνει τη δυνατότητα αν για παράδειγμα το παιδί έχει 1 ευρώ να πάρει ένα αντικείμενο αξίας 8 ευρώ, αφού με μια αποστολή μπορεί να πάρει ακόμη 5 ευρώ και άρα να φτάσει στα 6. Στο σενάριό μας, όμως, θα θεωρήσουμε ότι υπάρχουν 4 διαθέσιμα αντικείμενα, αξίας 2 ευρώ, 3 ευρώ, 5 ευρώ και 6 ευρώ.

Κεφάλαιο 4

Υλοποίηση σεναρίου στο Gostai Studio

4.1 Ανάλυση λειτουργιών προγράμματος

Όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο, το Gostai Studio είναι ένα πρόγραμμα ελέγχου ρομποτικών συστημάτων το οποίο βασίζεται στα διαγράμματα καταστάσεων. Επομένως, έχουμε ένα εντελώς καινούριο τρόπο προγραμματισμού και ελέγχου διάφορων ρομποτικών συστημάτων. Αναφέρουμε ενδεικτικά, κάποιες από τις ιδιαίτερες λειτουργίες που προσφέρει το Gostai Studio:

- Το πρόγραμμα δίνει δυο δυνατότητες προγραμματισμού ενός ρομπότ: η πρώτη είναι η συγγραφή ενός script στη γλώσσα Urbiscript (ένα αρχείο .u), το οποίο στην ουσία είναι προγραμματισμός με τη χρήση των εντολών της γλώσσας Urbiscript. Η μέθοδος αυτή δεν διαφέρει από τον προγραμματισμό του ρομπότ μέσω μιας γλώσσας προγραμματισμού (C, C++, Python κ.ά.) και η μεταφορά του κώδικα αυτού στο ρομπότ μέσω ασύρματης ή ενσύρματης σύνδεσης. Ο δεύτερος τρόπος προγραμματισμού ενός ρομπότ, είναι η δημιουργία ενός αρχείου Urbi Behavior Graph File (.ubg) το οποίο στην ουσία περιλαμβάνει μια γραφική αναπαράσταση ενός διαγράμματος καταστάσεων, βάση του οποίου θα γίνει ο προγραμματισμός και ο έλεγχος του ρομπότ.
- Για τη δημιουργία ενός διαγράμματος καταστάσεων, έχουμε στη διάθεσή μας διάφορους κόμβους και μεταβάσεις από ένα κόμβο στον άλλο. Κάθε κόμβος περιέχει κώδικα γραμμένο σε Urbiscript που στην ουσία είναι αυτό που θα εκτελείται από το ρομπότ όταν εισερχόμαστε σε αυτόν τον κόμβο, ενώ παράλληλα μπορούμε να εξοπλίσουμε τον κόμβο με εντολές που θα εκτελεστούν μόλις πριν μπούμε στον κόμβο αυτό αλλά και αμέσως μετά που θα φύγουμε από τον κόμβο. Για τη μετάβαση από ένα κόμβο σε άλλο χρησιμοποιούνται τόξα μεταβάσεων (transitions) τα οποία μπορούν να ελέγχουν μια συνθήκη. Δηλαδή, μπορούμε να μεταβούμε από τον ένα κόμβο στον επόμενο, αν ισχύει μια συνθήκη. Αν δεν ισχύει μπορούμε να παραμείνουμε στον τρέχοντα κόμβο ή να μεταβούμε σε έναν άλλο. Απαραίτητη προϋπόθεση για να λειτουργήσει το διάγραμμα καταστάσεων, είναι να θέσουμε ένα κόμβο ως αρχικό κόμβο.
- Κάθε κόμβος μπορεί να περιέχει διάφορους υποκόμβους. Δηλαδή, μπορούμε να χωρίσουμε τον κόμβο σε διάφορους μικρότερους όπου να ταξινομήσουμε άλλες υποενέργειες που πρέπει να γίνουν στο συγκεκριμένο κόμβο.
- Μπορούμε να δηλώσουμε και καθολικές και τοπικές μεταβλητές. Οι τοπικές μεταβλητές έχουν εμβέλεια μόνο εντός του κόμβου στον οποίο δηλώνονται. Οι καθολικές μεταβλητές έχουν εμβέλεια σε όλους τους κόμβους που βρίσκονται μετά από αυτόν στον οποίο δηλώνονται. Μια καλή τεχνική, είναι να δηλώνουμε τις καθολικές μας μεταβλητές στον αρχικό κόμβο ούτως ώστε στην ουσία να έχουν εμβέλεια σε όλο το διάγραμμα καταστάσεων.
- Με το πρόγραμμα αυτό μπορούμε να προγραμματίσουμε σε διάφορες ρομποτικές κονσόλες, όπως το Nao της Aldebaran που θα μας απασχολήσει εμάς.
- Υπάρχει η δυνατότητα εισαγωγής κάποιων ειδικών βιβλιοθηκών που διαθέτουν κάποιους εξει-

δικευμένους κόμβους. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε την generic library (generic.ubl), με την οποία μπορούμε να εισάξουμε ένα random κόμβο ο οποίος στην ουσία δίνει σε μια μεταβλητή την οποία ορίζουμε εμείς, τυχαία τιμή κάθε φορά που τρέχει το πρόγραμμά μας. Υπάρχουν διαθέσιμες 4 βιβλιοθήκες (generic.ubl, aibo.ubl, nao.ubl, robot.ubl) που διαθέτουν διάφορους κόμβους ενώ κάποιος μπορεί να κατασκευάσει και μια δική του βιβλιοθήκη και να την ενσωματώσει στο πρόγραμμα.

- Υπάρχει η δυνατότητα προσομοίωσης κάποιων καταστάσεων χωρίς τη σύνδεση με το ρομπότ. Μπορεί να γίνει εγκατάσταση ενός URBI server στο σύστημά μας και η εικονική στην ουσία δημιουργία ενός port που θα ενώνει το Gostai Studio με το server και επιτρέπει την εκτέλεση κάποιων εντολών. Έτσι μπορεί να γίνουν κάποιες δοκιμές προτού περάσουμε στην πράξη και τη μεταφορά του σεναρίου στη ρομποτική πλατφόρμα. Φυσικά, οι εντολές που μπορούν να λάβουν μέρος σε αυτήν την προσομοίωση είναι κάπως περιορισμένες, αφού όσες εντολές αφορούν καθαρά τη λειτουργία ενός ρομπότ δεν μπορούν να προσομοιωθούν στον εικονικό server.
- Το πρόγραμμα αυτό όπως και το Gostai Lab διατίθενται προς αγορά σε επαγγελματικές εκδόσεις και σε εκπαιδευτικές εκδόσεις. Πέρα όμως από αυτές, υπάρχουν διαθέσιμες και δοκιμαστικές εκδόσεις (trial version) οι οποίες δεν επιτρέπουν όλες τις λειτουργίες μιας πλήρους έκδοσης ενώ παράλληλα κάθε 10 λεπτά εμφανίζουν ένα προειδοποιητικό μήνυμα.
- Κατά την σύνδεση του προγράμματος με το ρομπότ, στην ουσία έχουμε μετατροπή του διαγράμματος καταστάσεων σε ένα απλό script κώδικα το οποίο και θα σταλεί στο ρομπότ, βάσει του οποίου θα εκτελεστούν όσα σχεδιάσαμε. Στην επαγγελματική και την εκπαιδευτική έκδοση του προγράμματος, υπάρχει η δυνατότητα να δούμε αυτό το script που παράγεται οπότε να τροποποιήσουμε αν θέλουμε κάτι πάνω στον κώδικά μας, ενώ στη δοκιμαστική έκδοση δεν υπάρχει αυτή η δυνατότητα.

4.2 Ανάλυση εντολών

4.2.1 Ανάλυση βασικών εντολών

Θα αναλύσουμε τώρα κάποιες βασικές εντολές που χρησιμοποιούνται από το Gostai Studio τις οποίες θα χρησιμοποιήσουμε και εμείς στο σενάριο το οποίο θα υλοποιήσουμε.

Βασικές Εντολές

```

/*Type of variables*/
42;                /*Integer literal*/
3.14;              /*Floating point number literal*/
"string";          /*Character string literal*/
[1, 2, "a", "b"]; /*List literal*/
["a" => 1];        /*Dictionary literal*/

/*Variable declaration */
var x = 42;

/*Scope of variables*/
/*What is inside the brackets {} has local scope*/
{
    var x = "test";
    echo(x);        /*Echo is used for printing in*/
};                 /*standard output*/

/*Flow control constructs*/

```

```

/*If structure*/
if (true)
    echo("ok");

if (x == 42)
    echo("ko")
else
    echo("ok");

/*While structure*/
while (x < 40)
{
    x += 10;
    echo(x);
};

/*For structure*/
for (var x = 2; x < 40; x += 10)
    echo(x);

/*Switch structure*/
var x = 42;
var y = 1;
switch(x)
{
    case 1: y = 10; /*If x=1 then y =10*/
    case 2: y = 20; /*If x=2 then y =20*/
};

```

Τώρα θα παρουσιάσουμε κάποιες βασικές διαδικασίες που χρειάζονται να γίνουν μεταξύ των κόμβων και των μεταβάσεων καταστάσεων. Αρχή θα κάνουμε με τη δήλωση μεταβλητών γενικής εμβέλειας (global variables).

Δήλωση μεταβλητών σε ένα κόμβο:

Οι δηλώσεις μεταβλητών σε ένα κόμβο γίνονται στο κομμάτι "Code executed before entering the node". Εδώ θα δηλώνουμε μεταβλητές που θέλουμε να είναι προσβάσιμες μόνο στον τρέχοντα κόμβο αλλά και στις μεταβάσεις που φεύγουν από αυτόν τον κόμβο, ενώ μπορούμε να δηλώσουμε και μεταβλητές γενικής εμβέλειας, οι οποίες θα εφαρμόζονται σε όλους τους κόμβους και μεταβάσεις καταστάσεων που βρίσκονται κάτω από τον τρέχοντα κόμβο στον οποίο γίνονται οι δηλώσεις μεταβλητές. Οι δηλώσεις μεταβλητών γίνονται με τον ακόλουθο τρόπο:

Δηλώσεις μεταβλητών σε κόμβους

```

/*Global variables declaration*/
if (!rootNode.hasLocalSlot("myVar")) /*This check is */
var rootNode.myVar = 42; /*for preventing the declaration*/
                        /*of the same variable*/

/*Local variables declaration for use for this node*/
/*and its transitions*/

```

```
if (!this.hasLocalSlot("myVar"))
var this.myVar = 42;
```

Μεταβάσεις κατάστασης:

Σε ένα βέλος μετάβασης κατάστασης, μπορούμε να μεταφερθούμε από τον ένα κόμβο στον άλλο μέσα από μια συνθήκη. Η συνθήκη αυτή μπορεί να είναι απλά το τελείωμα των εργασιών του πρώτου κόμβου ή το να ισχύει μια συνθήκη σχετικά με μια μεταβλητή. Δηλαδή, έχουμε:

Μεταβάσεις κατάστασης

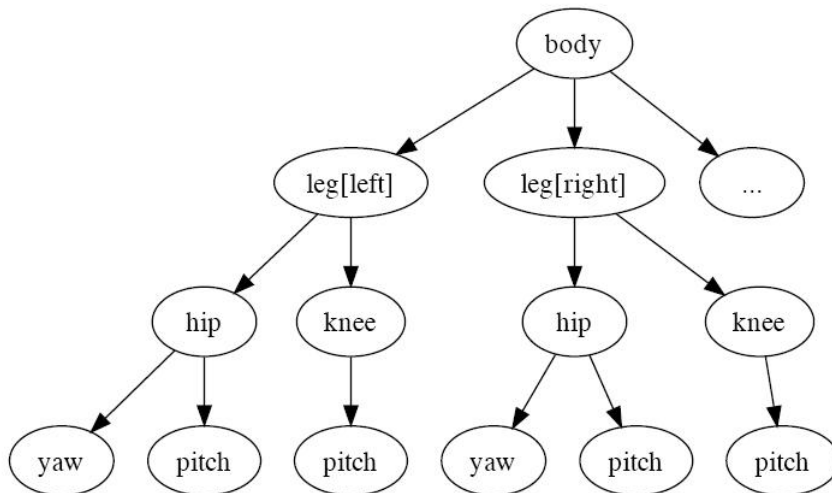
```
done? /*For transition to the next node when*/
      /*the job of the first node is finished*/
```

```
rootNode.myVar == 1 /*For transition to the next node when*/
                    /*the statement is true*/
```

4.2.2 Ανάλυση εντολών ρομποτικού πακέτου

Το Gostai Studio, όπως έχουμε αναφέρει, έχει τη δυνατότητα προγραμματισμού σε διάφορες ρομποτικές πλατφόρμες. Για κάθε μια από αυτές, προσφέρονται κάποια έτοιμα πακέτα που περιλαμβάνουν εντολές χειρισμού των συγκεκριμένων ρομπότ. Γενικά, όμως, υπάρχει μια συγκεκριμένη δομή που ακολουθεί το πρόγραμμα και αφορά όλες τις ρομποτικές πλατφόρμες. Ας δούμε, λοιπόν, κάποια γενικά πράγματα για το πως χειρίζεται ένα ρομπότ το Gostai Studio.

Το ρομπότ αντιμετωπίζεται σαν ένα δέντρο δομής το οποίο είναι ένα δέντρο οργανωμένο με ιεραρχική δομή. Η σχέση ενός φύλλου του δέντρου με το φύλλο του είναι ότι το δεύτερο είναι μέρος του πρώτου. Ένα παράδειγμα ενός ανθρωποειδούς ρομπότ σε μορφή δέντρου δομής, δίνεται παρακάτω, όπου παρατηρούμε ότι το σώμα του ρομπότ είναι ο αρχικός κόμβος ενώ τα χέρια του ρομπότ είναι τμήματα του σώματος του ρομπότ. Ομοίως γίνεται και με τα υπόλοιπα μέρη του ρομπότ.

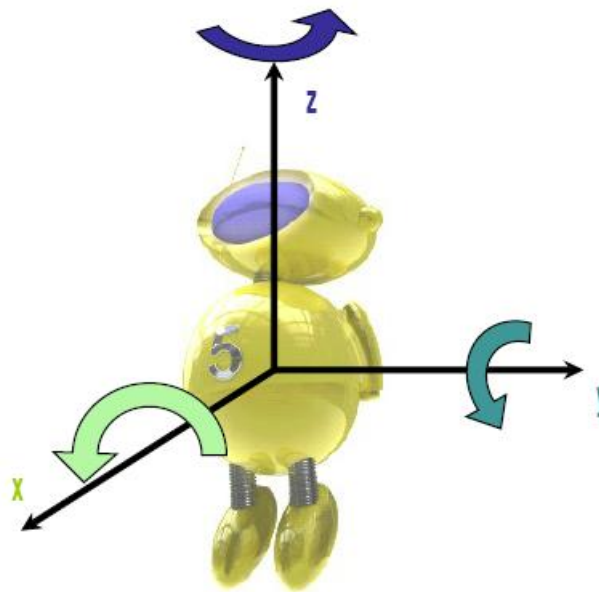


Σχήμα 4.1: Ένα δέντρο δομής ενός ανθρωποειδούς ρομπότ

Επομένως, με βάση τα πιο πάνω, αν θέλουμε να χρησιμοποιήσουμε κάποιο από τα τμήματα του ρομπότ που ανήκουν σε ένα από τα πρώτα φύλλα του δέντρου, θα χρησιμοποιήσουμε τις ακόλουθες εντολές:

```
body.leg [ left ].hip.yaw ;  
body.leg [ left ].knee.pitch ;  
body.leg [ right ].hip.pitch ;
```

Το πρόγραμμα δουλεύει με ένα πλαίσιο αναφοράς, μέσα από το οποίο χωρίζουμε το ρομπότ σε 3 κατευθύνσεις για να ξέρουμε κάθε φορά ακριβώς σε τι πράγματα αναφερόμαστε. Το πλαίσιο αναφοράς δίνεται από το πιο κάτω σχήμα:

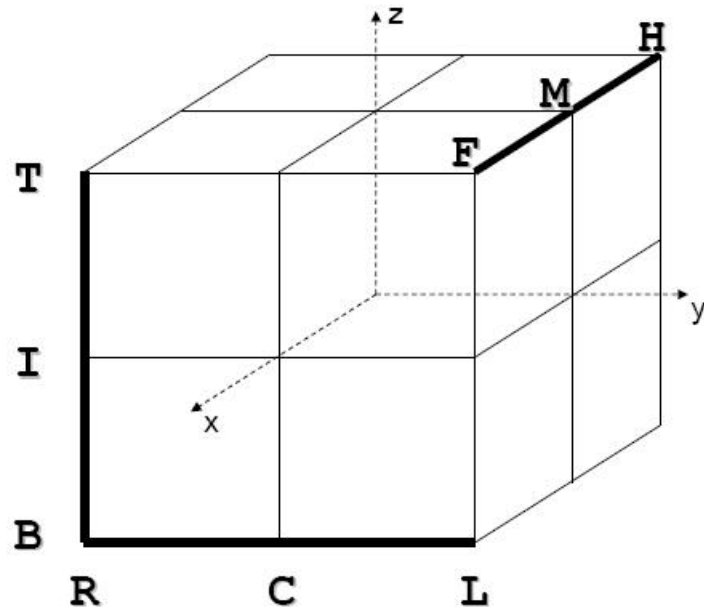


Σχήμα 4.2: Το πλαίσιο αναφοράς του ρομπότ

Με βάση το πιο πάνω σχήμα, έχουμε τον ακόλουθο διαχωρισμό των προσανατολισμών:

- Ο άξονας X προσανατολίζεται προς τα μπροστά του ρομπότ. Αν υπάρχει κάμερα, τότε το "μπροστά" ορίζεται ως η default κατεύθυνση της κάμερας, αλλιώς το "μπροστά" είναι ο φυσικός προσανατολισμός της μπροστινής κατεύθυνσης ενός αυτοκινούμενου ρομπότ.
- Ο άξονας Z προσανατολίζεται στην ακριβώς αντίθετη κατεύθυνση της βαρύτητας.
- Ο άξονας Y προσανατολίζεται για να δημιουργείται με τους άλλους 2 άξονες ένα δεξιόστροφο σύστημα συντεταγμένων.

Με βάση, τώρα, το σύστημα συντεταγμένων που αναφέραμε, μπορούμε να χρησιμοποιήσουμε κάποιες μεταβλητές τοπικότητας (που όπως είδαμε σε προηγούμενο παράδειγμα εμπερικλείονται σε αγκύλες []) για να εξακριβώσουμε για πιο ακριβώς τμήμα του ρομπότ μιλάμε. Αυτός ο διαχωρισμός των μεταβλητών τοπικότητας, γίνεται με βάση το ακόλουθο σχήμα:



Σχήμα 4.3: Μεταβλητές τοπικότητας του ρομπότ

Στο σχήμα αυτό, τα γράμματα που αναφέρονται έχουν την ακόλουθη σημασία:

- B: down - κάτω
- I: in, between - μέσα, ενδιάμεσα
- T: up - πάνω
- R: right - δεξιά
- C: center - κέντρο
- L: left - αριστερά
- F: front - μπροστά
- M: middle - στη μέση
- H: back - πίσω

Όταν έχουμε να χρησιμοποιήσουμε πολλές από αυτές τις μεταβλητές (δηλαδή για παράδειγμα θέλουμε να αναφερθούμε στο μπροστά-δεξιά πόδι ενός ρομπότ) τότε η σειρά προτεραιότητας των μεταβλητών τοπικότητας έχει ως ακολούθως: πρώτα βάζουμε την πληροφορία για το ύψος (B/I/D), μετά ακολουθεί η πληροφορία για το βάθος (F/M/H) και τέλος η πληροφορία για το πλάτος (R/C/L).

Με βάση τώρα όλα αυτά και τη γνώση μας για τα μέρη του ρομπότ Nao που θα χρησιμοποιήσουμε εμείς, παραθέτουμε παρακάτω έναν πίνακα ο οποίος περιέχει μια λίστα από τα μέρη που απαρτίζουν το Nao, ταξινομημένα με ιεραρχική δομή, από πάνω προς τα κάτω.

