



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Φορητού συστήματος επαυξημένης πραγματικότητας για εικονική εξιστόρηση
Όνοματεπώνυμο Φοιτητή	Στέφανος Κουγιουμτζής
Πατρώνυμο	Πέτρος
Αριθμός Μητρώου	ΜΠΣΠ/ 09061
Επιβλέπων	Παναγιωτόπουλος Θεμιστοκλής, Καθηγητής

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Περιεχόμενα

Περιεχόμενα.....	3
Περίληψη	5
Abstract	5
1: Εισαγωγή.....	6
2: State – of – the – art.....	7
3: Σχεδιασμός Εφαρμογής.....	17
Μέθοδοι και Προσέγγιση	17
Τεχνολογίες	17
Εξοπλισμός	17
4: Υλοποίηση Πρωτοτύπου	18
Τεχνολογίες Augmented Reality	18
Επιλογή Υλικού.....	19
Η κάμερα	19
Γυαλιά Video.....	19
ΦορητόςΥπολογιστής.....	20
Βασικές Βιβλιοθήκες.....	21
ARtoolkit	21
OpenSceneGraph.....	21
OSGart	21
Άλλα συστήματα Marker Based AR	21
Περιβάλλον προγραμματισμού	22
Επιλογή Περιβάλλοντος.....	22
Εγκαταστάσεις και χτίσιμο βιβλιοθηκών	22
Αρχιτεκτονική Λογισμικού	23
Κυριότερα χαρακτηριστικά λογισμικού.....	24

Tracker και Video Initialization	24
Widgets.....	27
Εμφάνιση και Απαλοιφή επιλογών μέσω Callbacks.....	28
Keyboard Handlers	29
Proximity Callbacks.....	30
Animation Handling	31
Custom Events	32
Αντικείμενα 3D.....	34
Αρχεία που χρησιμοποιήθηκαν.....	34
Εισαγωγή Animation και έλεγχος animation clips.....	35
5: Επαυξημένη Εξιστόρηση	44
Ανάλυση σεναρίου.....	44
Εξέλιξη ιστορίας, Βήμα – Βήμα.....	49
6: Πλεονεκτήματα – Συγκρίσεις	95
7: Επόμενα Βήματα	96
8: Επίλογος.....	96
9: Βιβλιογραφία	97

Περίληψη

Η παρούσα αναφορά έχει σκοπό να περιγράψει τη δημιουργία ενός τεχνητού περιβάλλοντος όπου ο χρήστης θα συμμετέχει σε μια εικονική εξιστόρηση. Για να δώσουμε στον χρήστη μια πειστική και κοντά στην πραγματικότητα εμπειρία, χρησιμοποιούνται τεχνικές που επαυξάνουν το περιβάλλον του. Με τον τρόπο αυτό πετυχαίνουμε να διατηρήσουμε τις φυσικές ιδιότητες της πραγματικότητας ενώ επιπρόσθετα εικονικά αντικείμενα τριών διαστάσεων και επιπρόσθετες υφές, την αλλάζουν σε πραγματικό χρόνο, σύμφωνα με τις ανάγκες ενός σεναρίου. Το επαυξημένο περιβάλλον δημιουργείται με τη χρήση της τεχνικής Marker Based augmentation και τα αντιπαραβαλλόμενα αντικείμενα προβάλλονται στον εξοπλισμό που φορά και μεταφέρει ο χρήστης. Στην παρούσα υλοποίηση έγινε η προσπάθεια να συμπεριληφθούν μη ελέγξιμοι βοηθητικοί χαρακτήρες και δημιουργήθηκε η υποδομή ώστε να υποστηρίζεται ο προγραμματισμός διαφορετικών αντιδράσεων για αυτούς, αναλόγως της ροής της ιστορίας. Ο χρήστης είναι ελεύθερος να εξερευνήσει το επαυξημένο περιβάλλον χωρίς περιορισμούς, μπορεί να ενεργήσει σε διάφορα στοιχεία του αλλά και να παρακολουθήσει τα διάφορα δρώμενα από όποιο σημείο επιθυμεί.

Abstract

The current report describes the creation of an artificial environment where the user participates in a virtual story telling. To pass to the user a feasible and near to reality experience we use techniques that augment his environment. In this way we achieve to maintain the physical properties of reality while additionally, virtual three dimensional objects and textures are added in the users visual field and change at real time. The augmented environment is created using the marker based technique and the augmentations are shown on the wearable equipment that the user carries. In this current approach we have tried to implement non playable characters as helpers to the storyline and we created the infrastructure that can hold different actions and reactions for them for different flows of the story. The use is able to explore the augmented environment without limitations, can perform actions in various objects and of course, is able to watch all the action from any desirable point of view.

1: Εισαγωγή

Η παρούσα αναφορά έχει σκοπό να περιγράψει τη δημιουργία ενός τεχνητού περιβάλλοντος όπου ο χρήστης θα συμμετέχει σε μια εικονική εξιστόρηση. Για να δώσουμε στον χρήστη μια πειστική και κοντά στην πραγματικότητα εμπειρία, χρησιμοποιούνται τεχνικές που επαυξάνουν το περιβάλλον του. Με τον τρόπο αυτό πετυχαίνουμε να διατηρήσουμε τις φυσικές ιδιότητες της πραγματικότητας ενώ επιπρόσθετα εικονικά αντικείμενα τριών διαστάσεων και επιπρόσθετες υφές, την αλλάζουν σε πραγματικό χρόνο, σύμφωνα με τις ανάγκες ενός σεναρίου. Το επαυξημένο περιβάλλον δημιουργείται με τη χρήση της τεχνικής Marker Based augmentation και τα αντιπαραβαλλόμενα αντικείμενα προβάλλονται στον εξοπλισμό που φορά και μεταφέρει ο χρήστης. Στην παρούσα υλοποίηση έγινε η προσπάθεια να συμπεριληφθούν μη ελέγξιμοι βοηθητικοί χαρακτήρες και δημιουργήθηκε η υποδομή ώστε να υποστηρίζεται ο προγραμματισμός διαφορετικών αντιδράσεων για αυτούς, αναλόγως της ροής της ιστορίας. Ο χρήστης είναι ελεύθερος να εξερευνήσει το επαυξημένο περιβάλλον χωρίς περιορισμούς, μπορεί να ενεργήσει σε διάφορα στοιχεία του αλλά και να παρακολουθήσει τα διάφορα δρώμενα από όποιο σημείο επιθυμεί.

Όσο η τεχνολογία προχωρά οι ανάγκες και οι σκοποί της εικονικότητας γίνονται όλο και πιο σαφείς. Οι εφαρμογές που έχουν ήδη κυκλοφορήσει εκτείνονται στους τομείς της εκπαίδευσης, της ψυχαγωγίας, της εξομοίωσης αλλά και της τηλεπαρουσίας. Αυτό που έχει αποτελέσει σημαντικό παράγοντα σε όλες τις παραπάνω υλοποιήσεις, είναι ο τρόπος που χρήστης αντιλαμβάνεται το τεχνητό περιβάλλον και με ποιούς τρόπους μπορεί αυτός να γίνει μέρος αυτού του περιβάλλοντος.

Μέσα από τα χρόνια της εξέλιξης της εικονικής πραγματικότητας, οι ερευνητές έχουν καταφέρει να μεταφέρουν τις αισθήσεις της όρασης, της ακοής, της αφής και πρόσφατα της όσφρησης, μέσα στους εικονικούς κόσμους, δίνοντας στον χρήστη τη δυνατότητα να ζήσει μια εμπειρία πραγματικά πολύ κοντά στην πραγματικότητα. Έτσι, η πρόοδος στον τομέα της αντίληψης, μπορούμε να πούμε, ότι έχει προχωρήσει σε αρκετά.

Όμως τι γίνεται για την αίσθηση της πραγματικής "ύπαρξης" μέσα σε ένα περιβάλλον; Αυτή η αίσθηση είναι συνολική και δημιουργείται όταν κινολυμαστέ ελεύθερα, όταν κάνουμε διάλογο και όταν λαμβάνουμε φυσιολογικές και έξυπνες συμπεριφορές από τους περιβάλλοντες χαρακτήρες. Πως είναι δυνατό να επιτευχθεί εμπύθιση όταν ο χρήστης δεν έχει τη δυνατότητα να κινηθεί φυσικά και χωρίς περιορισμούς και δεν έχει την αίσθηση του βάρους και του σχήματος των αντικειμένων όπως αν τα κρατούσε; Πώς θα άλλαζε μια εμπειρία εικονικής πραγματικότητας εάν δινόταν η δυνατότητα μιας συζήτησης που θα διατηρούσε τα φυσικά χαρακτηριστικά της, όπως γωνία θέασης ή η ένταση φωνής του εικονικού συνομιλητή;

Ο στόχος αυτής της εργασίας είναι η δημιουργία ενός περιβάλλοντος που δεν θα είναι περιορισμένο από την ύπαρξη μεγάλων οθονών, καλωδίων, μεγάλων κατασκευών ή σκοτεινών θαλάμων. Το περιβάλλον αυτό θα χρησιμοποιεί τεχνικές επαυξημένης πραγματικότητας που θα δένουν τον φυσικό κόσμο με τον εικονικό και θα παρουσιάζουν στον χρήστη μια εναλλακτική πραγματικότητα που θα φιλοξενεί δρώμενα συνθέτοντας ιστορίες ή εξομοιώσεις σεναρίων[9]. Τέλος, γίνεται η προσπάθεια να κάνουμε τον χρήστη να ξαναζήσει μια αλληλουχία γεγονότων σε έναν συγκεκριμένο τόπο και να ξανα ζήσει μια ιστορία σαν να ήταν κομμάτι της.

2: State – of – the – art

Μέχρι στιγμής έχει γίνει αρκετή έρευνα αλλά και πολλά ουσιαστικά βήματα στην αλληλεπίδραση του χρήστη εντός των εικονικών περιβαλλόντων. Η τάση που επικρατεί είναι προς την κατασκευή συστημάτων που είναι συνδεδεμένα με την καθημερινότητά μας και κατά κάποιο τρόπο επεκτείνουν την πληροφορία που παίρνουμε από τον φυσικό κόσμο. Παρακάτω αναφέρονται ερευνητικά θέματα φορητών συστημάτων, αφήγησης επαυξημένης πραγματικότητας και συσκευών αφής που έχουν καταφέρει εν μέρει τον σκοπό της ένωσης της εικονικότητας με την πραγματικότητα.

Οι τεχνικές της επαυξημένης πραγματικότητας έχουν ήδη απασχολήσει τον τομέα των ευφώνων περιβαλλόντων σαν μια πιθανότητα επέκτασης στον τρόπο που κατασκευάζονται οι τεχνικές πραγματικότητες[5]. Με τη συγχώνευση αυτών των δυο, διάφορα ζητήματα προέκυψαν σχετικά με το είδος του περιεχομένου σε αυτές τις πραγματικότητες, τις τεχνικές αλληλεπίδρασης, το είδος του εξοπλισμού και τέλος σε σχέση με τις εφαρμογές που θα μπορούσαν να εποφεληθούν από τέτοια υβριδικά συστήματα. Σε αυτή την εργασία γίνεται αναφορά στην οικογένεια των εφαρμογών που χρησιμοποιούν φορητές συσκευές και επιτρέπουν λειτουργικότητα ανά χρήστη και τελικά συνθέτουν μια εξατομικευμένη εμπειρία [5, 15].

Μια υλοποίηση για επαυξημένο περιεχόμενο μέσα από μια φορητή συσκευή (PDA) είναι το πρότζεκτ AR phone [6].



Με τη χρήση του ανοιχτού λογισμικού ARtoolkit[17] και με την βιβλιοθήκη OpenGL GLUT[18], ο M. Assad και η ομάδα του, κατάφεραν να δημιουργήσουν ένα προτότυπο σύστημα χρησιμοποιώντας βασικό και αρκετά συνηθισμένο εξοπλισμό. Αυτή η εφαρμογή, μαζί με άλλες αντίστοιχες που χρησιμοποιούν το ARtoolkit, απέδειξε πόσο αυτή η τεχνολογία έχει ωριμάσει και ότι πλέον μπορεί να χρησιμοποιηθεί για την κατασκευή εφαρμογών μεγαλύτερης κλίμακας.

Μια επίσης πρόσφατη και επιτυχής προσπάθεια μεταφοράς του ψηφιακού κόσμου στον πραγματικό, έγινε από τα MIT media labs με το πρότζεκτ SixthSense του κ.Pranav Mistry.



Αυτή η εφεύρεση επαυξάνει το περιβάλλον προβάλλοντας ψηφιακό περιεχόμενο πάνω σε διάφορες επιφάνειες και χρησιμοποιώντας αυτές για αλληλεπίδραση με το σύστημα. Για έλεγχο και εισαγωγή εντολών στο σύστημα, ο κ. Mistry συνδέει απλές χειρονομίες που ανιχνεύονται από μια φορητή κάμερα [7].

Μια άλλη, πολύ ενδιαφέρουσα υλοποίηση έχει γίνει από το πανεπιστήμιο South Australia με το σύστημα Tinmith [8] που χρησιμοποιείται ως πλατφόρμα συνεργασίας μεταξύ ομάδων που βρίσκονται σε εξωτερικό χώρο και ομάδων που βρίσκονται εγκατεστημένες σε εσωτερικό χώρο.



Οι τελευταίες κατέχουν στοιχεία αναφορικά με το περιβάλλον, συνολική εικόνα του πεδίου και άλλα τεχνολογικά βοηθήματα και μπορούν να μεταφέρουν όλη αυτή τη λειτουργικότητα στο εξωτερικό πεδίο έτσι ώστε να οργανώσουν και να καθοδηγήσουν τις απομακρυσμένες ομάδες προς συγκεκριμένα σημεία ενδιαφέροντος ή σημεία στόχους.

Οι τεχνολογίες επαυξημένης πραγματικότητας έχουν χρησιμοποιηθεί ξανά σε προτότυπα αφήγησης ιστοριών. Μέσα σε αυτά είναι τα Geist, Computer Supported Collaborative Interactive Storytelling, του κ N.Braun [10] αλλά και η πολύ επιτυχημένη και δημοφιλής προσέγγιση του HIT lab New Zealand με τίτλο

“Mixed Reality Book” των R. Grasset, A.Dünser και M. Billinghurst [11].

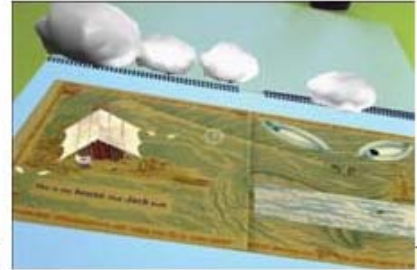
Στην πρώτη περίπτωση το Geist είναι μια εκπαιδευτική πλατφόρμα με ιστορικά δεδομένα. Οι χρήστες περιπλανιούνται μέσα σε μια προδιαγεγραμμένη τοποθεσία κρατώντας μια συσκευή όπου ενεργοποιείται σε συγκεκριμένα σημεία της τοποθεσίας. Μέσα από αυτή εμφανίζονται τρισδιάστατα μοντέλα ως φαντάσματα και φωνητικά αλληλεπιδρούν με τους χρήστες.



Στη δεύτερη περίπτωση συναντάμε το σύστημα του εργαστηρίου HIT lab New Zealand, όπου ο ο χρήστης αυτή τη φορά είναι στατικός ακριβώς μπροστά από ένα βιβλίο, όπου οι σελίδες του ζωντανεύουν και επαυξάνονται με τρισδιάστατο περιεχόμενο.



a)



b)



c)



d)



e)

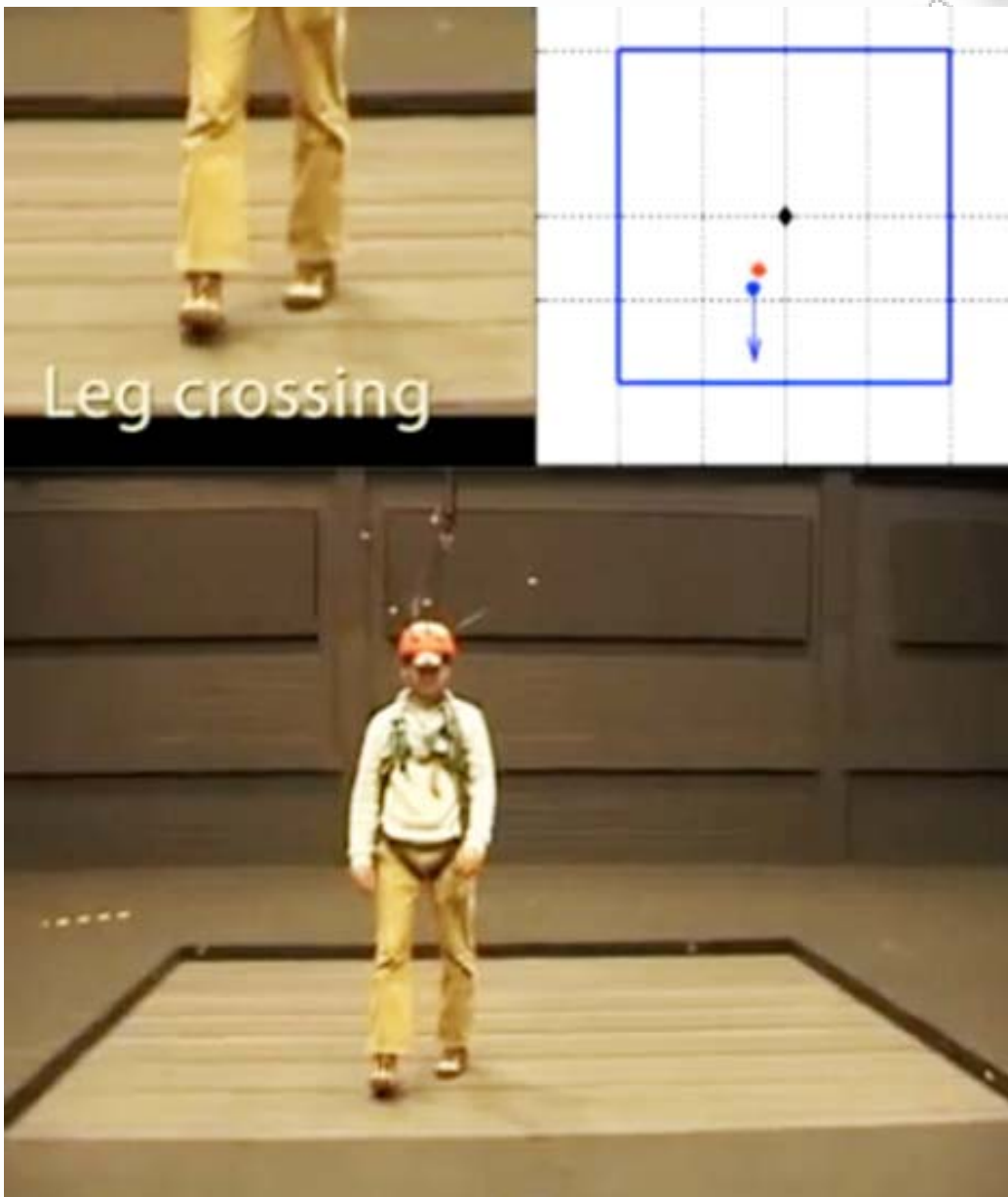


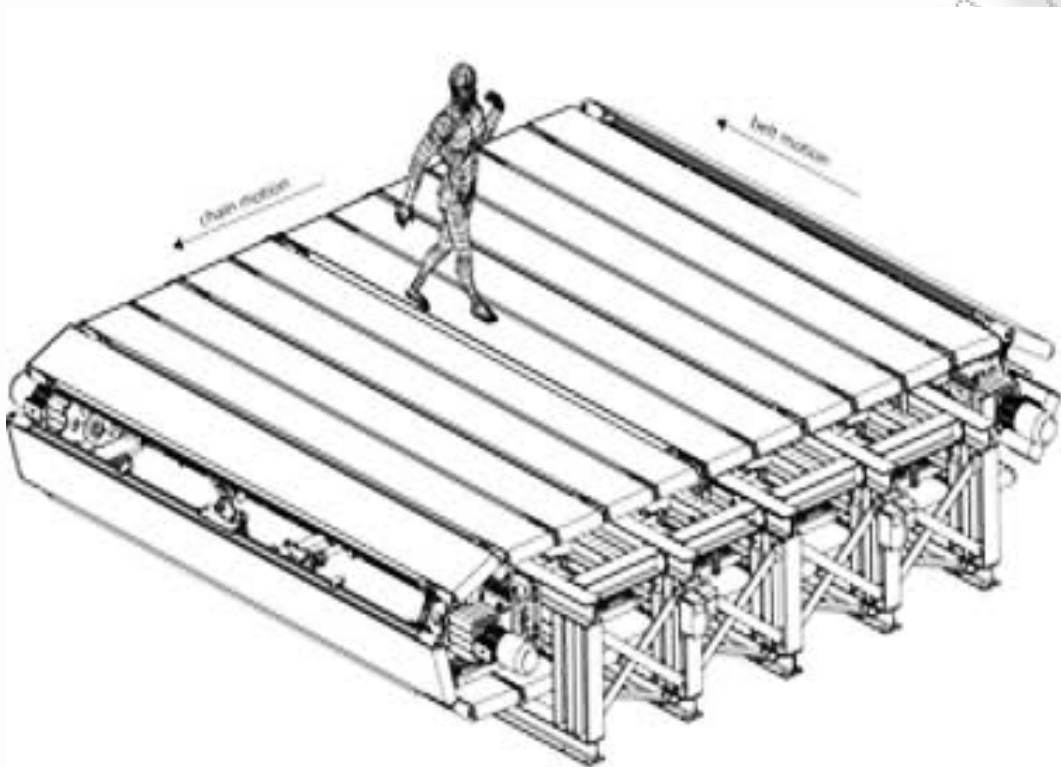
f)

Η δική μας περίπτωση βέβαια δεν είναι στατική όσον αφορά την τοποθεσία, αλλά δυναμική. Αυτό σημαίνει ότι ο χρήστης μετακινείται μέσα στο περιβάλλον πράγμα που αποτελεί ένα σοβαρό, ανοιχτό ζήτημα σε σχέση με τους εικονικούς κόσμους. Άλλες μέθοδοι πλοήγησης, πέρα από το πάτημα μερικών κουμπιών σε ένα χειριστήριο, έχουν κάνει την είσοδό τους πρόσφατα στα εικονικά περιβάλλοντα. Ένας αξιοσημείωτος και πολλά υποσχόμενος τρόπος είναι η πλοήγηση μέσω εγκεφαλογραφήματος (EEG-based navigation) [1, 2]



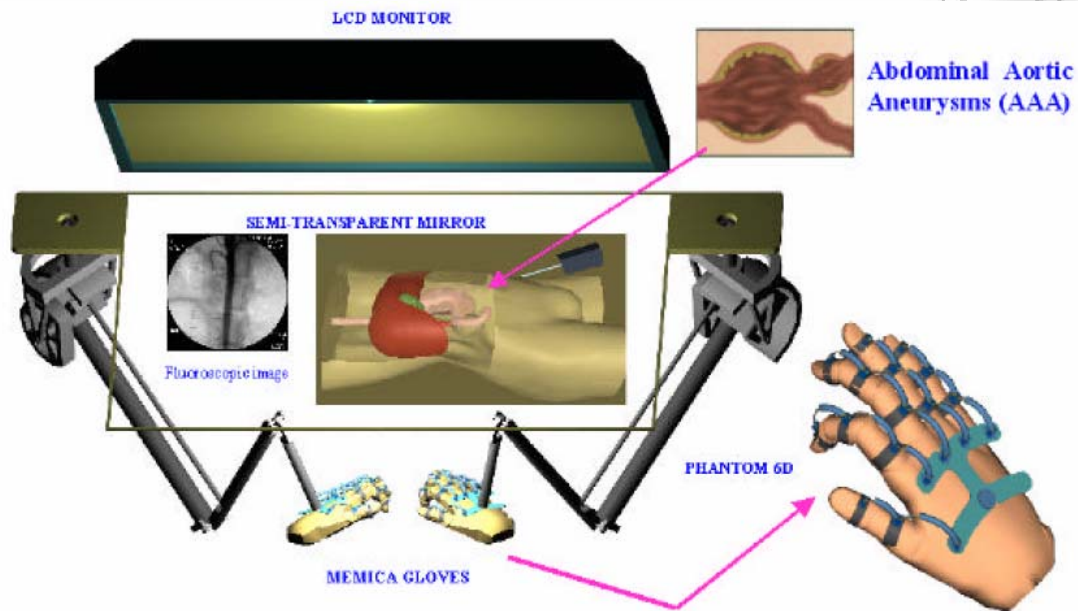
όπου ο χρήστης σκέφτεται με σκοπό να μετακινηθεί προς κάποιο σημείο του εικονικού κόσμου. Παρά τον πραγματικά καινοτόμο παράγοντα αυτής προσέγγισης, ακόμα ο χρήστης δεν μπορεί να χρησιμοποιήσει το περπάτημα, που αποτελεί και τον πιο φυσικό τρόπο πλοήγησης για τον άνθρωπο. Ένα υπαρκτό πρότζεκτ που ο χρήστης έχει τη δυνατότητα να μετακινείται με το περπάτημα και προς οποιαδήποτε κατεύθυνση, είναι το "Cyberwalk"[3].



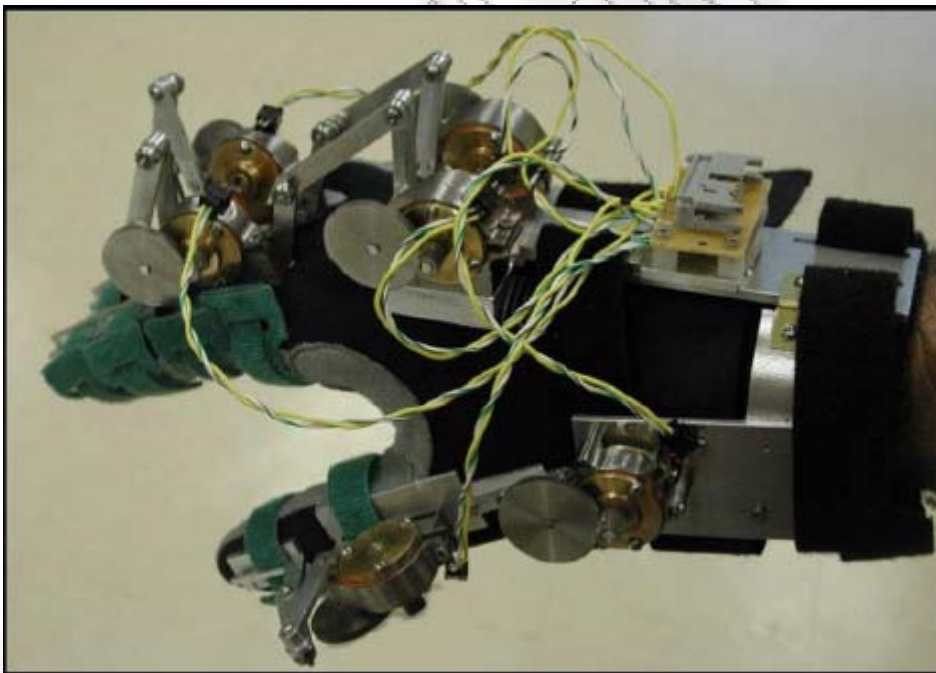


Σε αυτή την προσέγγιση γίνεται χρήση μιας αρκετά μεγάλης κατασκευής που στην ουσία είναι ένας διάδρομος περπατήματος δυο διαστάσεων, σε συνεργασία με ένα σύνολο από κάμερες και αισθητήρες που ανιχνεύουν την κίνηση και την κατεύθυνση του χρήστη.