Πίνακας 4.1: Μέρη του ρομπότ Nao

Όνομα	Περιγραφή	Υποκλάση του:	Περιέχει το:
robot	Είναι το βασικό μέρος του ρομπότ	-	body
body	Αυτό είναι το βασικό τμήμα του ρομπότ στο οποίο συνδέονται όλα τα μηχανικά και κινηματικά του μέρη	robot	arm, leg, head

leg	Το Nao διαθέτει 2 πόδια τα οποία είναι προσαρμοσμένα στο σώμα του ρομπότ	body	hip, knee, ankle
arm	Το Nao διαθέτει 2 χέρια τα οποία είναι προσαρμοσμένα στο σώμα του ρομπότ	body	shoulder, elbow
shoulder	Ο ώμος είναι το πάνω μέρος του χεριού	arm	pitch, roll
elbow	Ο αγκώνας ξεχωρίζει το άνω τμήμα του χεριού από το κάτω τμήμα	arm	roll, yaw
hand	Η παλάμη είναι η προέκταση του χεριού που συγκρατεί τα δάχτυλα	arm	-
hip	Ο γοφός είναι το πάνω μέρος του ποδιού και το συνδέει με το κυρίως σώμα	leg	pitch, roll
knee	Το γόνατο ξεχωρίζει το άνω τμήμα του ποδιού με το κάτω τμήμα	leg	pitch
ankle	Ο αστράγαλος συνδέει την πατούσα με το κάτω μέρος του ποδιού	leg	pitch, roll
head	Το κεφάλι του Nao	body, neck	pitch, yaw, camera
camera	Οι κάμερες που βρίσκονται τοποθετημένες στο κεφάλι του Nao	head, body	-
speaker	Τα ηχεία που βρίσκονται τοποθετημένα ως αυτιά του Nao	head, body	-
speech	Αφορά την αναγνώριση φωνής	robot	-
voice	Αφορά τη σύνθεση φωνής	robot	-

Το Gostai και γενικότερα η γλώσσα προγραμματισμού Urbi, επιτρέπει τη χρήση συμπαγούς συμβολισμού (compact notation) όπου δε χρειάζεται να αναφέρουμε όλες τις κλάσεις στις οποίες ανήκει ένα μέλος του ρομπότ. Για παράδειγμα, έχουμε τα ακόλουθα:

- *robot.body.head.yaw*: Το head και το yaw πρέπει να κρατηθούν γιατί είναι απαραίτητα για να ελέγξουμε αυτό που επιθυμούμε. Το robot και το body δεν χρειάζονται, γιατί είναι δεδομένο ότι το κεφάλι είναι συνδεδεμένο με το σώμα και αυτό με το robot. Επομένως, θα χρησιμοποιήσουμε το ακόλουθο: **headYaw**.
- *robot.body.leg[L].knee.pitch*: Στην περίπτωση αυτή, το knee έχει μόνο μια υποκλάση και αυτή είναι το pitch. Επομένως, δεν χρειάζεται να το συμπεριλάβουμε, όπως ούτε τα robot, body, leg αλλά είναι σημαντικό να αναφέρουμε το ότι αναφερόμαστε στο αριστερό πόδι. Επομένως, θα χρησιμοποιήσουμε το ακόλουθο: **kneeL**.

Αυτό το συμπαγές συμβολισμό θα χρησιμοποιήσουμε και εμείς κατά τη διάρκεια του σεναρίου μας, αφού είναι πολύ εύκολος και κατανοητός.

4.2.3 Ανάλυση οπτικής αναγνώρισης Naomark

Τα Naomark είναι τα ειδικά "νομίσματα" τα οποία μπορεί να αναγνωρίσει το Nao μέσα από το ειδικό τους σχήμα. Υπάρχουν 10 διαθέσιμα νομίσματα που μπορούν να αναγνωριστούν. Εμείς θα χρησιμοποιήσουμε 4 διαφορετικά νομίσματα τα οποία θα αναλύσουμε σε επόμενο κεφάλαιο.

Τα Naomark έχουν ένα ειδικό τρόπο αναγνώρισης από το Nao ¹. Η οπτική τους αναγνώριση

¹http://users.aldebaran-robotics.com/docs/site_en/reddoc/naomark_detection/naomark_

από το Nao στηρίζεται σε ένα module το οποίο δουλεύει πάνω στο NAOqi Framework. Το NAOqi Framework, αποτελεί ένα σύνολο βιβλιοθηκών σχεδιασμένο ειδικά για να προσφέρει στη ρομποτική πλατφόρμα παραλληλισμό, συγχρονισμό, διάφορα γεγονότα και διάφορες πηγές ². Το NAOqi βρίσκεται εγκατεστημένο σε κάθε ρομποτική πλατφόρμα Nao και με την ύπαρξη αυτού του Framework δίνεται η δυνατότητα χρήσης διάφορων modules τα οποία στην ουσία κάνουν τη λειτουργία πιο ευέλικτη. Ένα από αυτά τα modules είναι και το Naomark Detection Module (ALLandMarkDetection).

Το Naomark Detection Module χρησιμοποιεί ένα προκαθορισμένο τμήμα κώδικα για την οπτική αναγνώριση των Naomarks το οποίο είναι το ακόλουθο:

Naomark Detection Module

```

1  /*Optical recognition module*/
2
3  /*Create a proxy to ALLandMarkDetection*/
4  var markProxy = ALProxy("LandMarkDetection", [],
5  ["subscribe", "unsubscribe"]);
6
7  /*Subscribe to the ALLandMarkDetection extractor*/
8  var period = 500; /*The "period" parameter specifies—in milliseconds—*/
9                   /*how often ALLandMarkDetection tries to run its */
10                  /*detection method*/
11
12 /*Enable detection*/
13 markProxy.subscribe("Test_Mark", period, 0.0 );
14
15 /*Get data from landmark detection */
16 data = robot.ALMemory.getData("LandmarkDetected");
17
18 /*Disable detection*/
19 markProxy.unsubscribe("TestMark", period, 0.0);

```

Επομένως, από όλη αυτή τη διαδικασία, παίρνουμε μια μεταβλητή data που είναι τύπου λίστας. Αν ανιχνεύσει το Nao κάποιο Naomark τότε η λίστα αυτή δεν είναι πλέον κενή, αλλά αποτελείται από τα εξής 2 στοιχεία:

```
data = [ [ TimeStampField ] [ MarkInfo0 , MarkInfo1, . . . , MarkInfoN-1 ] ]
```

Τα πιο πάνω τμήματα της λίστας ερμηνεύονται ως ακολούθως:

- Το TimeStampField αποτελείται από δυο υπό-πεδία: TimeStampField = [TimeStampSeconds, TimestampMicroseconds]. Αυτές οι δυο μεταβλητές στην ουσία μας δίνουν το πότε έγινε η οπτική αναγνώριση.
- Τα διάφορα MarkInfoN μας δείχνουν πόσα Naomarks έχουν ανιχνευθεί. Αν έχουν ανιχνευθεί N Naomarks τότε θα έχουμε N τέτοιες μεταβλητές. Κάθε τέτοια μεταβλητή, αποτελείται από τα εξής υπό-πεδία: MarkInfo = [ShapeInfo, ExtraInfo]. Οι δυο αυτές μεταβλητές περιλαμβάνουν τα ακόλουθα:
 - ShapeInfo = [0, alpha, beta, sizeX, sizeY, heading]. Τα alpha και beta αντιπροσωπεύουν τη θέση των Naomarks σε γωνιές κάμερας, τα sizeX, sizeY είναι το μέγεθος των Naomarks σε γωνιές κάμερας και η γωνιά heading περιγράφει τον προσανατολισμό του Naomark γύρω από τον κάθετο άξονα σε σχέση με το κεφάλι του Nao.

detection.html

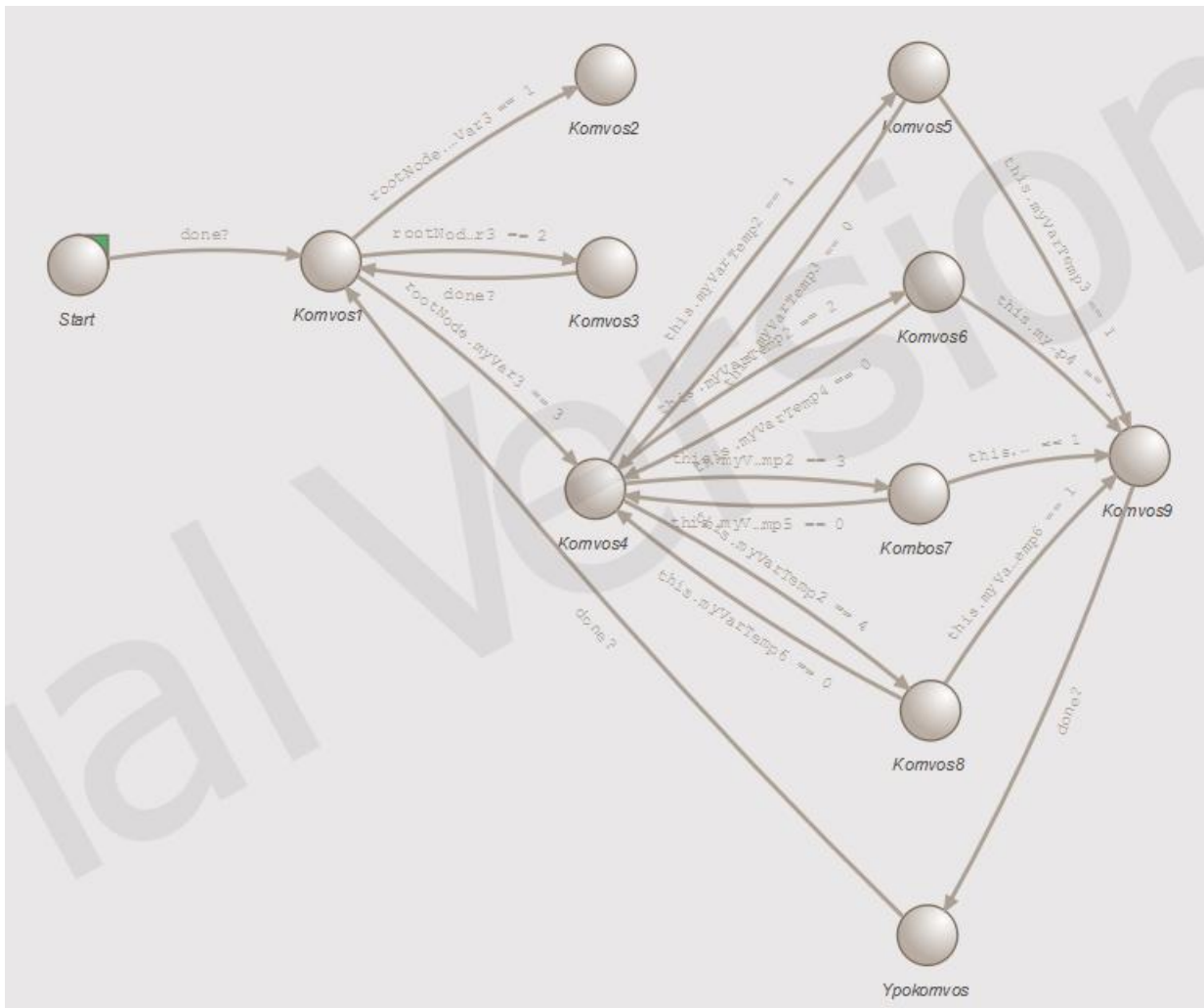
²http://users.aldebaran-robotics.com/docs/site_en/reddoc/framework/framework.html

- ExtraInfo = [MarkID]. Το MarkID δίνει τον αριθμό που αντιπροσωπεύει κάθε Naomark και είναι αυτό που στην ουσία μας ενδιαφέρει εμάς. Κάθε Naomark, δηλαδή, χαρακτηρίζεται από ένα ξεχωριστό αριθμό. Αυτός ο αριθμός, δίνεται από αυτό το πεδίο. Οπότε, στην ουσία εμείς από όλες αυτές τις διαθέσιμες μεταβλητές, αυτή είναι θα χρησιμοποιήσουμε, αφού αναγνωρίζοντας αυτό το μοναδικό για κάθε Naomark αριθμό, θα μπορούμε να αντιστοιχίσουμε το κάθε Naomark με ένα αριθμό νομίσιματος οπότε στην ουσία να αναγνωρίσουμε πιο νόμισμα μας έχει δώσει το παιδί.

4.3 Παρουσίαση της υλοποίησης

Θα παρουσιάσουμε τώρα, την υλοποίηση του σεναρίου που έχουμε φτιάξει, στο Gostai Studio.

Καταρχάς, παραθέτουμε μια εικόνα η οποία μας δείχνει τη γραφική υλοποίηση των διαγραμμάτων κατάστασης, όπως την έχουμε αναφέρει πιο πάνω, και όπως αυτή έχει υλοποιηθεί με τη χρήση του Gostai Studio.



Σχήμα 4.4: Γραφική αναπαράσταση σεναρίου

Όπως βλέπουμε, έχουμε δημιουργήσει τους 10 κόμβους που έχουμε αναφέρει στο προηγούμενο κεφάλαιο υλοποίησης του σεναρίου, ενώ χρειάστηκε και η δημιουργία ενός ακόμη ενδιαμέσου κόμβου (Υροκοινvos) απλά για τη μετάβαση από τον κόμβο 9 στον κόμβο 1.

Θα αναλύσουμε τώρα, τι ακριβώς εκτελείται από κάθε κόμβο ξεχωριστά και ποια είναι η συνθήκη που χρησιμοποιείται για τη μετάβαση από ένα κόμβο σε άλλο.

Κόμβος S:

Ο κόμβος αυτός έχει τεθεί ως ο αρχικός κόμβος του σεναρίου (Set as start node). Δεν εκτελείται οποιαδήποτε εντολή σε αυτόν τον κόμβο. Η χρησιμότητά του είναι καθαρά για λόγους τυπικότητας, για να τεθεί, δηλαδή, ως αρχικός κόμβος ένας ξεχωριστός κόμβος.

Για τη μετάβαση από τον κόμβο S στον κόμβο 1, χρησιμοποιείται ο ακόλουθος κώδικας:

Μετάβαση από S -> 1

```
1 done?      /*This means that we will go to the next node*/
2           /*when all the commands of the previous node*/
3           /*have been executed*/
```

Κόμβος 1:

Ο κόμβος αυτός, είναι ο κόμβος στον οποίο το ρομπότ θα ανακοινώσει το αντικείμενο και την τιμή του στο παιδί ανάλογα με μια επιλογή που θα έχει διαλέξει ο επιβλέπωντας. Έπειτα, το παιδί θα τοποθετήσει τα νομίσματα μπροστά στο οπτικό πεδίο του ρομπότ για να μπορέσει να εφαρμόσει οπτική αναγνώριση για να δει πόσα ακριβώς χρήματα του έχει βάλει το παιδί και ανάλογα από τα χρήματα αυτά θα μεταφερθούμε στον επόμενο κόμβο. Για να το πράξει αυτό, το ρομπότ θα σκύβει το κεφάλι του προς τα κάτω και θα εφαρμόζουμε το ειδικό module οπτικής αναγνώρισης που θα αναφέρουμε πιο κάτω.

Ο κώδικας του κόμβου αυτού, χωρίζεται σε δυο μέρη: στον κώδικα που αναγράφεται στο OnEnter κομμάτι του κόμβου (ο κώδικας αυτός θα εκτελεστεί πριν να μπούμε στον κόμβο) και στον κώδικα που θα εκτελεστεί όταν βρισκόμαστε στον κόμβο. Στο πρώτο μέρος, θα γίνουν όλες οι δηλώσεις των μεταβλητών οι οποίες είτε θα έχουν γενική εμβέλεια και θα χρησιμοποιηθούν και από τους υπόλοιπους κόμβους, είτε θα έχουν τοπική εμβέλεια και θα χρησιμοποιηθούν μόνο από το συγκεκριμένο κόμβο και από τις μεταβάσεις που φεύγουν από αυτόν τον κόμβο. Οπότε έχουμε:

OnEnter κώδικας του κόμβου 1

```
1 /*Code executed before entering the node*/
2 if (!rootNode.hasLocalSlot("myVar1")) var rootNode.myVar1=0;
3
4 if (!rootNode.hasLocalSlot("myVar2")) var rootNode.myVar2=0;
5
6 if (!rootNode.hasLocalSlot("myVar3")) var rootNode.myVar3=0;
7
8 if (!rootNode.hasLocalSlot("myPrice")) var rootNode.myPrice=0;
9
10 if (!rootNode.hasLocalSlot("myName")) var rootNode.myName=0;
11
12 if (!this.hasSlot("myVarTemp")) var this.myVarTemp=1;
13
14 if (!this.hasSlot("myPeriod")) var this.myPeriod=0;
15
```

```

16 if (!this.hasSlot("markProxy")) var this.markProxy=0;
17
18 if (!this.hasSlot("myData")) var this.myData=1;
19
20 if (!this.hasSlot("myExtra")) var this.myExtra1=0;
21
22 if (!this.hasSlot("myExtra")) var this.myExtra2=0;
23
24 if (!this.hasSlot("myExtra")) var this.myExtra3=0;

```

Τώρα, ακολουθεί ο κύριος κώδικας του κόμβου αυτού, όπου θα γίνουν όλες οι ενέργειες που έχουμε αναφέρει πιο πάνω:

Κυρίως κώδικας του κόμβου 1

```

1  /*Choose object by the value put in this.myVarTemp*/
2  switch(this.myVarTemp)
3  {
4  case 1:{rootNode.myPrice=2;
5           rootNode.myName="a";};
6  case 2:{rootNode.myPrice=3;
7           rootNode.myName="b";};
8  case 3:{rootNode.myPrice=5;
9           rootNode.myName="c";};
10 case 4:{rootNode.myPrice=6;
11          rootNode.myName="d";};
12 };
13
14 /*The name and the price of the object is announced */
15 tts.say("Object_is_%.0f" % rootNode.myName);
16 sleep(2s);
17 tts.say("The_price_is_%.0f" % rootNode.myPrice);
18
19 /*Turning the head towards the table*/
20 headPitch.val=10 time:2s;
21
22 /*Naomarks recognition*/
23 this.markProxy = ALProxy("LandMarkDetection", [],
24 ["subscribe","unsubscribe"]);
25
26 this.myPeriod = 500;
27
28 /*Enable detection*/
29 this.markProxy.subscribe("Test_Mark", this.myPeriod, 0.0 );
30
31 /*Get data from landmark detection */
32 this.myData = robot.ALMemory.getData("LandmarkDetected");
33
34 /*Repeat 10 times or until recognizing a mark */
35 var i = 1;
36 while ((this.myData == []) && (i<11))

```

```

37 {
38     this.myData = robot.ALMemory.getData("LandmarkDetected");
39     i += 1;
40 };
41
42 /*Wait for the coin to be recognised*/
43 sleep(3s);
44
45 /*Disable detection*/
46 this.markProxy.unsubscribe("TestMark", period, 0.0);
47
48 switch(this.myData[1][0][1][0])
49 {
50 case 64: this.myExtra1 = 1;
51 case 108: this.myExtra1 = 2;
52 case 85: this.myExtra1 = 4;
53 case 119: this.myExtra1 = 5;
54 };
55
56 switch(this.myData[1][1][1][0])
57 {
58 case 64: this.myExtra2 = 1;
59 case 108: this.myExtra2 = 2;
60 case 85: this.myExtra2 = 4;
61 case 119: this.myExtra2 = 5;
62 };
63
64 switch(this.myData[1][2][1][0])
65 {
66 case 64: this.myExtra3 = 1;
67 case 108: this.myExtra3 = 2;
68 case 85: this.myExtra3 = 4;
69 case 119: this.myExtra3 = 5;
70 };
71
72 if (this.myData[1][0][1][0])
73     rootNode.myVar2 = rootNode.myVar2 + this.myExtra1;
74
75 if (this.myData[1][1][1][0])
76     rootNode.myVar2 = rootNode.myVar2 + this.myExtra2;
77
78 if (this.myData[1][2][1][0])
79     rootNode.myVar2 = rootNode.myVar2 + this.myExtra3;
80
81
82 /*This is the value the child has to pay*/
83 rootNode.myVar1 = roodNode.myPrice - rootNode.myVar2;
84
85 /*Turning the head back to the initial state*/
86 body.headPitch.val=0 time:2s;
87

```

```

88 /*Comparing the price with the money given to the robot*/
89 if (rootNode.myVar1 == 0)
90     rootNode.myVar3 = 1
91     else if (rootNode.myVar1 > 0)
92         rootNode.myVar3 = 2
93         else if (rootNode.myVar1 < 0)
94             rootNode.myVar3 = 3;

```

Τέλος, έχουμε όλες τις μεταβάσεις που φεύγουν από τον κόμβο 1 προς τους γειτονικούς κόμβους:

Μεταβάσεις από τον κόμβο 1

```

1 /*For transition from Node 1 to Node 2*/
2 rootNode.myVar3 == 1
3
4 /*For transition from Node 1 to Node 3*/
5 rootNode.myVar3 == 2
6
7 /*For transition from Node 1 to Node 4*/
8 rootNode.myVar3 == 3

```

Κόμβος 2:

Ο κόμβος αυτός είναι και ο τελικός κόμβος, αφού εδώ θα μεταβούμε από τον κόμβο 1 αν η αξία των νομισμάτων που έχει δώσει το παιδί στο ρομπότ είναι ίση με την αξία του αντικειμένου που θέλει να πάρει το παιδί. Το ρομπότ τότε, θα αναφωνήσει τη λέξη "Right" δείχνοντας στο παιδί ότι έχει δώσει τα σωστά νομίσματα και ταυτόχρονα θα σηκώσει τα χέρια του ψηλά θέλοντας να δείξει στο παιδί ότι πλέον δικαιούται να παραλάβει το αντικείμενο που θέλει. Ο κώδικας του συγκεκριμένου κόμβου είναι ο ακόλουθος:

Κώδικας κόμβου 2

```

1 /*Robot says "Right"*/
2 tts.say("Right");
3
4 /*Nao lifts up its hands*/
5 /*Right hand*/
6 elbowRollR.val = -pi |
7 shoulderPitchR.val = -pi/2 |
8 wristYawR.val = -pi/2 |
9 handR.val = 1;
10
11 /*Left hand*/
12 elbowRollL.val = pi |
13 shoulderPitchL.val = -pi/2 |
14 wristYawL.val = pi/2 |
15 handL.val = 1;

```

Κόμβος 3:

Στον κόμβο αυτό, θα φθάσουμε από τον κόμβο 1 όταν το παιδί δώσει στο ρομπότ νομίσματα αξίας μεγαλύτερης από την πραγματική αξία του αντικειμένου που θέλει το παιδί να αγοράσει. Σε αυτή την περίπτωση, το ρομπότ αναφωνεί "Try again" δείχνοντας στο παιδί ότι στην ουσία έβαλε περισσότερα νομίσματα και ότι πρέπει να ξαναδοκιμάσει βάζοντας αυτή τη φορά το σωστό αριθμό νομισμάτων. Παραθέτουμε τον κώδικα του κόμβου αυτού όπως και τον κώδικα της μετάβασης στον κόμβο 1 όπου και θα επιστρέψουμε για να βάλει το παιδί το σωστό αριθμό νομισμάτων.

Κώδικας κόμβου 3 μαζί με τη μετάβαση στον κόμβο 1

```
1 /*Nao says "Try again"*/
2 tts.say("Try_again");
3
4 /*The transition from Node 3 back to Node 1*/
5 done?
```

Κόμβος 4:

Ο κόμβος αυτός, είναι ο κόμβος στον οποίο θα μεταβούμε αν στον κόμβο 1 η αξία των νομισμάτων που έχει δώσει το παιδί στο ρομπότ είναι μικρότερη από την αξία του αντικειμένου που θέλει το παιδί. Οπότε το ρομπότ, θα επαναλάβει στο παιδί την αξία του αντικειμένου δείχνοντάς του ότι πρέπει έτσι να αποκτήσει και άλλα νομίσματα και θα πρέπει να εκτελέσει μια αποστολή την οποία και θα του ανακοινώσει.

Και εδώ ο κώδικάς μας χωρίζεται σε δυο μέρη, στις δηλώσεις των μεταβλητών που γίνονται πριν εισέλθουμε στον κώδικα, και στο κύριο μέρος του κώδικα. Αυτά θα τα περιγράψουμε ως ένα κομμάτι κώδικα, οπότε για τον κόμβο 4 έχουμε τα ακόλουθα:

Κώδικας κόμβου 4

```
1 /*This is in the OnEnter part of the Node*/
2 if (!this.hasSlot("myVarTemp2")) var this.myVarTemp2=0;
3
4 if (!this.hasSlot("myVarTemp22")) var this.myVarTemp22=0;
5
6 if (!rootNode.hasLocalSlot("myNumber")) var rootNode.myNumber = 0;
7
8
9 /*This is the main part of the code of the Node*/
10 /*Robot says the price of the object*/
11 tts.say("The_price_is_%0f_coins" % rootNode.myPrice);
12
13 switch(rootNode.myVar1)
14 {
15 case 1: this.myVarTemp2=1;
16 case 2: this.myVarTemp2=2;
17 case 4: this.myVarTemp2=3;
18 case 5: this.myVarTemp2=4;
19 };
20
```



```

21 /*Robot announces the mission the child has to accomplish*/
22 tts.say("Mission_%.0f" % this.myVarTemp2);

```

Από τον κόμβο 4, ανάλογα με το πόσα χρήματα χρειάζεται το παιδί θα μεταβούμε σε μια από τις 4 διαθέσιμες αποστολές. Αυτό θα γίνει με μια κοινή μεταβλητή που ανάλογα από την τιμή που της έχουμε δώσει μέσα από ένα switch στο κυρίως κομμάτι του κώδικα του κόμβου 4, θα μας στέλνει στην κατάλληλη αποστολή. Επομένως, έχουμε τα ακόλουθα για τις μεταβάσεις που φεύγουν από τον κόμβο 4:

Μεταβάσεις που φεύγουν από τον κόμβο 4

```

1 this.myVarTemp2 == 1 /*For entering the first mission*/
2
3 this.myVarTemp2 == 2 /*For entering the second mission*/
4
5 this.myVarTemp2 == 3 /*For entering the third mission*/
6
7 this.myVarTemp2 == 4 /*For entering the fourth mission*/

```

Κόμβος 5:

Στον κόμβο αυτό θα μεταβούμε όταν το παιδί χρειάζεται ακόμη 1 επιπλέον ευρώ για να πάρει το αντικείμενο που χρειάζεται. Εδώ θα εκτελεσθεί η πρώτη αποστολή, που θα είναι να πει το παιδί στο ρομπότ πόσα χρήματα χρειάζεται ακόμη για να αποκτήσει το αντικείμενό του. Αν απαντήσει σωστά, πάμε στον κόμβο 9, αλλιώς επιστρέφουμε πίσω στον κόμβο 4. Οπότε έχουμε τα ακόλουθα κομμάτια κώδικα:

Κώδικας κόμβου 5

```

1 /*This is in the OnEnter part of the Node*/
2 if (!this.hasSlot("myVarTemp3")) var this.myVarTemp3=0;
3
4 if (!this.hasSlot("data1")) var this.data1=0;
5
6
7 /*This is the main part of the code of the Node*/
8 /*Nao asks the child to say how much money he needs*/
9 tts.say("How much money do you need");
10
11 sleep(5s); /*Delay 5 seconds*/
12
13 /*Voice Recognition*/
14 var asr=ALProxy("SpeechRecognition",[],["setLanguage",
15 "subscribe","unsubscribe"]);
16 asr.setLanguage("English");
17 var wordlist=["one"];
18 asr.setWordListAsVocabulary(wordlist);
19
20 /*Enable recognition*/
21 asr.subscribe("MyModule");
22

```

```

23 /*Get data from voice recognition*/
24 this.data1 = robot.ALMemory.getData_string("WordRecognized");
25
26 /*Repeat 10 times or until recognizing a mark */
27 var j = 1;
28 while ((this.data1[0] == "") && (j<11))
29 {
30     this.data1 = robot.ALMemory.getData_string("WordRecognized");
31     j += 1;
32 };
33
34
35 /*Wait for the word to be recognised*/
36 sleep(3s);
37
38 /*Disable recognition*/
39 asr.unsubscribe("MyModule");
40
41
42 if (this.data1[0] == "one") /*Checking the answer given*/
43     this.myVarTemp3 = 1
44     else {this.myVarTemp3 = 0;
45           tts.say("Wrong.Repeat");
46
47 sleep(2s);
48
49 /*Set this variable to 1 for use in Node 9*/
50 rootNode.myNumber = 1;

```

Για τις μεταβάσεις από τον κόμβο 5, έχουμε 2 περιπτώσεις. Αν το παιδί απαντήσει σωστά δίνεται στη μεταβλητή myVarTemp3 η τιμή 1 οπότε και πάμε στον κόμβο 9, ενώ αν απαντήσει λάθος, δίνεται στη μεταβλητή myVarTemp3 η τιμή 0 οπότε και μεταβαίνουμε πίσω στον κόμβο 4 για να ξαναπροσπαθήσει το παιδί.

Μεταβάσεις κόμβου 5

```

1 this.myVarTemp3 == 1 /*For entering Node 9*/
2
3 this.myVarTemp3 == 0 /*For entering Node 4*/

```

Κόμβος 6:

Στον κόμβο αυτό θα μεταβούμε όταν το παιδί χρειάζεται ακόμη 2 επιπλέον ευρώ για να πάρει το αντικείμενο που χρειάζεται. Εδώ θα εκτελεσθεί η δεύτερη αποστολή, που θα είναι να πει κτυπήσει τόσες φορές όσες τα εναπομείναντα χρήματα τον μπροστινό αισθητήρα αφής του κεφαλιού του Nao. Το κεφάλι του Nao είναι εφοδιασμένο με 3 αισθητήρες αφής, ένα στο μπροστινό μέρος του κεφαλιού, ένα στη μέση και ένα προς το πίσω μέρος του κεφαλιού. Εμείς ζητάμε από το παιδί να κτυπήσει στην ουσία 2 φορές τον αισθητήρα αυτό. Επειδή, όμως, υπάρχει ένα πρόβλημα ανάγνωσης διαδοχικών αγγιγμάτων από το ρομπότ, εισάξαμε μια καθυστέρηση 3 δευτερολέπτων πριν και μετά από τον έλεγχο του αισθητήρα. Επομένως, δίνουμε ένα χρονικό παράθυρο στο παιδί για να πατήσει τον αισθητήρα. Μάλιστα, για να δώσουμε στο παιδί να καταλάβει πότε πρέπει να πατήσει τον

αισθητήρα, βάλουμε το ρομπότ να ανοίγει τα δάχτυλα του αριστερού του χεριού 3 δευτερόλεπτα πριν να ανιχνευθεί το πάτημα και να κλείνει τα δάχτυλα 3 δευτερόλεπτα μετά. Έτσι, το παιδί, όταν θα δει τα δάχτυλα του αριστερού χεριού του Nao να ανοίγουν θα καταλαβαίνει ότι πρέπει να πατήσει μια φορά τον αισθητήρα αφής του κεφαλιού του Nao. Θα περιμένει έπειτα να κλείσουν τα δάχτυλά του και να ξαναανοίξουν για να ξαναπατήσει τον αισθητήρα. Η διαδικασία αυτή θα επαναληφθεί 5 φορές αλλά το παιδί καλείται να πατήσει τον αισθητήρα αφής μόνο 2 φορές, όσες και τα νομίσματα που χρειάζεται επιπλέον για να αποκτήσει το αντικείμενό του. Επομένως, έχουμε τον ακόλουθο κώδικα για τον κόμβο αυτό:

Κώδικας κόμβου 6

```

1  /*This is in the OnEnter part of the Node*/
2  if (!this.hasSlot("myVarTemp4")) var this.myVarTemp4=0;
3
4  if (!this.hasSlot("metavliti")) var this.metavliti=0;
5
6
7  /*This is the main part of the code of the Node*/
8  /*Robot asks the child to touch its front head sensor as many*/
9  /*times as the money he needs*/
10 tts.say("Touch_my_front_head_sensor_as_many_times_as_the_money_you_need");
11
12 /*Wait 5 seconds to start checking if the sensor is pressed*/
13 sleep(5s);
14
15 for (var x = 1; x < 6; x += 1)
16 {
17     handL.val = 1;
18     sleep(3s);
19     if (body.headTouchF.val == 1)
20         this.metavliti = this.metavliti + 1
21         else this.metavliti = this.metavliti + 0;
22     handL.val = 0;
23     sleep(3s);
24 };
25
26
27 if (this.metavliti == 2)
28     this.myVarTemp4 = 1
29     else {this.myVarTemp4 = 0;
30         tts.say("Wrong.Repeat");
31     }
32 sleep(2s);
33
34 /*Set this variable to 2 for use in Node 9*/
35 rootNode.myNumber = 2;

```

Για τις μεταβάσεις του κόμβου 6, έχουμε και πάλι 2 πιθανές μεταβάσεις, ανάλογα με την επιτυχία ή αποτυχία της αποστολής.

Μεταβάσεις κόμβου 6

```
1 this.myVarTemp4 = 1 /*For entering node 9 */
2
3 this.myVarTemp4 = 0 /*For entering node 4 */
```

Κόμβος 7:

Στον κόμβο αυτό θα μεταβούμε όταν το παιδί χρειάζεται ακόμη 4 επιπλέον ευρώ για να πάρει το αντικείμενο που χρειάζεται. Εδώ θα εκτελεσθεί η τρίτη αποστολή, που θα είναι σε συνεργασία με το Arduino και θα πρέπει το παιδί να διαλέξει το σωστό κουτί που αναγράφει την τιμή που χρειάζεται ακόμη για να αποκτήσει το αντικείμενο και έπειτα το ρομπότ να αναγνωρίσει ένα ήχο που θα ακούσει από το Arduino. Επομένως, έχουμε τα ακόλουθα:

Κώδικας κόμβου 7

```
1 /*This is in the OnEnter part of the Node*/
2 if (!this.hasSlot("myVarTemp5")) var this.myVarTemp5=0;
3
4 if (!this.hasSlot("data2")) var this.data2=0;
5
6
7 /*This is the main part of the code of the Node*/
8 /*The robot will say that the child must choice the right box*/
9 tts.say("Choose_the_box_which_has_the_amount_of_money_you_need");
10
11 sleep(5s); /*Delay 5 seconds*/
12
13 /*Voice Recognition*/
14 var asr=ALProxy("SpeechRecognition",[],["setLanguage",
15 "subscribe","unsubscribe"]);
16 asr.setLanguage("English");
17 var wordlist=["yes"];
18 asr.setWordListAsVocabulary(wordlist);
19
20 /*Enable recognition*/
21 asr.subscribe("MyModule");
22
23 /*Get data from voice recognition*/
24 this.data2 = robot.ALMemory.getData_string("WordRecognized");
25
26 /*Repeat 10 times or until recognizing a mark */
27 var k = 1;
28 while ((this.data1[0] == "") && (k<11))
29 {
30     this.data2 = robot.ALMemory.getData_string("WordRecognized");
31     k += 1;
32 };
33
34 /*Wait for the word to be recognised*/
35 sleep(3s);
36
```

```

37 /*Disable recognition*/
38 asr.unsubscribe("MyModule");
39
40 if (this.data2[0] == "yes")
41     this.myVarTemp5 = 1
42     else {this.myVarTemp5 = 0;
43           tts.say("Wrong. Repeat");
44
45 sleep(2s);
46
47 /*Set this variable to 3 for use in Node 9*/
48 rootNode.myNumber = 3;

```

Για τις μεταβάσεις του κόμβου 7, έχουμε και πάλι 2 πιθανές μεταβάσεις, ανάλογα με την επιτυχία ή αποτυχία της αποστολής.

Μεταβάσεις κόμβου 7

```

1 this.myVarTemp5 == 1 /*For entering node 9 */
2
3 this.myVarTemp5 == 0 /*For entering node 4 */

```

Κόμβος 8:

Στον κόμβο αυτό θα μεταβούμε όταν το παιδί χρειάζεται ακόμη 5 επιπλέον ευρώ για να πάρει το αντικείμενο που χρειάζεται. Εδώ θα εκτελεσθεί η τέταρτη αποστολή, που θα είναι σε συνεργασία με το Arduino και θα πρέπει το παιδί να πατήσει από ένα πληκτρολόγιο το κουμπί στο οποίο αναγράφεται η σωστή τιμή των χρημάτων που απομένουν για να πάρει το αντικείμενο και έπειτα το ρομπότ να αναγνωρίσει ένα ήχο που θα ακούσει από το Arduino. Επομένως, έχουμε τα ακόλουθα:

Κώδικας κόμβου 8

```

1 /*This is in the OnEnter part of the Node*/
2 if (!this.hasSlot("myVarTemp6")) var this.myVarTemp6=0;
3
4 if (!this.hasSlot("data2")) var this.data2=0;
5
6
7 /*This is the main part of the code of the Node*/
8 /*The robot will say that the child must push the right button*/
9 tts.say("Push_the_button_which_has_the_amount_of_money_you_need");
10
11 sleep(5s); /*Delay 5 seconds*/
12
13 /*Voice Recognition*/
14 var asr=ALProxy("SpeechRecognition",[],["setLanguage",
15 "subscribe","unsubscribe"]);
16 asr.setLanguage("English");
17 var wordlist=["yes"];
18 asr.setWordListAsVocabulary(wordlist);
19

```

```

20 /*Enable recognition*/
21 asr.subscribe("MyModule");
22
23 /*Get data from voice recognition*/
24 this.data3 = robot.ALMemory.getData_string("WordRecognized");
25
26 /*Repeat 10 times or until recognizing a mark */
27 var k = 1;
28 while ((this.data1[0] == "") && (k<11))
29 {
30     this.data3 = robot.ALMemory.getData_string("WordRecognized");
31     k += 1;
32 };
33
34 /*Wait for the word to be recognised*/
35 sleep(3s);
36
37 /*Disable recognition*/
38 asr.unsubscribe("MyModule");
39
40 if (this.data3[0] == "yes")
41     this.myVarTemp6 = 1
42     else {this.myVarTemp6 = 0;
43         tts.say("Wrong.Repeat");
44
45     sleep(2s);
46
47 /*Set this variable to 4 for use in Node 9*/
48 rootNode.myNumber = 4;

```

Για τις μεταβάσεις του κόμβου 8, έχουμε και πάλι 2 πιθανές μεταβάσεις, ανάλογα με την επιτυχία ή αποτυχία της αποστολής.

Μεταβάσεις κόμβου 8

```

1 this.myVarTemp6 == 1 /*For entering node 9 */
2
3 this.myVarTemp6 == 0 /*For entering node 4 */

```

Κόμβος 9:

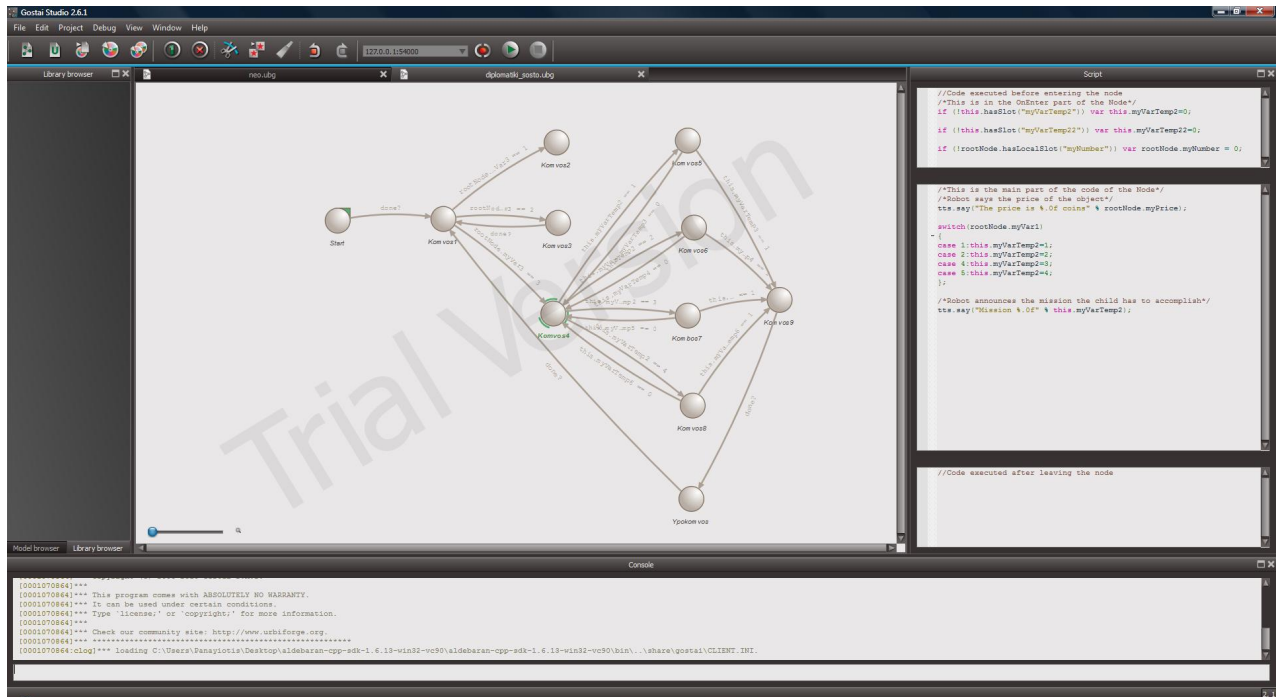
Στον κόμβο αυτό θα φτάνουμε με επιτυχή εκπλήρωση των αποστολών που πρέπει να εκτελέσει το παιδί. Αφού έχουμε λοιπόν επιτυχία της αποστολής που ανατέθηκε στο παιδί, το ρομπότ θα σηκώσει τα χέρια ψηλά ως ένδειξη επιτυχίας και θα φωνάξει στο παιδί ότι εκτέλεσε σωστά την αποστολή και ότι μπορεί να πάρει τα νομίσματα που χρειάζεται. Μετά το πέρας αυτών των εντολών, θα επιστρέψουμε στον κόμβο ένα για να επαναληφθεί η διαδικασία καταμέτρησης των νομισμάτων αφού πλέον το παιδί διαθέτει το σωστό αριθμό νομισμάτων και να μπορέσουμε έτσι να οδηγηθούμε στον τελικό κόμβο του σεναρίου. Σημειώνεται, ότι μεταξύ του κόμβου 9 και του κόμβου 1 παρεμβάλλεται ένας *Υποκόμβος* ο οποίος στην ουσία χρησιμοποιείται για λόγους σύνδεσης των δυο κόμβων, χωρίς καμιά πρακτική εφαρμογή. Ο κώδικας του κόμβου 9 μαζί με τον κώδικα της μετάβασης το στον κόμβο 1 είναι ο ακόλουθος:

Κώδικας κόμβου 9

```
1  /*This is in the OnEnter part of the Node*/
2  if (!this.hasSlot("myVarTemp7")) var this.myVarTemp7=0;
3
4
5  /*Main code of Node 9*/
6  /*The robot raises up its hands*/
7  /*Right hand*/
8  elbowRollR.val = -pi |
9  shoulderPitchR.val = -pi/2 |
10 wristYawR.val = -pi/2 |
11 handR.val = 1;
12
13 /*Left hand*/
14 elbowRollL.val = pi |
15 shoulderPitchL.val = -pi/2 |
16 wristYawL.val = pi/2 |
17 handL.val = 1;
18
19 switch(rootNode.myNumber)
20 {
21 case 1: this.myVarTemp7= 1;
22 case 2: this.myVarTemp7= 2;
23 case 3: this.myVarTemp7= 4;
24 case 4: this.myVarTemp7= 5;
25 };
26
27 /*Robot announces the correct result*/
28 tts.say("Right._Take_%.0f_coins" % this.myVarTemp7);
29
30 /*Wait for five seconds*/
31 sleep(5s);
32
33 /*Return to the initial potition of the hands*/
34 /*Right hand*/
35 elbowRollR.val = pi/2 |
36 shoulderPitchR.val = pi |
37 wristYawR.val = -pi |
38 handR.val = 0;
39
40 /*Left hand*/
41 elbowRollL.val = -pi/2 |
42 shoulderPitchL.val = pi |
43 wristYawL.val = pi |
44 handL.val = 0;
45
46
47 /*Transition code from Node 9 to Ypokomvos and */
48 /*from Ypokomvos to Node 1 */
49 done?
```

4.4 Παραδείγματα χρήσης

Στο κομμάτι αυτό θα παρουσιάσουμε ένα παράδειγμα χρήσης του Gostai Studio, όπου φαίνεται εμφανώς το διάγραμμα καταστάσεων που σχεδιάσαμε ενώ έχουμε επιλέξει τον Κόμβο 4 για να δείξουμε πώς γίνεται η εισαγωγή του κώδικα, τόσο του κώδικα που εκτελείται πριν μπούμε στον κόμβο όσο και αυτού που εκτελείται όσο βρισκόμαστε μέσα στον κόμβο. Επίσης, στο συγκεκριμένο παράδειγμα, έχουμε συνδεθεί με ένα τοπικό NaoQi Framework v.1.6.13 οπότε το μόνο που απομένει στον χρήστη είναι να πατήσει το κουμπί "Play" για να τρέξει το σενάριο.