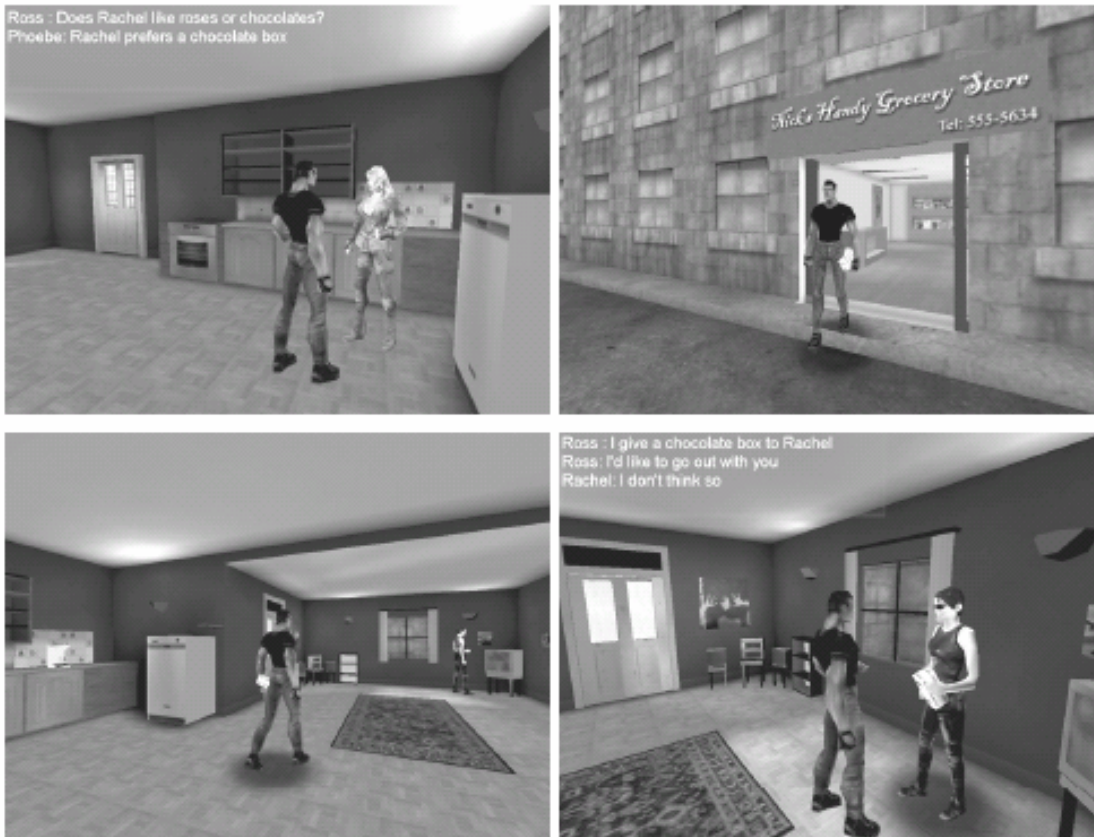
Σχετικά με τις διεπαφές αφής [13] και τις συσκευές αφής, υπάρχουν αρκετές υλοποιήσεις που μεταφέρουν την κίνηση των χεριών στον ψηφιακό κόσμο, όμως γίνονται αρκετά λιγότερες αν μιλήσουμε για αίσθηση βάρους ή ανάδραση υφής. Οι περισσότερες από αυτές, με εξελιγμένα συστήματα ανάδρασης, είναι σαν το σύστημα MEMICA[12] -που συναντάται σε χειρουργικές εφαρμογές - ή σαν το "MR glove" των J.Blake and H.B.Gurocak [4] που δημιουργήθηκε αποκλειστικά για εφαρμογές εικονικής πραγματικότητας. Στην περίπτωση του MEMICA η ανάγκη από είναι επιστημονική επιτήρηση κατά τη χρήση του είναι άκρως απαραίτητη.



Απεναντίας, το "MR glove" είναι πιο μικρό σε μέγεθος και έχει αποδειχθεί ότι μπορεί να μεταφέρει την αίσθηση της επαφής με κάποιο αντικείμενο βάζοντας σε λειτουργία το σύστημα φρένων που διαθέτει.



Συνεχίζοντας με την κατάσταση της έρευνας στα εικονικά, τα μεικτά και τα επαυξημένα περιβάλλοντα, ένας άλλος τομέας είναι η χρήση εικονικών πρακτόρων για αλληλεπίδραση με τους χρήστες. Σε αυτό το πεδίο, προηγούμενες προσεγγίσεις έχουν υλοποιηθεί από τον κ. Marc Cavazza και τους συνεργάτες του και εμφανίζονται στα άρθρα του πάνω στον προγραμματισμό συμπεριφορών πρακτόρων και πάνω στην αφηγηματική βασισμένη στους χαρακτήρες [14] μέσα σε εικονικούς κόσμους.



Δεδομένου ότι μια προ-αποφασισμένη ιστορία δεν είναι παρούσα, οι χαρακτήρες δημιουργούν την πλοκή δυναμικά σύμφωνα με τις ιεραχικά δομημένες ανάγκες τους. Μέσα από την διαλογική αλληλεπίδραση με τον χρήστη, μπορούν να αναδυθούν πληροφορίες σχετικά με τις ανάγκες των πρακτόρων και αυτές να καταλήξουν να αλλάξουν τις προτεραιότητές τους. Με τον τρόπο αυτό ο χρήστης έρχεται αντιμέτωπος με ένα δυναμικό σενάριο όπου η αλλαγή στην πλοκή είναι αναμενόμενη και δρα ως διεγερτικός παράγοντας για την εμπειρία του χρήστη. Τέτοια υλοποίηση κυκλοφόρησε για ένα περιβάλλον μεικτής πραγματικότητας στο 4^ο διεθνές εργαστήριο IVA 2003 όπου ο χρήστης καταγράφεται από κάμερα και μεταφέρεται μέσα στην εφαρμογή. Οι χειρονομίες του και η ομιλία του αναλύονται από την αφηγηματική μηχανή της εφαρμογής έτσι ώστε να εξελιχθεί το σενάριο [16].

Συνοψίζοντας τα παραπάνω, έχουν συγκεντρωθεί οι παράγοντες που οδήγησαν σε αυτή την υλοποίηση. Από τη μία μεριά έχουμε την εμφάνιση των φορητών συσκευών που έχουν αρκετή υπολογιστική ισχύ για να εμφανίσουν και να μεταχειριστούν τρισδιάστατα γραφικά και να κάνουν ανάλυση εικόνας παράλληλα.

Από την άλλη, υπάρχουν τα υπάρχοντα προβλήματα σε σχέση με την μεταφορά της ανθρώπινης κίνησης και αφής μέσα στα εικονικά περιβάλλοντα.

Εν τέλει, η πρόοδος στην εικονική αφήγηση και στους διαδραστικούς ευφυείς εικονικούς πράκτορες δεν έχει μεταφερθεί στα συστήματα διάχυτης νοημοσύνης που τελικά προσφέρουν εμπύθιση στους χρήστες. Σε αυτήτην κατεύθυνση προτείνεται η κατασκευή ενός συστήματος επαυξημένου περιβάλλοντος όπου ο χρήστης εξοπλισμένος με ένα φορητό σύστημα θα μπορεί να περιπλανιέται στον χώρο, έχοντας τη δυνατότητα να αλληλεπιδράσει στις προκαθορισμένες από εμάς τοποθεσίες με μοναδικά και αναγνωρίσιμα σημάδια. Τα σημάδια αυτά χρησιμοποιούνται ως δείκτες για τη θέση των εικονικών πρακτόρων, είτε ως σημάδια σε αντικείμενα που θέλουμε να αλλάξουμε την όψη τους, έτσι ώστε να αντιπροσωπεύουν κάτι διαφορετικό για τις ανάγκες του σεναρίου μας.

3: Σχεδιασμός Εφαρμογής

Στην προσπάθεια να αντιμετωπίσουμε τα προβλήματα που αναφέρθηκαν νωρίτερα σχετικά με την ελεύθερη κίνηση και την αφή, πρέπει να λάβουμε υπόψιν ότι πρέπει να διατηρηθούν οι φυσικές ιδιότητες του κόσμου και των αντικειμένων του. Για να επιτευχθεί αυτό επιλέχθηκε η δημιουργία ενός εικονικού περιβάλλοντος πάνω από πραγματικό. Το αποτέλεσμα αυτής της προσέγγισης είναι ότι ο χρήστης έρχεται σε επαφή με ένα φυσικό περιβάλλον που είναι οπτικά αλλαγμένο και φυσικά έχει τη δυνατότητα να αναπαράγει γεγονότα μέσα από τον έλεγχο της παρούσας εφαρμογής.

Μέθοδοι και Προσέγγιση

Τεχνολογίες

Για την κατασκευή του εν λόγω περιβάλλοντος, χρησιμοποιούνται επαυξησεις σε προκαθορισμένα σημεία όπου έχουν εγκατασταθεί εκτυπώσεις με συγκεκριμένα μοτίβα. Όσο ο χρήστης περιπλανιέται στον προετοιμασμένο από εμάς χώρο, μια ψηφιακή κάμερα εγκατεστημένη στο κεφάλι του, καταγράφει μια ροή από εικόνες όπου με τη σειρά τους αναλύονται με αλγορίθμους υπολογιστικής όρασης. Όταν ένα από τα προκαθορισμένα μοτίβα σαρωθεί από την κάμερα, η βιβλιοθήκη ανίχνευσης είναι υπεύθυνη να αναγνωρίσει το συγκεκριμένο μοτίβο και να υπολογίσει την γωνία θέασής του, έτσι ώστε να μετακινησει (translate), να περιστρέψει (rotate) και να δώσει την σωστή κλίμακα (scale) στο τρισδιάστατο αντικείμενο ή στην υφή που του έχουμε εμείς ορίσει αναλόγως.

Με τη εκτενή χρήση αυτών των συγκεκριμένων μοτίβων (markers) στο σύνολο του χώρου, καταφέρνουμε να δημιουργήσουμε μια πλήρη σκηνή με τρισδιάστατα αντικείμενα. Το παραγόμενο περιβάλλον αποτελεί την δική μας αλλαγμένη πραγματικότητα. Επιπροσθέτως, οι markers δρούν και ως σημεία αναφοράς και ανιχνεύουν την παρουσία του χρήστη, πράγμα που χρησιμεύει ώστε να πυροδοτούνται γεγονότα της ιστορίας ή/και μελλοντικά, τρισδιάστατος ήχος.

Εξοπλισμός

Ο χρήστης πρέπει να φορά τον απαραίτητο εξοπλισμό καθ' όλη τη διάρκεια της εφαρμογής. Οι συσκευές που αποτελούν το σύστημα είναι : Γυαλιά video για την εικόνα του περιβάλλοντος αλλά και των επαυξησεων πάνω σε αυτό. Στην περίπτωση μας τα γυαλιά είναι της εταιρείας Vuzix, μοντέλο wrap 310xl με ανάλυση 640x480. Επόμενο μέρος του εξοπλισμού είναι μια ψηφιακή κάμερα usb για την ανίχνευση των προκαθορισμένων σημείων στο χώρο. Η κάμερα που χρησιμοποιήθηκε είναι της εταιρείας Microsoft, μοντέλο HD3000 ανάλυσης 2048x1080 με ενσωματωμένο μικρόφωνο για την περίπτωση ενσωμάτωσης φωνητικών ενολών ή διαλόγων στο μέλλον. Τέλος, ένα laptop αναλαμβάνει να δημιουργήσει όλα τα

τρισεπίστυχα γραφικά και τις υφές ενώ παράλληλα συντονίζει το tracking των προαποφασισμένων μοτίβων (markers), την μεταφορά της εικόνας στα γυαλιά video και γενικά εκτελεί όλη την εφαρμογή.

4: Υλοποίηση Πρωτοτύπου

Τεχνολογίες Augmented Reality

Πολλές και διαφορετικές τεχνικές είναι αυτές που επιτρέπουν την ένωση εικονικών αντικειμένων με την πραγματικότητα. Οι πιο διαδεδομένες τεχνικές είναι:

- χρήση των φυσικών χαρακτηριστικών του ανιχνεύοντος περιβάλλοντος
- χρήση δέκτη GPS
- μικροηλεκτρονικές συσκευές όπως επιταχυνσιόμετρα, ψηφιακές πυξίδες και ανιχνευτές κατεύθυνσης
- ανίχνευση προσυμφωνημένων μοτίβων

Κάθε περίπτωση χρησιμοποιείται και για διαφορετική περίπτωση και είναι σαφές ότι και οι τέσσερις τεχνικές έχουν πλεονεκτήματα και μειονεκτήματα. Είναι χαρακτηριστικό για παράδειγμα ότι για να εντοπιστούν φυσικά χαρακτηριστικά του περιβάλλοντος χρειάζεται ευρεία εκπαίδευση του συστήματος για να ανιχνεύει το που τοποθετείται κάθε φορά το επίπεδο. Το θετικό βέβαια είναι ότι το πεδίο δεν πρέπει να έχει τύχει οποιασδήποτε προετοιμασίας. Στην δεύτερη περίπτωση η χρήση του δέκτη gps μπορεί να δώσει αρκετά καλές ενδείξεις θέσης, ταχύτητας και κατεύθυνσης του χρήστη, όμως δεν έχει τη δυνατότητα να λειτουργήσει σε εσωτερικό χώρο. Επίσης, ακόμα και αν η εφαρμογή εκτελείται σε εξωτερικό χώρο, δεν υπάρχει η δυνατότητα να ανιχνεύσει το σύστημα τη γωνία θέασης του οριζοντα οπότε δεν γίνεται να εφαρμοστούν οι επαυξησεις πάνω σε συγκεκριμένα σημεία του φυσικού περιβάλλοντος.

Η τεχνική που κάνει χρήση των μικροηλεκτρονικών συσκευών θεωρείται αρκετά ακριβής και έχει και δυνατότητα να ανιχνεύσει ακριβώς το που κοιτάζει ο χρήστης. Με τον τρόπο αυτό τα εικονικά αντικείμενα μπορούν να τοποθετηθούν ακριβώς όπου επιλέξει ο διαχειριστής του συστήματος. Το μειονέκτημα της τεχνικής είναι ότι τα ψηφιακά επιταχυνσιόμετρα και οι ανιχνευτές στρέψης χρειάζονται ρύθμιση αρχικής θέσης αλλά και επαναρύθμιση σε συγκεκριμένα σημεία στον χώρο και κατά τη διάρκεια της εφαρμογής. Σε αντίθετη περίπτωση, η διαφορά στους υπολογισμούς μεγαλώνει αυξητικά και το αποτέλεσμα είναι οι ψηφιακές επαυξησεις να εμφανίζονται σε λάθος σημείο.

Η τελευταία τεχνική χρησιμοποιεί συγκεκριμένα μοτίβα για να υπολογίσει τη θέση και την περιστροφή κάθε ψηφιακού αντικειμένου. Με τη συγκεκριμένη αντιμετώπιση όλα τα αντικείμενα τοποθετούνται σωστά στον χώρο και ο χρόνος υπολογισμού και εμφάνισης είναι αρκετά μικρός. Το μειονέκτημα είναι ότι πρέπει το περιβάλλον να είναι γνωστό, να έχει προετοιμαστεί και να έχουν εγκατασταθεί τα προεπιλεγμένα μοτίβα στις σωστές τοποθεσίες.

Στην δική μας περίπτωση, επειδή σκοπός της διπλωματικής εργασίας δεν είναι η κατασκευή μιας βιβλιοθήκης επαυξημένης πραγματικότητας αλλά η κατασκευή ενός περιβάλλοντος για επαυξημένη εξιστόρηση, επιλέχθηκε η τελευταία τεχνική με την χρήση των προεπιλεγμένων μοτίβων ή αλλιώς marker based augmented reality. Ένα άλλο πλεονέκτημα αυτής της τεχνικής είναι ότι υπάρχει διαθέσιμη η βιβλιοθήκη ARtoolkit που είναι ευρέως διαδεδομένη και υπάρχει αρκετά καλή τεκμηρίωση σε σχέση με τη χρήση και την παραμετροποίησή της.

Επιλογή Υλικού

Για τη λειτουργία της βιβλιοθήκης ARtoolkit αλλά και γενικότερα της εφαρμογής όπως αυτή σχεδιάστηκε, είναι απαραίτητα κάποια βασικά μέρη εξοπλισμού. Επιγραμματικά αυτά είναι, ένας φορητός υπολογιστής, μια κάμερα, ένα ζευγάρι γυαλιά βίντεο, ένας μετατροπέας βύσματος miniDisplay port σε VGA D-Sub και ένας μετατροπέας σήματος από ψηφιακό video σε αναλογικό. Για την επιλογή των συγκεκριμένων υλικών προηγήθηκαν αρκετοί έλεγχοι με διαφορετικά υλικά. Τα συμπεράσματα σχετικά με όλες τις δοκιμές ακολουθούν παρακάτω.

Η κάμερα

Η κάμερα χρησιμοποιείται στο σύστημα για να λαμβάνονται οι εικόνες του περιβάλλοντος και συγκεκριμένα για να ανιχνεύονται τα εκτυπωμένα προεπιλεγμένα μοτίβα (markers). Αρχικά χρησιμοποιήθηκε μια κοινή ψηφιακή κάμερα χαμηλού κόστους, βασικής ανάλυσης 640x480, με χειροκίνητη εστίαση, της εταιρείας Mustek μοντέλο m6512. Το πρόβλημα που έχει το συγκεκριμένο μοντέλο –όπως και τα άλλα μοντέλα της κατηγορίας- είναι ότι η ταχύτητα λήψης στην ανάλυση 640x480 είναι 15 frames το δευτερόλεπτο πράγμα που αποτελεί πρόβλημα όταν η εικόνα έρχεται αρκετά κοντά στα μάτια του χρήστη. Το αποτέλεσμα είναι η θολή εικόνα και ασυνεχής ροή video. Το πρόβλημα λύνεται εάν ο χρήστης κάνει αρκετά αργές και σταθερές κινήσεις. Το δεύτερο πρόβλημα με αυτή την κάμερα είναι η χειροκίνητη εστίαση. Είναι σαφές ότι ο χρήστης δεν μπορεί να ρυθμίζει την εστίαση κατά τη διάρκεια της εφαρμογής πράγμα που αποδεικνύεται απαραίτητο γιατί εάν δεν γίνει σωστή εστίαση στο σημείο που κοιτά ο χρήστης, είναι αρκετά πιθανό να μην αναγνωριστεί κάποιος marker, ή ακόμα χειρότερα να αναγνωριστεί κάποιος marker λανθασμένα και να εμφανιστεί κάποια υφή ή κάποιο αντικείμενο που είναι καταχωρημένο σε άλλον marker.

Με την χρήση μιας κάμερας υψηλής ανάλυσης της Microsoft, μοντέλο HD 3000 τα παραπάνω προβλήματα λύνονται σε μεγάλο βαθμό. Η κάμερα μπορεί να καταγράψει ροή video σε ανάλυση 640x480 στα 30 frames το δευτερόλεπτο και έχει τη δυνατότητα να καταγράψει ακόμα και σε ανάλυση 720p στα 20 frames το δευτερόλεπτο. Το βίντεο που λαμβάνει ο χρήστης είναι αρκετά σταθερό, χωρίς διακοπές και με αρκετά πιο καθαρή εικόνα. Επιπροσθέτως, η συγκεκριμένη κάμερα διαθέτει μηχανισμό αυτόματης εστίασης πράγμα που διευκολύνει στην ανίχνευση των markers που μπορούν να είναι τοποθετημένοι σε οποιοδήποτε βάθος στον χώρο. Τέλος ένα ακόμη χαρακτηριστικό της συσκευής είναι το ενσωματωμένο μικρόφωνο που μπορεί να χρησιμοποιηθεί στο μέλλον είτε για να περνάμε στο σύστημα την ομιλία του χρήστη, είτε άλλα ηχητικά σήματα.

Γυαλιά Video

Για την μεταφορά της εικόνας μπροστά στα μάτια του χρήστη, χρησιμοποιήθηκαν τα γυαλιά video της nuzix, μοντέλο wrap 310. Τα γυαλιά αυτά μπορούν μόνο να δείξουν αναλογικά σήματα video και δεν πρέπει να συγχέονται με άλλες ανάλογες συσκευές που χρησιμοποιούνται σε εφαρμογές εικονικής πραγματικότητας. Πρέπει να αναφερθεί ότι δεν έχουν ανιχνευτές κινήσεων του κεφαλιού του χρήστη και σίγουρα δεν είναι ημι διάφανα. Όταν ο χρήστης τα φορά δεν μπορεί να δει παρα μόνο τις δυο μικρο-οθόνες τις συσκευής. Τα συγκεκριμένα επιλέχθηκαν με μοναδικό κριτήριο το κόστος. Είναι η φθηνότερη υλοποίηση γυαλιών video από την εν λόγω εταιρεία. Υπάρχουν στην αγορά μοντέλα που υποστηρίζουν μεγαλύτερες αναλύσεις -ακόμα και HD 1080p - και η αναπαραγωγή γίνεται απευθείας από ψηφιακό σήμα. Το κόστος για τέτοια μοντέλα μπορεί να ξεπεράσει τα 600 ευρώ, πράγμα που τα θέτει εκτός τεχνικών όρων για τη συγκεκριμένη υλοποίηση. Η ανάλυση που υποστηρίζεται από τα wrap 310 είναι 640x480 και η ανανέωση οθόνης είναι στα 60hz μέγιστο. Για την συγκεκριμένη εφαρμογή αυτό που είναι απαραίτητο είναι η ανανέωση της οθόνης να μην είναι μικρότερη των 30hz και αυτό διότι τα τρισδιάστατα μοντέλα που

έχουν animation θα απεικονίζονται με διακοπές και ενδεχομένως να χάνεται η πληροφορία που πρέπει να δει ο χρήστης. Ένα από τα βασικά μειονεκτήματα του μοντέλου που δεν μπορεί να αντιμετωπισθεί είναι η απουσία ρύθμισης της απόστασης μεταξύ των δυο μικρο-οθονών στον οριζόντιο άξονα. Αυτή η ρύθμιση είναι αρκετά σημαντική για μοντέλα με παλιότερα lcd panels γιατί η γωνία θέασης είναι αρκετά περιορισμένη. Εάν κάποιος χρήστης έχει μεγαλύτερη απόσταση από μάτι σε μάτι, σε σχέση με την προκαθορισμένη από την εταιρεία, θα βρεθεί αντιμέτωπος με σκιάσεις στα άκρα της εικόνας ή με σκιάσεις στο μέσο αυτής.

Φορητός Υπολογιστής

Για την εκτέλεση της εφαρμογής ήταν απαραίτητη η χρήση ενός φορητού υπολογιστή. Στη συγκεκριμένη περίπτωση χρησιμοποιήθηκε ένα laptop της εταιρείας dell μοντέλο L502x με επιπρόσθετη κάρτα γραφικών nvidia gt540 2gb, με σκοπό την υποστήριξη όλων των εκδόσεων openGL αλλά και υποστήριξη των εντολών επέκτασης openGL GLSL 4.0, που στη συνέχεια θα επεξηγηθεί ο λόγος της χρησιμότητάς τους. Μετά από έρευνα στα διάφορα format αρχείων 3D, που θα αναλυθεί στη συνέχεια, επιλέχθηκε ο τύπος FBX. Για την σωστή αναπαραγωγή των υφών των αρχείων FBX -με τη χρήση του παρεχόμενου SDK από την εταιρεία Autodesk- είναι απαραίτητο η κάρτα γραφικών να υποστηρίζει τις εντολές επέκτασης open GLShading Language (GLSL). Σε αντίθετη περίπτωση, τα μοντέλα είτε είναι ολοκληρωτικά μαύρα λόγω λανθασμένης μετάφρασης της φωτεινής πηγής, είτε δεν έχουν καθόλου υφές και εμφανίζονται ολοκληρωτικά λευκά. Στις ενσωματωμένες κάρτες γραφικών με διαμοιραζόμενη μνήμη, οι οδηγοί που τις συνοδεύουν δεν δίνουν υποστήριξη για το openGL GLSL 4.0 καθώς θεωρείται τεχνολογία απεικόνισης υψηλής ποιότητας και αποτελεί λειτουργία που βρίσκεται εκτός απαιτήσεων, ειδικά για φορητούς υπολογιστές εργασίας που δεν προορίζονται για αναπαραγωγή παιχνιδιών ή για επεξεργασία 3D γραφικών.

Στα απολύτως απαραίτητα χαρακτηριστικά είναι μια ψηφιακή έξοδος ή μια αναλογική έξοδος της εικόνας της οθόνης. Είναι σύνηθες να υπάρχει κάποιου είδους διασύνδεση με εξωτερική συσκευή απεικόνισης στα περισσότερα laptops. Το πρόβλημα που αντιμετωπίστηκε με το εν λόγω laptop είναι ότι δεν διέθετε αναλογική έξοδο video –ή αλλιώς TV-out-. Για την επίλυση του προβλήματος χρησιμοποιήθηκε η ψηφιακή έξοδος miniDisplay port σε συνδυασμό με έναν μετατροπέα από miniDisplay σε VGA. Στη συνέχεια, μια συσκευή μετατροπής του ψηφιακού σήματος σε αναλογικό, με έξοδο σε κοινό βύσμα RCA – composite video μεταφέρει το αναλογικό σήμα της εικόνας στα γυαλιά video.

Τέλος, η λειτουργία της εφαρμογής απαιτεί υπολογιστική δύναμη, μεγαλύτερη του μέσου όρου των φορητών υπολογιστών της αγοράς της τρέχουσας περιόδου και αυτό διότι γίνονται δυο αρκετά επίπονες διαδικασίες ταυτόχρονα. Η πρώτη είναι επεξεργασία και ανάλυση εικόνας με ταχύτητα 30 εικόνες το δευτερόλεπτο και η δεύτερη είναι η απεικόνιση τρισδιάστατων γραφικών σε πραγματικό χρόνο. Επίσης, είναι απαραίτητο να αναφερθεί ότι η βιβλιοθήκη ARtoolkit δεν είναι υλοποιημένη για εκτέλεση σε συστήματα με πολλαπλούς πυρήνες και αυτό έχει ως αποτέλεσμα να μην χρησιμοποιείται όλη η ισχύς φορητών υπολογιστών τελευταίας γενιάς που αποτελούνται από 4 πυρήνες και πάνω αλλά και να φαίνεται στο σύστημα ότι μόνο ένας επεξεργαστής λειτουργεί και αυτός στο μέγιστο δυνατό.

Βασικές Βιβλιοθήκες

ARtoolkit

Είναι η βιβλιοθήκη που επιλέχθηκε για την αναγνώριση των markers και την σωστή τοποθέτηση των γραφικών στο γράφημα σκηνής. Το ARtoolkit επιλέχθηκε γιατί είναι δωρεάν και έχει μέχρι στιγμής αρκετά μεγάλο κοινό που ασχολείται μαζί του. Βασικό επίσης προτέρημα είναι η ύπαρξη documentation -στα αγγλικά- αλλά και οδηγιών που επεξηγούν τις βασικές λειτουργίες του. Η βιβλιοθήκη χρησιμοποιεί τον πιο βασικό και διαδεδομένο οδηγό για την ενεργοποίηση της κάμερας με το όνομα DirectShow video library και εμφάνιση των γραφικών γίνεται μέσω της βιβλιοθήκης GLUT που είναι ουσιαστικά μια έκδοση γραφήματος σκηνής για μετάφραση εντολών OpenGL. Το OpenGL είναι μια βιβλιοθήκη γραφικών ανοιχτού κώδικα και έχει το πλεονέκτημα να χρησιμοποιείται αρκετά σε διάφορα περιβάλλοντα για επεξεργασία 3D μοντέλων και για κατασκευή 3D κόσμων. Αυτός είναι και ο βασικός λόγος που επιλέχθηκε το ARtoolkit. Η γλώσσα που είναι γραμμένη η βιβλιοθήκη είναι η C++ και εμείς θα χρησιμοποιήσουμε αυτή την υλοποίηση παρόλο που το ίδιο λογισμικό έχει μεταφραστεί και σε άλλες γλώσσες όπως java και flash.

OpenSceneGraph

Το OpenSceneGraph είναι το λογισμικό που χρησιμοποιήθηκε για γράφημα σκηνής. Είναι ένα από τα πιο διαδεδομένα λογισμικά για απεικόνιση σκηνών τρισδιάστατων γραφικών και διαθέτει μια αρκετά ενεργή κοινότητα που συμβάλει στην εξέλιξή του. Και αυτό είναι βασισμένο στις εντολές OpenGL μόνο που με τις κλάσεις του ο προγραμματισμός γραφικών γίνεται σε αρκετά ανώτερο επίπεδο από το να γράφει κανείς εντολές OpenGL. Άλλο πλεονέκτημα που έχει το συγκεκριμένο software είναι η υποστήριξη plugins και extensions. Με τη χρήση των προαναφερθέντων μπορεί κανείς να χρησιμοποιήσει αρκετά βοηθήματα όπως υποστήριξη πολλών format σε 3D αρχεία (π.χ. ASE, 3DS, MAX, MD5, FBX κ.α.), υποστήριξη βιβλιοθηκών για animation, υποστήριξη για κανόνες φυσικής και υποστήριξη τρισδιάστατου ήχου. Επιπροσθέτως, είναι και αυτό γραμμένο σε C++, πράγμα που διευκολύνει αρκετά την επικοινωνία με το ARtoolkit. Το βασικότερο πλεονέκτημα όμως σε σχέση με την επιλογή του είναι η ύπαρξη της βιβλιοθήκης OSGart που δένει τελικά το ARtoolkit με το OpenSceneGraph και περιγράφεται αμέσως μετά.