Σχήμα 4.5: Παράδειγμα χρήσης του Gostai Studio

Επίσης, είναι σημαντικό να αναφέρουμε, ότι για τη δοκιμή αρκετών από των εντολών, προτού δημιουργήσουμε το σενάριο στην τελική του μορφή όπως έχει παρουσιαστεί πιο πάνω, χρησιμοποιήσαμε ένα άλλο πρόγραμμα διαθέσιμο από την Gostai, το Gostai Console, με το οποίο στην ουσία μπορείς να στείλεις μεμονωμένες εντολές είτε σε έναν τοπικό server είτε στο Nao το ίδιο. Έτσι, πρώτου χρησιμοποιήσουμε απευθείας το Gostai Studio με τη χρήση των διαγραμμάτων κατάστασης, δοκιμάζαμε όλες τις εντολές μας στο Gostai Console, όπου μπορούν να δοκιμαστούν μια προς μια οι εντολές αλλά υπάρχει και η δυνατότητα να δοκιμαστεί μια σειρά απο εντολές, ένα τμήματ ολόκληρου κώδικα. Επομένως, το Gostai Console έπαιξε βασικό ρόλο στην επιβεβαίωση της ορθότητας των εντολών μας. Ακολουθεί ένα παράδειγμα χρήσης του Gostai Console με μια τυπική χρήση εντολών, ενώ μάλιστα είναι συνδεδεμένο σε ένα τοπικό NaoQi Framework και στο Choregraphe, τα οποία θα αναλύσουμε σε επόμενο κεφάλαιο.


```
[0001720076]*** *****  
[0001720076]*** Urbi SDK version 2.0 rev. 96a4b2f  
[0001720076]*** Copyright (C) 2005-2010 Gostai S.A.S.  
[0001720076]***  
[0001720076]*** This program comes with ABSOLUTELY NO WARRANTY.  
[0001720076]*** It can be used under certain conditions.  
[0001720076]*** Type `license;' or `copyright;' for more information.  
[0001720076]***  
[0001720076]*** Check our community site: http://www.urbiforge.org.  
[0001720076]*** *****  
[0001720076:clog]*** loading C:\Users\Panayiotis\Desktop\aldebaran-cpp-sdk-1.6.13-  
win32-vc90\aldebaran-cpp-sdk-1.6.13-win32-vc90\bin\..\share\gostai\CLIENT.INI.  
=> elbowRollL.val = pi |  
shoulderPitchL.val = -pi/2 |  
wristYawL.val = pi/2 |  
handL.val = 1;  
[0001723065]1
```

Σχήμα 4.6: Παράδειγμα χρήσης του Gostai Console

Κεφάλαιο 5

Αποστολές με Arduino

5.1 Ανάλυση των Αποστολών

Όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο, οι αποστολές που είναι διαθέσιμες για να εκτελέσει το παιδί σε περίπτωση που χρειάζεται περισσότερα νομίσματα για να αποκτήσει το αντικείμενο που επιθυμεί, είναι 4. Εξ' αυτών οι 2 είναι αποστολές που αφορούν καθαρά το ρομπότ Nao και οι άλλες δυο αποτελούν μια συνεργασία του Nao με την πλατφόρμα Arduino. Σε αυτό το κεφάλαιο θα ασχοληθούμε αποκλειστικά με τις αποστολές που περιλαμβάνουν και τη συμμετοχή του Arduino, αφού οι δυο πρώτες αποστολές έχουν μελετηθεί σε προηγούμενο κεφάλαιο και δεν χρίζουν περαιτέρω αναφοράς.

Η πρώτη αποστολή που θα μας απασχολήσει (η τρίτη κατά σειρά από τις 4) είναι η αποστολή που θα δώσει στο παιδί 4 ευρώ για να αποκτήσει το αντικείμενο που θέλει. Στην αποστολή αυτή, το παιδί θα πρέπει να διαλέξει από τρία διαθέσιμα κουτιά, αυτό στο οποίο αναγράφεται η σωστή τιμή των χρημάτων που χρειάζεται το παιδί για να αποκτήσει το αντικείμενο που θέλει (δηλαδή στην περίπτωση αυτή η τιμή 4). Το ρομπότ, θα ανακοινώσει καταρχάς στο παιδί ότι πρέπει να διαλέξει το κουτί στο οποίο αναγράφεται η σωστή τιμή των χρημάτων που χρειάζεται το παιδί. Θα υπάρχουν 3 κουτιά εκ των οποίων το ένα θα αναγράφει τη σωστή τιμή και τα άλλα 2 θα γράφουν λάθος τιμές. Τα 3 αυτά κουτιά, θα είναι συνδεδεμένα με διακόπτες πίεσης οι οποίοι θα ανιχνεύουν τότε ανοίγει το κουτί. Όταν το παιδί ανοίξει ένα κουτί, τότε ο διακόπτης του κουτιού που θα είναι συνδεδεμένος με το Arduino θα στέλνει ένα σήμα με το οποίο θα ενημερώνεται το Arduino ότι έχει ανοίξει το συγκεκριμένο κουτί. Τότε, θα ενεργοποιείται από το Arduino ένα κύκλωμα παραγωγής φωνής το οποίο κατασκευάσαμε και θα εξηγήσουμε στη συνέχεια, το οποίο ανάλογα με το πιο κουτί άνοιξε το παιδί, θα λέει και μια συγκεκριμένη λέξη. Αν το κουτί που έχει ανοίξει το παιδί είναι το σωστό κουτί, τότε το κύκλωμα παραγωγής φωνής θα λέει τη λέξη "Yes". Αν το παιδί έχει ανοίξει λάθος κουτί, τότε θα παραχθεί η λέξη "No". Και στις δυο περιπτώσεις, το Nao με την ικανότητα αναγνώρισης φωνής που διαθέτει (voice recognition) θα αναγνωρίσει ποια από τις δυο λέξεις έχει λεχθεί από το ρομπότ. Αν αναγνωρίσει τη λέξη "Yes", αυτό σημαίνει επιτυχία της αποστολής οπότε θα μεταβούμε στον κόμβο 9 για να πάρει τα νομίσματα που χρειάζεται το παιδί. Αν η λέξη που θα αναγνωριστεί είναι το "No" τότε έχουμε ανεπιτυχή ολοκλήρωση της αποστολής με αποτέλεσμα να επιστρέψουμε πίσω στον κόμβο για να επαναληφθεί η δοκιμασία.

Η δεύτερη αποστολή (η τελευταία από τις 4 διαθέσιμες) είναι η αποστολή που θα δώσει στο παιδί 5 ευρώ για να αποκτήσει το αντικείμενο που θέλει. Στην αποστολή αυτή, το παιδί θα καλείται να χρησιμοποιήσει ένα χειροποίητο πληκτρολόγιο το οποίο εμείς κατασκευάσαμε, πατώντας μέσω αυτού, το κουμπί στο οποίο αναγράφεται η σωστή τιμή των χρημάτων που χρειάζεται το παιδί ακόμη για να αποκτήσει το αντικείμενο που ζητάει. Το ρομπότ, θα ανακοινώνει και πάλι στο παιδί ότι πρέπει να πατήσει το κουμπί στο οποίο αναγράφεται η σωστή τιμή των χρημάτων που χρειάζεται

(τα 5 ευρώ δηλαδή). Έπειτα, το παιδί θα πρέπει να πατήσει έναν από τους 4 διαθέσιμους διακόπτες αυτού του πληκτρολογίου που έχουμε κατασκευάσει. Οι διακόπτες αυτοί είναι συνδεδεμένοι με το Arduino. Ο ένας από αυτούς είναι η σωστή επιλογή για το παιδί ενώ οι άλλοι 3 είναι λάθος επιλογές. Όταν το παιδί θα πατήσει ένα διακόπτη, τότε θα σταλεί ένα σήμα στο Arduino με το οποίο είναι ενωμένοι και οι 4 διακόπτες και θα ενεργοποιηθεί το ίδιο κύκλωμα παραγωγής φωνής της προηγούμενης αποστολής. Ομοίως με την προηγούμενη αποστολή, αν πατηθεί ο σωστός διακόπτης θα έχουμε παραγωγή της λέξης "Yes" ενώ αν πατηθεί ένας από τους 3 λάθος διακόπτες θα παραχθεί η λέξη "No". Μετέπειτα θα ακολουθήσει η αναγνώριση αυτής της φωνής από το Nao για να διαπιστώσει αν είναι επιτυχής ή όχι η αποστολή οπότε θα μεταβούμε ανάλογα στον αντίστοιχο κόμβο.

5.2 Υλοποίηση Αποστολών

5.2.1 Κύκλωμα παραγωγής φωνής

Καταρχάς, προτού δούμε εκτενώς την υλοποίηση των δυο αυτών αποστολών, θα ασχοληθούμε με τη δημιουργία του κυκλώματος παραγωγής φωνής, το οποίο θα χρησιμοποιηθεί και στις δυο αποστολές και θα είναι στην ουσία η διασύνδεση της πλατφόρμας Arduino με το Nao, το οποίο στην ουσία θα ανιχνεύει αν η αποστολή έχει επιτευχθεί με επιτυχία ή όχι.

Για πολλά χρόνια, η σύνθεση ομιλίας από τους υπολογιστές, ήταν μια πρόκληση για τους μηχανικούς. Το πρόβλημα όμως της μικρής υπολογιστικής ισχύος, απέτρεπε την παραγωγή φωνής μέσω του λογισμικού. Επομένως, τα παλαιότερα χρόνια, για τη σύνθεση φωνής μέσω ενός υπολογιστή, χρησιμοποιείτο ένα ολοκληρωμένο κύκλωμα. Με την πάροδο όμως του χρόνου και τη ραγδαία ανάπτυξη της υπολογιστικής ισχύος, το ολοκληρωμένο αυτό αντικαταστάθηκε από την παραγωγή φωνής μέσω του λογισμικού, επομένως η παραγωγή αυτού του chip σταμάτησε. Αυτό, όμως, δημιούργησε πρόβλημα στους προγραμματιστές του Arduino, αφού χρειάζονταν κάτι πέραν του λειτουργικού για να παράξουν φωνή. Προς τη λύση του προβλήματος αυτού, ήρθε η Sparkfun η οποία και κατασκεύασε το SpeakJet chip ¹. Το ολοκληρωμένο αυτό κύκλωμα, είναι ένα εντελώς αυτόνομο chip παραγωγής φωνής και σύνθετου ήχου. Χρησιμοποιεί ένα μαθηματικό ηχητικό αλγόριθμο για τον έλεγχο ενός εσωτερικού συνθέτη φωνής 5 καναλιών για την παραγωγή φωνής και σύνθετων ήχων. Το SpeakJet, πέραν των δυνατοτήτων που έχει για ρύθμιση των pitch, rate, bend και volume parameters, δεν παράγει ξεχωριστά γράμματα του αλφαβήτου της αγγλικής διαλέκτου, αλλά παράγει 72 αλλόφωνα. Τα αλλόφωνα είναι ένας συνδυασμός πολλαπλών δυνατών ήχων ομιλίας που χρησιμοποιούνται για την προφορά ενός φωνήματος ². Επομένως, για να παράξουμε μια λέξη, πρέπει να συνδυάσουμε αυτά τα αλλόφωνα, τα οποία περιέχονται στο data sheet του SpeakJet ³.

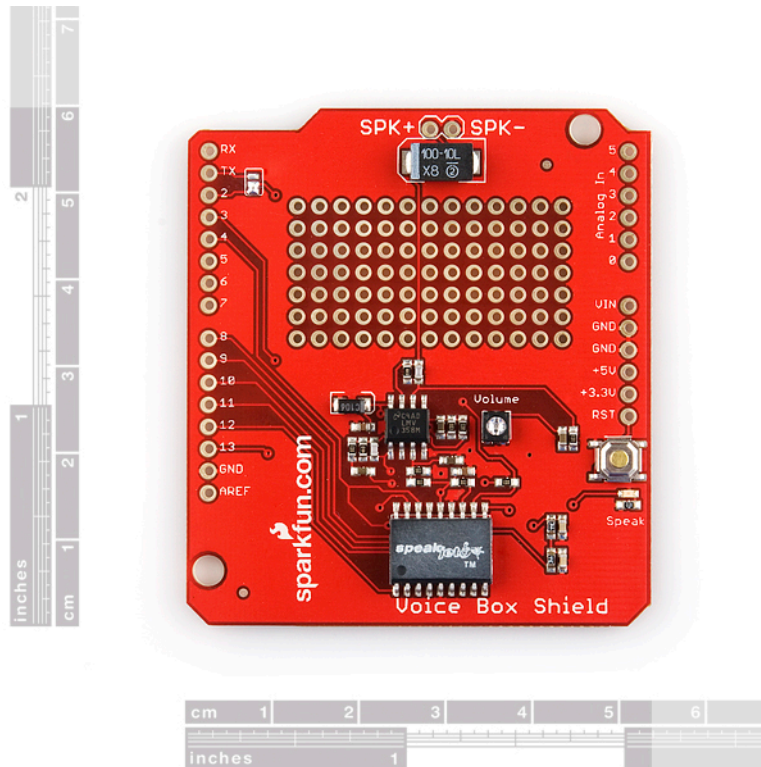
Εμείς, θα χρησιμοποιήσουμε αυτό το ολοκληρωμένο κύκλωμα για την παραγωγή των λέξεων που χρειαζόμαστε. Συγκεκριμένα, αντί να δημιουργήσουμε μόνοι μας ένα κύκλωμα παραγωγής φωνής το οποίο θα περιλαμβάνει το ολοκληρωμένο αυτό κύκλωμα μαζί με ένα κύκλωμα ενισχυτή φωνής, θα χρησιμοποιήσουμε μια έτοιμη πλακέτα της εταιρίας Sparkfun, το Voice Box Shield ⁴. Το Voice Box Shield περιλαμβάνει το Speakjet συνδεδεμένο κατάλληλα, ένα ενισχυτή ήχου και ένα βαθυπερατό φίλτρο 2 πόλων. Μια εικόνα του Voice Box Shield και το σχηματικό διάγραμμά του δίνονται παρακάτω:

¹<http://www.sparkfun.com/products/9578>

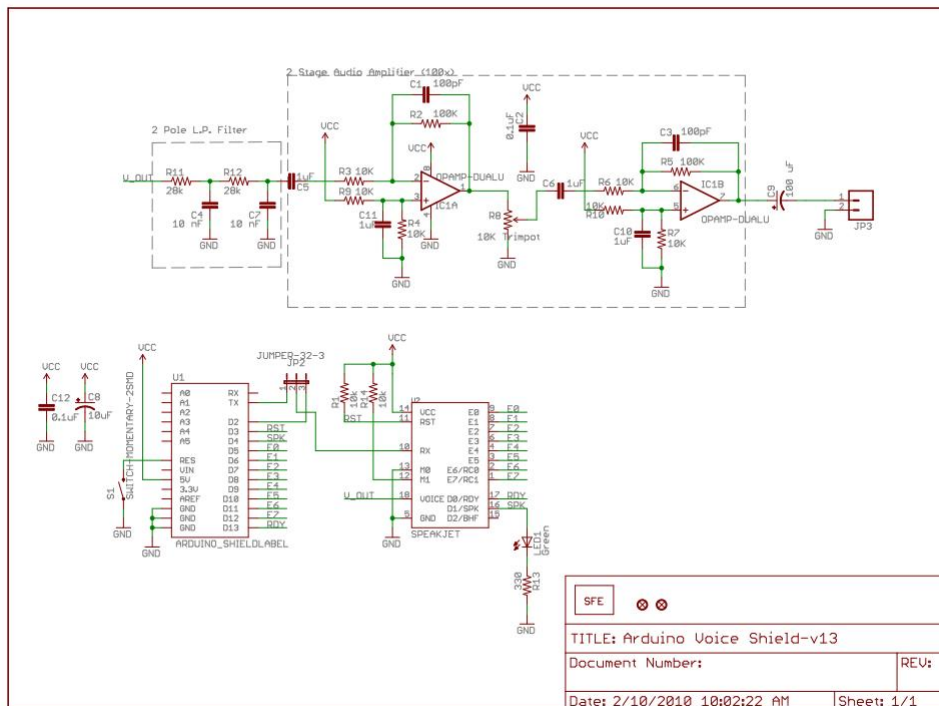
²<http://en.wikipedia.org/wiki/Allophone>

³<http://www.sparkfun.com/datasheets/Components/General/speakjet-usermanual.pdf>

⁴<http://www.sparkfun.com/products/9799>



Σχήμα 5.1: Το Voice Box Shield της Sparkfun



Σχήμα 5.2: Το σχηματικό διάγραμμα του Voice Box

Επομένως, όπως βλέπουμε και από το σχηματικό διάγραμμα, για να λειτουργήσει το Voice Box, το μόνο που χρειάζεται είναι να το τροφοδοτήσουμε με ρεύμα 5V και με γείωση μέσω του Arduino, να ενώσουμε το pin 2 του Voice Box με ένα pin του Arduino και να συνδέσουμε ένα κλασικό μεγάφωνο αντίστασης 8 Ω (Ohm) στα pins SPK+ και SPK- του Voice Box. Ένα απλό παράδειγμα

παραγωγής της λέξης "Arduino" δίνεται στη συνέχεια. Αξίζει να σημειώσουμε, ότι για να παράξουμε την επιθυμητή μας λέξη, όπως αναφέραμε και προηγουμένως, θα ανατρέξουμε στο data sheet του SpeakJet και θα βρούμε τα κατάλληλα φωνήματα που αντιστοιχούν στη λέξη που θέλουμε. Στη συγκεκριμένη περίπτωση, η λέξη "arduino" αποτελείται από τα φωνήματα AWRR, RR, DO, WW, IY, NE, OWWW τα οποία έχουν αντίστοιχες τιμές bytes 152, 148, 175, 147, 128, 141, 164. Επομένως, έχουμε τον ακόλουθο κώδικα:

Παράδειγμα χρήσης Voice Box με τη λέξη "arduino"

```

1 #include <SoftwareSerial.h>
2
3 #define rxPin 2
4 #define txPin 3 /*Connect voice box pin 2 with*/
5                 /*arduino pin 3*/
6
7 SoftwareSerial speakJet = SoftwareSerial(rxPin, txPin);
8
9 /*Define Word Pause length*/
10 #define WP 6
11
12 byte message [] = {
13     /* arduino */ 152, 148, 175, 147, 128, 141, 164, WP
14 };
15
16 int messageSize = sizeof(message);
17
18 void setup ()
19 {
20     pinMode(rxPin, INPUT);
21     pinMode(txPin, OUTPUT);
22     speakJet.begin(9600);
23     speakJet.print(20, BYTE);
24     speakJet.print(96, BYTE);
25     delay(1000);
26 }
27
28 void loop ()
29 {
30     int i;
31     for (i=0; i<messageSize; i++)
32     {
33         speakJet.print(message1[i], BYTE);
34     }
35     delay(5000);
36 }

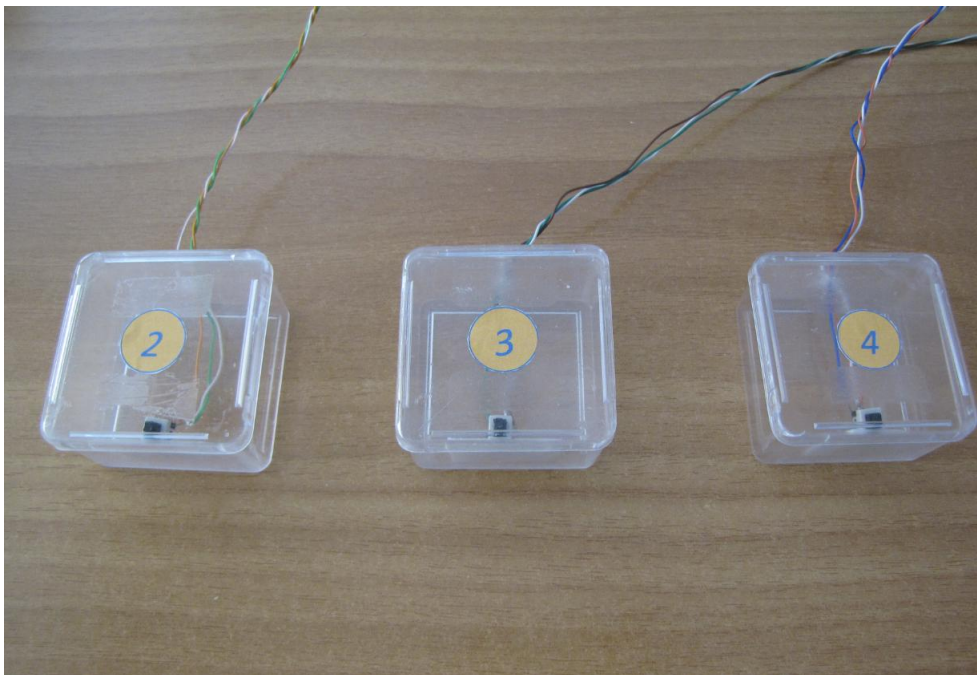
```

Άρα αυτό θα είναι το κύκλωμα και ο κώδικας που θα χρησιμοποιήσουμε για να παράξουμε τις λέξεις που θέλουμε. Όπως αναφέραμε πιο πάνω, εμείς θέλουμε με επιτυχή ολοκλήρωση να παράγεται η λέξη "yes" και με αποτυχία της αποστολής να παράγεται η λέξη "no". Οι λέξεις αυτές, βάσει του data sheet του SpeakJet, αποτελούνται από τα ακόλουθα αλλόφωνα και αντιστοιχούν στα ακόλουθα bytes:

- "Yes": IYEH, SE, SE και σε bytes: 158, 187, 187
- "No": NO, OWWW και σε bytes: 142, 164

5.2.2 3η Αποστολή

Στην αποστολή αυτή, όπως έχουμε αναφέρει πιο πάνω, το παιδί πρέπει να διαλέξει μεταξύ τριών κουτιών, να ανοίξει αυτό στο οποίο αναγράφεται η σωστή τιμή των χρημάτων που χρειάζεται ακόμη για να αποκτήσει το αντικείμενο που χρειάζεται, δηλαδή το κουτί με τον αριθμό 4 επάνω. Επομένως, για την αποστολή αυτή, χρειάστηκε να κατασκευάσουμε 3 κουτιά εφοδιασμένα με διακόπτες πίεσης ούτως ώστε να ξέρουμε πότε ανοίγει το καπάκι ενός κουτιού και άρα ποιο κουτί επιλέγει κάθε φορά το παιδί. Οι διακόπτες αυτοί, είναι μεταγωγικοί διακόπτες δυο καταστάσεων οι οποίοι όταν πατηθούν και μετά αφεθούν επιστρέφουν στην αρχική τους κατάσταση. Κάθε διακόπτης διαθέτει τρία ποδαράκια, ένα για την τάση τροφοδοσίας, ένα για τη γείωση και ένα που θα μας δείχνει αν ο διακόπτης είναι πατημένος ή όχι. Στη μια κατάσταση του διακόπτη, ενώνεται η τάση τροφοδοσίας με το μεσαίο ποδαράκι οπότε έχουμε ένδειξη ύπαρξης τάσης, ενώ στην άλλη κατάσταση του διακόπτη, ενώνεται η γείωση με το μεσαίο ποδαράκι και έχουμε ένδειξη μηδενικής τάσης. Για να σιγουρέψουμε μάλιστα τη διάκριση του αν η τάση που αντιλαμβάνεται το Arduino είναι HIGH ή LOW, παρεμβάλουμε μεταξύ της τροφοδοσίας και του διακόπτη ένα μια αντίσταση 10 kΩ, γνωστή και ως pull up resistor. Η ύπαρξη αυτής της αντίστασης, λόγω της μεγάλης της τιμής, στην ουσία μηδενίζει την τάση από τυχόν απώλειες σε περίπτωση που έχουμε LOW κατάσταση, με αποτέλεσμα να μην υπάρχει πρόβλημα στην ανίχνευση των δυο καταστάσεων από το Arduino. Έτσι, ενώσαμε τους τρεις διακόπτες σε κάθε κουτί έτσι ώστε όταν είναι κλειστά τα κουτιά, οι διακόπτες να είναι πατημένοι και να έχουμε τη ανίχνευση της μιας κατάστασης από το Arduino. Όταν το παιδί ανοίξει ένα κουτί, τότε ο διακόπτης επανέρχεται στην αρχική του κατάσταση με αποτέλεσμα να ανιχνεύσουμε την αλλαγή κατάστασης και ανάλογα από το αν είναι το σωστό κουτί που ανοίχθηκε ή όχι, να ενεργήσουμε ανάλογα.



Σχήμα 5.3: Τα κουτιά της 3ης αποστολής

Ο κώδικας που χρησιμοποιήσαμε, στην ουσία ανιχνεύει την κατάσταση που βρίσκεται κάθε ένας από τους διακόπτες. Αν ο διακόπτης του σωστού κουτιού μεταβεί από HIGH σε LOW, δηλαδή το

παιδί έχει ανοίξει αυτό το κουτί, τότε θα ενεργοποιείται το σύστημα παραγωγής φωνής και θα λείπει τη λέξη "Yes" δείχνοντας στο παιδί ότι έχει ανοίξει το σωστό κουτί. Αν ένας από τους διακόπτες των άλλων δυο κουτιών μεταβούν από HIGH σε LOW, τότε και πάλι θα ενεργοποιηθεί το σύστημα παραγωγής φωνής αλλά αυτή τη φορά θα παραχθεί η λέξη "No" που θα δείχνει στο παιδί ότι έχει ανοίξει λάθος κουτί. Αυτές τις λέξεις θα τις ανιχνεύει μετέπειτα το Nao και ανάλογα θα ενεργεί.

Κώδικας 3ης αποστολής

```

1  /* Mission 3 */
2
3  #include <SoftwareSerial.h>
4
5  #define rxPin 2
6  #define txPin 3
7
8  SoftwareSerial speakJet = SoftwareSerial(rxPin, txPin);
9
10
11 byte message1[] = {
12     158, 187, 187
13 }; // Yes
14
15 byte message2[] = {
16     142, 164
17 }; // No
18
19 int messageSize1 = sizeof(message1);
20 int messageSize2 = sizeof(message2);
21
22 int inputPin1 = 4; /* pushbutton 1-2 euros */
23 int inputPin2 = 5; /* pushbutton 2-3 euros */
24 int inputPin3 = 6; /* pushbutton 3-4 euros */
25
26 void setup() {
27     pinMode(inputPin1, INPUT); /* declare pushbutton as input */
28     pinMode(inputPin2, INPUT); /* declare pushbutton as input */
29     pinMode(inputPin3, INPUT); /* declare pushbutton as input */
30     pinMode(rxPin, INPUT);
31     pinMode(txPin, OUTPUT);
32     speakJet.begin(9600);
33     speakJet.print(20, BYTE);
34     speakJet.print(96, BYTE);
35     delay(1000);
36 }
37 void loop(){
38     if (digitalRead(inputPin1)==HIGH && digitalRead(inputPin2)==HIGH
39 && digitalRead(inputPin3)==LOW){
40         int i;
41         for (i=0; i<messageSize1; i++)
42             {
43                 speakJet.print(message1[i], BYTE);
44             }

```



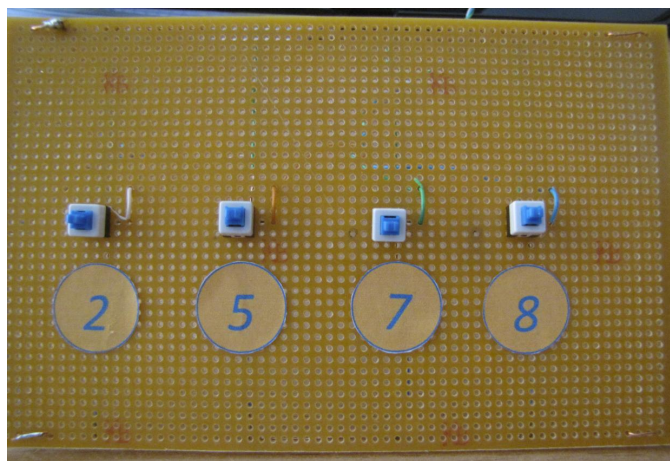
```

45   delay(2000);
46 }else if (digitalRead(inputPin1)==HIGH && digitalRead(inputPin2)==LOW
47 && digitalRead(inputPin3)==HIGH){
48   int i;
49   for (i=0; i<messageSize2; i++)
50   {
51     speakJet.print(message2[i], BYTE);
52   }
53   delay(2000);
54 }else if (digitalRead(inputPin1)==LOW && digitalRead(inputPin2)==HIGH
55 && digitalRead(inputPin3)==HIGH){
56   int i;
57   for (i=0; i<messageSize2; i++)
58   {
59     speakJet.print(message2[i], BYTE);
60   }
61   delay(2000);
62 }
63 }

```

5.2.3 4η Αποστολή

Στην αποστολή αυτή, το παιδί πρέπει να πατήσει από ένα πληκτρολόγιο που κατασκευάσαμε εμείς, το κατάλληλο κουμπί το οποίο αντιστοιχεί στην τιμή των χρημάτων που χρειάζεται το παιδί για να αποκτήσει το αντικείμενο που θέλει, δηλαδή το κουμπί που θα έχει την ένδειξη των 5 ευρώ επάνω του. Οπότε, για την αποστολή αυτή, κατασκευάσαμε ένα δικό μας πληκτρολόγιο το οποίο αποτελείται από 4 κουμπιά τα οποία υλοποιήθηκαν με τη βοήθεια μεταγωγικών διακοπών δυο καταστάσεων. Η διαφορά των διακοπών αυτής της αποστολής από τους διακόπτες που χρησιμοποιήσαμε στην προηγούμενη αποστολή, είναι ότι αυτοί οι διακόπτες, όταν πατηθούν, παραμένουν σε εκείνη την κατάσταση έως ότου πατηθούν ξανά. Η διαδικασία σύνδεσης των διακοπών αυτών είναι η ίδια με της προηγούμενης άσκησης ενώ και πάλι χρησιμοποιήσαμε αντίσταση 10 kΩ ως pull up resistor μεταξύ της τροφοδοσίας και του διακόπτη. Όλα αυτά υλοποιήθηκαν σε μια πλακέτα, ούτως ώστε να δώσουμε μια πιο ρεαλιστική αντίληψη ενός πληκτρολογίου όπου το παιδί καλείται να πατήσει τα ανάλογα κουμπιά.



Σχήμα 5.4: Το πληκτρολόγιο της 4ης αποστολής

Το παιδί, πρέπει να πατήσει ένα από τα 4 κουμπιά. Αν πατηθεί το λάθος κουμπί τότε θα σταλεί σήμα HIGH στο Arduino και θα ενεργοποιηθεί το σύστημα παραγωγής φωνής παράγοντας τη λέξη "No". Αυτό θα ισχύσει αν το παιδί πατήσει κάποιο από τα 3 λάθος κουμπιά. Αν πατηθεί το σωστό κουμπί τότε θα παραχθεί η λέξη "Yes" που θα αναγνωρίσει το Nao ως επιτυχή ολοκλήρωση της αποστολής. Όταν πατήσει ένα κουμπί το παιδί, αν είναι λάθος το κουμπί που πάτησε και χρειάζεται να πατήσει ξανά το σωστό, τότε πρέπει να πατήσει πρώτα αυτό που είχε πατήσει στην αρχή για να επανέλθει στην αρχική του κατάσταση και μετά να πατήσει το σωστό κουμπί. Επομένως, ο κώδικας που θα χρησιμοποιήσουμε, περιλαμβάνει ελέγχους και των τεσσάρων κουμπιών, αφού για να ενεργοποιηθεί το σύστημα παραγωγής φωνής, πρέπει να έχει πατηθεί μόνο ένα από τα 4 κουμπιά και άρα τα υπόλοιπα θα βρίσκονται στην αρχική τους κατάσταση. Έτσι, ο κώδικας που χρησιμοποιήσαμε για την υλοποίηση της αποστολής αυτής στο Arduino, είναι ο ακόλουθος:

Κώδικας 4ης αποστολής

```

1  /* Mission 4 */
2
3  #include <SoftwareSerial.h>
4
5  #define rxPin 2
6  #define txPin 3
7
8  SoftwareSerial speakJet = SoftwareSerial(rxPin, txPin);
9
10
11 byte message1 [] = {
12     158, 187, 187
13 }; /* Yes */
14
15 byte message2 [] = {
16     142, 164
17 }; /* No */
18
19 int messageSize1 = sizeof(message1);
20 int messageSize2 = sizeof(message2);
21
22 int inputPin1 = 7; /* pushbutton 1-2 euros */
23 int inputPin2 = 8; /* pushbutton 2-5 euros */
24 int inputPin3 = 9; /* pushbutton 3-7 euros */
25 int inputPin4 = 10; /* pushbutton 4-8 euros */
26
27
28 void setup() {
29     pinMode(inputPin1, INPUT); /* declare pushbutton as input */
30     pinMode(inputPin2, INPUT); /* declare pushbutton as input */
31     pinMode(inputPin3, INPUT); /* declare pushbutton as input */
32     pinMode(inputPin4, INPUT); /* declare pushbutton as input */
33     pinMode(rxPin, INPUT);
34     pinMode(txPin, OUTPUT);
35     speakJet.begin(9600);
36     speakJet.print(20, BYTE);
37     speakJet.print(96, BYTE);
38     delay(1000);

```

```

39 }
40 void loop(){
41   if (digitalRead(inputPin1)==LOW && digitalRead(inputPin2)==HIGH
42   && digitalRead(inputPin3)==LOW && digitalRead(inputPin4)==LOW){
43     int i;
44     for (i=0; i<messageSize1; i++)
45     {
46       speakJet.print(message1[i], BYTE);
47     }
48     delay(2000);
49   } else if (digitalRead(inputPin1)==HIGH && digitalRead(inputPin2)==LOW
50   && digitalRead(inputPin3)==LOW && digitalRead(inputPin4)==LOW){
51     int i;
52     for (i=0; i<messageSize2; i++)
53     {
54       speakJet.print(message2[i], BYTE);
55     }
56     delay(2000);
57   } else if (digitalRead(inputPin1)==LOW && digitalRead(inputPin2)==LOW
58   && digitalRead(inputPin3)==HIGH && digitalRead(inputPin4)==LOW){
59     int i;
60     for (i=0; i<messageSize2; i++)
61     {
62       speakJet.print(message2[i], BYTE);
63     }
64     delay(2000);
65   } else if (digitalRead(inputPin1)==LOW && digitalRead(inputPin2)==LOW
66   && digitalRead(inputPin3)==LOW && digitalRead(inputPin4)==HIGH){
67     int i;
68     for (i=0; i<messageSize2; i++)
69     {
70       speakJet.print(message2[i], BYTE);
71     }
72     delay(2000);
73   }
74 }

```

Κεφάλαιο 6

Υλοποίηση του σεναρίου στη ρομποτική πλατφόρμα Nao

6.1 Προσομοίωση στο Choregraphe

Προτού περάσουμε στη μεταφορά του σεναρίου μας στη ρομποτική πλατφόρμα Nao, χρησιμοποιήσαμε ένα περιβάλλον προσομοίωσης για να ελέγξουμε την ορθότητα του σεναρίου. Το περιβάλλον προσομοίωσης που χρησιμοποιήσαμε, είναι το Choregraphe¹ της Aldebaran, της γαλλικής εταιρείας, δηλαδή, που κατασκευάζει το ίδιο το Nao. Το Choregraphe, είναι ένα περιβάλλον προσομοίωσης το οποίο μπορούμε να ενώσουμε είτε με ένα τοπικό NaoQi server είτε με το ίδιο το Nao και να δούμε στην πράξη μέσω ενός τριδιάστατου μοντέλου που προσφέρεται από το πρόγραμμα, κάποιες λειτουργίες του Nao. Για να δουλέψει το πρόγραμμα, χρειάζεται να έχουμε εγκαταστήσει ένα τοπικό NaoQi server, ο οποίος διατίθεται από τη σελίδα των χρηστών της Aldebaran². Επομένως, αν θέλουμε να προσομοιώσουμε κάποιες λειτουργίες του Nao χωρίς να έχουμε ενωθεί με το ίδιο το ρομπότ, χρειάζεται να ξεκινήσουμε τον τοπικό NaoQi server και έπειτα από το Choregraphe να εκτελέσουμε τις εντολές που θέλουμε.

Το Choregraphe και το NaoQi Framework, υποστηρίζουν και τα δυο τον κώδικα URBI και συνεπώς αυτά που γράψαμε στο πρόγραμμα Gostai Studio. Γενικά, υπάρχουν κάποια προβλήματα συμβατότητας με τις διάφορες εκδόσεις του NaoQi Framework όσον αφορά τη συμβατότητα με το URBI. Εμείς στην αρχή, δοκιμάσαμε να χρησιμοποιήσουμε το NaoQi v.1.10.44 που είναι και η πιο πρόσφατα διαθέσιμη έκδοση, αλλά υπήρξε ένα πρόβλημα όταν προσπαθήσαμε να ξεκινήσουμε το σχετικό module που αφορά το URBI. Επομένως, αναγκαστήκαμε να καταφύγουμε σε παλαιότερη έκδοση, αυτήν του NaoQi v.1.6.13 το οποίο είναι πλήρως συμβατό με το URBI και συνεπώς και το Gostai. Επομένως, ακολουθήσαμε τα ακόλουθα βήματα:

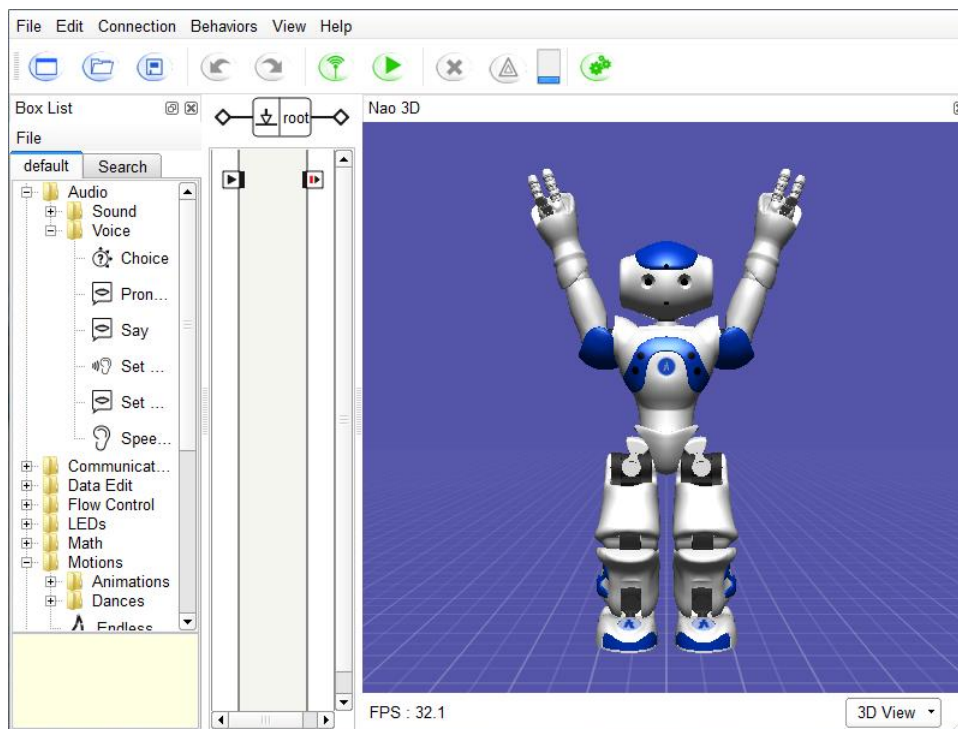
1. Εγκαταστήσαμε το NaoQi Framework v.1.6.13 όπως και το Choregraphe v.1.10.44 (το γεγονός ότι η έκδοση του Choregraphe που χρησιμοποιήσαμε ήταν πολύ νεότερη από αυτήν του NaoQi Framework δεν μας δημιούργησε κάποιο πρόβλημα αλλά κάλλιστα μπορούσαμε να χρησιμοποιήσουμε και μια πιο παλιά έκδοση του Choregraphe για να υπάρχει απόλυτη συμβατότητα των δυο).
2. Βρήκαμε τα αρχεία autoload.ini και των δυο προγραμμάτων (για το μεν NaoQi βρίσκεται στο φάκελο /preferences ενώ για το Choregraphe στο /Aldebaran/Choregraphe 1.10.44/preferences) και κάναμε και στα δυο uncommment τη γραμμή που περιείχε το "urbistarter". Έτσι στην ουσία, κάθε φορά που θα ξεκινούμε τα δυο προγράμματα θα ξεκινά μαζί τους και ο server του URBI.