OSGart

Η βιβλιοθήκη OSGart είναι ο συνδετικός κρίκος μεταξύ του ARtoolkit και του OpenSceneGraph. Το λογισμικό έχει αναπτυχθεί από το HIT New Zealand και δίδεται δωρεάν μέσα από την διαδικτυακή τους διεύθυνση μαζί με κάποιες κατευθύνσεις και παραδείγματα για τη χρήση του. Πρέπει να σημειωθεί ότι χωρίς το συγκεκριμένο λογισμικό δεν θα μπορούσε να υλοποιηθεί το συγκεκριμένο project. Το OSGart πετυχαίνει να μεταφέρει τις συντεταγμένες όπως τα διανύσματα θέσης, περιστροφής και κλίμακας από το ARtoolkit, στη μορφή που μπορεί το OpenSceneGraph να καταλάβει καθώς κάνει την ανανεώση του γραφήματος σκηνής ανά frame. Με τον τρόπο αυτό μπορούμε να εκμεταλευτούμε την αρκετά καλή λειτουργία του ARtoolkit και την επεκτασιμότητα και τις δυνατότητες του OpenSceneGraph. Έχουν γίνει αρκετές προσπάθειες να συνδυαστούν βιβλιοθήκες επαυξημένης πραγματικότητας με γραφήματα σκηνής, όμως το αποτέλεσμα είναι αρκετά πιο ελλειπές από τη συγκεκριμένη υλοποίηση.

Άλλα συστήματα Marker Based AR

Στην υλοποίηση του NgarToolkit δεν υπάρχει κάποια σύνδεση με κάποιο εξελιγμένο γράφημα σκηνής και όλο το documentation είναι στην Ιαπωνική γλώσσα. Το Jmonkey3D έχει μια υλοποίηση επαυξημένης πραγματικότητας όμως αντιμετωπίζει αρκετά προβλήματα καθώς την έχουν υλοποιήσει φοιτητές ωε πτυχιακή τους εργασία και το λογισμικό δεν έχει ανανεωθεί για αρκετό καιρό, συν ότι πολλά bugs παραμένουν αδιόρθωτα. Το ogre3D έχει μια υλοποίηση για επαυξημένη πραγματικότητα που όμως και αυτή τυγχάνει ελλειπούς τεκμηρίωσης με αποτέλεσμα να μην είναι δυνατόν να τη χρησιμοποιήσει κάποιος

που δεν έχει επαφή με το περιβάλλον του Ogre3D. Τη στιγμή που έγινε η παρούσα έρευνα υπήρχε η περίπτωση του FLARtoolkit η οποία φάνηκε να εξελίσσεται αργότερα και να αποκτά και commercial άδεια, πράγμα που σημαίνει ότι το λογισμικό θα έγινε αρκετά λειτουργικό. Το μειονέκτημα σε αυτή την προσέγγιση είναι ότι οι εφαρμογές που θα φτιαχτούν θα είναι όλες σε περιβάλλον flash πράγμα που δεν ενδείκνυται για την περίπτωση του project της επαυξημένης εξιστόρησης. Εάν η υλοποίηση μεγαλώσει αρκετά και τελικά γίνει μια μεσαία με μεγάλη εφαρμογή η πλατφόρμα του flash θα αποδειχθεί λίγη και αρκετά περιορισμένη. Τέλος, η υλοποίηση που θα μπορούσε να ακολουθηθεί αλλά είχε και αυτή αρκετά προβλήματα documentation και σφαλμάτων ήταν μέσα από το περιβάλλον του λογισμικού Unity και το project είχε και ακόμα έχει την ονομασία UnityAR. Το συγκεκριμένο project έχει αρκετές προοπτικές καθώς η πλατφόρμα της Unity είναι ένα σύγχρονο περιβάλλον που υποστηρίζει αρκετά συστήματα (win, android, ios), τρισδιάστατο ήχο, animation και διαχείριση κόσμου.

Περιβάλλον προγραμματισμού

Επιλογή Περιβάλλοντος

Η υλοποίηση του λογισμικού έγινε σε γλώσσα C++ και συγκεκριμένα με την δωρεάν έκδοση του Visual Studio 2008, λεγόμενη και Express. Το συγκεκριμένο περιβάλλον επιλέχθηκε μετά από διερεύνηση μεταξύ των: visual studio 2005, visual studio 2008 και visual studio 2010. Σε κάθε ένα από αυτά δοκιμάστηκε το compile του OpenSceneGraph και του FBX sdk που διανέμει η εταιρεία Autodesk. Αυτό ήταν απαραίτητο γιατί έπρεπε να μπορούν να διαβαστούν σωστά τα FBX αρχεία που χρησιμοποιούνται για 3D modelling και για animation. Έγιναν οι απαραίτητες δοκιμές με όλους τους συνδυασμούς visual studio και με τις διάφορες εκδόσεις του FBX sdk. Τελικά το επιθυμητό αποτέλεσμα ήρθε με τη χρήση του visual Studio 2008, με το compile εξ αρχής του OpenSceneGraph 3.0 και με την ενσωμάτωση του fbx sdk , έκδοση 2012.1 .

Στις περιπτώσεις που ήταν λάθος η έκδοση του visual studio εμφανίστηκαν προβλήματα linking μεταξύ των καλούμενων lib και dlls του OpenSceneGraph. Στην περίπτωση που η έκδοση του FBX sdk ήταν λάθος, τα μοντέλα εμφανίζονταν με λανθασμένο προσανατολισμό αλλά και με αρκετά προβλήματα στη σύνδεση των κλειδώσεων του σκελετού που τα χαρακτήριζε . Ήταν φανερό ότι η βιβλιοθήκη διερμηνέας της autodesk λειτουργούσε με διαφορετικό τρόπο. Το λογισμικό που χρησιμοποιήθηκε για την επεξεργασία των 3D μοντέλων είναι το Autodesk 3D Studio Max 2011 και για τα animations χρησιμοποιήθηκε το Autodesk Motion Builder 2013, λόγω του ότι έχει τον καλύτερο και πιο ανανεωμένο exporter για αρχεία FBX.

Εγκαταστάσεις και χτίσιμο βιβλιοθηκών

Μετά από την κατάληξη για χρήση του Visual Studio 2008 είναι απαραίτητη η αναφορά στην εγκατάσταση και τον έλεγχο του ARtoolkit. Η έκδοση που χρησιμοποιήθηκε είναι η 2.72.1 και είναι και η τελευταία έκδοση που κυκλοφόρησε ως δωρεάν, με ανοιχτό κώδικα. Το ARtoolkit για να εγκατασταθεί σωστά πρέπει κανείς να ακολουθήσει τις οδηγίες που δίνονται στη σελίδα documentation και να προσθέσει την βιβλιοθήκη DSVL και το GLUT. Για την εκτέλεση σε περιβάλλον Microsoft υπάρχουν και έτοιμες οι βιβλιοθήκες του, που σημαίνει ότι δεν είναι απαραίτητο να γίνουν compile από εξ αρχής. Στην περίπτωση που κάποιος επιθυμεί να κάνει compile τη βιβλιοθήκη πρέπει οπωσδήποτε να το κάνει σε debug mode καθώς και οτιδήποτε άλλο χτίζεται τοπικά, πρέπει να γίνει σε debug mode. Με τον τρόπο αυτό αποφεύγονται λάθη που εμφανίζει η εφαρμογή κατά την εκτέλεση, τα οποία καταλήγουν σε unhandled exceptions και τερματισμό της λειτουργίας του προγράμματος – φυσικά χωρίς προφανή λόγο.

Η Εγκατάσταση και το χτίσιμο του OpenSceneGraph αποδείχθηκε το πιο σύνθετο κομμάτι αυτής της

διαδικασίας. Για το συγκεκριμένο project δοκιμάστηκαν οι εκδόσεις 2.8.0, 2.8.2, 2.8.3 και 3.0.0. Κάθε έκδοση έγινε compile με τις εκδόσεις Visual Studio 2005 και 2008. Οι εκδόσεις 2.8.x προχωρούσαν σωστά σε όλη τη διάρκεια του compilation του OpenSceneGraph με χρήση του Visual Studio 2005. Το πρόβλημα που αντιμετωπίστηκε ήταν η μη υποστήριξη του VS2005 από την εταιρεία Autodesk και το SDK για τα αρχεία FBX. Εφόσον όμως τα αρχεία αυτά πρέπει να χρησιμοποιηθούν για το project, ήταν απαραίτητο να γίνει η μεταφορά σε VS2008. Με την αλλαγή αυτή παρατηρήθηκαν προβλήματα στις εντολές pre-processor του project OpenSceneGraph 2.8.x και τα modules osgAnimation και osgWidget παρουσίαζαν σφάλματα. Το αποτέλεσμα ήταν να δοκιμαστεί η εγκατάσταση της έκδοσης OpenSceneGraph 3.0.0 που τελικά αποδείχτηκε η πιο απροβλημάτιστη. Αν εξαιρεθεί το πρόβλημα των paths που αντιμετωπίζεται στα Windows 7 και πρέπει να δηλώνονται απόλυτες διαδρομές προς τα modules, έτσι ώστε να ανιχνεύονται σωστά τα αρχεία source, τότε το OpenSceneGraph είναι η λύση στα προαναφερθέντα προβλήματα εκδόσεων και ασυμβατότητας.

Το project δημιουργείται μέσα από το λογισμικό CMAKE το οποίο αναλαμβάνει να φτιάξει την απαραίτητη δομή για το VS 2008. Είναι πολύ σημαντικό να δηλωθούν οι προαπαιτούμενες βιβλιοθήκες σε όλα τα πεδία του cmake γιατί εάν παραλειφθούν, τότε το project θα εμφανιστεί ελλιπές και υπάρχει περίπτωση μέσα στο solution –σύνολο από projects του VS2008- θα διαπιστωθούν ελλείψεις. Το cmake κατά την παραγωγή του project (generation procedure), δεν πρέπει να βγάλει κανένα μήνυμα λάθους σχετικά με τα επιμέρους project. Εάν κάτι τέτοιο συμβεί είναι πολύ πιθανό κάποιο module να μην συμπεριλαμβάνεται στα projects προς compilation. Μαζί με τον πηγαίο κώδικα του OpenSceneGraph δίνεται και το 3rd Party tools που είναι απαραίτητο να υπάρχει καθώς μέσα σε αυτό το αρχείο υπάρχουν τα plug ins για αρχεία jpeg, tiff, ttf (γραμματοσειρές) και για τα osgWidgets.

Τελευταίο μέρος της εγκατάστασης είναι το OSGart, που έχει φτάσει μέχρι την έκδοση 2.0 RC 3. Για την δημιουργία του project χρησιμοποιήθηκε και πάλι το cmake. Κατά τη δημιουργία του project από το cmake είναι απαραίτητο να δηλωθούν σωστά τα αρχεία των βιβλιοθηκών του Artoolkit και του OpenSceneGraph. Για επιβεβαίωση της σωστής δημιουργίας του project, είναι απαραίτητο να εκτελεστούν τα παραδείγματα osgart_simple και osgart_viewer. Για να γίνουν σωστά τα compiles αυτών των project, θα πρέπει να έχουν συνδεθεί τα lib αρχεία μέσα από το μενού additional libraries. Εάν δεν γίνει αυτό θα βρεθούμε αντιμέτωποι με προβλήματα στη διαδικασία linking. Στην περίπτωση που ενώ έχουν δηλωθεί σωστά οι βιβλιοθήκες εξακολουθούν να εμφανίζονται μηνύματα αρχείων που λείπουν, τότε είναι απαραίτητο να προσθέσουμε στην Environment Variable : Path τα directories που έχουν παραχθεί τα dlls του Artoolkit και του OpenSceneGraph. Σε αντίθετη περίπτωση μπορούμε να αντιγράψουμε τα προαπαιτούμενα dlls στο ίδιο directory που παράγεται και το εκτελέσιμο πρόγραμμα μέσα από το VS2008.

Αρχιτεκτονική Λογισμικού

Ο σχεδιασμός του λογισμικού έγινε με γνώμονα την αναπαραγωγή ενός σεναρίου που δεν έχει προαποφασισμένα γεγονότα και ροή, παρόλο που το τρέχον σενάριο έχει έναν συγκεκριμένο σκελετό. Με τον τρόπο αυτό, αν και ο προγραμματισμός έγινε αρκετά πιο περίπλοκος, η μετάβαση σε μια ιστορία πιο δυναμική, θα μπορεί να γίνει πιο εύκολα.

Η εφαρμογή ξεκινά με ένα κεντρικό δομικό στοιχείο που είναι η κλάση StoryController. Η κλάση αυτή είναι υπεύθυνη για να αρχικοποιήσει το περιβάλλον και να ενορχηστρώσει όλες τις κλάσεις που είναι απαραίτητες για να ξεκινήσει η εξιστόρηση. Η πρώτη φάση αφορά την αρχικοποίηση της κάμερας και το setup του plug in του Artoolkit που λέγεται Tracker. Το αντικείμενο αυτό είναι υπεύθυνο για την ανίχνευση των προεπιλεγμένων μοτίβων και για να λειτουργήσει χρειάζεται όπως είναι αναμενόμενο τα μοτίβα και το stream της κάμερας.

Η επόμενη φάση του StoryController αφορά τον χρήστη και συγκεκριμένα τη σύνθεση του παραθύρου που

είναι στο οπτικό του πεδίο, έτσι ώστε να ενημερώνεται από το σύστημα με διάφορες πληροφορίες αλλά και για να λαμβάνει απαντήσεις μετά από τις διάφορες πράξεις του. Η σύνθεση αποτελείται από αντικείμενα `osgWidgets` και μια δεύτερη κάμερα του `OpenSceneGraph`, αυτή τη φορά στατική. Προγραμματιστικά ορίζουμε την εικονική κάμερα στο σημείο `0.0.0` και παίρνουμε ως αποτέλεσμα το λεγόμενο HUD ή αλλιώς `Heads Up Display`.

Σειρά έχει η δημιουργία κόμβων του `OpenSceneGraph`, ενωμένων με τα 3D γραφικά αλλά και με τα `transform matrice` του `Tracker`. Έτσι, εισάγονται τα γραφικά στο `project` και εγκαθίσταται η επικοινωνία τους με τον `Tracker`. Με αυτή τη διαδικασία πετυχαίνουμε την αλλαγή της θέσης των αντικειμένων, σύμφωνα με τους υπολογισμούς του `Artoolkit`. Εφόσον ο `Tracker` μεταφέρει τις πληροφορίες μεταβολής του μοτίβου στο πρόγραμμά μας ανά `frame`, το `OpenSceneGraph` λαμβάνει τη νέα θέση με την ίδια ταχύτητα και την εφαρμόζει στο 3D αντικείμενο. Το αποτέλεσμα είναι η αλλαγή της θέσης του αντικειμένου σύμφωνα με την οπτική γωνία της κάμερας, σε πραγματικό χρόνο.

Μετά την αρχικοποίηση του HUD και των `markers` σειρά έχει η ένωση των δράσεων που θα μπορεί να έχει ο χρήστης καθώς βρίσκεται μέσα στο περιβάλλον. Το `OpenSceneGraph`, όπως αναφέρθηκε νωρίτερα είναι μόνο ένα σύστημα για την αναπαραγωγή ενός γραφήματος σκηνής. Για περισσότερες λειτουργίες όπως `input` κινήσεων του χρήστη από το πληκτρολόγιο ή το ποντίκι, πυροδότηση συγκεκριμένων οπτικών `events`, έλεγχο `animation` κλπ, χρησιμοποιεί τον μηχανισμό των `callbacks`. Προς στιγμή θα πούμε μόνο ότι τα `callbacks` είναι ένας τρόπος για να σταματήσει το `rendering` του γραφήματος σκηνής, να τρέξει ένα ξεχωριστό κομμάτι κώδικα και μόλις αυτό τελειώσει, να συνεχιστεί το `rendering`. Κάθε ενέργεια του χρήστη πάνω σε αντικείμενο του εικονικού κόσμου, έχει κωδικοποιηθεί μέσα σε ένα `callback`. Έτσι, επιστρέφοντας στην επεξήγηση της αρχιτεκτονικής του συστήματος ανφέρουμε ότι επόμενη φάση είναι η ένωση όλων των δράσεων του χρήστη με το γράφημα σκηνής μέσω των `callbacks`.

Τέλος, αρχικοποιούνται δυο βάσεις γνώσης με μορφή συνδεδεμένων λιστών. Αυτό γίνεται διότι όσο ο χρήστης μέσα από τις δράσεις του, προχωρά στο σενάριο της ιστορίας, είναι απαραίτητο να διατηρηθεί η κατάσταση. Έτσι, η βάση γνώσης του χρήστη υφίσταται όσο εκτελείται το πρόγραμμα και είναι μοναδική για κάθε χρήστη. Αντίστοιχα, διατηρείται μια βάση γνώσης για δεδομένα που αφορούν το περιβάλλον και κατά συνέπεια όλους τους χρήστες που μπορεί να συμμετέχουν ταυτόχρονα στην ιστορία. Με τον τρόπο αυτό μπορεί να διατηρηθεί η ροή της ιστορίας και να είναι πάντα διαθέσιμη η κατάσταση του χρήστη και του περιβάλλοντος. Η λειτουργικότητα των δυο βάσεων γνώσεων θα αναλυθεί περαιτέρω στη συνέχεια.

Κυριότερα χαρακτηριστικά λογισμικού

Tracker και Video Initialization

Σε αυτό το σημείο θα αναλυθούν τα μέρη που αναφέρθηκαν λίγο νωρίτερα στην αρχιτεκτονική του συστήματος με σκοπό την καλύτερη κατανόηση της λειτουργίας της εφαρμογής.

Η αρχικοποίηση του γράφου και η τοποθέτηση της εικόνας της κάμερας ως `background` στον `viewer` του χρήστη

```
osg::ref_ptr<osg::Group> root = new osg::Group;  
viewer.setSceneData(root.get());
```

Αρχικοποίηση του Video

```
int _video_id =
osgART::PluginManager::instance()->load("osgart_video_artoolkit2");
    // preload the video and tracker
osg::ref_ptr<osgART::Video> video =
    dynamic_cast<osgART::Video*>(osgART::PluginManager::instance()->
        get(_video_id)); // Load a video plugin.
```

Αρχικοποίηση Έλεγχος για την έναρξη του video stream

```
if (!video.valid()){
    osg::notify(osg::FATAL) <<
        "Could not initialize video plugin!" <<
        std::endl;
    exit(-1);
}
```

Τοποθέτηση του video stream ως background. Το αντικείμενο Cam είναι και αυτό κόμβος του γράφου και συγκεκριμένα είναι η κάμερα που ενώνεται με τον viewer. Ο χρήστης παρακολουθεί μέσα από την συγκεκριμένη κάμερα.

```
osg::ref_ptr<osg::Group> videoBackground =
createImageBackground(video.get());
videoBackground->getOrCreateStateSet()->setRenderBinDetails(0,
"RenderBin");
cam->addChild(videoBackground.get());
```

Στη συνέχεια ακολουθεί η αρχικοποίηση του Tracker και η σύνδεση των κόμβων με τον Tracker και την κάμερα.

Φόρτωση του plug in και δημιουργία του αντικειμένου tracker

```
osg::ref_ptr<osgART::Tracker> tracker =
    dynamic_cast<osgART::Tracker*>(osgART::PluginManager::instance()->
        get(_tracker_id));

int _tracker_id =
osgART::PluginManager::instance()->load("osgart_tracker_artoolkit2");
```

Άνοιγμα του αρχείου calibration για τις παραμέτρους του artoolkit.

```
osg::ref_ptr<osgART::Calibration> calibration = tracker->
getOrCreateCalibration();

// load a calibration file
if (!calibration->load(std::string("C:/Data/camera_para.dat")))
{
    // the calibration file was non-existing or couldnt be loaded
    osg::notify(osg::FATAL)
    << "Non existing or incompatible calibration file"
    << std::endl;
    exit(-1);
}
```

Ένωση του video stream από την κάμερα, με τον tracker

```
tracker->setImage(video.get());
```

Πρόσθεση ενός callback που θα εκτελείται σε κάθε ανανέωση του γράφου. Αυτό είναι το σημείο που ο tracker ενώνεται με τον γράφο σκηνής και έτσι μεταφέρει δεδομένα προς το σύστημα frame by frame

```
osgART::TrackerCallback::addOrSet(root.get(), tracker.get());
```

Κάθε κόμβος που προστίθεται από εμάς και θέλουμε να ανιχνεύεται πρέπει να αρχικοποιείται μέσα από τον tracker

```
osg::ref_ptr<osgART::Marker> newMarker =
    tracker->addMarker( markerFilename );
```

Στη συνέχεια πρέπει να αρχικοποιηθεί η κάμερα σύμφωνα με το calibration αρχείο που έχει παραχθεί από τον tracker

```
osg::ref_ptr<osg::Camera> cam = calibration->createCamera();
```

Η κάμερα που δημιουργήθηκε είναι τμήμα του κεντρικού γράφου και συγκεκριμένα ο δεύτερος κόμβος. Για να εμφανίζονται στο γράφημα σκηνής και τα δικά μας μοντέλα πρέπει να ενωθούν με τον κεντρικό γράφο και συγκεκριμένα ακριβώς κάτω από την κάμερα. Έτσι εκτελείται η παρακάτω μέθοδος.

```
int initializeResult = camInitialization(cam);
```

Όπου μέσα της γίνεται η ένωση για κάθε αντικείμενο που έχει προστεθεί στο σύστημα και υπάρχει μέσα στη λίστα με τους κόμβους.


```

for ( TmarkerTransformVector::size_type i=0;
      i < markerTransformVector.size(); i++)
{
    cam->addChild(markerTransformVector[i].get());
}
return 1;

```

Widgets

Η βιβλιοθήκη `osgWidgets` είναι ένα plug in για το `OpenSceneGraph` και δίνει τη δυνατότητα εμφάνισης μνημάτων σε διάφορα σχήματα, μεγέθη, χρώματα και γραμματοσειρές μέσα στο κεντρικό παράθυρο του `OpenSceneGraph`, τον λεγόμενο `viewer`. Μαζί με τα μνήματα έρχονται και επιπρόσθετες κλάσεις που επιτρέπουν τον γενικότερο έλεγχο πάνω σε αυτά τα αντικείμενα αλλά και την τοποθέτησή τους χωροταξικά, πάνω στο παράθυρο του `viewer`. Για την εν λόγω εφαρμογή, τα `widgets` χρησιμοποιούνται α)για την εμφάνιση της οριζόντιας σταθερής μπάρας πληροφοριών, β)για την εμφάνιση των επιλογών του χρήστη πάνω σε κάποιο αντικείμενο, γ)για την εμφάνιση μνημάτων απάντησης προς τις πράξεις του χρήστη.

Σε κάθε περίπτωση πρέπει να αναφερθεί ότι χρησιμοποιείται το αντικείμενο `windowManager` που είναι υπεύθυνο για να φιλοξενήσει αυτά τα μνήματα κειμένου. Όπως ονομάζεται σε διάφορα συστήματα, η κλάση `windowManager` είναι ένας `container` για μνήματα τύπου `widget`. Τα μνήματα αυτά δεν είναι μόνο οπτικά εφέ, αλλά μπορούν να αποτελέσουν και δομικό ρόλο σε εφαρμογές, καθώς μπορούν να συνδεθούν με το `interface`. Ο χρήστης μπορεί να δράσει πάνω στα `widgets` κάνοντας κλικ πάνω τους, τραβώντας τα με `drag` ή `drop` ή ακόμα και να πληκτρολογήσει μέσα τους.

Στο συγκεκριμένο πρόγραμμα η λειτουργία των `widgets` περιορίζεται στην εμφάνιση μνημάτων δυναμικά, κατά τη διάρκεια της ιστορίας. Η αρχικοποίηση κατά κανόνα γίνεται με τις παρακάτω εντολές

```

osgWidget::Label* label = new osgWidget::Label( "", "" );
label->setFont( "fonts/Vera.ttf" );
label->setFontSize(25);
label->setFontColor(1.5f, 1.5f, 1.5f, 1.5f);
label->setLabel(1);
label->setName( labelName );
label->setColor(1.0f, 0.5f, 0.0f, 0.8f);
label->setPadding(5.0f);
label->addSize(12.0f, 12.0f);

_dialogBox->addWidget( label );

```

Όπως φαίνεται και στον κώδικα, το `widget` που εμφανίζεται είναι τύπου `label`, του δίνουμε αρχικές τιμές σχετικά με τη μορφή που θέλουμε να έχει και τελικά το προσθέτουμε στο `dialogBox` που είναι τύπου `osgWidget::Box` και είναι το στοιχείο που έχει ήδη προστεθεί σε συγκεκριμένο `windowManager`.

Τα μνήματα που εμφανίζονται στην εφαρμογή πρέπει και να εξαφανίζονται όταν πλέον πληροφορία δεν είναι χρήσιμη για τον χρήστη. Η διαγραφή ενός μνημάτος γίνεται για δυο λόγους. Είτε επειδή το μνήμα ήταν μενού επιλογών και ο χρήστης έστρεψε την κάμερα προς κάποια κατεύθυνση που ο `marker` επιλογών δεν είναι πια ορατός, είτε επειδή κάποιο προαποφασισμένο χρονικό διάστημα έχει παρέλθει. Για την απαλοιφή ενός μνημάτος από τον `viewer` είναι απαραίτητος ο μοναδικός του προσδιορισμός, έτσι ώστε το η διαδικασία απαλοιφής να διαγραφεί μόνο το συγκεκριμένο μνήμα και όχι οποιοδήποτε. Ο

προσδιορισμός των μυνημάτων γίνεται κατά τη δημιουργία τους και συγκεκριμένα από την τιμή της private ιδιότητας Name του αντικειμένου Label. Όταν η συνθήκη απενεργοποίησης είναι αληθής τότε :

```
if ( _dialogBox->getObjects().size() > 0 ){
    osgWidget::Widget* w = _dialogBox->getObjects()[0].get();
    if( (w->getName().compare("AthenaInfoDialog1") == 0) )
        _dialogBox->removeWidget(w);
}
```

Γίνεται έλεγχος για το εάν όντως υπάρχουν μυνήματα για απενεργοποίηση, εάν ναι, ανακτάται το υπάρχον και γίνεται σύγκριση με το όνομα που περιμένουμε να λάβουμε καθώς θέλουμε να αφαιρεθεί μόνο το συγκεκριμένο μήνυμα. Εάν ταιριάζει το όνομα του μυνήματος που υπάρχει στην οθόνη τότε το αφαιρούμε από το dialogBox και αυτομάτως εξαφανίζεται από την οθόνη.

Εμφάνιση και Απαλοιφή επιλογών μέσω Callbacks

Το επόμενο μέρος προς ανάλυση είναι η δημιουργία των callbacks για την εμφάνιση των μενού επιλογών στα αντικείμενα που ο χρήστης μπορεί να επιδράσει. Η παραδοχή που έγινε είναι ότι ένα 3D αντικείμενο θα φαίνεται από μεσαία και κοντινή απόσταση ενώ οι επιλογές του πρέπει να είναι ορατές μόνο όταν ο χρήστης είναι αρκετά κοντά. Έτσι, αποφασίστηκε ότι τα μενού επιλογών θα είναι συνδεδεμένα με μικρότερους markers και όταν ο χρήστης πλησιάζει και γίνεται η ανίχνευσή τους, θα εμφανίζεται και το μενού επιλογών. Εάν ο χρήστης κοιτάξει προς άλλη κατεύθυνση, ο marker του μενού επιλογών δεν θα ανιχνεύεται πια και το μενού θα εξαφανίζεται.

Η ανάθεση της κλάσης callback γίνεται όταν αρχικοποιούνται και τα υπόλοιπα στοιχεία του marker, μέσα στην συνάρτηση addMarkerFunctionality. Στη συνάρτηση αυτή μετά την αρχικοποίηση του marker χρησιμοποιείται ο κόμβος σύνδεσης με τον γράφο που είναι το αντικείμενο newMarkerTransform και καλείται η συνάρτηση :

```
attachCustomCallback( newMarker.get(), newMarkerTransform.get(), cbType);
```

Στο σώμα αυτής, ανάλογα με το πιο callback αρχικοποιείται, εκτελείται και το τμήμα όπως το παρακάτω, για να συνδεθεί η κλάση του νέου callback με τον επιθυμητό κόμβο του γράφου. Αυτό σημαίνει πως το συγκεκριμένο κομμάτι κώδικα θα εκτελείται όταν θα γίνεται η διάσχιση του γράφου (traverse) και μόνο όταν ο έλεγχος θα φτάνει σε αυτόν τον συγκεκριμένο κόμβο.

```
osgART::addEventCallback(
MarkerTransform.get(),
new MarkerIsTracedCallback(marker.get(),
this->dialogBox,
this->theUser,
avmcVector)
);
```

Όταν εκτελούνται τα αναφερόμενα callbacks, που είναι συνδεδεμένα με τους markers με λειτουργία εμφάνισης των επιλογών (option markers), τότε γίνεται έλεγχος για το εάν είναι widget που αναγράφει τις διαθέσιμες επιλογές. Η λειτουργία των widgets (παραθύρων πληροφοριών) γίνεται με την πρόσθεση ενός νέου παραθύρου στην προαποφασισμένη περιοχή διαλόγου (_dialogBox) που έχει κοινοποιηθεί στην κλάση κατά τη δημιουργία της. Εάν είναι άδεια η περιοχή διαλόγου

```
if (_dialogBox->getObjects().size() < 1 ) {
```

τότε δημιουργείται ένα νέο παράθυρο πληροφοριών

```
athinaOptionslabel = createLabel(ATHINAOPTIONSLABEL_1);
```

Η συνάρτηση `createLabel` είναι υπεύθυνη για την δημιουργία του νέου παραθύρου και να αρχικοποιήσει τα χαρακτηριστικά του όπως γραμματοσειρά, μέγεθος γραμμάτων, χρώμα χαρακτήρων, χρώμα background.