¹<http://www.aldebaran-robotics.com/en/programmable>

²<http://users.aldebaran-robotics.com/>

3. Ξεκινήσαμε πρώτα το NaoQi Framework δίνοντας την εντολή `naoqi -p 5400`, όπου το "-p 5400" λέει στο NaoQi να ξεκινήσει στο port 5400 που είναι το port που "ακούει" το Gostai Console.
4. Στη συνέχεια, ανοίξαμε το Gostai Studio και ενωθήκαμε στη διεύθυνση "127.0.0.1:5400".
5. Τέλος, ανοίξαμε το Choregraphe και ενωθήκαμε στο local Nao που δημιουργήθηκε στο port 5400.

Έτσι, μπορούσαμε να τρέξουμε οποιεσδήποτε εντολές επιθυμούσαμε από το Gostai Studio και να προσομοιώσουμε τη λειτουργία τους στο Choregraphe. Το Choregraphe γενικά, μπορεί να προσομοιώσει όλες τις λειτουργίες ενός ρομπότ, είτε αυτές περιλαμβάνουν κίνηση είτε αλληλεπίδραση με τους αισθητήρες του ρομπότ. Η αλληλεπίδραση όμως του ρομπότ με τους αισθητήρες του, αντιμετωπίζει κάποια προβλήματα στο Choregraphe. Επομένως, στην ουσία, με την προσομοίωση αυτή ελέγξαμε όλες τις μηχανικές κινήσεις που θα χρειαστεί να κάνει το ρομπότ για το σενάριό μας. Την αλληλεπίδραση με τους αισθητήρες (παραγωγή ήχου, ανίχνευση ήχου) θα την εξετάσουμε στην πράξη με το ίδιο το ρομπότ.



Σχήμα 6.1: Παράδειγμα χρήσης του Choregraphe

6.2 Διασύνδεση με το Nao

Όλες οι ρομποτικές πλατφόρμες Nao είναι εφοδιασμένες με ένα δικό τους ενσωματωμένο NaoQi Framework. Αυτό το NaoQi που περιέχει το Nao, έχει τη δυνατότητα να συνδεθεί με URBI οπότε στην ουσία να λάβει το ρομπότ εντολές από προγράμματα όπως το Gostai Studio που χρησιμοποιούμε εμείς. Για να ενεργοποιήσουμε, όμως, αυτή τη δυνατότητα στο Nao, χρειάζεται να ακολουθήσουμε τα ακόλουθα βήματα:

1. Πρέπει καταρχάς να συνδεθούμε από τον υπολογιστή μας με το Nao μέσω της χρήσης Secure Shell (ssh) χρησιμοποιώντας την ακόλουθη εντολή: `ssh nao@myNao`
Έτσι, έχουμε πρόσβαση στα αρχεία του Nao.

2. Βρίσκουμε το αρχείο ‘/opt/naoqi/preferences/autoload.ini’, προσθέτουμε μια γραμμή που περιέχει την εντολή *urbistarter* και αποθηκεύουμε το αρχείο.
3. Επανεκκινούμε το Nao.
4. Πλέον είμαστε έτοιμοι να συνδεθούμε από το Gostai Studio στη διεύθυνση "147.102.11.197:54000" στην οποία "ακούει" το Nao (147.102.11.197 είναι η διεύθυνση IP του Nao, η οποία είναι δυναμική και αλλάζει κάθε φορά οπότε για να το μάθουμε πατάμε το κουμπί στο σώμα του Nao και θα το ακούσουμε από τα ηχεία του).
5. Πατώντας το κουμπί "Play" στο Gostai Studio, θα σταλθεί ο κώδικάς μας σε μορφή Urbiscript (.u) και θα εκτελεστεί από το Nao.

Δυστυχώς, όμως, το Nao του Εργαστηρίου μας το οποίο και θα χρησιμοποιούσαμε στην υλοποίηση της διπλωματικής αυτής εργασίας, δεν ήταν διαθέσιμο από την έναρξη αυτής της διπλωματικής εργασίας, λόγω εργασιών αναβάθμισης και συντήρησης που έπρεπε να γίνουν στο εργαστήριο της Aldebaran στη Γαλλία. Το ρομπότ μας είχε επιστραφεί λίγες μέρες πριν την παρουσίαση αυτής της διπλωματικής εργασίας. Επομένως, δεν είχαμε τη χρονική ευχέρεια να δοκιμάσουμε σε πλήρη έκταση όλη τη λειτουργία του σεναρίου, παρά μόνο τα κομμάτια αποσπασματικά. Τα αποτελέσματα αυτών των δοκιμών θα αναφερθούν παρακάτω.

6.3 Ειδικά Νομίσματα

6.3.1 Κατασκευή Ειδικών Νομισμάτων

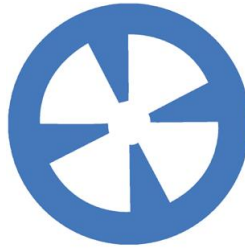
Όπως έχουμε δει στα προηγούμενα κεφάλαια, θα χρησιμοποιήσουμε κάποια ειδικά "νομίσματα" ως τα νομίσματα τα οποία θα δίνει το παιδί στο ρομπότ για να γίνει η οπτική τους αναγνώριση. Τα ειδικά αυτά "νομίσματα" είναι προϊόν της ίδιας της εταιρίας κατασκευής του Nao, της γαλλικής εταιρίας Aldebaran, και ονομάζονται Naomarks. Υπάρχουν 10 διαφορετικά Naomarks και βρίσκονται όλα στο References CD του Nao. Κάθε ένα από αυτά τα νομίσματα, έχει διαφορετικό σχήμα και ένας ξεχωριστός αριθμός αντιστοιχεί σε αυτά τα νομίσματα ούτως ώστε το κάθε νόμισμα με την οπτική αναγνώριση να μετατρέπεται στην ουσία σε αυτόν τον αριθμό και να μπορούμε έτσι να ξεχωρίσουμε τα νομίσματα μεταξύ τους. Εμείς, από τα 10 διαθέσιμα νομίσματα, χρησιμοποιήσαμε τα Naomark 64, 108, 85 και 119 και τα αντιστοιχήσαμε με νομίσματα αξίας 1, 2, 4 και 5 ευρώ αντίστοιχα. Τα 4 αυτά Naomarks, είναι τα ακόλουθα:



Σχήμα 6.2: Το Naomark 64 που αντιστοιχεί στο 1 ευρώ



Σχήμα 6.3: Το Naomark 108 που αντιστοιχεί στα 2 ευρώ

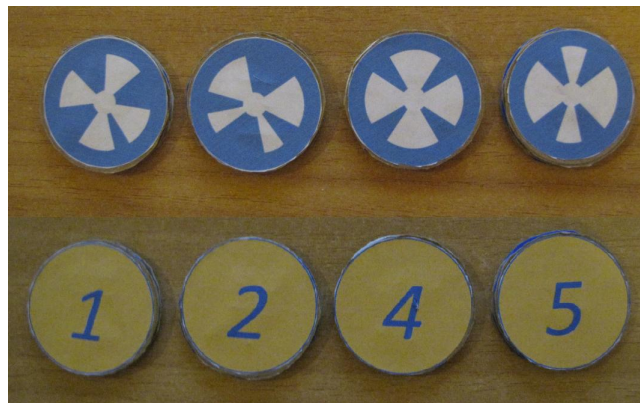


Σχήμα 6.4: Το Naomark 85 που αντιστοιχεί στα 4 ευρώ



Σχήμα 6.5: Το Naomark 119 που αντιστοιχεί στα 5 ευρώ

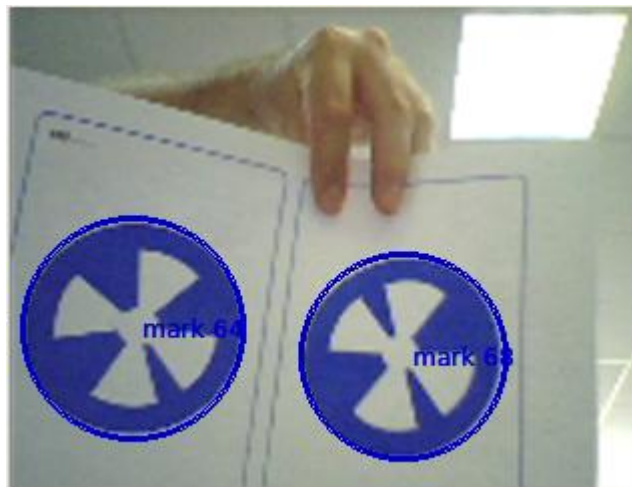
Επομένως, κατασκευάσαμε 4 νομίσματα και από τη μια μεριά κολλήσαμε ένα αυτοκόλλητο με το σήμα του κάθε Naomark και από την άλλη κολλήσαμε τον αριθμό στον οποίο αντιστοιχεί το κάθε Naomark για να μπορεί το παιδί να ξέρει την αξία των νομισμάτων που κρατάει. Αυτά τα νομίσματα είναι τα νομίσματα που θα έχει στη διάθεση το παιδί του και θα δίνει στο Ναο για οπτική αναγνώριση (στην ουσία εμάς μας ενδιαφέρει η αναγνώριση αυτού του μοναδικού αριθμού που αντιστοιχεί σε κάθε νόμισμα) αλλά και τα νομίσματα που θα παίρνει το παιδί αν έχει ολοκληρώσει με επιτυχία μια αποστολή.



Σχήμα 6.6: Τα νομίσματα που κατασκευάσαμε εμείς

6.3.2 Προσομοίωση οπτικής αναγνώρισης στο Telepathe

Για να δοκιμάσουμε τα νομίσματα που κατασκευάσαμε, χρησιμοποιήσαμε στην αρχή ένα ειδικό πρόγραμμα που παρέχει η Aldebaran, το Telepathe, το οποίο αφορά αποκλειστικά την εισαγωγή και επεξεργασία εικόνων και βίνετο από τις δυο κάμερες του Nao. Επιλέγοντας από το πρόγραμμα την επιλογή "mark detection" δοκιμάσαμε την οπτική αναγνώριση των ειδικών νομισμάτων που κατασκευάσαμε. Στην αρχή, δοκιμάσαμε μικρά σχετικά νομίσματα (περίπου 4 εκατοστά διάμετρο το κάθε ένα) και συνειδητοποιήσαμε ότι το Nao δεν μπορεί να εφαρμόσει οπτική αναγνώριση σε τόσο μικρά αντικείμενα. Επομένως, χρησιμοποιήσαμε πιο μεγάλα νομίσματα, περίπου 9 εκατοστά διάμετρο το κάθε ένα. Και όπως δείχνουν οι φωτογραφίες πιο κάτω, το πρόγραμμα αναγνώρισε με επιτυχία το νόμισμα στην εικόνα ενώ μάλιστα μας έδινε και το ID του που είναι αυτό που θα χρησιμοποιήσουμε για να ξεχωρίσουμε τα νομίσματα το ένα από το άλλο. Επομένως, αφού διαπιστώσαμε ότι η οπτική αναγνώριση των νομισμάτων είναι εφικτή στο Nao, προχωρήσαμε στην πρακτική εφαρμογή της.



Σχήμα 6.7: Οπτική αναγνώριση νομισμάτων από το Telepathe

6.4 Πειράματα

Όπως αναφέραμε στην αρχή της διπλωματικής αυτής εργασίας, το σενάριο εκμάθησης βασικών αριθμητικών εννοιών που δημιουργήσαμε απευθύνεται σε παιδιά προσχολικής και πρώτης σχολικής ηλικίας. Έτσι, το συγκεκριμένο σενάριο μπορεί να εφαρμοστεί μέσα από μια σειρά πειραμάτων σε παιδιά τέτοιας ηλικίας, για να διαπιστώσουμε τη σωστή του λειτουργία αλλά και την αποτελεσματικότητά του. Τα πειράματα που μπορούν να γίνουν με τη βοήθεια μιας ομάδας παιδιών για να εξακριβώσουμε την εύρυθμη λειτουργία και αποτελεσματικότητα του σεναρίου, είναι τα ακόλουθα:

1. Καταρχάς θα χρησιμοποιήσουμε παιδιά και των δυο φύλων αλλά ηλικιών από 4 μέχρι 6 χρονών με στόχο να διαπιστώσουμε αν η μέθοδος αυτή εκπαιδευτικής μάθησης με τη χρήση ρομποτικών συστημάτων, είναι περισσότερο αποδεκτή σε παιδιά μικρότερης προσχολικής ηλικίας ή σε παιδιά πρώτης σχολικής ηλικίας τα οποία έχουν ήδη μια εμπειρία από τη μάθηση στο Δημοτικό Σχολείο.
2. Θα εξετάσουμε τα ποσοστά επιτυχίας των παιδιών σε κάθε μια από τις πιθανές αποστολές που σχηματίσαμε. Έτσι, θα διαπιστώσουμε αν υπάρχει κάποια αποστολή η οποία είναι δύσκολη για τα παιδιά και δεν μπορούν να την κατανοήσουν οπότε χρειάζεται να την αντικαταστήσουμε με μια άλλη ενώ παράλληλα μπορεί να υπάρχουν και αποστολές που ήταν πολύ

εύκολες για τα παιδιά οπότε στην ουσία δεν προσέφεραν ουσιαστική εκπαιδευτική μάθηση στα παιδιά.

3. Αξίζει να πειραματιστούμε με δυο κατηγορίες παιδιών, αυτά που έχουν επίγνωση του τι εστί ρομπότ ίσως μέσα από κινούμενα σχέδια, από τηλεοπτικές εκπομπές και άλλες συναφείς πηγές πληροφορίας, και σε παιδιά που δεν έχουν λάβει στη ζωή τους τη γνώση για το τι εστί ρομπότ, οπότε αντικρίζουν στην ουσία πρώτη φορά ένα ρομποτικό σύστημα.

6.5 Αποτελέσματα

Θα αναφέρουμε τώρα, τα αποτελέσματα που βγάλαμε από τις προσομοιώσεις που κάναμε στο Choregraphe και από τη υλοποίηση του κάθε τμήματος ξεχωριστά στη ρομποτική πλατφόρμα του εργαστηρίου μας.

- Όλες οι κινήσεις των μηχανικών μερών του ρομπότ (κινήσεις χεριών και κεφαλιού) δουλεύουν άψογα τόσο στην προσομοίωση με το Choregraphe όσο και σε πραγματικό χρόνο στο Nao του εργαστηρίου μας.
- Όλες οι εντολές, οι δομές και το γενικότερο σχήμα του διαγράμματος καταστάσεων του σεναρίου που φτιάξαμε, έχει απόλυτη εφαρμογή και στην προσομοίωση και στο ρομπότ μας.
- Οι εντολές παραγωγής ήχου (όταν το ρομπότ πρέπει να πει κάποιες λέξεις ή προτάσεις για να δώσει στο παιδί να καταλάβει κάποια πράγματα), όπως αναφέραμε πιο πάνω δεν δουλεύουν στην προσομοίωση αλλά στο ρομπότ μας δουλεύουν κανονικά. Επομένως οι φωνητικές εντολές που δίνει το Nao στο παιδί δίνονται κανονικά.
- Όπως και η παραγωγή ήχου, έτσι και η αναγνώριση φωνής (voice recognition) δεν δουλεύει στην προσομοίωση στο Choregraphe. Σε πραγματικό χρόνο στο Nao, γενικά γίνεται αναγνώριση φωνής αλλά δεν μπορούμε να πούμε ότι τα αποτελέσματα είναι εντελώς ικανοποιητικά. Γενικά, η δυνατότητα αναγνώρισης φωνής που διαθέτει το Nao εμφανίζει αρκετά προβλήματα στην εφαρμογή της. Υπάρχουν περιπτώσεις που αυτό που λέμε ανιχνεύεται απόλυτα σωστά από το ρομπότ ενώ υπάρχουν άλλες περιπτώσεις που έχουμε αναγνώριση άλλης λέξης/πρότασης από αυτό που έπρεπε να αναγνωρίσει το ρομπότ. Λόγω του ότι κάθε άτομο, και δει τα παιδιά, έχει διαφορετική χροιά φωνής και διαφορετική τονικότητα, είναι πολλές φορές εξαιρετικά δύσκολο να αναγνωρίσει το Nao αυτό που ακριβώς του είπε το παιδί. Το παιδί, χρειάζεται να μιλά με συγκεκριμένο, αργό και σταθερό τρόπο και με δυνατή ένταση ούτως ώστε να γίνει αντιληπτό από το ρομπότ αλλά και πάλι δεν γίνεται πάντα επιτυχής αναγνώριση. Όσον αφορά τώρα, την αναγνώριση της φωνής που παράγεται από το VoiceBox του Arduino και πάλι τα πράγματα δεν ήταν ικανοποιητικά, αφού ενώ έδειχνε να αναγνωρίζει την ύπαρξη κάποιων λέξεων, αδυνατούσε κάποιες φορές να ξεχωρίσει ποια λέξη παραγόταν. Παρόλο που για να τελειοποιήσουμε τα αποτελέσματά μας, χρησιμοποιήσαμε ηχεία μεγαλύτερης έντασης τα οποία τοποθετήσαμε όσο το δυνατόν πιο κοντά στο μικρόφωνο που βρίσκεται τοποθετημένο στο μέτωπο του Nao, ενώ παράλληλα η όλη διαδικασία ηχητικής αναγνώρισης έγινε σε όσο το δυνατόν καλύτερες συνθήκες ηχομόνωσης από εξωτερικούς θορύβους, δυστυχώς και πάλι δεν καταφέραμε να έχουμε τα επιθυμητά αποτελέσματα.
- Η οπτική αναγνώριση των νομισμάτων, έγινε με απόλυτη επιτυχία και στην προσομοίωση στο Telepathe αλλά και στην πράξη στη συνεργασία του Nao με το Gostai Studio. Επιτύχαμε να παίρνουμε όσο έτρεχε το module για την οπτική αναγνώριση, μια μεταβλητή που μας έδινε στοιχεία για τα νομίσματα που ανιχνεύονταν με αποτέλεσμα να μπορούσαμε να χρησιμοποιήσουμε αυτή τη μεταβλητή για να εξακριβώσουμε την τιμή του κάθε νομίσματος. Το μόνο μειονέκτημα στην οπτική αναγνώριση, είναι το γεγονός ότι κάποιες φορές, ανάλογα με τη γωνία τοποθέτησης του νομίσματος αλλά και το φωτισμό, το νόμισμα δεν αναγνωριζόταν απευθείας και χρειαζόταν να το μετακινήσουμε έως ότου αναγνωριστεί από το ρομπότ. Όσες φορές, πάντως, το ρομπότ ανίχνευσε το αντικείμενο (οι οποίες ήταν πολύ περισσότερες από τις αποτυχημένες προσπάθειες οπτικής αναγνώρισης) έδινε με απόλυτη επιτυχία την ανίχνευση του είδους του

νομίματος, επομένως μπορούσαμε αμέσως να ξεχωρίσουμε πιο νόμισμα αναγνωρίστηκε.