Η κλάσεις των callbacks που αναφέρονται σε markers επιλογών, εκτός από την εμφάνιση αυτών των επιλογών, είναι υπεύθυνες και για την απαλοιφή των επιλογών από την οθόνη. Έτσι, όταν κάποιος marker επιλογής πάψει να είναι ορατός και να ανιχνεύεται από την κάμερα, το callback θα εκτελεστεί και εφόσον ο marker δεν θα ανιχνεύεται πια, τότε είναι υπεύθυνο να καθαρίσει την οθόνη του χρήστη. Το κομμάτι κώδικα που κάνει αυτή τη λειτουργία είναι το παρακάτω:

```
}else if (!_marker->valid()){
    /*DE-ACTIVATE dialog box*/
    if (_dialogBox->getObjects().size() > 0 ){
        osgWidget::Widget* w = _dialogBox-
>getObjects()[0].get();
        if (w->getName().compare("athinaOptionsLabel") == 0 )
            _dialogBox->removeWidget(w);
    }
}
```

Το παραπάνω block τσεκάρει την ύπαρξη του συγκεκριμένου marker καθώς και την ύπαρξη του παραθύρου επιλογών μέσα από τη λίστα αντικειμένων του προεπιλεγμένου χώρου εμφάνισης (dialogBox). Εάν αυτή η λίστα είναι μηδενική, τότε δεν υπάρχει κάποιο παράθυρο για απαλοιφή. Αλλιώς, θα πρέπει το υπάρχον παράθυρο να είναι αυτό που δημιουργήθηκε από την ύπαρξη του συγκεκριμένου marker -σε κάποια προηγούμενη φάση- και εάν βρεθεί, αφαιρείται από τον προεπιλεγμένο χώρο εμφάνισης (dialogBox).

Keyboard Handlers

Για την εισαγωγή των επιλογών του χρήστη, χρησιμοποιείται το πληκτρολόγιο. Για την ανίχνευση των χτυπημάτων πληκτρολογίου του χρήστη αλλά και για την εκτέλεση συγκεκριμένου κώδικα ανάλογα με την επιλογή αυτή, το OpenSceneGraph διαθέτει την κλάση `GUIEventHandler` που διαβάζει είτε εισαγωγές από το πληκτρολόγιο, είτε από το ποντίκι. Για κάθε περίπτωση που θέλουμε να εκτελούνται διαφορετικά τμήματα κώδικα ανάλογα με τις επιλογές του χρήστη, δημιουργηθηκαν και ξεχωριστές κλάσεις που κληρονομούν την `GUIEventHandler`. Η λογική που ακολουθείται για τις κλάσεις των handlers είναι ότι υπάρχει μια ξεχωριστή κλάση handler για κάθε αντικείμενο που μπορεί ο χρήστης να αλληλεπιδράσει. Κατά την εκκίνηση της εφαρμογής συνδέονται οι κλάσεις των keyboard Handlers με το παράθυρο του viewer που θα λαμβάνει και τα inputs του χρήστη. Για να είναι εφικτός ο διαχωρισμός των αποτελεσμάτων κάθε πατήματος, αναλόγως το virtual αντικείμενο που κοιτά ο χρήστης, σε κάθε κλάση handler γίνεται και έλεγχος εάν το συνδεδεμένο αντικείμενο είναι ορατό. Με την ακόλουθη εντολή γίνεται η σύνδεση του handler με έναν συγκεκριμένο marker και τελικά με τον viewer.

```
viewer.addEventHandler( new AthinaKeyboardHandler (
getMarkerByDescription("Athina's options"),mainMessageDialogBox,
theUser) );
```

Στα ορίσματα υπάρχει ο συνδεδεμένος Marker, η περιοχή εμφάνισης μυνημάτων (mainMessageDialogBox)

και φυσικά το αντικείμενο με τις πληροφορίες του τρέχοντος χρήστη (theUser).

Ένας keyboard Handler, στην ουσία είναι ένα callback που συμβαίνει όταν δημιουργηθεί ένα event μέσα από graphical user interface (GUI). Στη δική μας περίπτωση αναλαμβάνουμε να μεταφράσουμε το κάθε event που συμβαίνει στο GUI του χρήστη. Εάν είναι ένα από αυτά που μας ενδιαφέρουν, τότε εκτελούμε και τον υπόλοιπο κώδικα της κλάσης.

```
switch (ea.getEventType()) {
    case osgGA::GUIEventAdapter::KEYDOWN : {
        switch (ea.getKey()) {
            case osgGA::GUIEventAdapter::KEY_1: {
```

Γίνεται έλεγχος για το εάν πρόκειται για το εάν πρόκειται για Event τύπου keydown. Το event το λαμβάνουμε από την κληρονομηθείσα κλάση και συγκεκριμένα από το αντικείμενο EventAdapter. Εάν το event είναι του πληκτρολογίου, γίνεται έλεγχος για το ποιό είναι το πλήκτρο που πατήθηκε. Εφόσον μπορούμε να ξεχωρίσουμε τα πλήκτρα που πατιούνται τότε είναι εύκολο να συνδέσουμε και τον κώδικα που θέλουμε στην εκάστοτε περίπτωση. Στην εφαρμογή το σύνηθες είναι όταν πατιέται κάποια από τα πλήκτρα να εμφανίζεται ένα μήνυμα στην οθόνη και εφόσον ο χρήστης λαμβάνει μια νέα πληροφορία, ανανεώνεται η βάση γνώσης του ή το status του περιβάλλοντός του. Είναι αυτονόητο ότι σε κάθε κλάση πληκτρολογίου που εμφανίζει μηνύματα, υπάρχει και το αντίστοιχο κομμάτι κώδικα που μετά από συγκεκριμένο χρονικό διάστημα τα εξαφανίζει.

Proximity Callbacks

Η αρχιτεκτονική του OSgart, επιτρέπει να συμπεριφερόμαστε στους markers σαν αντικείμενα με συγκεκριμένη θέση στον virtual κόσμο. Εάν ανάγουμε αυτό το γεγονός στο ότι σε κάθε marker συνδέεται και ένα virtual 3D αντικείμενο, τότε είναι κατανοητό ότι με την βοήθεια των markers μπορούμε πάντα να ξέρουμε τη σχετική με την κάμερα, θέση των αντικειμένων. Με τον τρόπο αυτό μπορούμε να δημιουργήσουμε events στο σενάριό μας που έχουν ως βάση την θέση ενός αντικειμένου σε σχέση με ένα άλλο. Στο σενάριο για παράδειγμα η εικονική φοιτήτρια ζητά ένα αντικείμενο controller και μόνο όταν ο χρήστης το τοποθετήσει αρκετά κοντά, το σενάριο μπορεί να συνεχιστεί. Τεχνικά αυτή η λειτουργία εξηγείται από τον υπολογισμό της απόστασης μεταξύ των δυο δυανισμάτων των markers. Δεδομένου ότι μιλάμε για απόσταση δυο αντικειμένων στον συνολικό κόσμο, ο υπολογισμός της απόστασης πρέπει να γίνεται καθ' όλη τη διάρκεια της εφαρμογής και σε κάθε διάσχιση του γραφήματος σκηνής. Για τον λόγο αυτό ο υπολογισμός είναι μέσα σε callback που καλείται σε κάθε update του γράφου.

```
root->addUpdateCallback(
new ControllerToAthenaProximityUpdateCallback(
    getMarkerTransformByDescription("Controller"),
    getMarkerTransformByDescription("Athena"),
    350.0f,
    mainMessageDialogBox,
    theEnvironment,
    theUser,
    getMarkerByDescription("Athena"),
    getMarkerByDescription("Controller"),
    avmcVector )
```



```
);
```

Όσα αναφέρθηκαν για τον υπολογισμό της απόστασης, αποτυπώνονται στον κώδικα όπως φαίνεται παρακάτω

```
osg::Vec3 posAthina = _mtAthina->getMatrix().getTrans();
osg::Vec3 posController = _mtController->getMatrix().getTrans();
float distance = 0;

if(posAthina.length() != 0 && posController.length() != 0){
    osg::Vec3 offset = posAthina - posController;
    distance = offset.length();
}
```

Γίνεται ανάκτηση και των δυο πινάκων θέσεων των δυο virtual αντικειμένων και στη συνέχεια παίρνουμε το μήκος των δυο διανυσμάτων που δεν πρέπει να είναι 0, γιατί αυτό σημαίνει πως δεν είναι υπαρκτά στον κόσμο. Επόμενο βήμα είναι ο υπολογισμός του διανύσματος της διαφοράς μεταξύ των δυο θέσεων και τέλος βλέπουμε την ανάκτηση του μήκους του διανύσματος.

```
if (_mAthina->valid() && _mController->valid()){
    if (distance <= _mThreshold){
```

Για να ισχύει το αποτέλεσμα της απόστασης και εφόσον μιλάμε για απόσταση σε σχέση με την κάμερα, πρέπει και οι δυο markers να είναι ορατοί. Εάν όντως είναι τότε γίνεται η σύγκριση της εν λόγω απόστασης με την προαποφασισμένη τιμή threshold. Εάν και μόνο εάν εκπληρωθεί η συνθήκη σε σχέση με το threshold τότε εκτελείται και το block κώδικα που προχωρά το σενάριο στην επόμενη φάση.

Animation Handling

Άλλο σημαντικό μέρος του κώδικα που πρέπει να αναφερθεί, είναι η διαχείριση των animations κατά τη διάρκεια της ιστορίας. Όπως αναφέρθηκε και παραπάνω, για να είναι ο μεικτός κόσμος ρεαλιστικός, μια προϋπόθεση είναι η ύπαρξη διάφορων animations στα 3D μοντέλα. Υπάρχουν αρκετά σημεία της ιστορίας που τέτοια animations εκτελούνται και είναι αποτέλεσμα κάποιας δράσης του χρήστη. Για την χρήση πολλών animations ανά μοντέλο χρησιμοποιήθηκε το format fbx, όπου ένα αρχείο μπορεί να περιέχει πολλά clips. Το module του OpenSceneGraph που χρησιμοποιήθηκε για την διαχείριση των animations λέγεται osgAnimation.

Η βασική αρχή λειτουργίας αυτής της βιβλιοθήκης είναι η εκτέλεση μιας scanning διαδικασίας στο κάθε 3D αρχείο και η υλοποίηση ενός controller αντικειμένου, που μπορεί να χειρίζεται όλα τα σκαναρισμένα clips. Ένας από τους λόγους που επιλέχθηκε το format fbx ήταν και αυτός. Το osgAnimation μπορεί να ανοίγει σωστά τα αρχεία και να αναπαράγει το κάθε clip κατά απαίτηση του χρήστη χωρίς λάθη.

Για κάθε μοντέλο δημιουργείται και ένα αντικείμενο που θα το σκανάρει και θα εντοπίσει τα περικλειόμενα σε αυτό animations. Το σκανάρισμα γίνεται με την μέθοδο accept.

```
AnimationManagerFinder finder;
newMarkerTransform->accept(finder);
```

Το αντικείμενο AnimationManagerFinder είναι ένα struct τυπου osg::NodeVisitor, που σημαίνει ότι θα λειτουργήσει σε κάθε παιδί του κόμβου που εκτελεί την μέθοδο accept. Για κάθε κόμβο που επισκέπτεται κάνει και έλεγχο για το εάν υπάρχει κάποιο updateCallback. Όταν βρεθεί αυτό το callback σημαίνει ότι αυτός ο κόμβος έχει τεθεί ως κεντρικός και το block κώδικα που εμπεριέχει στο update του, ελέγχει και όλους τους κόμβους στο animation. Για τον λόγο αυτό, το συγκεκριμένο callback αποτελεί και τη βάση των animations ή αλλιώς το animationManager Base.

Στη συνέχεια πρέπει να αρχικοποιηθεί ένας controller που θα ελέγχει την εκτέλεση αυτών των animations όπως φαίνεται παρακάτω,

```
AnimtkViewerModelController* avmc = new AnimtkViewerModelController();
```

Εάν βρεθεί ο animationManager, για να εκτελείται ως updateCallback, πρέπει να συνδεθεί και με το κεντρικό γράφημα σκηνής και συγκεκριμένα με τον κόμβο που θα κινήσει. Εάν δεν γίνει αυτό, ο κόμβος θα παραμένει ακίνητος καθώς το updateCallback δεν θα εκτελείται ποτέ.

```
if (finder._am.valid()) {
    newMarkerTransform-
    >setUpdateCallback(finder._am.get());
}
```

Επόμενη κίνηση είναι να συνδεθεί με τον συγκεκριμένο controller, το συγκεκριμένο μοντέλο

```
avmc->setModel(finder._am.get());
```

Τέλος, για δική μας διευκόλυνση αποθηκεύουμε τον controller σε ένα vector, μαζί με μια περιγραφή για να μπορούμε να έχουμε εύκολη και γρήγορη πρόσβαση μεταξύ των μοντέλων, όταν αυτά είναι περισσότερα από ένα.

```
AvmcData* avmcData = new
AvmcData(avmc, markerDescription);
avmcVector.push_back(avmcData);
}
```

Όταν στην ιστορία απαιτείται να εκτελεστεί ένα animation τότε ο έλεγχος του μοντέλου γίνεται ως εξής, πρώτα βρίσκουμε τον model controller μέσα από τη λίστα vector που είναι όλοι οι controllers καταχωρημένοι και έπειτα χρησιμοποιούμε την μέθοδο του controller playByName

```
AnimtkViewerModelController* mc = avmcd->_avmc;
mc->playByName("program");
```

Το όνομα κάθε animation δίδεται κατά την δημιουργία του μέσα από το πρόγραμμα της Autodesk, Motion Buillder.

Custom Events

Για την δημιουργία μιας εξιστόρησης, είναι αναγκαίο να υπάρχει η δυνατότητα στον διαχειριστή του συστήματος ή στον προγραμματιστή, να μπορεί να δημιουργήσει ή/και να πυροδοτήσει δρώμενα κατά βούληση. Για τον λόγο αυτό έγινε έρευνα ώστε να εξετασθεί εάν υπάρχει αυτή η δυνατότητα με το OpenSceneGraph. Το παράδειγμα που υλοποιήθηκε για να αποδειχθεί η δυνατότητα των custom events είναι η υλοποίηση ενός χρονογράφου με την παραγωγή ενός event ανα μια συγκεκριμένη περιοδικότητα.

Για να λειτουργήσει μια τέτοια υλοποίηση πρέπει να ακολουθηθεί η διαδικασία που υποστηρίζει το OpenSceneGraph. Σύμφωνα με αυτή θα πρέπει να δημιουργηθεί μια κλάση που θα ενωθεί με τον viewer και θα εκτελείται κατά την διάσχιση του γραφήματος σκηνής, μέσω ενός callback. Κατά συνέπεια δημιουργήθηκε η κλάση TimerHandler.

Η παραπάνω κλάση ορίζεται ως απόγονος της osgGA::GUIEventHandler που είναι και η default κλάση του OpenSceneGraph για όλους τους handlers του γραφικού περιβάλλοντος και του interface. Στη συγκεκριμένη περίπτωση δεν αναλύεται η αρχιτεκτονική εκτέλεσης του γραφήματος σκηνής, οπότε θα αρκεστούμε στο γεγονός ότι όλες οι κλάσεις που κληρονομούν την GUIEventHandler υλοποιούν την virtual μέθοδο handle που εκτελείται μέχρι και ανά frame στο γράφημα σκηνής και μέσα σε αυτήν μπορούμε να εκτελέσουμε τον κώδικα που επιθυμούμε.

Σχεδιαστικά αυτό που γίνεται στην handle συνάρτηση, είναι σε κάθε frame να γίνεται ένας έλεγχος σε έναν τοπικό μετρητή και εάν η προδιαγεγραμμένη περίοδος (τιμή μετρητή ανά εκατό ακεραίους) έχει παρέλθει, να εισάγεται στον viewer ένα νέο event του χρήστη. Στη συνέχεια, το αντικείμενο EventAdapter που συγκεντρώνει τα events σε μια λίστα FIFO (first in first out) ανιχνεύει το event του χρήστη και εμείς κάνοντας τον απαραίτητο έλεγχο στο event που έρχεται προς διαχείριση, εκτελούμε το block κώδικα που προορίζεται για την αντίστοιχη περίπτωση.

Στο σκέλος που δημιουργείται το event του χρήστη και εισάγεται στη λίστα του EventAdapter, πρέπει να αναφερθεί ότι εισάγεται και ένα αντικείμενο που έχει αποθηκευμένες πληροφορίες του χρήστη. Το αντικείμενο αυτό για να μπορεί να χρησιμοποιηθεί πρέπει να κληρονομεί την κλάση Referenced του OpenSceneGraph. Αυτή η δυνατότητα αποδुकνείται ιδιαίτερα βολική καθώς η χρήστης μπορεί να μεταφέρει δεδομένα οποιουδήποτε τύπου, από την φάση που δημιουργεί το event και να τα χρησιμοποιήσει κατά το handling. Στην παρούσα φάση τα δεδομένα του χρήστη που μεταφέρονται είναι μόνο η τιμή του μετρητή τη στιγμή του event.

Στον κώδικα η κλάση ορίζεται ως εξής,

```
class TimerHandler : public osgGA::GUIEventHandler
{
    ...
protected:
    unsigned int _count;
}
```

Και η συνάρτηση Handle ως εξής,

```
bool TimerHandler::handle( const osgGA::GUIEventAdapter &ea,
osgGA::GUIActionAdapter &aa )
{
    ...
}
```

Τα events του eventAdapter ξεχωρίζονται εάν είναι συστήματος ή χρήστη από το παρακάτω switch,

```
switch ( ea.getEventType() ){
    case osgGA::GUIEventAdapter::FRAME:
        ...

    case osgGA::GUIEventAdapter::USER:
        ...
}
```

Τέλος, το αντικείμενο με τις πληροφορίες του χρήστη ορίζεται όπως φαίνεται παρακάτω,

```
struct TimerInfo : public osg::Referenced
{
    TimerInfo( unsigned int c ) : _count(c) {}
    unsigned int _count;
};
```

Αντικείμενα 3D

Παρόλο που μέχρι τώρα περιγράφηκαν προγραμματιστικές δομές και αρχιτεκτονικές που μας δίνουν έλεγχο πάνω σε augmented reality εφαρμογές, το βασικότερο στοιχείο του μεικτού κόσμου είναι τα γραφικά. Χωρίς αυτά, η μεικτή πραγματικότητα δεν θα μπορούσε να υπάρχει και για αυτό το λόγο θα γίνει εκτενής αναφορά στο πως χρησιμοποιούνται τα γραφικά στην εφαρμογή, πως συνδέονται με το γράφημα σκηνής, ποιοί τύποι αρχείων γραφικών χρησιμοποιήθηκαν και γιατί, ποιά προγράμματα συνέβαλαν στη δημιουργία των μοντέλων και με ποιές δυσκολίες. Τέλος θα γίνει αναφορά στη χρήση του animation, πως αυτό μπορεί να υλοποιηθεί και σε ποιά μοντέλα ώστε να εκτελέσουν κινήσεις μπροστά από τον χρήστη.

Αρχεία που χρησιμοποιήθηκαν

Τα αρχεία που χρησιμοποιήθηκαν για να εκτελεστούν παραδείγματα ήταν τα osg που αποτελούν και το αυθεντικό format που υποστηρίζεται από OpenSceneGraph. Το πρόβλημα που αντιμετωπίστηκε σε αυτά τα αρχεία ήταν η υποστήριξη πολλαπλών animations μέσα σε ένα αρχείο και η χρήση τους κατά παραγγελία του χρήστη. Ενώ υπάρχουν προγράμματα που μπορούν να εξάγουν 3D μοντέλα σε osg, δεν είναι σαφές το εάν μπορούν να δημιουργήσουν και osg αρχεία με animation clips, ειδικά εάν μιλάμε για εργαλεία ανοιχτού κώδικα ή τουλάχιστον που να μπορούν να χρησιμοποιηθούν κάτω από ακαδημαϊκή άδεια.

Μετά από έρευνα στα πακέτα που επεξεργάζονται graphics και animation αλλά και μετά από αρκετές συζητήσεις στο επίσημο forum του OpenSceneGraph, οι δημιουργοί του γραφήματος σκηνής πρότειναν να χρησιμοποιηθεί το format fbx για οτιδήποτε έχει να κάνει με animation, καθώς το εργαλείο που χρησιμοποιείται για editing είναι αρκετά ολοκληρωμένο, είναι της Autodesk και έχει και ακαδημαϊκή άδεια χρήσης. Το εργαλείο είναι το MotionBuilder και τελικά χρησιμοποιήθηκε η νεότερη έκδοσή του που είναι το MotionBuilder 2013. Ένα άλλο πολύ θετικό πλεονέκτημα στην υπόθεση του fbx είναι ότι η Autodesk

δίνει δωρεάν το SDK πακέτο που μπορεί να συνδεθεί με το OpenSceneGraph και τελικά να μπορούν τα fbx να διαβάζονται και να μεταφράζονται σωστά από το γράφημα σκηνής.

Σχετικά με τα 3D μοντέλα που δεν πρέπει να εκτελούν κάποιο animation, επι το πλείστον χρησιμοποιήθηκε το 3DS. Το μέγεθος των αρχείων 3DS είναι σχετικά μικρότερο από το FBX και αυτό βοηθά στην καλύτερη απόδοση του γραφήματος σκηνής που πρέπει να εμφανίσει το μοντέλο σε πραγματικό χρόνο τη στιγμή που ανιχνεύεται ο marker. Επίσης, στα αρχεία 3DS η παραμετροποίηση της κάμερας και πάνω από όλα του φωτισμού είναι αρκετά πιο εύκολη και φαίνεται να μεταφράζεται με μεγαλύτερη ακρίβεια από το OSG σε σχέση με το FBX. Παρόλο που έγιναν δοκιμές με αρκετά αρχεία FBX τα μοντέλα φαινόταν σκοτεινά, κάτι που υποδηλώνει ότι ο φωτισμός είχε προβλήματα. Επειδή το εν λόγω έργο έχει σκοπό τη σύνθεση του κόσμου και όχι την αφεγάδιαστη απεικόνιση, δεν έγινε επίλυση του συγκεκριμένου θέματος και τελικά επιλέχθηκε το format 3DS που είχε αρκετά καλά αποτελέσματα κατά την εκτέλεση όσον αφορά την ποιότητα απεικόνισης αλλά και την τχύτητα φόρτωσης των μοντέλων.

Τα μοντέλα που χρησιμοποιήθηκαν προέρχονται από το website turbosquid και είναι διαθέσιμα δωρεάν για όποιον θέλει να τα κατεβάσει. Σε μερικά από αυτά έπρεπε να γίνουν μερικές διορθώσεις μέσα από το 3D Studio Max. Οι διορθώσεις είχαν σχέση με την κλίμακα του μοντέλου και την αρχική του θέση σε σχέση με τους άξονες. Σε περιπτώσεις όπως της οθόνης και του controller χρειάστηκε να αφαιρεθούν και ορισμένα μέρη του μοντέλου που δεν ήταν απαραίτητα για εμάς.

Τέλος η περισσότερη δουλειά έγινε στα μοντέλα των τριών χαρακτήρων της ιστορίας όπου εκεί έπρεπε να φτιαχτούν τα αντίστοιχα animations. Περισσότερα σχετικά με αυτή τη διαδικασία αναφέρονται αναλυτικά στην επόμενη παράγραφο.

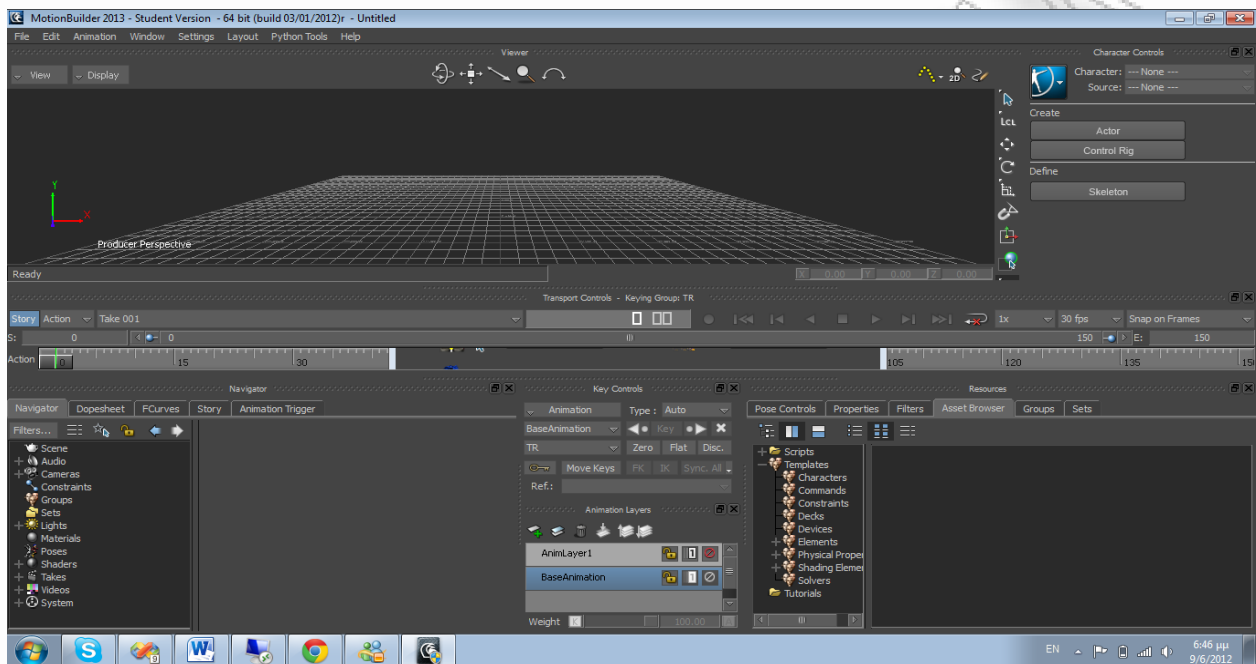
Εισαγωγή Animation και έλεγχος animation clips

Skeletal Animation

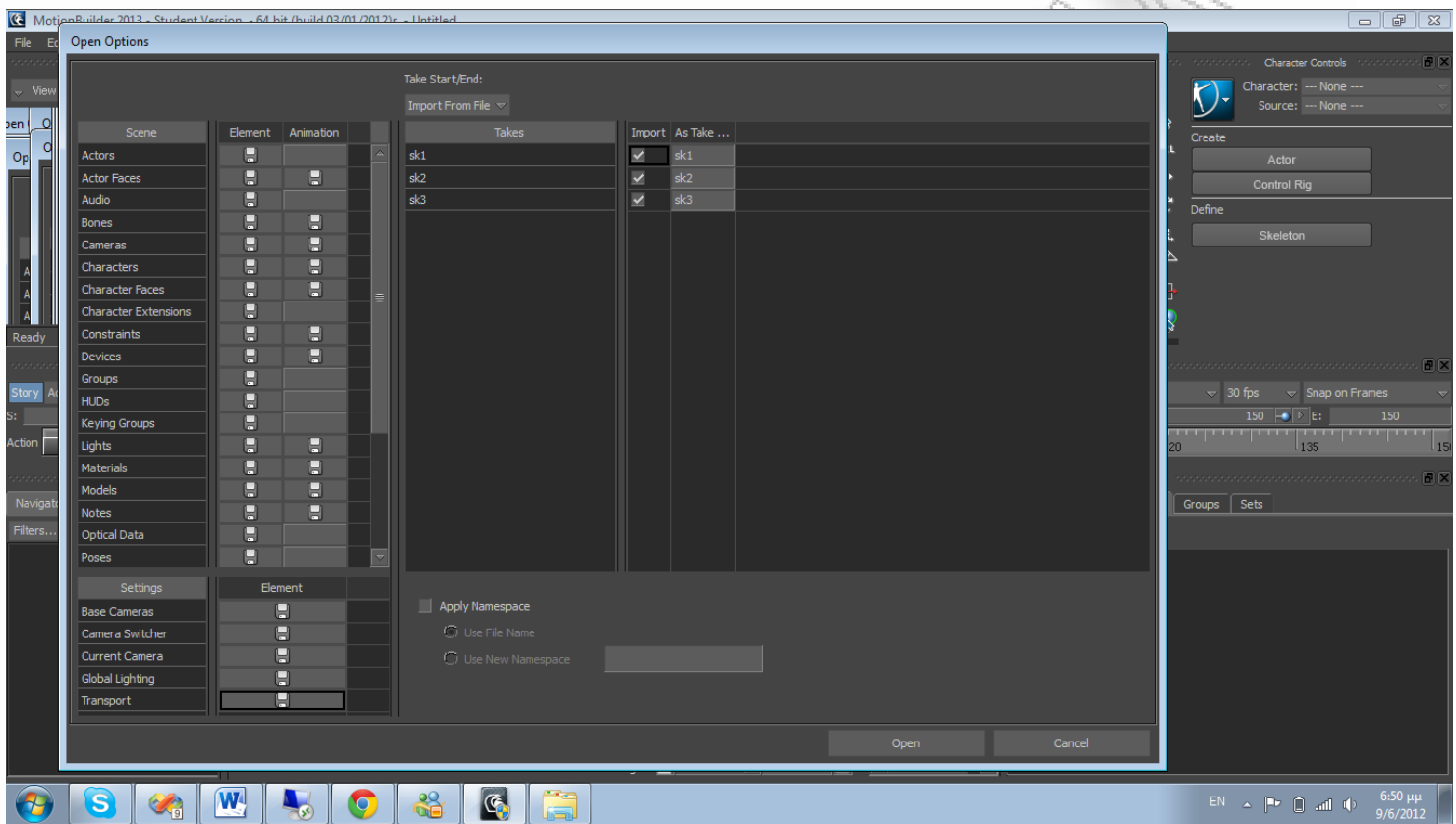
Η διαδικασία της δημιουργίας των animations δεν ήταν ιδιαίτερα απλή και αποδείχθηκε να κρύβει παγίδες που δεν ήταν εμφανείς στην αρχή. Αυτά που έπρεπε να κατασκευαστούν ήταν απλά και η χρήση τους έχει στόχο ακαδημαϊκό και όχι τον εντυπωσιασμό και την διασέδαση του χρήστη. Επιλέχθηκαν μοντέλα που είχαν δημιουργηθεί με βάση το 3D Studio Max και που μπορεί να εφαρμοστεί πάνω τους ένας σκελετός για animation. Το skeletal animation έχει αρκετά προτερήματα σε σχέση με το rigged (απευθείας πάνω στα πολύγωνα) καθώς δίνει τη δυνατότητα να μεταχειριστούμε το μοντέλο σαν ολοκληρωμένη οντότητα και όχι σε κάθε κομμάτι ξεχωριστά. Το αποτέλεσμα αυτής της προσέγγισης είναι ότι το παραγόμενο animation είναι αρκετά ομαλό και το σώμα κινείται φυσιολογικά χωρίς να φαίνονται σπασίματα και κοψίματα στην κίνηση. Επιπροσθέτως, υπάρχει η δυνατότητα να συνδεθούν στο skeletal μοντέλο οι λεγόμενοι actors που πρόκειται για σκελετους ανεξαρτήτως μοντέλου που εκτελούν συγκεκριμένα animation clips. Με τη σύνδεση ενός actor στο μοντέλο μας δεν είναι απαραίτητος ο σχεδιασμός ενός animation ξανά και ξανά. Η κίνηση του σκελετού του actor μεταφέρεται στον σκελετό του δικού μας μοντέλου και αυτό μπορεί να γίνει για κάθε μοντέλο με τον αντίστοιχο σκελετό.

Εισαγωγή και προετοιμασία μοντέλου

Το μοντέλο που επιλέχθηκε σε πρώτη φάση πρέπει να εισαχθεί στον Motion Builder και να μπορέσει το σύστημα να διαβάσει σωστά όλες τις πληροφορίες από το αρχείο FBX. Εάν αυτή η διαδικασία του import αποτύχει θα είναι σχεδόν απίθανο το μοντέλο αυτό να μπορεί να δεχθεί skeletal animation και να λειτουργήσει σωστά.

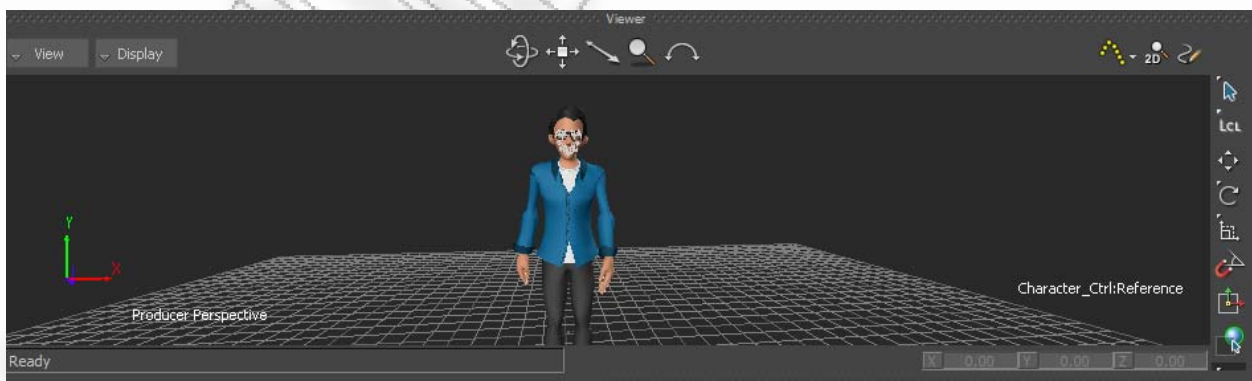


Η αρχική οθόνη του Motion Builder. Δεν υπάρχει κάποιο φορτωμένο μοντέλο.

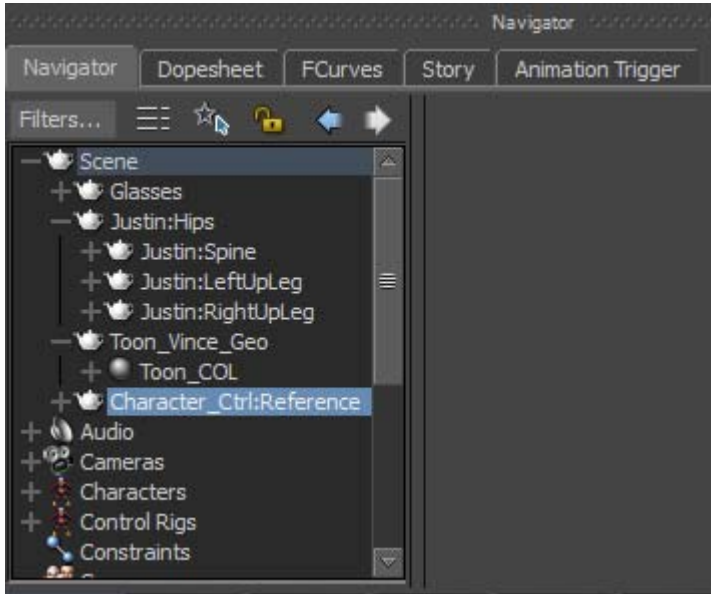


Ακολουθώντας το γνωστό file -> Open επιλέγουμε το αρχείο fbx που θέλουμε να επεξεργαστούμε. Στην οθόνη που εμφανίζεται μας δίνεται η δυνατότητα να εισάγουμε όποια από τα animations του αρχείου, αλλά και κάμερες, textures, φώτα και γενικά όλα τα χαρακτηριστικά του γραφήματος σκηνής που χρειαζόμαστε μέσα από το FBX. Στη δική μας περίπτωση τα επιλέγουμε όλα και συνεχίζουμε με το κουμπί Open.

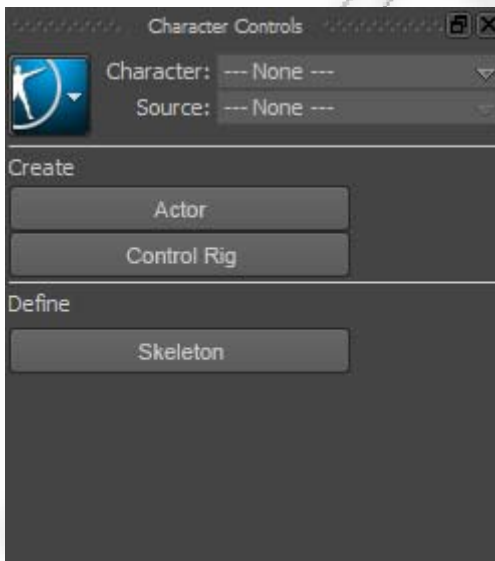
Στο παράθυρο viewer εμφανίζεται ο επιλεγμένος χαρακτήρας



Στο παράθυρο Navigator, εμφανίζονται τα μέρη του γραφήματος σκηνής. Παρατηρούμε ότι το μοντέλο έχει τέσσερις κεντρικούς κόμβους αυτοί είναι οι glasses, hips, toon_vince_geo και character_ctrl: reference. Το σώμα του μοντέλου που είναι και αυτό που μας ενδιαφέρει είναι κάτω από τον κόμβο hips.



Εφόσον σιγουρευτήκαμε ότι το μοντέλο έχει κάποιο κεντρικό κόμβο και ότι τα μέλη του είναι συνδεδεμένα ιεραρχικά, πρέπει να ορίσουμε και τον σκελετό που θα τα ελέγχει. Αυτή η διαδικασία γίνεται από το παράθυρο Character controls που φαίνεται και παρακάτω

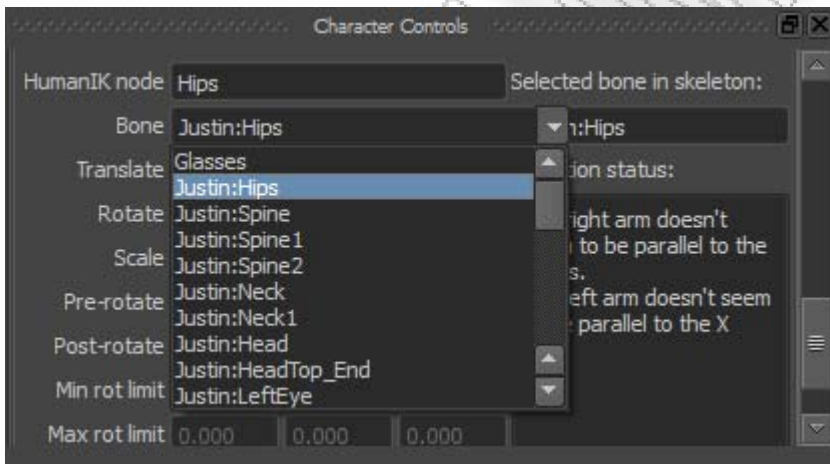


Επιλέγοντας το define skeleton το πρόγραμμα μας ενημερώνει ότι δεν υπάρχει σκελετός και ότι πρέπει να τον δημιουργήσουμε. Επιλέγοντας τη συνέχεια της διαδικασίας εμφανίζεται το παράθυρο για τον ορισμό του σκελετού. Σε αυτό το παράθυρο θα γίνει η σύνδεση των κόμβων του σκελετού με τα αντίστοιχα 3D

μέρη του σώματος του χαρακτήρα.



Για κάθε κόκκαλο του σκελετού επιλέγουμε και τον αντίστοιχο κόμβο όπως φαίνεται παρακάτω,

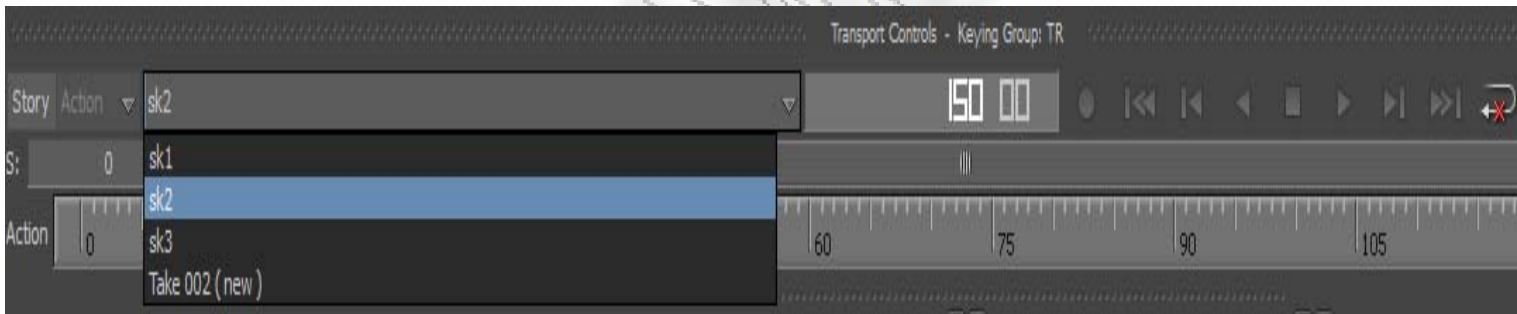


Με την ολοκλήρωση της αντιστοίχισης το πρόγραμμα μας ειδοποιεί ότι ο σκελετός έχει οριστεί σωστά και πλέον μπορούμε να χειριστούμε το μοντέλο με skeletal animation. Στο παράθυρο Character controls θα εμφανίζεται ο σκελετός όπως στην επόμενη εικόνα



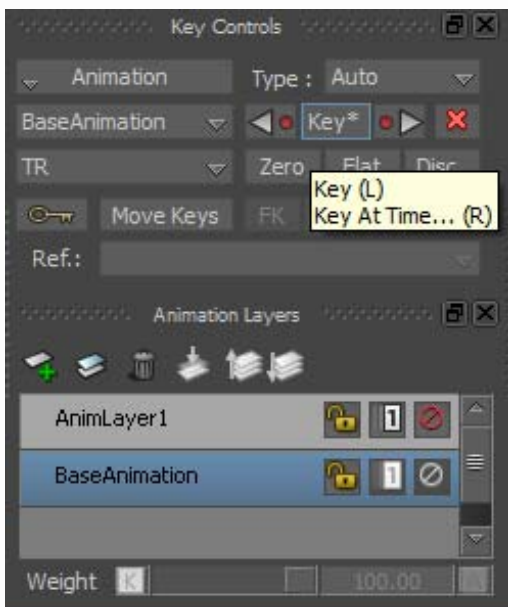
Δημιουργία animations

Τώρα που ο χαρακτήρας μας έχει ολοκληρωθεί μπορούμε να ξεκινήσουμε να παράγουμε animations. Στο παράθυρο Transport Controls – Keying Group επιλέγουμε την ανάπτυξη του drop down box που κρατά τα actions του χαρακτήρα.

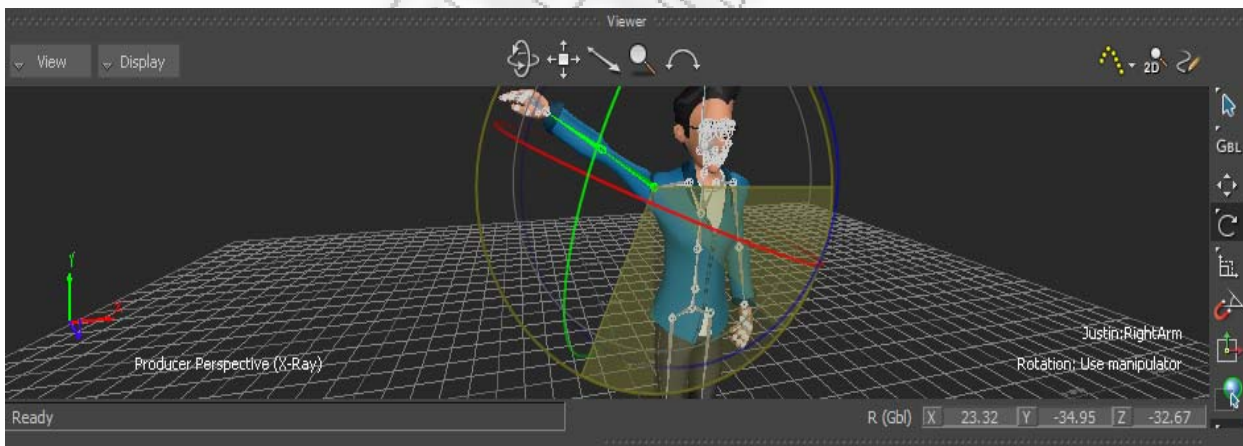


Στο συγκεκριμένο φαίνονται όλα τα ήδη υπάρχοντα animations αλλά και μια ακόμα επιλογή με το όνομα Take 002 (new). Αυτή η επιλογή θα μας δώσει τη δυνατότητα να φτιάξουμε ένα νέο animation σε ένα τελείως κενό timeline.

Η διαδικασία της δημιουργίας ενός νέου animation χωρίζεται σε δυο φάσεις για κάθε κίνηση που θέλουμε να βλέπουμε στι κλιπ μας. Στην πρώτη φάση θέτουμε το σώμα του μοντέλου εκεί που θέλουμε να ξεκινήσει η κίνηση, δηλαδή στην αρχική θέση. Η αρχική μας θέση για το παράδειγμα θα είναι και η αρχική θέση του μοντέλου, οπότε δεν θα πρέπει αυτή τη φορά να μετακινηθεί κάποιος κόμβος. Μόλις η πόζα είναι έτοιμη, επιλέγουμε να θέσουμε στον χρόνο εκκίνησης ένα key frame. Στην συγκεκριμένη περίπτωση ο χρόνος εκκίνησης είναι το 0.

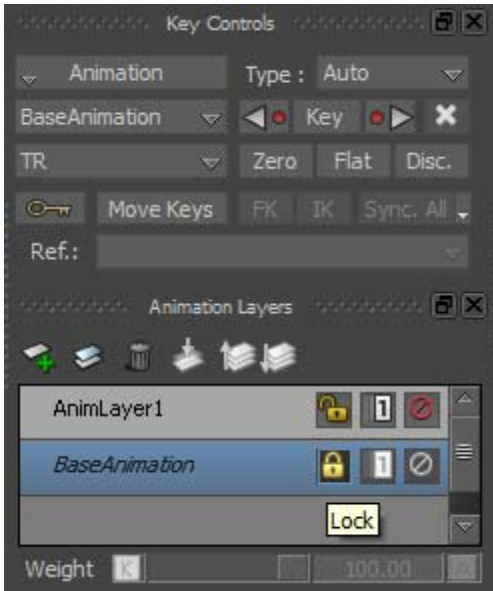


Στη συνέχεια επιλέγουμε στο μοντέλο τους κόμβους ή τον κόμβο που θέλουμε να έχει κίνηση και εμεταβάλλουμε τη θέση του. Επιλέγουμε την τελική θέση του κόμβου και τον χρόνο που θέλουμε αυτός ο κόμβος να βρίσκεται στη θέση αυτή. Στο παράθυρο key controls θέτουμε ένα νέο key frame που σηματοδοτεί και το τέλος της κίνησης.

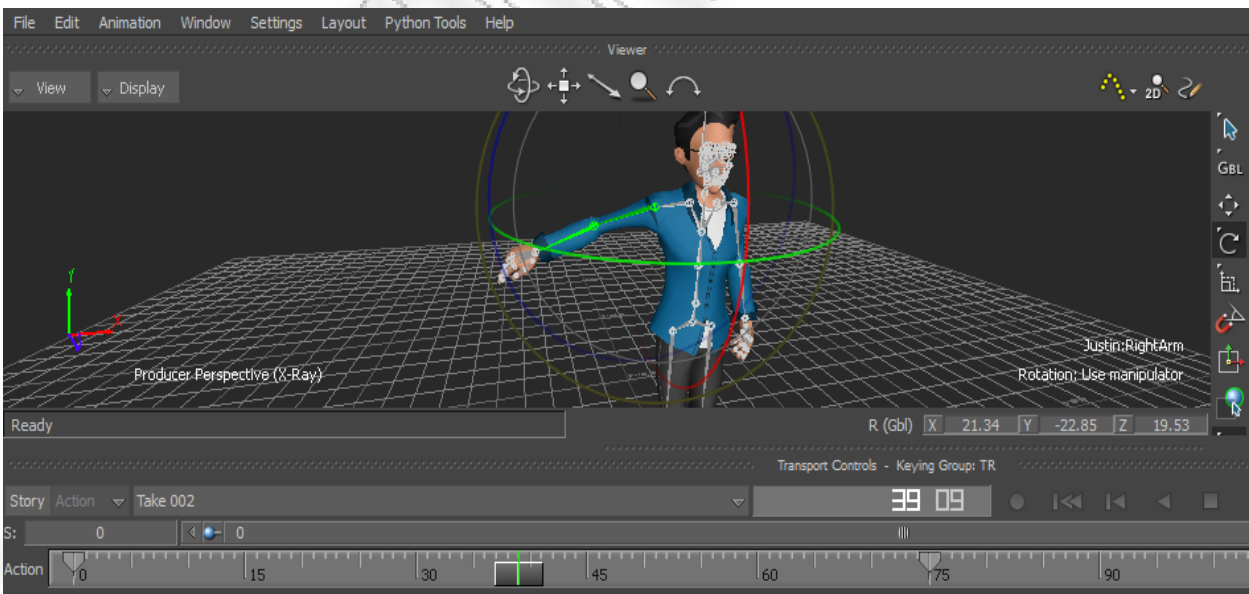


Δοκιμές και Export μοντέλου

Για δοκιμάσουμε το νέο animation πρέπει να κλειδωθεί το animation layer της κίνησης που κάνουμε edit.

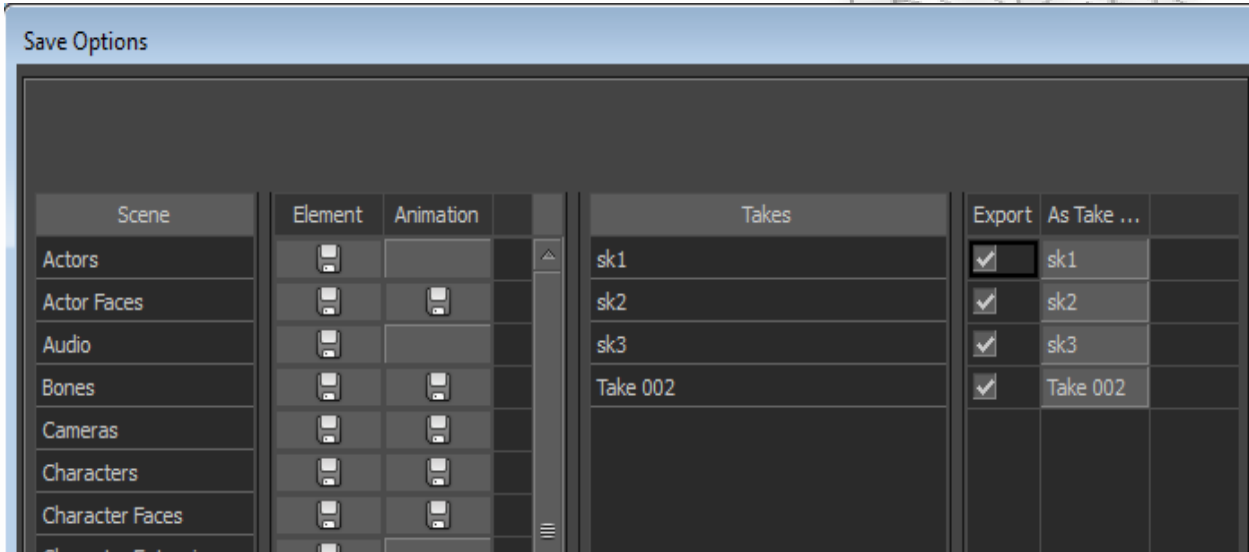


Στη συνέχεια θέτουμε τον κέρσορα του timeline στην τιμή 0 και πατάμε την επιλογή αναπαραγωγής. Βλέπουμε το χέρι του μοντέλου να εκτελεί την κίνηση από την αρχική θέση προς την τελική.



Για να εξάγουμε το animation clip μαζί με τα υπόλοιπα clips του αρχείου, επιλέγουμε να σώσουμε το αρχείο και εμφανίζεται το παράθυρο επιλογών σύνθεσης του FBX. Στο παράθυρο επιλογών εμφανίζονται όλα τα animation clips ως Takes. Στη λίστα με τα διαθέσιμα Takes υπάρχει και το νέο animation που

δημιουργήθηκε για το παράδειγμα με το όνομα Take 002. Στη στήλη As Take... επιλέγουμε μια ονομασία για το clip. Η ονομασία αυτή είναι σημαντική γιατί μέσα από το πρόγραμμά μας καλούμε το εκάστοτε animation σύμφωνα με το όνομα αυτό.



Το αρχείο FBX που δημιουργείται είναι σαν ένα αρχείο .zip. Όταν καλείται από το OpenSceneGraph γίνεται το άνοιγμά του και διαβάζονται τα μέρη του. Το άνοιγμα του αρχείου είναι διαδικασία που καθυστερεί την εκτέλεση του προγράμματος και θα ήταν καλύτερο να μπορεί να γίνει κάποια στιγμή κατά την έναρξή του. Παρόλο που οι πληροφορίες που είναι αποθηκευμένες στο αρχείο είναι πολλές και ίσως πλεονάζουν, το format αυτό έχει τις δυνατότητες που είναι απαραίτητες για την εφαρμογή.

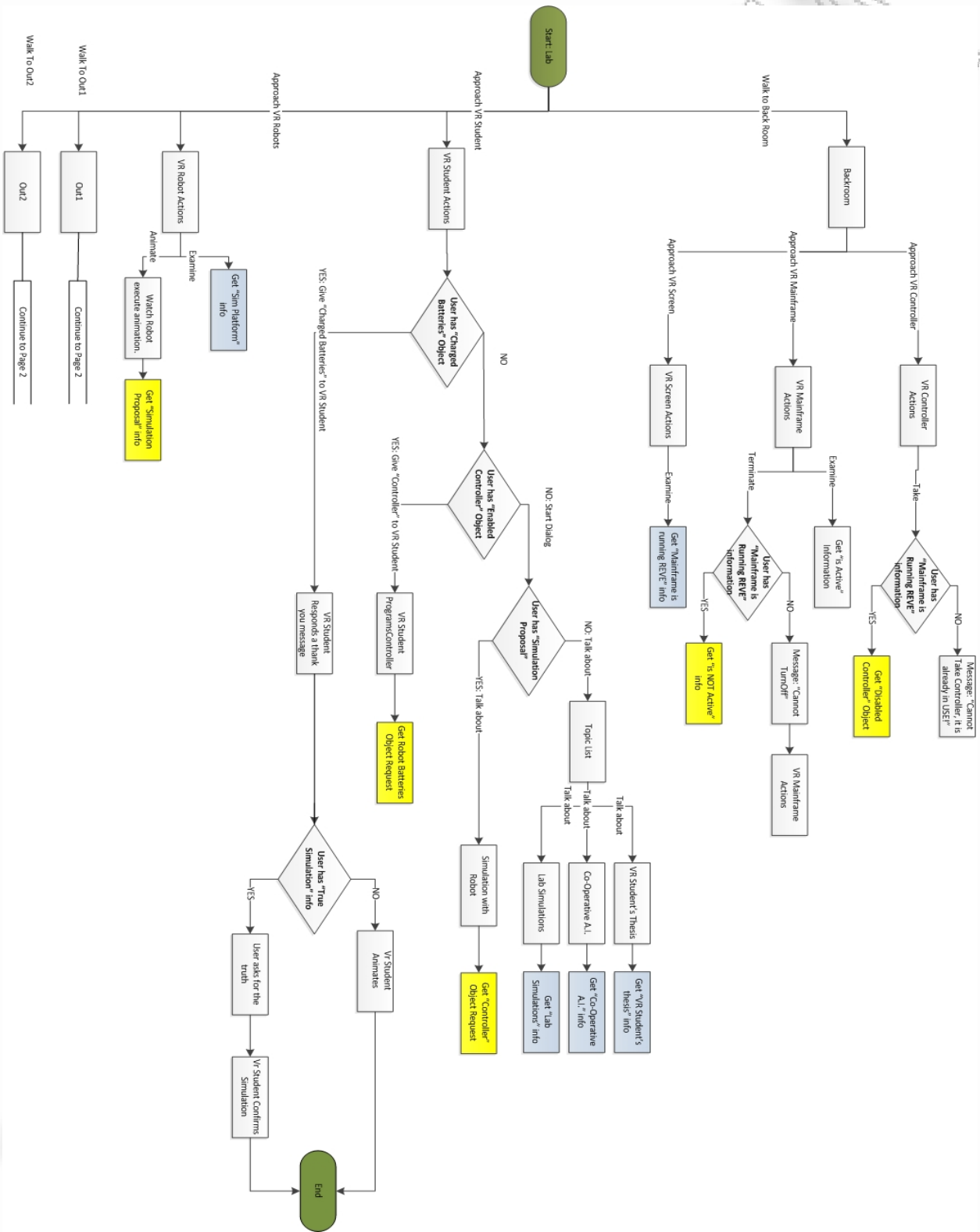
5: Επαυξημένη Εξιστόρηση

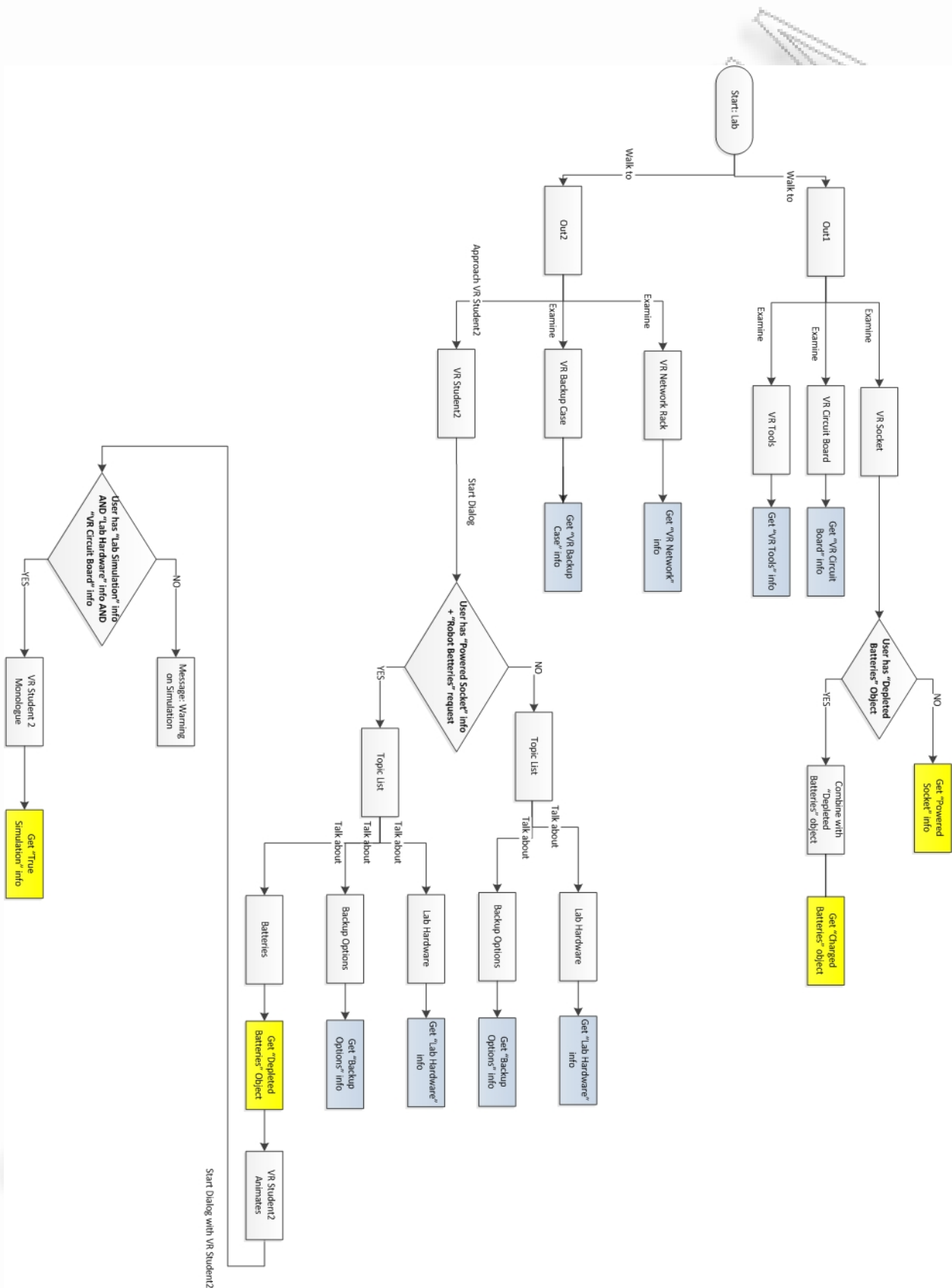
Ανάλυση σεναρίου

Για να φανούν τα όσα τεχνικά αναφέρθηκαν παραπάνω αλλά και για να δείξουμε το πως είναι στην πραγματικότητα μια ιστορία μεικτής πραγματικότητας, δημιουργήθηκε ένα μικρό ενδεικτικό σενάριο. Μέσα σε αυτό, ο χρήστης παίρνει τη θέση ενός φοιτητή του Πανεπιστημίου και καλείται να δράσει σύμφωνα με τα όσα συναντά στην πορεία του.