- Η δεύτερη αποστολή που έπρεπε να εκτελέσει το παιδί, περιελάμβανε αλληλεπίδραση του παιδιού με τον μπροστινό αισθητήρα αφής του κεφαλιού του ρομπότ. Όπως διαπιστώσαμε και στην πράξη, επειδή η αναγνώριση της αφής ξεκινάει αμέσως με το που γίνεται ο έλεγχος για το αν έχει πατηθεί ή όχι ο αισθητήρας, αναγκαστήκαμε να δημιουργήσουμε ένα χρονικό παράθυρο ανάμεσα στις φορές που πρέπει το παιδί να πατάει τον αισθητήρα. Δηλαδή, πριν αρχίσει ο έλεγχος για το αν έχει πατηθεί ο αισθητήρας, εισάγουμε μια χρονοκαθυστέρηση των 5 δευτερολέπτων. Επίσης, επειδή δεν ήταν εφικτό το παιδί να πατάει με τη συχνότητα που θέλει τον αισθητήρα του ρομπότ αφού έτσι είχαμε πολλές φορές σφάλματα και το ρομπότ μπορούσε ενώ το παιδί πατούσε τον αισθητήρα μια φορά, να το εκλάβει ότι είχε πατηθεί δυο ή και περισσότερες φορές, δημιουργήσαμε ένα σύστημα το οποίο θα δείχνει στο παιδί πότε πρέπει να πατάει το διακόπτη. Έτσι, κατά την έναρξη του ελέγχου, ανοίγαμε τα δάχτυλα του αριστερού χεριού του Nao και έπειτα εισαγάγαμε χρονική καθυστέρηση 3 δευτερολέπτων. Στη συνέχεια το παιδί, είχε 3 δευτερόλεπτα για να πατήσει τον αισθητήρα και θα τον άφηνε μόλις τα δάχτυλα του αριστερού χεριού του Nao κλείνανε. Τότε θα υπήρχε ακόμη μια καθυστέρηση 3 δευτερολέπτων έως ότου να επαναληφθεί η διαδικασία αναγνώρισης. Συνολικά, επαναλάβαμε αυτή τη διαδικασία με το ανοιγοκλείσιμο των δαχτύλων του Nao 5 φορές, δίνοντας έτσι στο παιδί τη δυνατότητα να πατήσει τον αισθητήρα όσες φορές χρειάζεται, δηλαδή 2 φορές. Έτσι καταφέραμε να ολοκληρώσουμε με επιτυχία αυτό το κομμάτι που στην αρχή μας παρουσίασε πρόβλημα.
- Οι αποστολές που πρέπει να εκτελεστούν σε συνεργασία με το Arduino δουλεύουν στην εντέλεια όσον αφορά το κομμάτι που αφορά αποκλειστικά το Nao. Όταν το παιδί ανοίγει ένα κουτί για την αποστολή 3 ή πατά ένα διακόπτη για την αποστολή 4 τότε ανάλογα παράγονται οι αντίστοιχες λέξεις από το Arduino, όπως ακριβώς τις παρουσιάσαμε στο προηγούμενο κεφάλαιο. Η συνεργασία του Arduino με το Nao, γίνεται αποκλειστικά μέσω της αναγνώρισης φωνής από το Nao των λέξεων που θα παράγει το Arduino σε κάθε περίπτωση. Όπως αναφέραμε και προηγουμένως, τις περισσότερες φορές έχουμε επιτυχή αναγνώριση των λέξεων επομένως και η συνεργασία του Arduino με το Nao είναι άψογη αφού το ρομπότ αντιλαμβάνεται πλήρως αν η αποστολή έχει εκτελεστεί σωστά ή όχι. Στις περιπτώσεις όμως που δεν έχουμε σωστή αναγνώριση της λέξης, τότε αυτόματα το Nao εκλαμβάνει ότι η αποστολή έχει αποτύχει επομένως ζητά να επαναληφθεί η διαδικασία. Αυτό είναι πρόβλημα μόνο στην περίπτωση που η αποστολή έχει ολοκληρωθεί με επιτυχία αλλά το ρομπότ δεν αντιλαμβάνεται σωστά τη λέξη και θεωρεί ότι η αποστολή ήταν αποτυχής και ζητά από το παιδί να την επαναλάβει.

Όπως τονίσαμε και πιο πάνω, λόγω έλλειψης χρόνου από την καθυστερημένη άφιξη του Nao στο εργαστήριό μας, δεν προλάβσαμε να ελέγξουμε το σενάριο στην ολότητά του παρά μόνο αποσπασματικά. Αυτό συνέβηκε, κυρίως, γιατί όπως είδαμε, αρκετά τμήματα του σεναρίου που φτιάξαμε δεν είχαν απόλυτη επιτυχία (οπτική αναγνώριση νομισμάτων και ηχητική αναγνώριση φωνής). Επομένως, και λόγω της έλλειψης του χρόνου που μας απέτρεψε να εφαρμόσουμε τυχόν διαφοροποιήσεις στο σενάριο μέσω διαφόρων βελτιωτικών κινήσεων, δεν καταφέραμε να υλοποιήσουμε παρά μόνο μερικές φορές, το σενάριο με συνοχή. Έχουμε όμως αποδείξει ότι το σενάριο που δημιουργήσαμε δουλεύει, επομένως, μπορεί σε μια επόμενη διπλωματική εργασία να μελετηθεί η τελειοποίηση του σεναρίου ούτως ώστε να δουλεύει τέλεια σαν μια οντότητα όπως επίσης και να γίνει εφαρμογή των πειραμάτων τα οποία αναφέραμε πιο πάνω, ούτως ώστε να πάρουμε την άποψη και των παιδιών για το σενάριο που δημιουργήσαμε, που στην ουσία είναι για αυτά που προορίζεται το σενάριο.

Κεφάλαιο 7

Προβλήματα - Βελτιώσεις

7.1 Προβλήματα

Όπως έχουμε δει στα προηγούμενα κεφάλαια, κατά τη διάρκεια σχεδιασμού και υλοποίησης του σεναρίου, προέκυψαν διάφορα προβλήματα. Κάποια αυτά προσπαθήσαμε να τα επιλύσουμε με τον καλύτερο δυνατό τρόπο ενώ για κάποια άλλα θα μπορούσαμε να κάνουμε κάποιες βελτιώσεις τις οποίες θα αναλύσουμε στη συνέχεια. Τα προβλήματα που αντιμετωπίσαμε είναι τα ακόλουθα:

- Το πρώτο ερώτημα που μας προβληματίσε, ήταν το πόσα και ποια νομίσματα πρέπει το παιδί να χρησιμοποιήσει για τις αποστολές του. Λόγω του ότι το σενάριο αφορά παιδιά προσχολικής και πρώτης σχολικής ηλικίας, τα νομίσματα που έπρεπε να κατασκευάσουμε, έπρεπε να έχουν μικρή αξία αφού τα παιδιά τέτοιας ηλικίας έχουν λίγες γνώσεις αριθμητικής. Επίσης, επειδή εμείς κατασκευάσαμε τα νομίσματά μας, περιορίστηκαμε στην κατασκευή μόνο ενός από κάθε είδος. Έτσι είχαμε περιορισμό στις τιμές των αντικειμένων και συνεπώς των πράξεων που έπρεπε να κάνει το παιδί αφού τα νομίσματα μας είχαν όλα μαζί μέγιστη αξία τα 11 ευρώ.
- Ένα από τα αρχικά προβλήματα που αντιμετωπίσαμε ήταν ο αριθμός των νομισμάτων που μπορούσε να δώσει το παιδί στο ρομπότ. Στη θεωρία, το παιδί από τις προηγούμενες του αποστολές, μπορούσε να έχει αποκτήσει απεριόριστο αριθμό νομισμάτων. Το Nao, όμως, αν και στη θεωρία όπως τονίζει η κατασκευαστική εταιρία, αναγνωρίζει πολλά μαζί νομίσματα, έχει στην ουσία περιορισμένο αριθμό νομισμάτων που μπορεί να ανιχνεύσει, όσα δηλαδή βρίσκονται στο οπτικό πεδίο της κάμεράς του. Λόγω και του μεγάλου μεγέθους των νομισμάτων, επιλέξαμε το παιδί να τα τοποθετεί στο τραπέζι όπου το ρομπότ θα σκύβει και θα εφαρμόζει οπτική αναγνώριση. Αυτό όμως, περιορίζει το οπτικό πεδίο του Nao με αποτέλεσμα στην ουσία να μην μπορεί να ανιχνεύσει πολλά νομίσματα μαζί. Έτσι, θέσαμε τον περιορισμό, ότι το παιδί θα μπορεί να βάλει μέχρι και 3 ταυτόχρονα νομίσματα μαζί. Έτσι, και με δεδομένο την ποσότητα και την αξία των νομισμάτων που δημιουργήσαμε, το παιδί μπορούσε να βάλει στο τραπέζι για αναγνώριση νομίσματα συγκεκριμένης αξίας. Αυτό περιορίζει τις επιλογές μας όσον αφορά τις τιμές που έπρεπε να καθορίσουμε για το κάθε αντικείμενο αλλά και τα νομίσματα που θα έπαιρνε με ολοκλήρωση κάθε αποστολής.
- Ένα άλλο πρόβλημα που κληθήκαμε να επιλύσουμε, είναι η επικοινωνία του Arduino με το Nao. Καταλήξαμε τελικά σε ένα κύκλωμα παραγωγής ήχου το οποίο θα παράγει μια ηχητική εντολή (μια λέξη) την οποία μέσω του προγράμματος φωνητικής αναγνώρισης που διαθέτει το Nao, το ρομπότ θα το αναγνωρίζει και επομένως έτσι θα γίνεται η επικοινωνία ρομπότ με Arduino. Όπως είδαμε όμως στην πράξη, η ικανότητα αυτή που έχει το Nao αναγνώρισης της φωνής δεν είναι ακόμη απόλυτα τελειοποιημένη ενώ μάλιστα εμφανίζει και πολλά προβλήματα. Αν και προσπαθήσαμε να τελειοποιήσουμε το περιβάλλον με τη χρήση ρομποτικής φωνής και την ηχητική απομόνωση του χώρου, και πάλι υπήρχαν πολλές φορές που το ρομπότ δεν ανίχνευε σωστά τη λέξη που θέλαμε και έτσι είχαμε λάθος αποτέλεσμα. Αυτό, συνέβει και στην πρώτη αποστολή όπου το παιδί έπρεπε να πει στο ρομπότ την αξία των χρημάτων που χρειάζεται για

να αποκτήσει το αντικείμενο που θέλει. Τα πράγματα εδώ ήταν ακόμη χειρότερα, αφού η φωνή των ανθρώπων διαφέρει από άνθρωπο σε άνθρωπο ενώ μάλιστα τα παιδιά έχουν λεπτότερη φωνή, πράγμα που κάνει την αναγνώριση της φωνής από το ρομπότ ακόμη δυσκολότερη.

- Ένα άλλο πρόβλημα που ήρθε στην επιφάνεια κατά τη διαδικασία της υλοποίησης, αφορά την οπτική αναγνώριση των νομισμάτων. Δυστυχώς, δεν είχαμε απόλυτη επιτυχία στην οπτική αναγνώριση των νομισμάτων αφού κάποιες φορές η θέση που ήταν τοποθετημένο το νόμισμα αλλά και ο φωτισμός του περιβάλλοντος δεν επέτρεπαν στο ρομπότ να αναγνωρίσει το αντικείμενο, αν και αυτό προσπαθήσαμε να το αντιμετωπίσουμε με επαναλαμβανόμενη προσπάθεια καταγραφής των δεδομένων, μετακινώντας ταυτόχρονα και τη θέση του νομίσματος.
- Όπως είδαμε στην υλοποίησή μας, οι φωνητικές εντολές που έπαιρνε αλλά και έδινε το ρομπότ, ήταν όλες στα Αγγλικά. Δυστυχώς, ακόμη η εταιρία κατασκευής του ρομπότ δεν έχει εφοδιάσει το Nao με παραγωγή φωνητικών εντολών στα ελληνικά αλλά και ανίχνευση ελληνικών λέξεων. Αυτό είναι ένα πρόβλημα αν το σενάριο αφορά αποκλειστικά παιδιά που μιλούν μόνο ελληνικά, αν και για να είμαστε ρεαλιστές οι λέξεις που χρησιμοποιούνται είναι πολύ απλές και μπορούν να θεωρηθούν ως μια πρόωρη μάθηση κάποιων λέξεων στα αγγλικά.
- Όπως παρατηρήσαμε στην πράξη, για κάποια πράγματα το παιδί πρέπει πρώτα να έχει δει κάποια πράγματα για να μπορεί να τα εκτελέσει σωστά. Για παράδειγμα, όταν θα κληθεί να πατήσει τον αισθητήρα αφής του κεφαλιού του Nao πρέπει να γνωρίζει ότι θα αφήσει το χέρι του πατημένο στον αισθητήρα για το διάστημα στο οποίο το αριστερό χέρι του Nao θα έχει ανοιχτά τα δάχτυλά του. Επίσης, στην αποστολή με το εικονικό πληκτρολόγιο, θα πρέπει να γνωρίζει ότι όταν πατήσει λάθος κουμπί, για να πατήσει πρώτα μια νέα επιλογή πρέπει να επαναφέρει τον πρώτο διακόπτη στην αρχική του κατάσταση. Αυτά φυσικά, μπορούν να επιλυθούν με τη βοήθεια του επιβλέποντα ο οποίος και θα βρίσκεται συνέχεια μαζί με το παιδί.

7.2 Βελτιώσεις

Για κάποια από τα προβλήματα που αναφέραμε πιο πάνω, σκεφτήκαμε και παραθέτουμε κάποιες λύσεις, οι οποίες αν υλοποιούνταν ίσως να επιλύαν αυτά τα προβλήματα.

- Όσον αφορά το πρόβλημα που προέκυψε με την ποσότητα και το είδος των νομισμάτων, το πιο σωστό θα ήταν να δημιουργήσουμε πολλά νομίσματα διαφόρων αξιών (μέχρι τουλάχιστον το 10 και όχι μέχρι το 5 που κατασκευάσαμε εμείς), οπότε να δίνεται η δυνατότητα να εκτελεστούν περισσότερες και πιο περίπλοκες αριθμητικές πράξεις από το παιδί. Για να αντιμετωπίσουμε παράλληλα τον περιορισμό που προέκυψε και αφορά τον αριθμό των ταυτόχρονων νομισμάτων που μπορεί να αναγνωρίσει το Nao, μπορούμε να δημιουργήσουμε ένα βρόγχο στον οποίο θα βρίσκεται η διαδικασία οπτικής αναγνώρισης, και αυτό θα επαναλαμβάνεται κάποιες φορές ούτως ώστε να μπορούν να αναγνωριστούν όλα τα νομίσματα που θα δώσει το παιδί στο ρομπότ. Θα μπορούσαμε, για παράδειγμα να βάλουμε τη διαδικασία να εκτελείται 5 φορές, οπότε θα μπορούσαμε να καλύψουμε μέχρι και 15 νομίσματα που είναι ένα αρκετά μεγάλος αριθμός νομισμάτων. Αποφύγαμε να τα υλοποιήσουμε αυτά γιατί θεωρήσαμε ότι σε πρώτο στάδιο είναι σημαντικό να δούμε την εφαρμογή του σεναρίου για μικρές πράξεις και για λίγα νομίσματα. Οπότε αυτά τα βήματα μπορούν να εφαρμοστούν ως εξέλιξη του δικού μας σεναρίου.
- Μια καλύτερη μέθοδος επικοινωνίας του Arduino με το Nao θα ήταν η επικοινωνία με bluetooth. Υπάρχει διαθέσιμο ένα Bluetooth Module ¹ φτιαγμένο για Arduino το οποίο μπορούμε να χρησιμοποιήσουμε για να στέλνουμε απευθείας σήμα στο ρομπότ το οποίο θα το αναγνωρίσει και θα ξέρει αν έχει ολοκληρωθεί με επιτυχία ή όχι η αποστολή. Επίσης, υπάρχει και τρόπος σύνδεσης μέσω δικτύου, όπου το Arduino θα συνδεθεί μέσω Ethernet Shield ² στο τοπικό δίκτυο που βρίσκεται το ρομπότ και να στείλει έτσι πληροφορίες στο Nao. Αυτές οι μέθοδοι φυσικά, δεν υλοποιήθηκαν σε αυτή τη διπλωματική λόγω του υψηλού κόστους υλοποίησής τους αλλά και

¹<http://arduino.cc/en/Main/ArduinoBoardBluetooth>

²<http://www.arduino.cc/en/Main/ArduinoEthernetShield>

της δυσκολίας τους. Εάν όμως θέλουμε καλύτερα αποτελέσματα και μετεξέλιξη του παρόντος σεναρίου, αυτές οι προαναφερθέντες μέθοδοι είναι ο καλύτερος τρόπος αλληλεπίδρασης με το ρομπότ.

- Η αποστολή στην οποία ζητείται από το παιδί να πει την τιμή των χρημάτων που χρειάζεται, θα μπορούσε να αντικατασταθεί από μια άλλη αποστολή στην οποία να μην υπάρχει το κατά κάποιο τρόπο προβληματικό σύστημα αναγνώρισης φωνής του ρομπότ ενώ παράλληλα δεν θα έπρεπε και το παιδί να χρησιμοποιήσει την αγγλική γλώσσα για την υλοποίηση της αποστολής. Μια πιθανή αποστολή που θα μπορούσε να υλοποιηθεί, είναι να ζητηθεί από το παιδί να πάει σε ένα συγκεκριμένο τόπο μέσα στο δωμάτιο. Τότε, μέσα από το Arduino το οποίο θα ήταν συνδεδεμένο με ειδικές κάμερες, θα αναγνωριζόταν η θέση του παιδιού και θα δινόταν το κατάλληλο σήμα στο ρομπότ. Άλλη πιθανή αποστολή, θα ήταν να υλοποιήσουμε μια οθόνη αφής στην οποία το παιδί θα έπρεπε να γράψει με μια ειδική γραφίδα τον αριθμό των νομισμάτων που χρειαζόταν για να πάρει το επιθυμητό αντικείμενο. Αυτή, μάλιστα, λαμβάνοντας υπόψη και την εξοικείωση των σημερινών παιδιών με παιχνίδομηχανές που έχουν τέτοιες οθόνες, θα ήταν πολύ εύκολη στην αντίληψη για το παιδί. Φυσικά, αυτές οι αποστολές έχουν αρκετά υψηλό κόστος και μεγάλο βαθμό δυσκολίας όσον αφορά την υλοποίηση γι' αυτό και εμείς προτιμήσαμε να υλοποιήσουμε κάτι πιο απλό, στα πλαίσια πάντα μιας διπλωματικής εργασίας.

Επομένως, όπως είδαμε πιο πάνω, υπάρχουν λύσεις για τα προβλήματά μας τα οποία θα τελειοποιούσαν τη λειτουργία του σεναρίου και θα μας επέτρεπαν να το εφαρμόσουμε με συνοχή και έτσι να δοκιμάσουμε τα πειράματα τα οποία αναφέραμε. Αλλά κάποιοι λόγοι (έλλειψη χρόνου, μεγάλο κόστος, υψηλός βαθμός δυσκολίας υλοποίησης) μας ανάγκασαν να χρησιμοποιήσουμε τις λύσεις που αναφέραμε στο αρχικό μας σενάριο, με τις οποίες το σενάριο δουλεύει απλά παρουσιάζονται κάποιες φορές κάποια προβλήματα τα οποία μπορεί να δυσκολέψουν τη χρήση του.

Κεφάλαιο 8

Αξιολόγηση - Συμπεράσματα

8.1 Γενική Αξιολόγηση - Συμπεράσματα

Για να αξιολογήσουμε το σενάριο που υλοποιήσαμε, ας θυμηθούμε ποιος ήταν ο αρχικός μας στόχος: θέλαμε να δημιουργήσουμε ένα σενάριο εκμάθησης για παιδιά προσχολικής και πρώτης σχολικής ηλικίας σε συνεργασία με τη ρομποτική πλατφόρμα Nao του Εργαστηρίου μας, αξιοποιώντας της έννοιες της εκπαιδευτικής ρομποτικής, της θεωρίας του Νου, της παρακολούθησης του βλέμματος (gaze following) και της θεωρίας ροής παιχνιδιών σοβαρού σκοπού (Flow Theory), βασιζόμενοι σε ένα νέο προγραμματιστικό περιβάλλον, το Gostai Studio. Έτσι, κατασκευάσαμε ένα σενάριο στο οποίο το παιδί θα αλληλεπιδρά με ένα ρομπότ με στόχο τη μάθηση κάποιων βασικών αριθμητικών εννοιών. Για το σκοπό αυτό, χρησιμοποιήσαμε το Nao, ένα ανθρωπόμορφο ρομπότ το οποίο είναι πιο οικείο σε παιδιά, ενώ μάλιστα μας προσφέρει πλήρη αξιοποίηση των εννοιών της παρακολούθησης του βλέμματος και της θεωρίας του Νου, αφού το Nao με τη μορφή που διαθέτει έχει την ικανότητα να υποκαθιστά ένα ανθρώπινο ον και έτσι το παιδί να βλέπει στα μάτια του αυτά που συναντά στο δάσκαλό του που σε αυτή την ηλικία είναι αυτός που τον εφοδιάζει με γνώσεις. Έτσι, δημιουργήσαμε στην ουσία ένα παιχνίδι σοβαρού σκοπού, με το οποίο το παιδί θα έπαιζε αλλά κατά τη διάρκεια του παιχνιδιού θα αναγκαζόταν να κάνει κάποιες μαθηματικές πράξεις για να προχωρήσει στο επόμενο επίπεδο. Άρα, καταφέραμε να δημιουργήσουμε ένα σενάριο μέσα από το οποίο το παιδί θα εξασκούσε κάποιες βασικές αριθμητικές πράξεις πρόσθεσης και αφαίρεσης, εφάμιλλες με αυτές που συναντά ένα παιδί της ηλικίας του.