Η ιστορία εκτυλίσσεται στο υποτιθέμενο εργαστήριο "Τεχνολογίας Γνώσης" του Πανεπιστημίου και συμπεριλαμβάνει τέσσερις τοποθεσίες. Η κεντρική τοποθεσία είναι το εσωτερικό του εργαστηρίου και οι άλλες τρεις είναι το πίσω δωμάτιο του εργαστηρίου, ο χώρος ακριβώς έξω από το εργαστήριο και τέλος ένας ακόμα χώρος έξω από το εργαστήριο αλλά λίγο μακρύτερα. Στο διάγραμμα του σεναρίου οι τοποθεσίες αναφέρονται ως Lab, backroom, Out1 και Out2.

Ακολουθεί το flow chart του σεναρίου που είναι χωρισμένο σε δυο διαγράμματα. Το πρώτο περιλαμβάνει τις τοποθεσίες Lab και BackRoom ενώ το δεύτερο τις τοποθεσίες Out1 και Out2.





Στη συνέχεια περιγράφονται οι κινήσεις του χρήστη όπως φαίνονται και στο διάγραμμα ροής του σεναρίου.

Τοποθεσία "Lab", φάση 1η

Ο χρήστης ξεκινά από την τοποθεσία Lab και έρχεται σε επαφή με την εικονική φοιτήτρια Αθηνά και με μια πλατφόρμα που πάνω της υπάρχει ένα εικονικό ρομπότ. Ο χρήστης παρατηρεί το ρομπότ και μιλά με την εικονική φοιτήτρια. Μαθαίνοντας πληροφορίες για την εργασία της Αθηνάς αλλά και εξετάζοντας το ρομπότ, του δίνεται η δυνατότητα να προτείνει στην Αθηνά να δοκιμάσει την εργασία της πάνω στο ρομπότ. Αυτή από την πλευρά της δέχεται την πρόταση του χρήστη και ζητά το αντικείμενο controller που θα την βοηθήσει να προγραμματίσει το ρομπότ. Αυτό το αντικείμενο, σύμφωνα με το σενάριο, είναι μια προγραμματιζόμενη συσκευή που μπορεί να φιλοξενήσει αλγορίθμους τεχνητής νοημοσύνης. Ο χρήστης πρέπει να ψάξει και να βρεί το αντικείμενο.

Τοποθεσία "Backroom"

Συνεχίζοντας την εξερεύνηση του περιβάλλοντος, ο χρήστης εισέρχεται στην τοποθεσία "Backroom" και παρατηρεί ένα μεγάλο υπολογιστικό σύστημα σε λειτουργία, μια οθόνη και το αντικείμενο controller που ζήτησε η Αθηνά. Με μια πρώτη κίνηση αφαίρεσης του αντικειμένου controller από τη θέση του, ο χρήστης μαθαίνει ότι δεν μπορεί να το κάνει γιατί ακόμα το mainframe που τον χρησιμοποιεί είναι εν εργασία. Προσπαθώντας ο χρήστης να κλείσει το σύστημα έρχεται αντιμέτωπος με ένα μήνυμα ότι δεν γνωρίζει αν αυτή η πράξη είναι ασφαλής και ότι πρέπει πρώτα να σιγουρευτεί για το αποτέλεσμα του κλεισίματος του mainframe καθώς ο Καθηγητής μπορεί να διεξάγει κάποια σημαντική εξομοίωση. Κοιτάζοντας και εξερευνώντας την εικονική οθόνη, λαμβάνει τη γνώση ότι το υπολογιστικό σύστημα εκτελεί ένα σενάριο για το REVE worlds. Αυτό σημαίνει για τον χρήστη ότι πλέον είναι ασφαλές να τερματιστεί το σύστημα και ότι δεν θα προκληθεί ζημιά σε κάποιο πείραμα. Εν συνέχεια, ο controller δεν θα είναι σε λειτουργία και θα μπορεί αποδεσμευτεί ώστε χρήστης να τον μεταφέρει προς την Αθηνά.

Τοποθεσία "Lab", φάση 2η

Επιστρέφοντας στον κεντρικό χώρο του εργαστηρίου και αφήνοντας κοντά στην Αθηνά τον controller, η εικονική φοιτήτρια ξεκινά να τον προγραμματίζει και ζητά πλέον μπαταρίες, έτσι ώστε να μπορεί να κινηθεί το ρομπότ. Ο χρήστης πρέπει να ψάξει και να φέρει πίσω τις μπαταρίες.

Τοποθεσία "Out1", φάση 1η

Στην πρώτη εξωτερική τοποθεσία, ο χρήστης βρίσκει τρία αντικείμενα. Μια πρίζα, μια πλακέτα και ένα σετ εργαλείων. Εξετάζοντάς τα ένα ένα, ο χρήστης μαθαίνει ότι η πρίζα έχει ρεύμα και ότι μπορεί να χρησιμοποιηθεί. Επίσης από τα σχόλια του χρήστη, καταλαβαίνουμε ότι είναι περιέργο που τα εργαλεία και η πλακέτα βρίσκονται εγκαταλελημένα έξω από το εργαστήριο. Στην παρούσα κατάσταση δεν υπάρχει κάποια άλλη κίνηση για τον χρήστη στο χώρο και πρέπει αυτός να κινηθεί προς την τελευταία τοποθεσία.

Τοποθεσία "Out2"

Στην τελευταία τοποθεσία ο χρήστης διακρίνει ένα κράνος εικονικής πραγματικότητας, μια συσκευή backup για μεγάλο όγκο δεδομένων και τον δεύτερο εικονικό φοιτητή, τον Θάνο. Εξετάζοντας τα δυο αντικείμενα η περιέργεια στα σχόλια του χρήστη, σχετικά με το σκοπό της παρουσίας του εξοπλισμού χωρίς κάποια φύλαξη, είναι ακόμα πιο έντονη και εμφανής. Συζητώντας με τον εικονικό φοιτητή, ο χρήστης αναφέρει ότι χρειάζεται την μπαταρία για τα ρομπότ και κατ' επέκταση για την εργασία της Αθηνάς. Ο Θάνος δίνει στον χρήστη την μπαταρία αλλά τον προειδοποιεί ότι πρέπει να την φορτίσει. Μετά από την καθοριστική συζήτηση και εφόσον ο χρήστης ήταν διερευνητικός, εάν έχει μαζέψει τις πληροφορίες από τον εξοπλισμό και τις εξομοιώσεις του εργαστηρίου, ο Θάνος, παραδεχόμενος ότι ο χρήστης έχει μαζέψει ήδη αρκετές πληροφορίες, του αποκαλύπτει την αλήθεια. Τα λεγόμενα του εικονικού φοιτητή μας πληροφορούν ότι εμείς οι ίδιοι είμαστε κομμάτι μιας εξομοίωσης. Για την συνέχεια του σεναρίου ο χρήστης πρέπει να

πλοηγηθεί προς τον πρώτο εξωτερικό χώρο.

Τοποθεσία "Out1", φάση 2η

Ο χρήστης έχοντας την μπαταρία την τοποθετεί κοντά στην πρίζα και αυτή πλέον φορτίζεται. Η φορτισμένη, πλέον, μπαταρία πρέπει να επιστραφεί στην Αθηνά για να συνεχιστεί το σενάριο.

Τοποθεσία "Lab", φάση 3η

Στο τέλος του σεναρίου ο χρήστης επιστρέφει στην αρχική τοποθεσία και πρέπει να παραδώσει την φορτισμένη μπαταρία στην Αθηνά. Μόλις η μπαταρία βρεθεί κοντά στην Αθηνά, αυτή την παίρνει και την τοποθετεί στο ρομπότ. Το ρομπότ πλέον είναι προγραμματισμένο και με την φορτισμένη μπαταρία ξεκινά να κινείται. Στην περίπτωση που ο χρήστης έχει πληροφορηθεί από τον φοιτητή Θάνο για την εξομοίωση, η Αθηνά παραδέχεται και από μέρους της πλέον την υπόθεση και αποκαλύπτει και αυτή με τη σειρά της, ότι ο χρήστης είναι μέρος μιας εξομοίωσης και προτρέπει τον χρήστη να συνεχίσει σε μια ακόμα ιστορία...

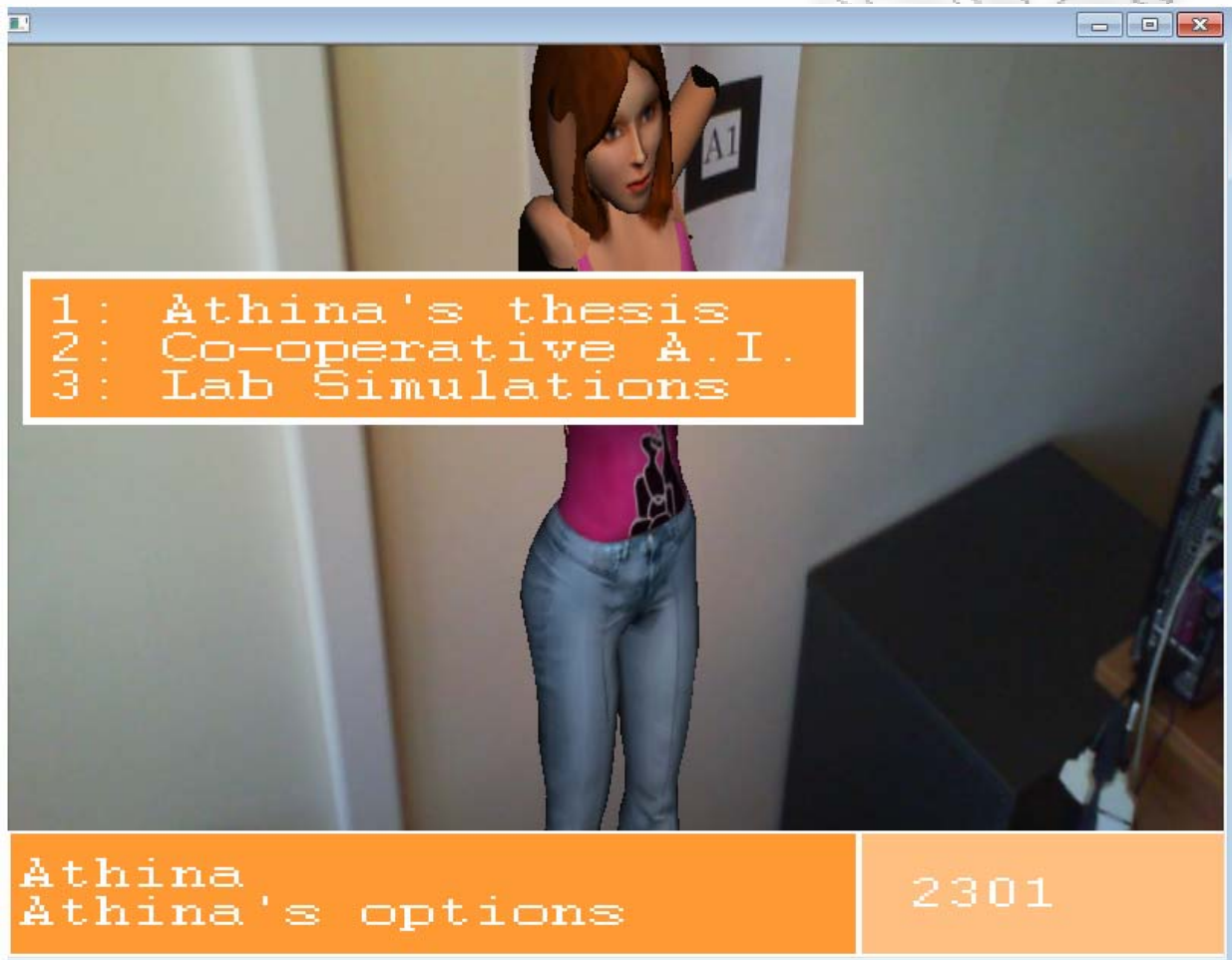
Εξέλιξη ιστορίας, Βήμα – Βήμα

Σε αυτό το σημείο θα γίνει η περιγραφή της ιστορίας βήμα προς βήμα με τη βοήθεια των οθονών που τραβήχτηκαν κατά την εκτέλεση του σεναρίου. Οι markers τοποθετήθηκαν σε τέσσερις διαφορετικές τοποθεσίες για να γίνεται αντιληπτή και η μετακίνηση στον χώρο.

Στην τοποθεσία Lab συναντάμε την Αθηνά,



Πλησιάζοντας εμφανίζονται και οι επιλογές που έχουμε σε σχέση με την εικονική φοιτήτρια.



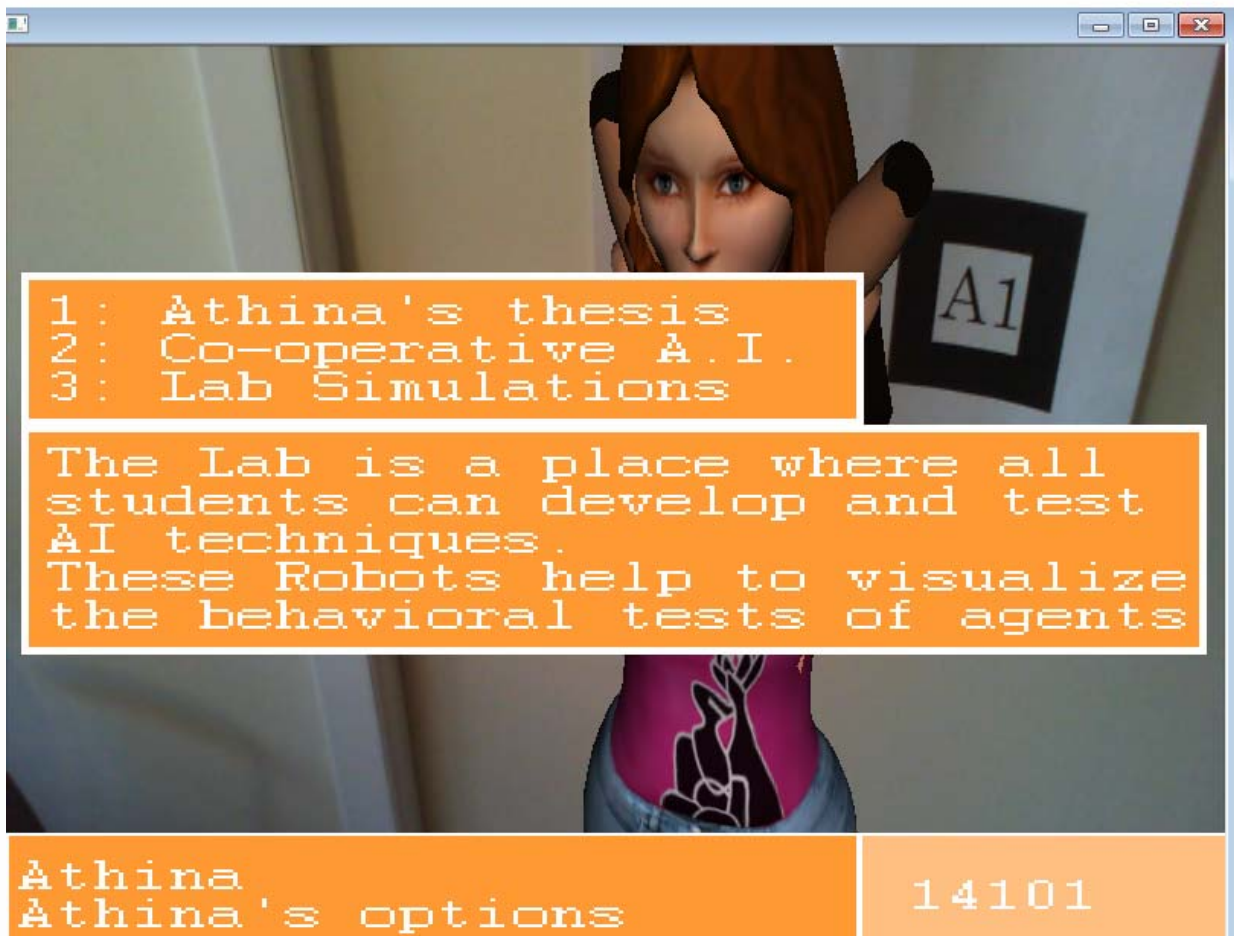
Την 1^η φορά που ερχόμαστε σε επαφή μαζί της, εκτελεί ένα animation χαιρετισμού, όπως φαίνεται και στην εικόνα, με τα χέρια ψηλά. Σχετικά με την εργασία της μας πληροφορεί ότι το θέμα είναι σχετικό με την δημιουργία πλάνων ενεργειών για συμπεριφορές τεχνητής νοημοσύνης



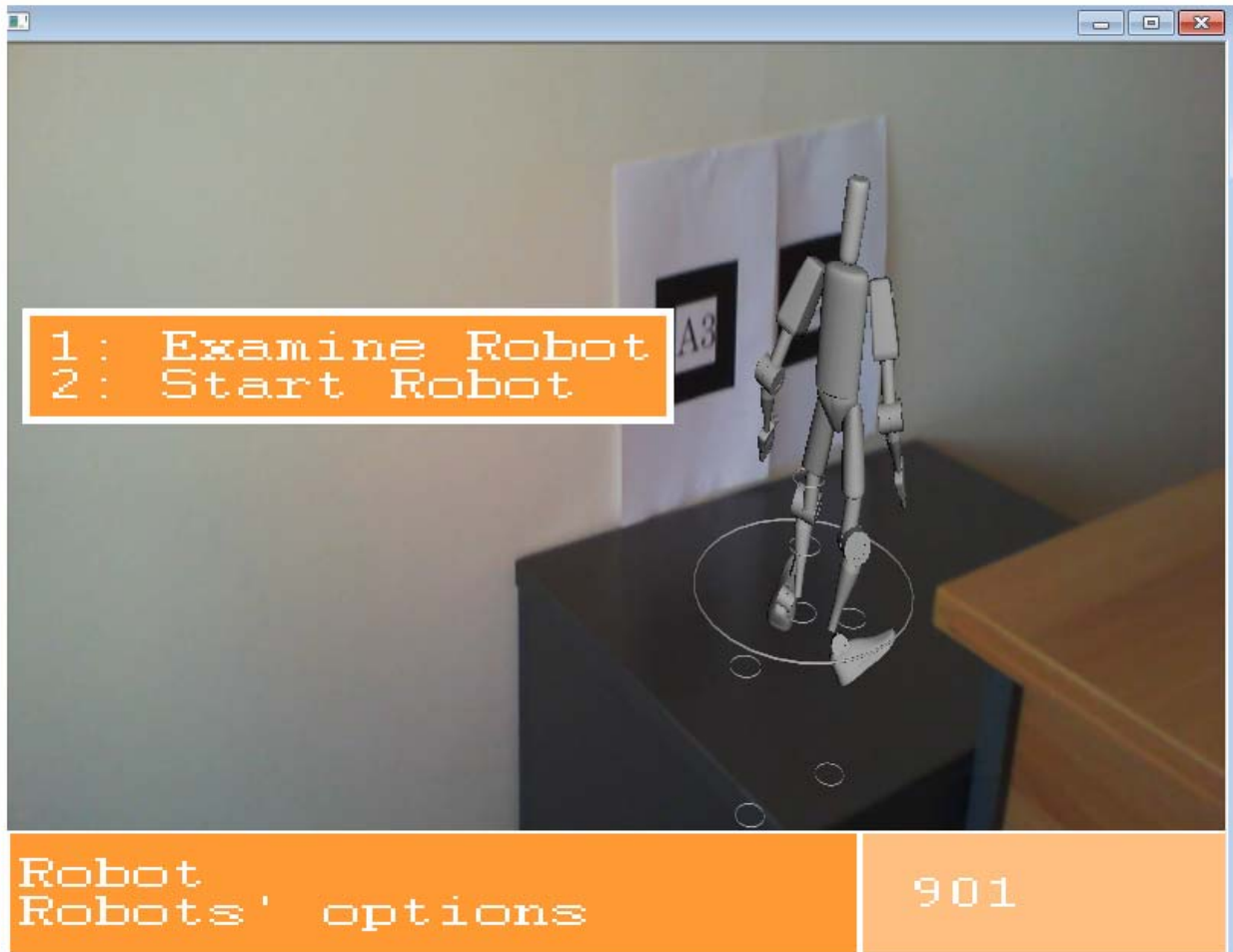
Με την επόμενη επιλογή ζητάμε να μας εξηγήσει μερικά πράγματα παραπάνω για την συνεργατική τεχνητή νοημοσύνη. Η Αθηνά μας εξηγεί ότι πρόκειται για πράκτορες τεχνητής νοημοσύνης που λειτουργούν σε ομάδες και συνεργάζονται ώστε να επιτύχουν έναν κοινό σκοπό.



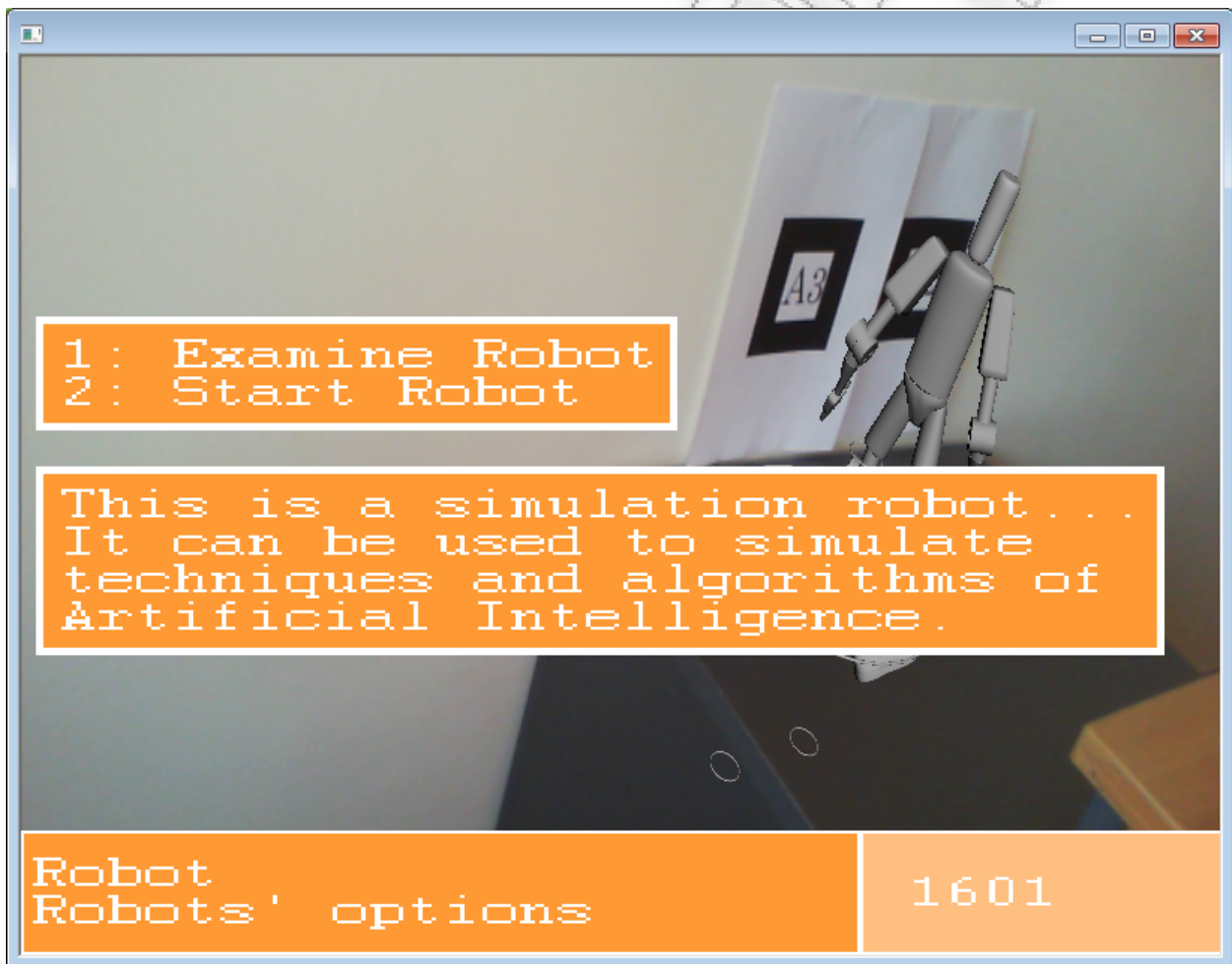
Με την τελευταία επιλογή ενημερώνομαστε για τα πειράματα/ εξομοιώσεις που γίνονται σε αυτό το εργαστήριο. Η Αθηνά δίνει μας δίνει ένα στοιχείο, λέγοντας πως το εργαστήριο μπορεί να χρησιμοποιηθεί από όλους τους φοιτητές που θέλουν να αναπτύξουν τα δικά τους συστήματα ή να δοκιμάσουν τεχνικές τεχνητής νοημάσυνας που οι ίδιοι έχουν αναπτύξει. Στη συνέχεια με ενημερώνει ότι με την χρήση του robot που υπάρχει δίπλα της οι φοιτητές μπορούν να οπτικοποιήσουν τους αλγόριθμους που κατασκευάζουν, στην πράξη.



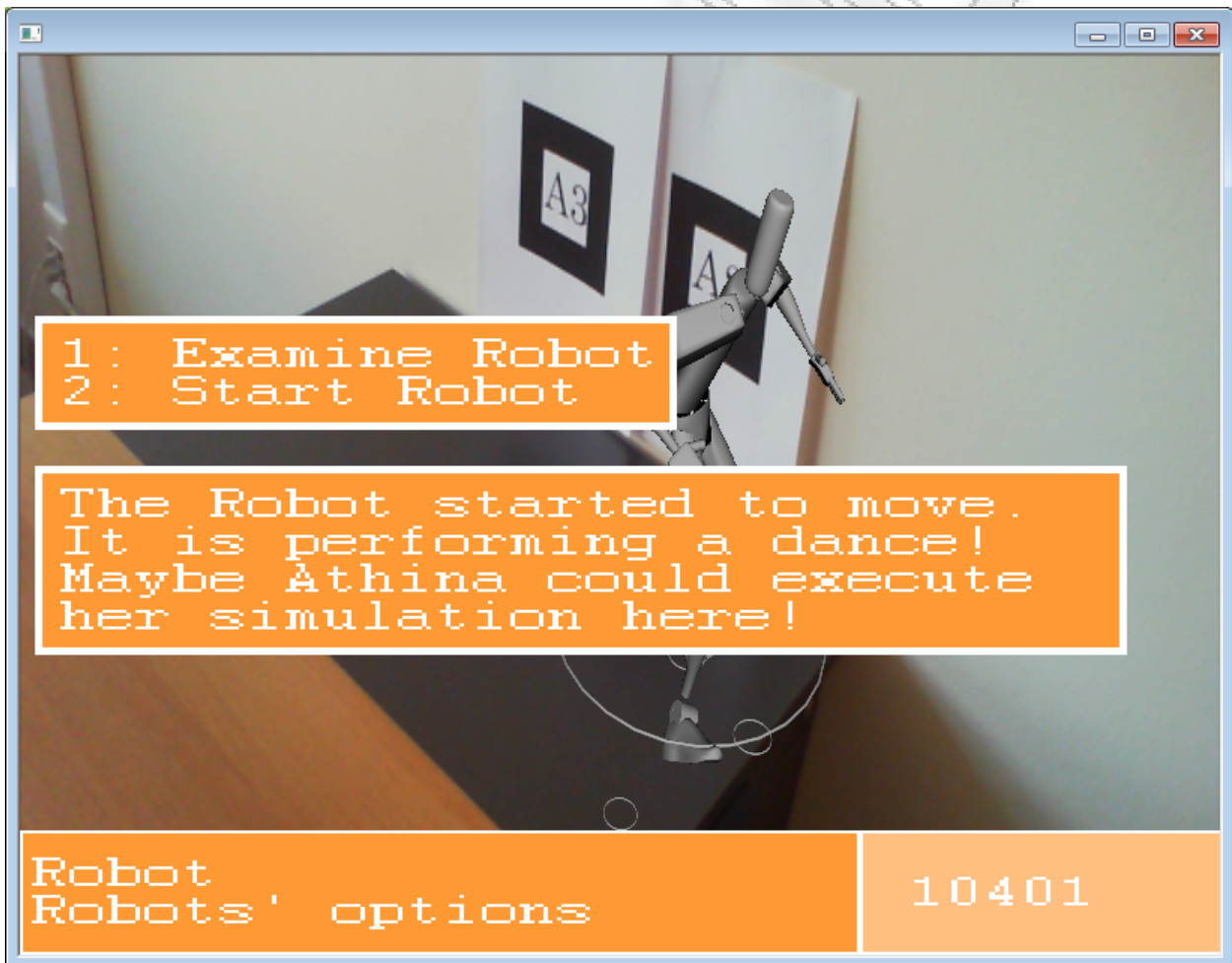
Κινούμαστε προς το robot που μας υπέδειξε η Αθηνά και βλέπουμε το παρακάτω,



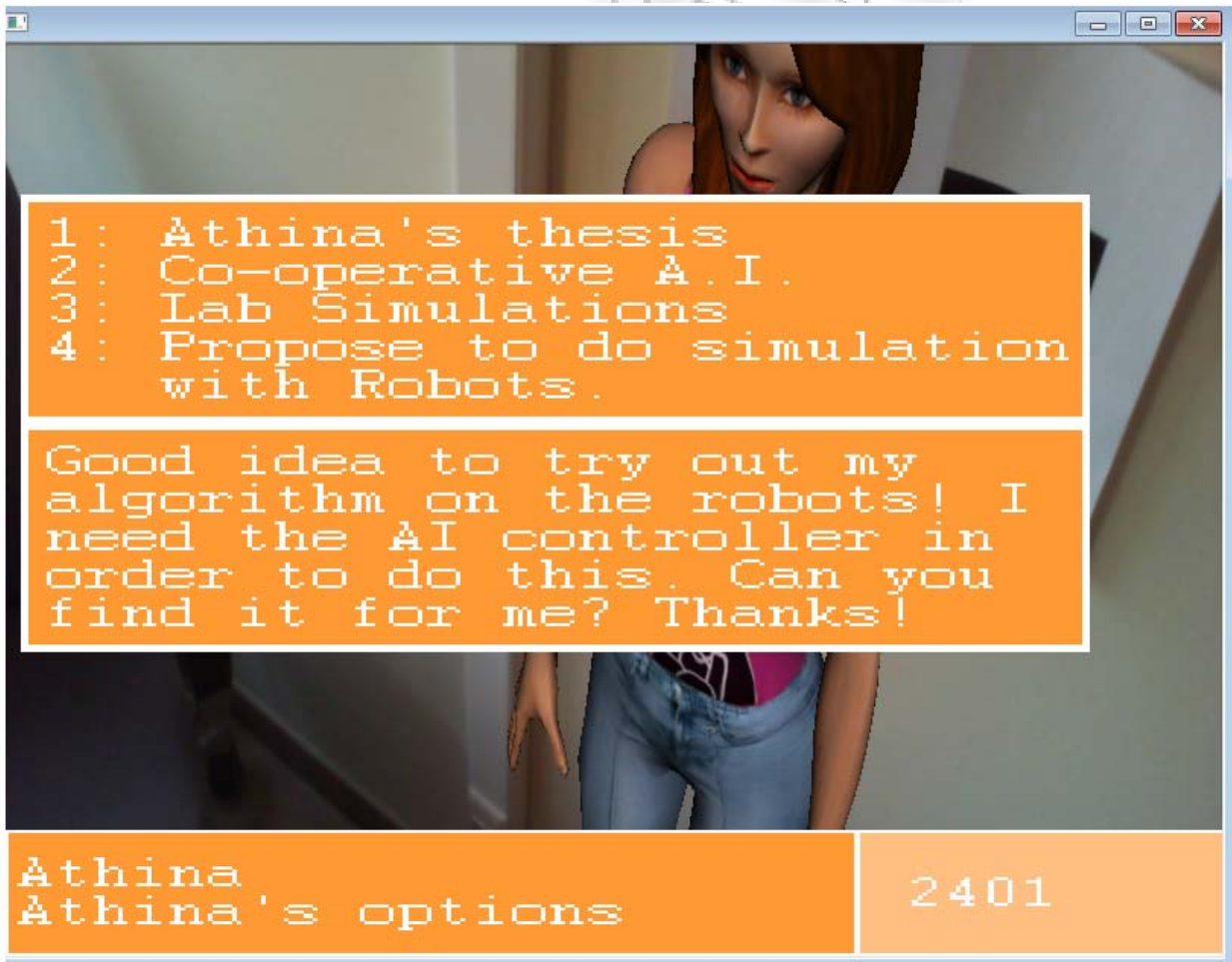
Το ρομπότ είναι δίπλα της και παραμένει ακίνητο. Αν εξετάσουμε το ρομπότ αυτό, επιβεβαιώνουμε ότι είναι ένα εργαλείο για εξομοιώσεις αλγορίθμων, όπως φαίνεται και στο ακόλουθο μήνυμα που εμφανίζεται.



Προχωρώντας στην επόμενη επιλογή που είναι η εκκίνηση του ρομπότ, το βλέπουμε να κινείται εκτελώντας μια χορευτική φιγούρα και σκεφτόμαστε ότι ίσως θα μπορούσαμε να προτείνουμε στην Αθηνά να δοκιμάσει την εργασία της πάνω σε αυτό το ρομπότ για καλύτερα αποτελέσματα. Μετά από αυτή την επιλογή μας στρεφόμαστε ξανά προς την Αθηνά για λίγη ακόμα συζήτηση.

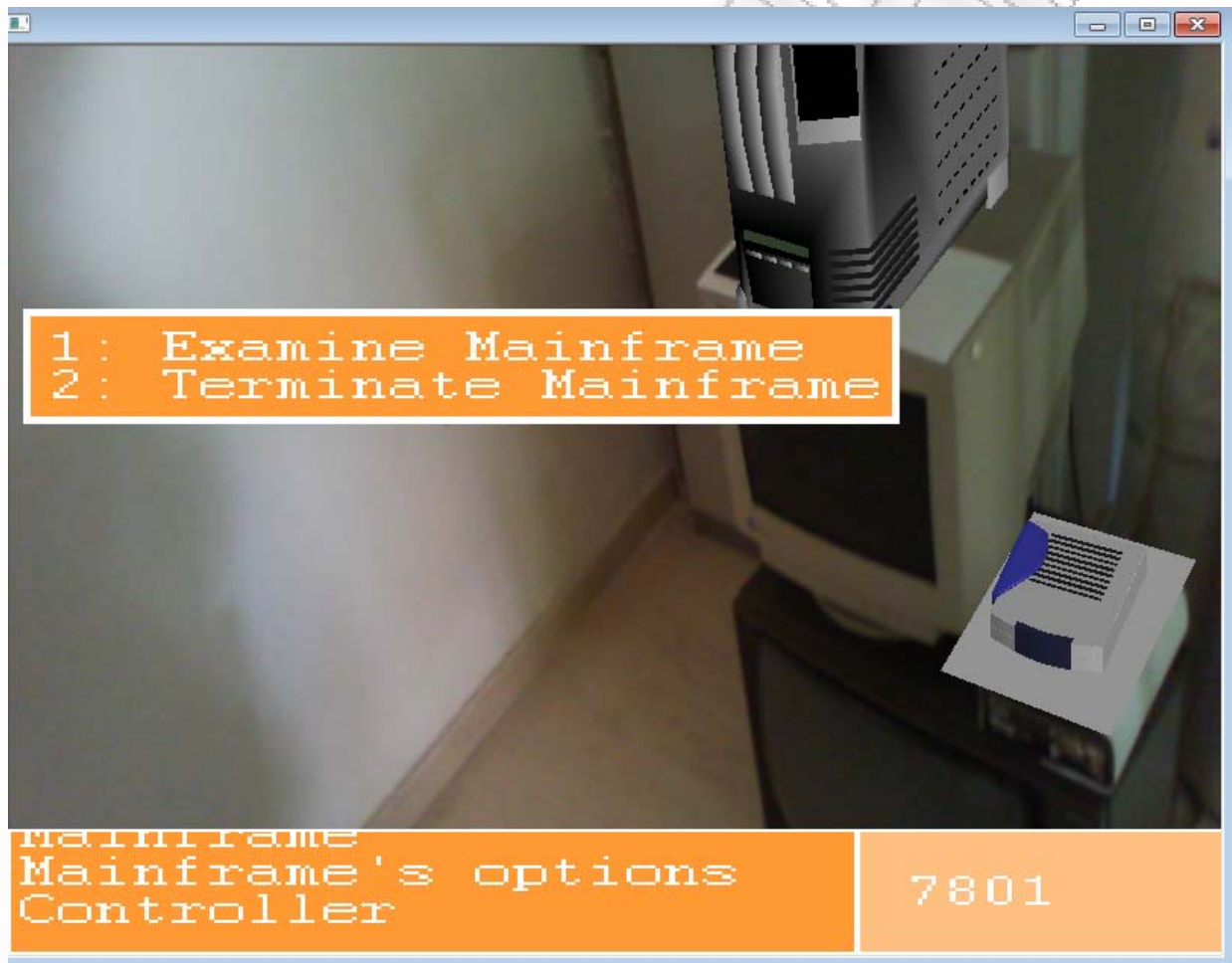


Αφού το ρομπότ ολοκληρώσει την κίνησή του, πηγαίνουμε προς την Αθηνά και διαπιστώνουμε ότι πλέον υπάρχει ακόμα μια επιλογή στις δράσεις που μπορούμε να επιλέξουμε. Πατώντας το νούμερο 4 στο πληκτρολόγιο, προτείνουμε στην Αθηνά να χρησιμοποιήσει το ρομπότ του εργαστηρίου για την εργασία της και αυτή με τη σειρά της απαντά πως την αποδέχεται. Για να προχωρήσει όμως στην υλοποίηση χρειάζεται το αντικείμενο controller που είναι μια προγραμματιζόμενη συσκευή. Μέσα σε αυτόν τον controller θα περάσει τον αλγόριθμό της και στη συνέχεια θα τον συνδέσει με το ρομπότ. Έτσι, μας ζητάει να την βοηθήσουμε και συνεπώς αποκτούμε ένα νέο σκοπό στην ιστορία.

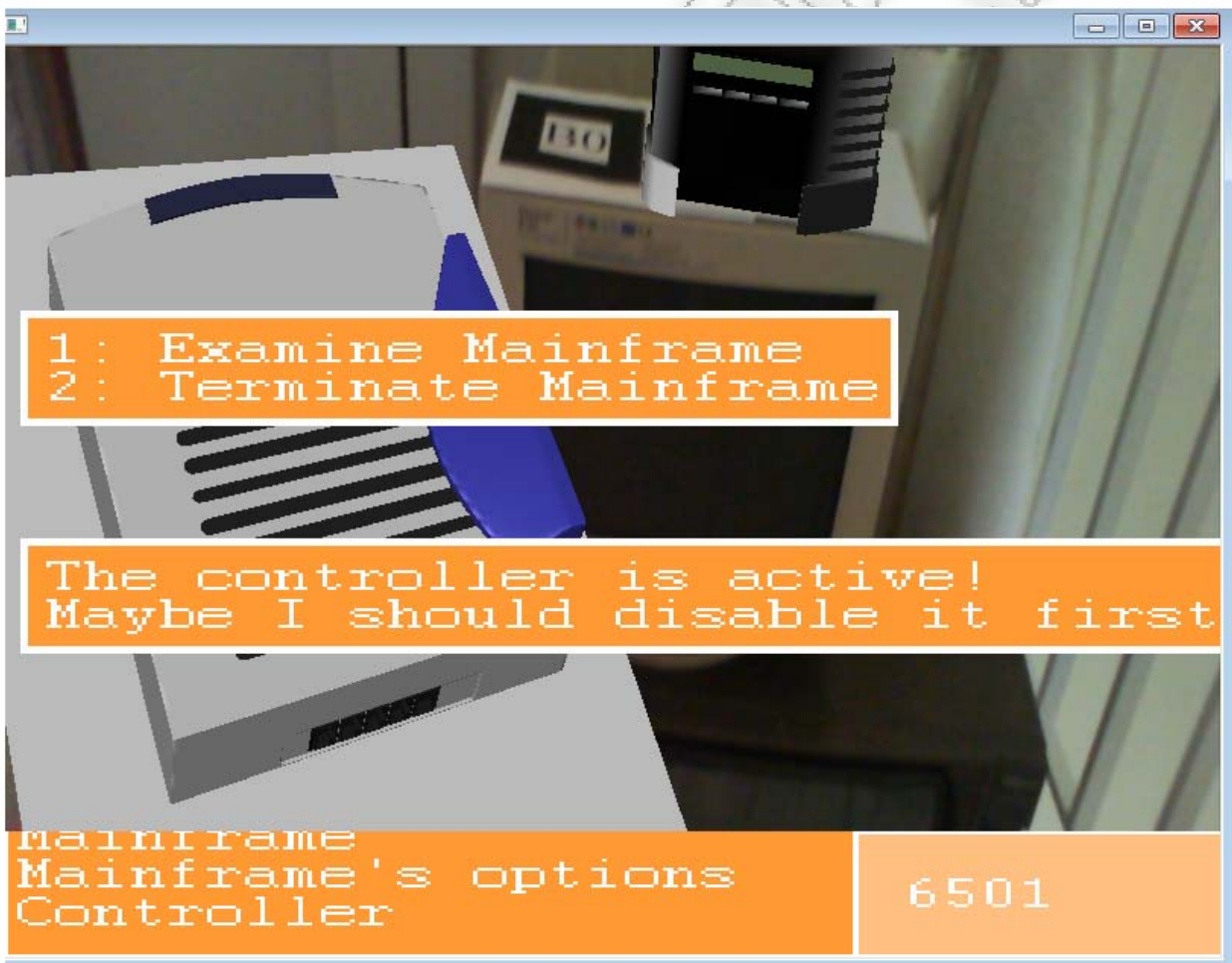


Όπως φαίνεται δεν μπορούμε να προχωρήσουμε περαιτέρω χωρίς την απόκτηση του controller. Για τον λόγο αυτό συνεχίζουμε την περιήγησή μας στον επόμενο χώρο που είναι το πίσω δωμάτιο του εργαστηρίου.

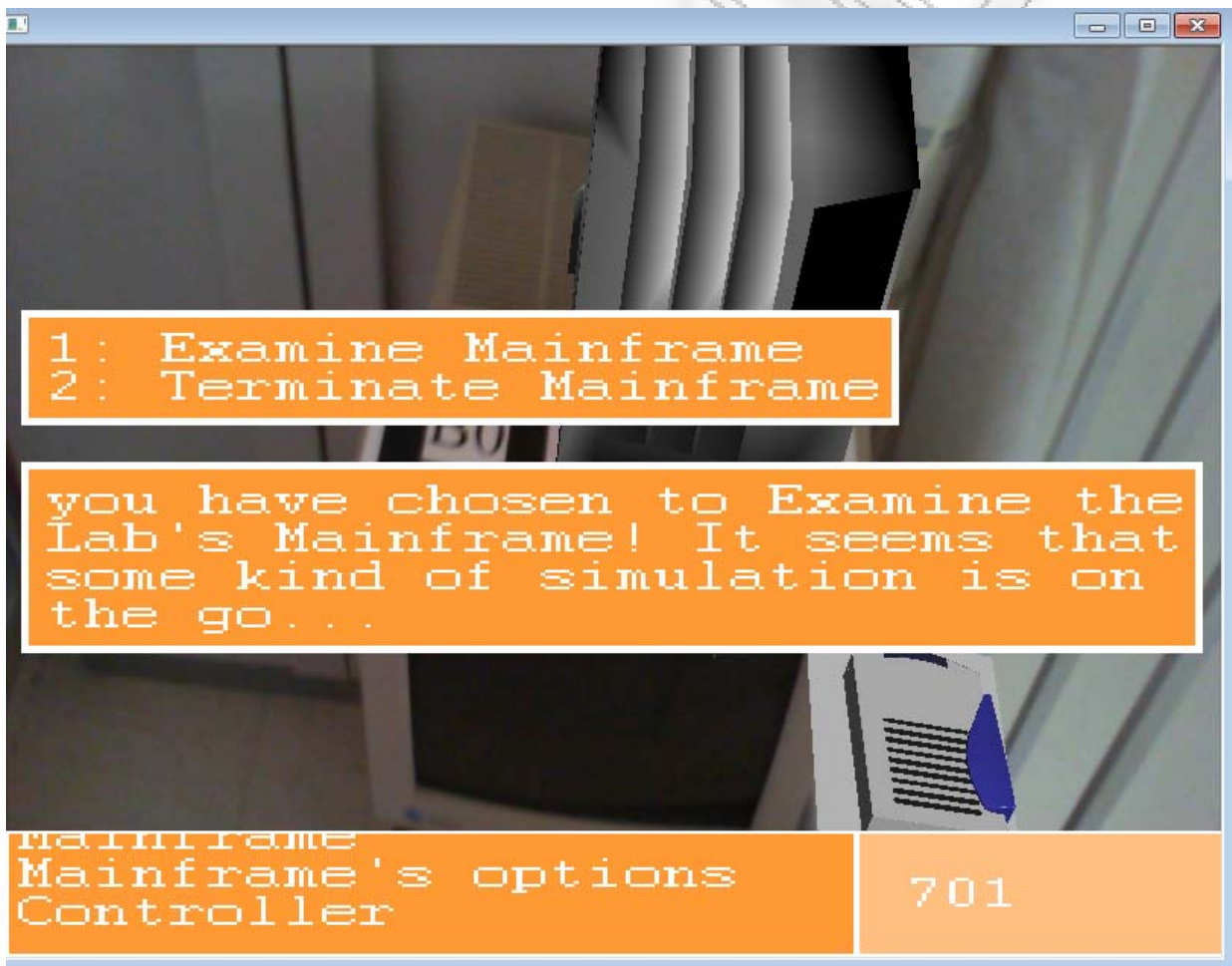
Στον νέο χώρο συναντάμε έναν μεγάλο server, μια οθόνη και τον controller.



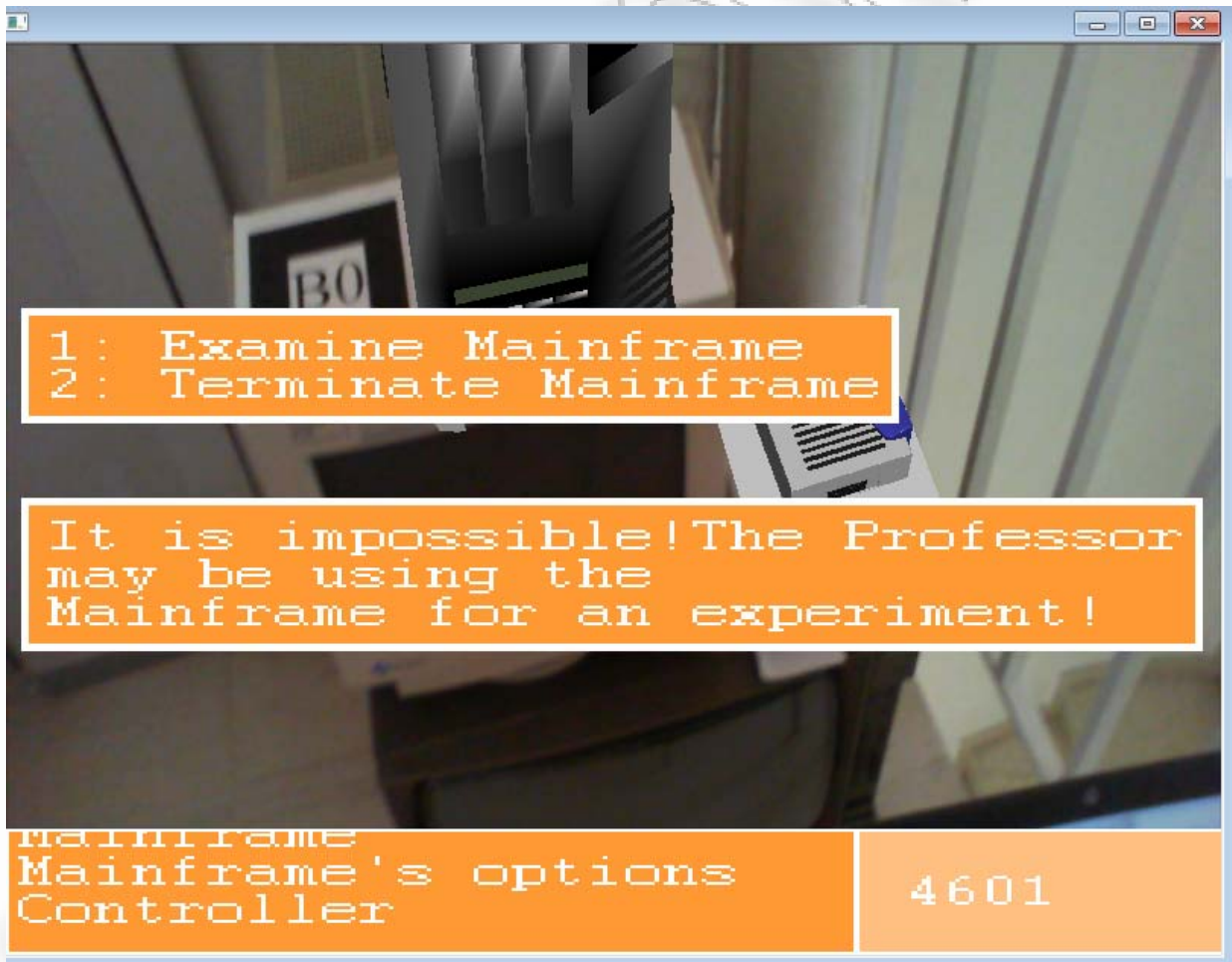
Η πρώτη κίνηση που κάνουμε αφού βρήκαμε τον controller είναι να τον πάρουμε μαζί για να τον επιστρέψουμε στην Αθηνά. Πολύ σύντομα όμως καταλαβαίνουμε ότι είναι σε λειτουργία, ότι δεν μπορούμε να τον μετακινήσουμε σε αυτή τη φάση και ότι πρέπει πρώτα να τον απενεργοποιήσουμε.



Μετά την πρώτη απόπειρα εξετάζουμε το mainframe για να πάρουμε περισσότερες πληροφορίες. Φαίνεται ότι είναι σε λειτουργία.



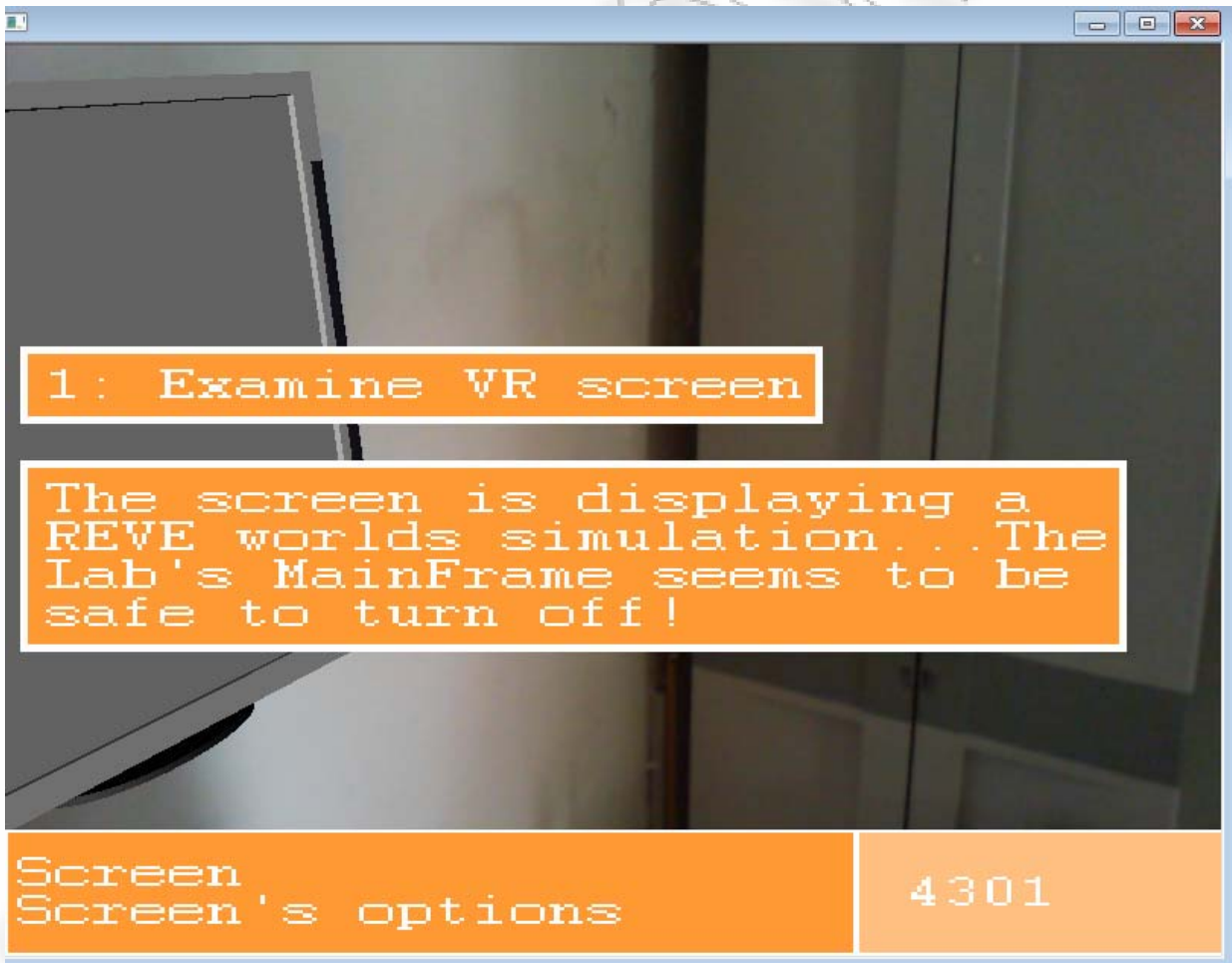
Δοκιμάζουμε να απενεργοποιήσουμε το mainframe. Η προσπάθειά μας όμως, είναι μάταιη. Ο χαρακτήρας μας ενημερώνει ότι δεν γνωρίζουμε τι εκτελείται σε αυτόν τον server σε αυτή τη φάση. Το μήνιμα μας προειδοποιεί ότι ίσως δημιουργηθεί πρόβλημα σε ένα από τα πειράματα του καθηγητή.