Όπως, όμως, τονίσαμε στο προηγούμενο κεφάλαιο, δεν ήταν στα πλαίσια της παρούσας διπλωματικής εργασίας να εκτελέσουμε μια σειρά πειραμάτων για να εξακριβώσουμε από τα ίδια τα παιδιά την αποτελεσματικότητα αυτού του σεναρίου που κατασκευάσαμε, ενώ δεν είχαμε και το χρόνο λόγω της καθυστερημένης άφιξης του ρομπότ να δοκιμάσουμε το σενάριο στην ολότητά του έστω και σε μεμονωμένες περιπτώσεις. Παρόλα αυτά, έχουμε εντοπίσει κάποια στοιχεία τα οποία πολύ πιθανόν να αναδεικνύονταν από τα ίδια τα παιδιά κατά την πρακτική εφαρμογή του σεναρίου:

- Οι πράξεις τις οποίες ένα παιδί θα έπρεπε να κάνει, είναι αρκετά απλές. Δεδομένο, ότι η εκπαιδευτική μάθηση των παιδιών έχει προχωρήσει πολύ σήμερα και τα παιδιά αρχίζουν να μαθαίνουν από πολύ μικρά τις βασικές έννοιες αριθμητικής, το σενάριο που φτιάξαμε μπορεί να φανεί πολύ εύκολο στα παιδιά. Επομένως, καλό θα ήταν να εισάξουμε και πιο πολύπλοκες μαθηματικές πράξεις πρόσθεσης και αφαίρεσης, αριθμών μεγαλύτερων του 10.
- Το παιδί στην αρχή μπορεί να δυσκολευτεί να αφομοιώσει την αλληλεπίδραση με το ρομπότ λόγω του ότι είναι πιθανόν να βλέπει κάποια πράγματα πρώτη φορά. Επομένως, η παρουσία του διδάσκοντα ή του γονιού, είναι απαραίτητη κατά τη διάρκεια υλοποίησης του σεναρίου, αφού χρειάζεται να δοθούν κάποιες διευκρινήσεις στο παιδί.
- Είναι σχεδόν σίγουρο ότι θα υπάρξει ένα πρόβλημα με τη χρήση της Αγγλικής γλώσσας από το ρομπότ αφού θα είναι κάτι εντελώς καινούριο για το παιδί. Αν και στις μέρες μας, τα παιδιά

τέτοιας ηλικίας γνωρίζουν πολύ καλά τις λέξεις yes, no, right που θα χρησιμοποιήσει το ρομπότ οπότε στην ουσία απλά θα χρειαστεί η παρέμβαση του διδάσκοντα για τη μετάφραση κάποιων πιο μεγάλων εκφράσεων που θα χρησιμοποιήσει το ρομπότ. Αυτό είναι και ένα από τα προβλήματα το οποίο αναλύσαμε στο προηγούμενο κεφάλαιο.

- Ίσως κάποια παιδιά, να αναφέρουν ότι θα περίμεναν περισσότερες κινήσεις από το ρομπότ, κυρίως πέραν των χεριών του. Δηλαδή, ίσως θα ήταν καλύτερο να ενσωματώναμε στο σενάριο μας κάποιες μετακινήσεις του ρομπότ, ένα περπάτημα, να καθίσει κάτω και μετά να σηκωθεί. Πράγματα που για τα παιδιά είναι άκρως εντυπωσιακά και θα κεντρίσουν περισσότερο το ενδιαφέρον τους.

Παρόλες όμως αυτές τις παρατηρήσεις που ενδεχομένως να έκαναν τα παιδιά, θεωρούμε ότι ο στόχος μας έχει επιτευχθεί με επιτυχία. Δηλαδή, το σενάριο που κατασκευάσαμε, αποτελεί μια πιο ευχάριστη και συνάμα ενδιαφέρουσα προσέγγιση στη μάθηση και την εκπαίδευση των παιδιών. Ένα παιδί μπορεί μέσω του σεναρίου που κατασκευάσαμε να παίζει το παιχνίδι του και συνάμα να εξασκήσει και να διευρύνει τις γνώσεις του στις αριθμητικές του ικανότητες, που είναι κάτι στο οποίο εξασκούνται πολύ τα παιδιά σε αυτήν την ηλικία. Αυτή είναι μια μικρή αρχή που μπορεί να γίνει για να εισαχθεί σιγά σιγά στα σχολεία η εκπαιδευτική ρομποτική η οποία θα βοηθήσει πολύ την εκπαιδευτική διαδικασία. Σίγουρα, το ρομπότ που χρησιμοποιήσαμε είναι σχετικά ακριβό με αυτά που μπορεί να διαθέσει ένα σχολείο, αλλά το σενάριο μας έχει γενικότερο σκοπό οπότε μπορεί να εφαρμοστεί και σε διάφορες άλλες ρομποτικές πλατφόρμες.

Πέρα από το βασικό μας σκοπό τον οποίο αναλύσαμε πιο πάνω, στόχος αυτής της διπλωματικής ήταν να αναδείξουμε το νέο προγραμματιστικό περιβάλλον του Gostai Studio ως ένας νέος και αρκετά εύκολος τρόπος επικοινωνίας και αλληλεπίδρασης με διάφορα ρομπότ, και συγκεκριμένα με το Nao το οποίο διαθέτει το Εργαστήριό μας. Όπως διαπιστώσαμε, το πρόγραμμα αυτό έχει πάρα πολλές δυνατότητες ενώ προσφέρει μεγαλύτερη ευχέρεια προγραμματισμού με τη χρήση των διαγραμμάτων κατάστασης. Πλέον, το πρόγραμμα αυτό είναι πλήρως συμβατό με το συγκεκριμένο ρομπότ αλλά και με τα περισσότερα ρομπότ της αγοράς. Παρόλα αυτά, έχουμε διαπιστώσει κάποια προβλήματα τα οποία κρίναμε σημαντικό να θίξουμε στο σημείο αυτό.

- Παρότι πλέον το πρόγραμμα αυτό που βασίζεται στην πλατφόρμα ανοιχτού κώδικα URBI, είναι αναγνωρισμένο από την εταιρία κατασκευής του Nao, πολλές είναι οι περιπτώσεις όπου υπάρχουν ασυμφωνίες συμβατότητας ακόμη και στις εκδόσεις τελευταίων προγραμμάτων. Για παράδειγμα, στην αρχή χρησιμοποιήσαμε το NaoQi Framework v.1.10.44 που είναι το τελευταίο που έχει βγει από την Aldebaran και αναμέναμε ότι η URBI θα ήταν πλήρως συμβατό με αυτό, όπως ακριβώς είχε ανακοινώσει η Aldebaran στο Release Note της συγκεκριμένης έκδοσης. Προς μεγάλη μας έκπληξη, όμως, η έκδοση αυτή του NaoQi δεν ήταν συμβατή με το URBI και έτσι χρειαστήκαμε να χρησιμοποιήσουμε μια παλαιότερη έκδοση με λιγότερες δυνατότητες. Προβλήματα παρουσιάστηκαν και στα ίδια τα προγράμματα της Gostai τα οποία περιείχαν αρκετά bugs παρότι έχει σχεδόν 5 χρόνια που κυκλοφορούν στην αγορά.
- Το εγχειρίδιο χρήσης του Gostai, είναι πολύ γενικό και αναφέρει πολύ λίγα για τη σχέση του Gostai με το Nao. Μάλιστα, σε κάποιες φορές, τα πιο γενικά πράγματα που αναφέρονται στο εγχειρίδιο αυτό, δεν έχουν πρακτική εφαρμογή με το Nao. Έτσι, πολλές ήταν οι φορές που χρειαστήκαμε να ψάξουμε μόνοι μας πώς θα γίνει η αλληλεπίδραση του προγράμματος με το Nao (κλασικά παραδείγματα είναι η οπτική αναγνώριση των νομισμάτων και η αναγνώριση φωνής που δεν αναφέρονταν καθόλου στο συγκεκριμένο εγχειρίδιο).
- Το σχετικό forum της εταιρίας ¹ στο οποίο μπορούν να γίνουν απορίες για διάφορα θέματα σχετικά, συμμετέχουν οι προγραμματιστές της URBI αλλά δυστυχώς η βοήθεια που προσφέρουν είναι ανεπαρκής. Πολλές ήταν οι φορές που ενώ απευθυνθήκαμε σε αυτό το forum είτε πείραμε καθυστερημένη απάντηση είτε δεν μπορούσαν να μας επιλύσουν το πρόβλημά μας.

¹<http://forum.gostai.com/>

Επομένως, το Gostai Studio, είναι ένα πρόγραμμα το οποίο έχει απεριόριστες δυνατότητες όσον αφορά την αλληλεπίδραση με ρομποτικές πλατφόρμες και δει το Nao. Το δυνατό του σημείο είναι η διαγραμματική λειτουργία (state diagrams) κάτι εντελώς πρωτοποριακό στο χώρο αυτό. Δυστυχώς, όμως, η βοήθεια που προσφέρεται από τους δημιουργούς του έχει ακόμη πάρα πολλά προβλήματα και τεράστια περιθώρια βελτίωσης. Παρόλα αυτά, αν αναλογιστούμε ότι το συγκεκριμένο πρόγραμμα έχει μόνο 5 χρόνια που έχει δημιουργηθεί, θεωρούμε ότι στο μέλλον θα αποτελέσει ένα από τα πιο δυνατά χαρτιά όσον αφορά τον προγραμματισμό των ρομποτικών συστημάτων.

Τέλος, αξίζει να αναφερθούμε και στο μικροελεγκτή ανοιχτού κώδικα Arduino. Το Arduino, όπως είδαμε πολύ καλά στις αποστολές μας, έχει απίστευτες δυνατότητες και εξελίσσεται συνεχώς. Πλέον, μπορείς να το χρησιμοποιήσεις για να φτιάξεις ότι φανταστείς. Επίσης, γράφονται συνεχώς πάρα πολλά βιβλία τα οποία σχετίζονται με διάφορα έργα σε Arduino, ενώ παράλληλα βγαίνουν συνεχώς νέα υλικά τα οποία μπορείς να χρησιμοποιήσεις για να ολοκληρώσεις την εργασία σου. Από την προσωπική μας εμπειρία με το Arduino, είναι κάτι το οποίο αξίζει να ασχοληθεί κάποιος, ενώ μάλιστα θα ήταν εξαιρετικά χρήσιμη η εισαγωγή του και στα Πανεπιστήμια και η αντικατάσταση παλαιότερων μικροελεγκτών, αφού πλέον το Arduino αποτελεί την τελευταία λέξη της τεχνολογίας στον τομέα αυτό.

8.2 Η δική μας συνεισφορά

Αρκετά από τα θέματα που αναλύσαμε και υλοποιήσαμε στο σενάριο εκμάθησης βασικών αριθμητικών γνώσεων που υλοποιήσαμε, είδη έχουν αναλυθεί και επεξεργαστεί σε διάφορα άλλα επίσημα έγγραφα. Θα δούμε όμως τώρα, τι ξεχωριστό κάναμε εμείς που αποτελεί τη δική μας προσωπική συνεισφορά και ένα επιπλέον λιθαράκι στην όλη διαδικασία της εκπαιδευτικής ρομποτικής. Έτσι, έχουμε τα ακόλουθα σημεία:

- Σε αντίθεση με ό,τι έχει χρησιμοποιηθεί μέχρι τώρα, εμείς δημιουργήσαμε ένα σενάριο εκμάθησης το οποίο ήταν βασισμένο σε διαγράμματα καταστάσεων. Ως επί το πλείστον, μέχρι τώρα, παρόμοιες μελέτες γίνονταν χωρίς τη χρήση σεναρίου. Εμείς, εισάγοντας την ιδέα του σεναρίου μέσω διαγραμμάτων κατάστασης, καταφέραμε να δημιουργήσουμε μια γενικότερη ιδέα που αφορά την εκπαιδευτική ρομποτική, η οποία μπορεί να εφαρμοστεί όχι μόνο στο Nao και το Gostai Studio, αλλά γενικότερα σε διάφορες ρομποτικές πλατφόρμες και περιβάλλοντα προγραμματισμού.
- Για τον προγραμματισμό του Nao, χρησιμοποιήσαμε το Gostai Studio το οποίο δεν είναι ευρέως διαδεδομένο ακόμη. Γενικά, είναι ένας νέος τρόπος αλληλεπίδρασης με ρομποτικά συστήματα ο οποίος σιγά σιγά εξελίσσεται και μετατρέπεται σε ένα πολύ δυνατό εργαλείο στα χέρια των προγραμματιστών αφού έχει απίστευτες δυνατότητες ενώ πάνω από όλα είναι το μόνο που χρησιμοποιεί τον προγραμματισμό με βάση τα διαγράμματα καταστάσεων.
- Για τις αποστολές που έπρεπε να εκτελέσει το παιδί, δημιουργήσαμε μια συνεργασία του Nao με το Arduino, το οποίο αν και αναπτύσσεται ραγδαία, γενικά ακόμη δεν είναι ευρέως γνωστό και δεν χρησιμοποιείται από πολλούς. Έτσι, αναδείξαμε αυτήν την καινούρια πλακέτα μικροελεγκτή και δείξαμε ότι έχει πάρα πολλά να προσφέρει ενώ μάλιστα η συνεργασία της με το Nao είναι και που διευκολύνει κατά πολύ τη δημιουργία σεναρίων αλλά και άλλων εργασιών.
- Δημιουργήσαμε κάποια κομμάτια κώδικα τα οποία δεν αναφέρονται σε κάποιο εγχειρίδιο ή σε κάποιο σχετικό forum. Για παράδειγμα, η οπτική αναγνώριση των Naomarks αλλά και η αναγνώριση φωνής στη γλώσσα urbiscript, είναι αποκλειστικά δική μας δημιουργία αφού δεν υπήρχε κάτι παρόμοιο. Οπότε συνδυάσαμε τη γνώση μας για τα συγκεκριμένα σε άλλες γλώσσες προγραμματισμού και παρήγαμε αυτό το αποτέλεσμα.
- Κατά τη διάρκεια επεξεργασίας του σεναρίου, ανακαλύψαμε διάφορα προβλήματα με το Gostai Studio τα οποία και αναφέραμε στην κατασκευάστρια εταιρία. Δηλαδή, διαπιστώσαμε προβλήματα τύπου bugs στις παλαιότερες εκδόσεις του Gostai Studio όσον αφορά τις μεταβά-

σεις καταστάσεων, ενώ ακόμη ανακαλύψαμε πρόβλημα ασυμβατότητας του NaoQi Framework v.1.10.44 με το URBI, παρόλο που η κατασκευάστρια εταιρία αναφέρει ότι είναι συμβατό με URBI.

Συμπερασματικά, καταφέραμε να επιτύχουμε το στόχο μας για τη δημιουργία ενός σεναρίου εκμάθησης βασικών αριθμητικών γνώσεων με τη χρήση μιας ρομποτικής πλατφόρμας που θα συμβάλει πολύ στην προαγωγή της εκπαιδευτικής ρομποτικής, επιτυγχάνοντας παράλληλα να εισάξουμε κάποιες νέες έννοιες οι οποίες μπορούν να χρησιμοποιηθούν σε μελλοντικές έρευνες και εργασίες για να διευκολύνουν παρόμοιες διαδικασίες.

Βιβλιογραφία

- [SOL1] Σολομωνίδου Χριστίνα, *Εκπαιδευτική τεχνολογία: μέσα, υλικά: διδακτική χρήση και αξιοποίηση*, εκδ. Καστανιώτης, Αθήνα 1999
- [SOL2] Σολομωνίδου Χριστίνα, *Νέες τάσεις στην εκπαιδευτική τεχνολογία: εποικοδομητισμός και σύγχρονα περιβάλλοντα μάθησης*, Εκδ.Μεταίχμιο, Αθήνα 2006
- [ITM] Smaldino, Russel, Heinich, Molenda, *Instructional Technology and Media for Learning*, 2004
- [BMP] X. Βρασίδης, Μ. Ζεμπύλας, Α. Πέτρου, *Σύγχρονα παιδαγωγικά μοντέλα και ο ρόλος της εκπαιδευτικής τεχνολογίας*, <http://vrasidas.intercol.edu/educ534/vrasZembPetrou.doc>
- [EDR] Βεργίνης, Φράγκου *Διδακτικές Προσεγγίσεις και Εργαλεία για τη Διδασκαλία της Πληροφορικής*, 2009
- [EDR2] Σ. Αναγνωστάκης, Α. Μαργετουσάκη, Π. Γ. Μιχαηλίδης *Δυνατότητα Εργαστηρίου Εκπαιδευτικής Ρομποτικής στα Σχολεία*, 2008, http://www.ecedu.upatras.gr/didinfo/eishghseis_DIDINFO08/DIDINFO08_243_252.pdf
- [EDR3] <http://edurobotics.weebly.com/index.html>
- [EDR4] S.Napper, R.Seaman, *Applications of Robots in Rehabilitation*
- [EDR5] R.Severson, S. Carlson, *Behaving as or behaving as if? Children's conceptions of personified robots and the emergence of a new ontological category*, 2010
- [EDR6] H. Kozima, C. Nakagawa, Y. Yasuda, *Children-robot interaction: a pilot study in autism therapy*, 2007
- [EDR7] B. Denis, S. Hubert, *Collaborative learning in an educational robotics environment*, 2001
- [EDR8] T. Beran, A. Ramirez-Serrano, R. Kuzyk, M. Fior, S. Nugent, *Understanding how children understand robots: Perceived animism in child-robot interaction*, 2011
- [THM] Baron-Cohen, Simon, *Mindblindness: An Essay on Autism and Theory of Mind*, 1995.
- [GZT] A. Meltzoff, R. Brooks, A. Shonb, R. Raob, *“Social” robots are psychological agents for infants: A test of gaze following*, 2010
- [FSG] Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*, London: Harper Perennial, 1990
- [FTH] Isabelle Astic, Coline Aunis *A Ubiquitous Mobile Edutainment Application for Learning Science through Play*, 2011, http://conference.archimuse.com/mw2011/papers/a_ubiquitous_mobile_edutainment_application_fo

- [SCE] B. Robins, E. Ferrari, K. Dautenhahn, G. Kronreif, B. Prazak-Aram, G. Gelderblom, B. Tanja, F. Caprino, E. Laudanna, P. Marti, *Human-centred design methods: Developing scenarios for robot assisted play informed by user panels and field trials*, 2010
- [GOS] *Gostai Official Site*, <http://www.gostai.com/products/studio/>
- [URB] *URBI Official Site*, <http://www.urbiforge.org/>
- [NAO] *Aldebaran's Nao*, <http://www.aldebaran-robotics.com/en/naoeducation>
- [NAOD] *Documentation for Nao*, http://users.aldebaran-robotics.com/docs/site_en/index_doc.html
- [ARD] *Arduino Official Site*, <http://www.arduino.cc/>
- [URB2] Gostai, *The Urbi Software Development Kit*, 2011, <http://www.gostai.com/downloads/urbi/2.7.1/doc/urbi-sdk.pdf>
- [URBF] *Urbi Gostai Forum*, <http://forum.gostai.com/>
- [NAOM] *Naomark Detection*, http://users.aldebaran-robotics.com/docs/site_en/reddoc/naomark_detection/naomark_detection.html
- [SPK] *Sparkfun's SpeakJet*, <http://www.sparkfun.com/products/9578>
- [VOB] *Sparkfun's VoiceBox Shield*, <http://www.sparkfun.com/products/9799>
- [ARD1] *Sparkfun Inventor's Guide* <http://www.sparkfun.com/products/10173>, <http://www.oomlout.com/a/products/ardx/>
- [ARD2] J. Ozer, H. Blemings, *Practical Arduino: Cool Projects for Open Source Hardware*, 2009
- [ARD3] M. McRoberts, *Beginning Arduino*, 2010