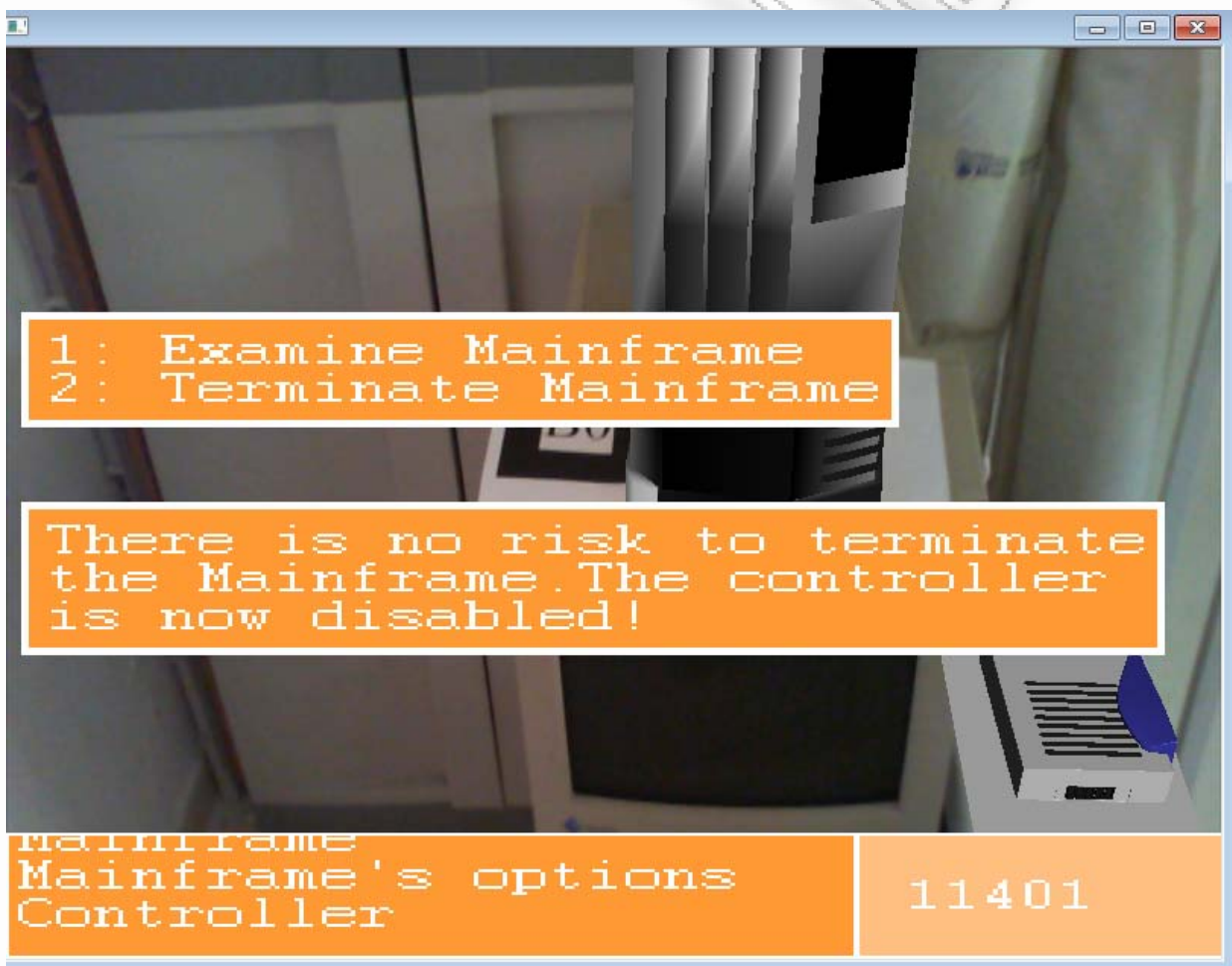
Εφόσον δεν υπάρχουν άλλες επιλογές εξετάζουμε και τον υπόλοιπο χώρο του δωματίου. Ένα ακόμα αντικείμενο δεν έχει εξετασθεί και αυτό είναι η οθόνη που είναι τοποθετημένη στον τοίχο.



Εξετάζοντας την οθόνη μαθαίνουμε ότι δείχνει το πρόγραμμα που εκτελεί το mainframe και ότι πρόκειται για ένα απλό σενάριο του συστήματος REVE worlds. Γνωρίζοντας τι τρέχει στο mainframe πλέον μπορούμε να το απενεργοποιήσουμε.

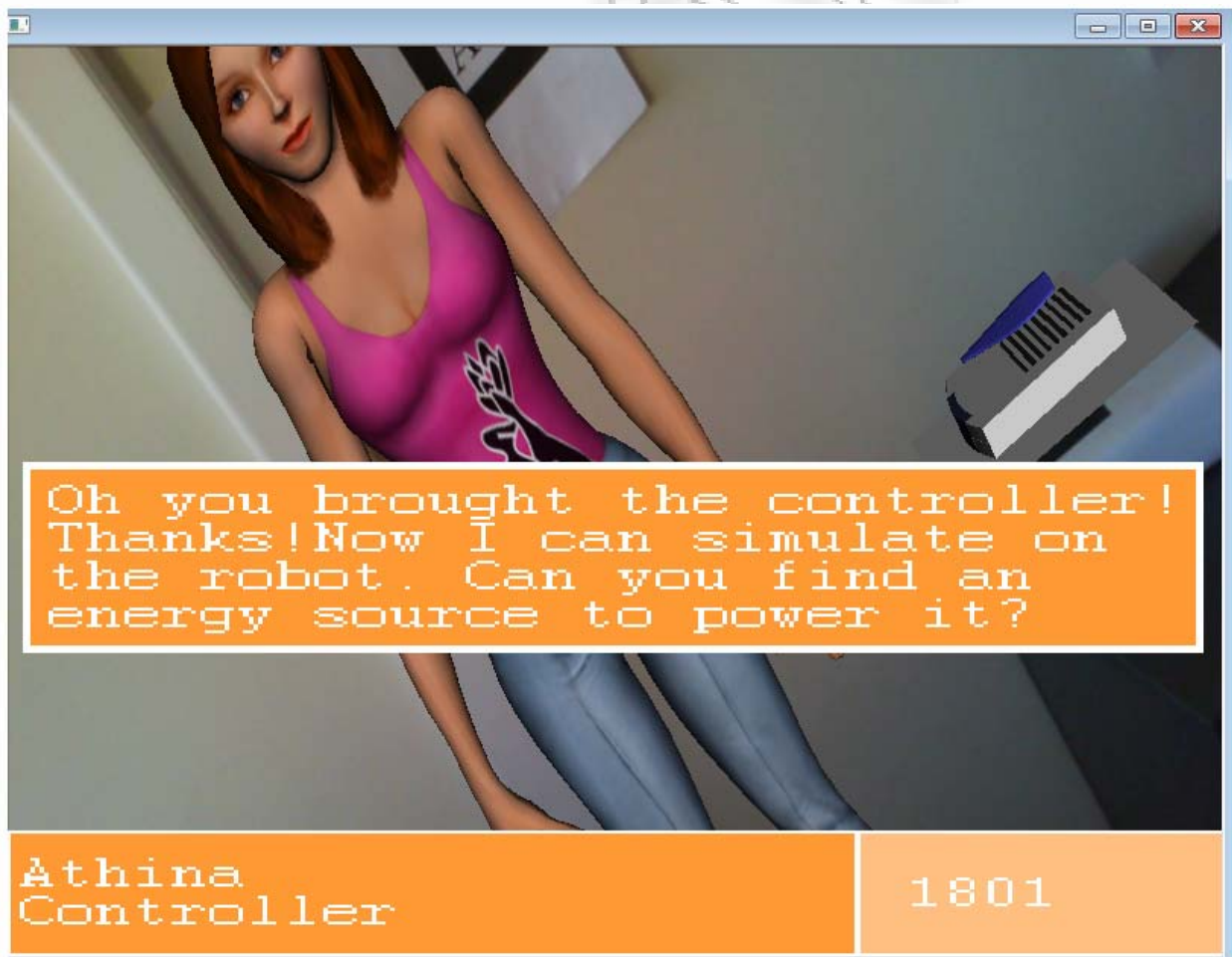


Επόμενο βήμα είναι η προσπάθεια απενεργοποίησης του mainframe που δεσμεύει τον controller.

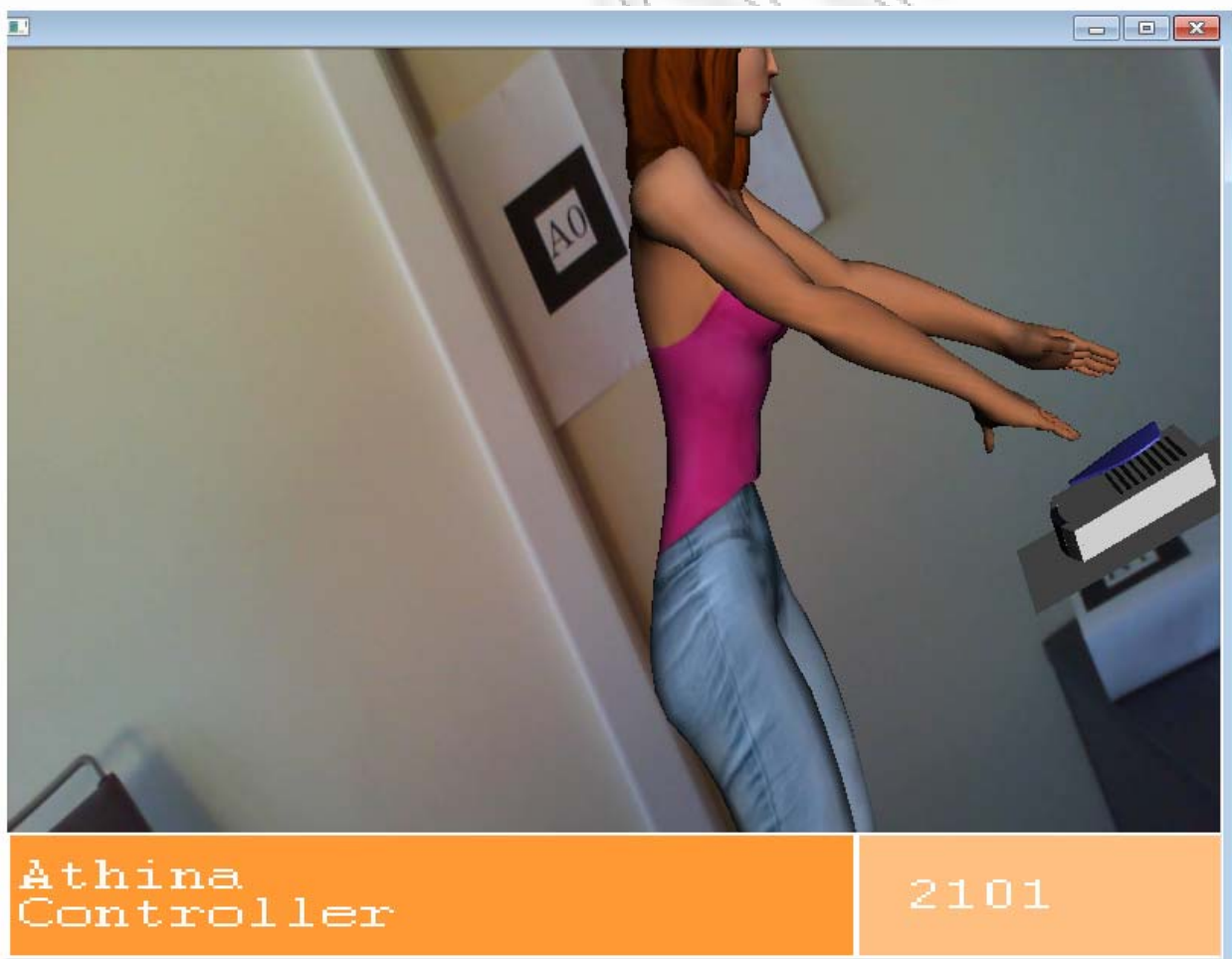


Το mainframe πλέον είναι απενεργοποιημένο και εμείς μπορούμε να μεταφέρουμε τον controller στην Αθηνά για να τον προγραμματίσει.

Μεταφέροντας τον controller στην τοποθεσία Lab και αφήνοντάς τον κοντά στην Αθηνά αυτή μας ευχαριστεί και ζητά το επόμενο αντικείμενο που της είναι απαραίτητο για την εργασία της. Το αντικείμενο είναι μια μπαταρία που θα χρησιμοποιηθεί για να κινητοποιήσει το ρομπότ.



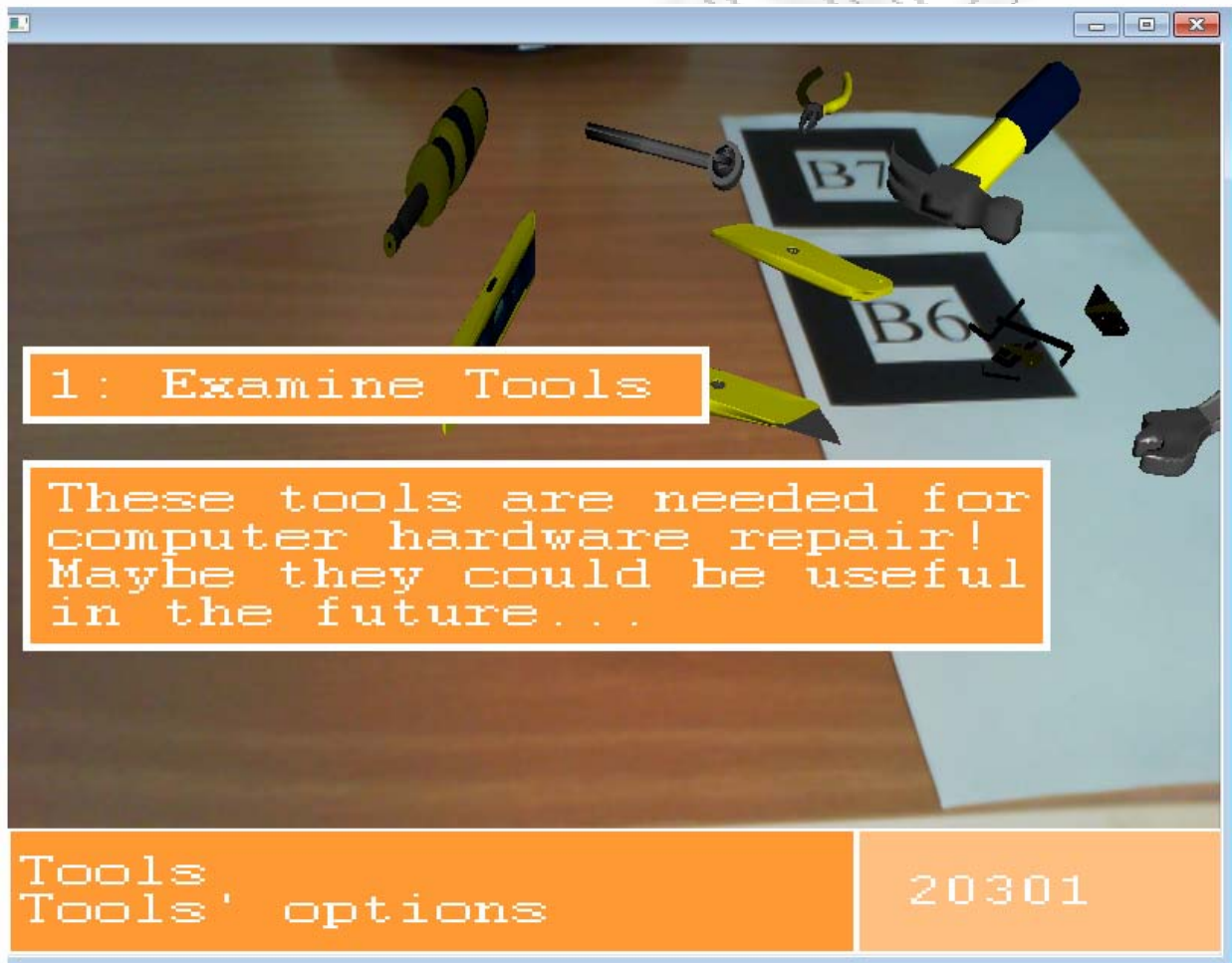
Αφού το μήνυμα εξαφανιστεί και εμείς έχουμε πλέον νέο σκοπό στην ιστορία, η Αθηνά ξεκινά να προγραμματίζει τον controller με ένα σχετικό animation.



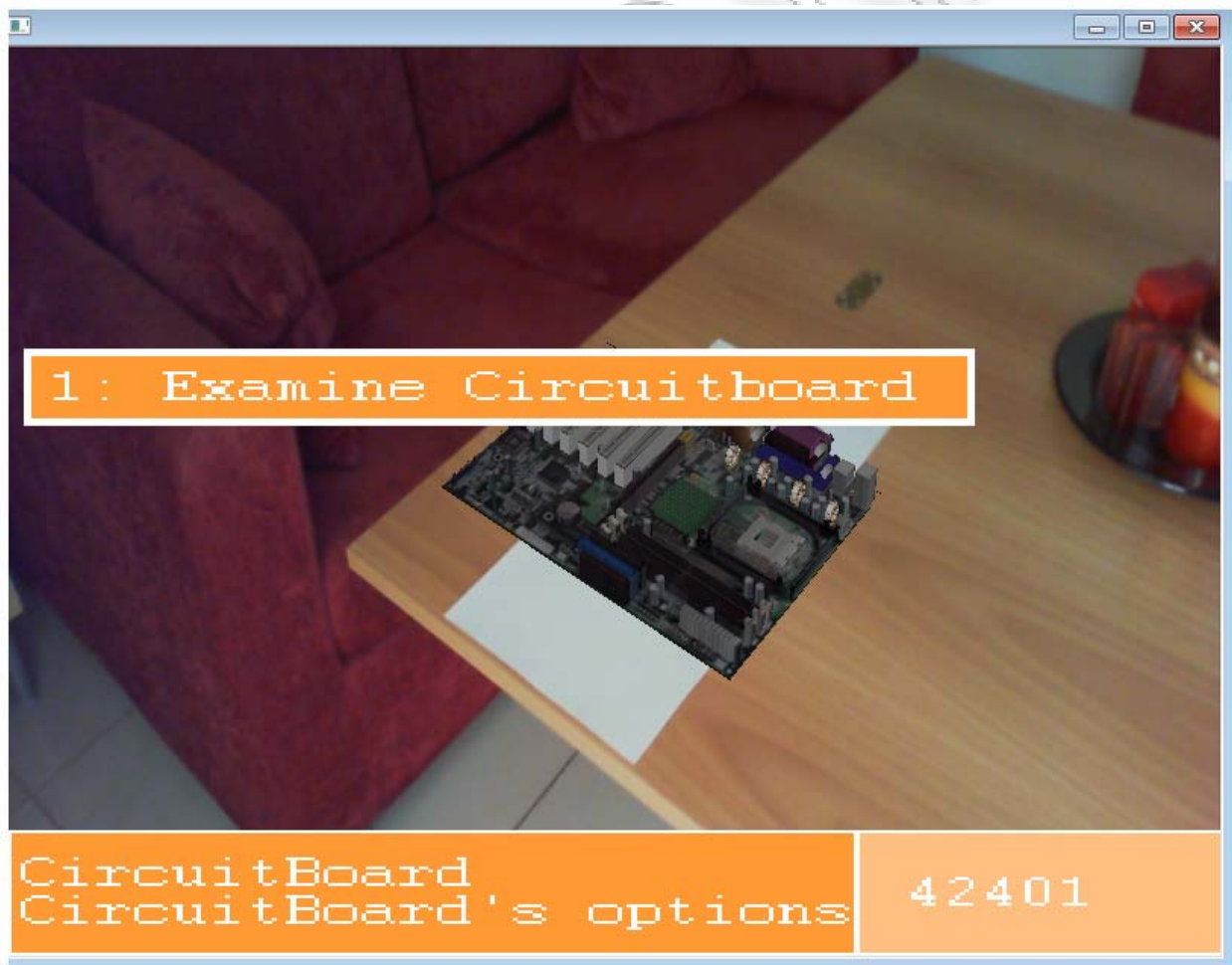
Με σκοπό την αναζήτηση της μπαταρίας που ζήτησε η εικονική φοιτήτρια περιπλανούμαστε στον χώρο και επισκεπτόμαστε την τοποθεσία Out1 που αποτελεί έναν εξωτερικό χώρο και συγκεκριμένα ένα lobby ακριβώς έξω από το εργαστήριο. Κοιτάζοντας τον χώρο βρίσκουμε ένα σετ από εργαλεία.



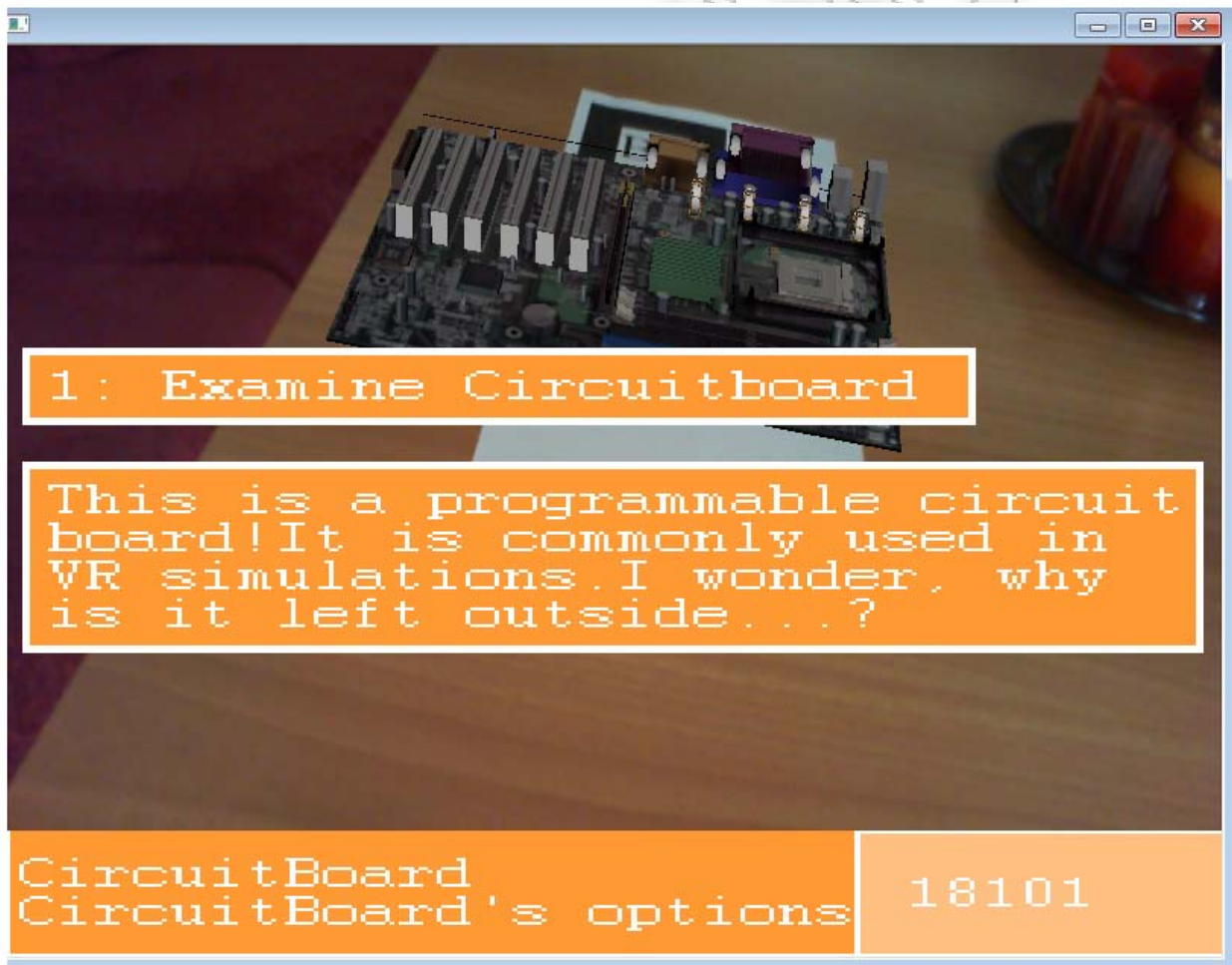
Εξετάζοντας τα εργαλεία μαθαίνουμε ότι είναι εργαλεία για την επιδιόρθωση υπολογιστών και ο χαρακτήρας μας μας ειδοποιεί ότι ίσως φανούν χρήσιμα στο μέλλον.



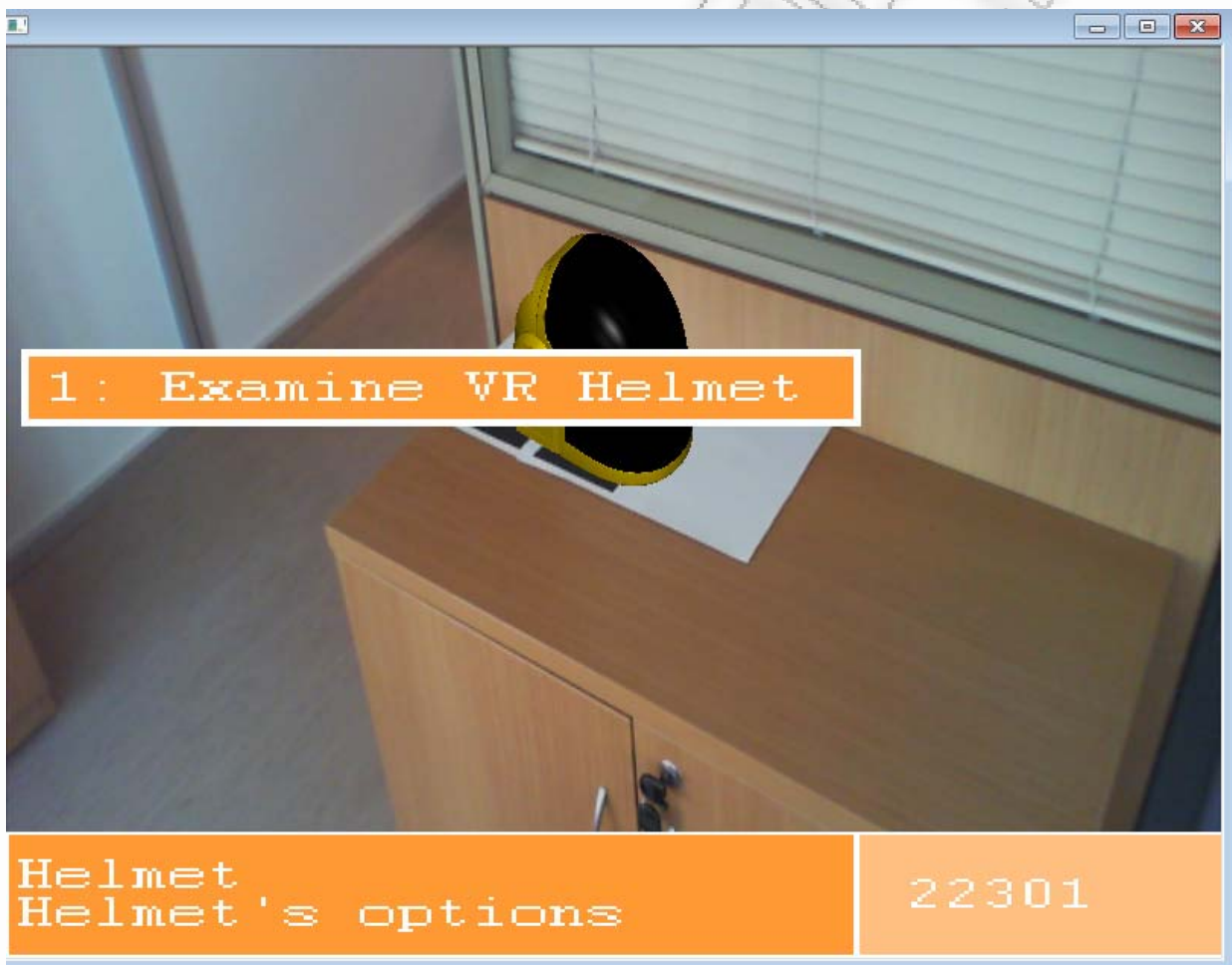
Επόμενο αντικείμενο στον ίδιο χώρο είναι μια πλακέτα. Πλησιάζοντάς την μας δίνεται η επιλογή να την εξετάσουμε.



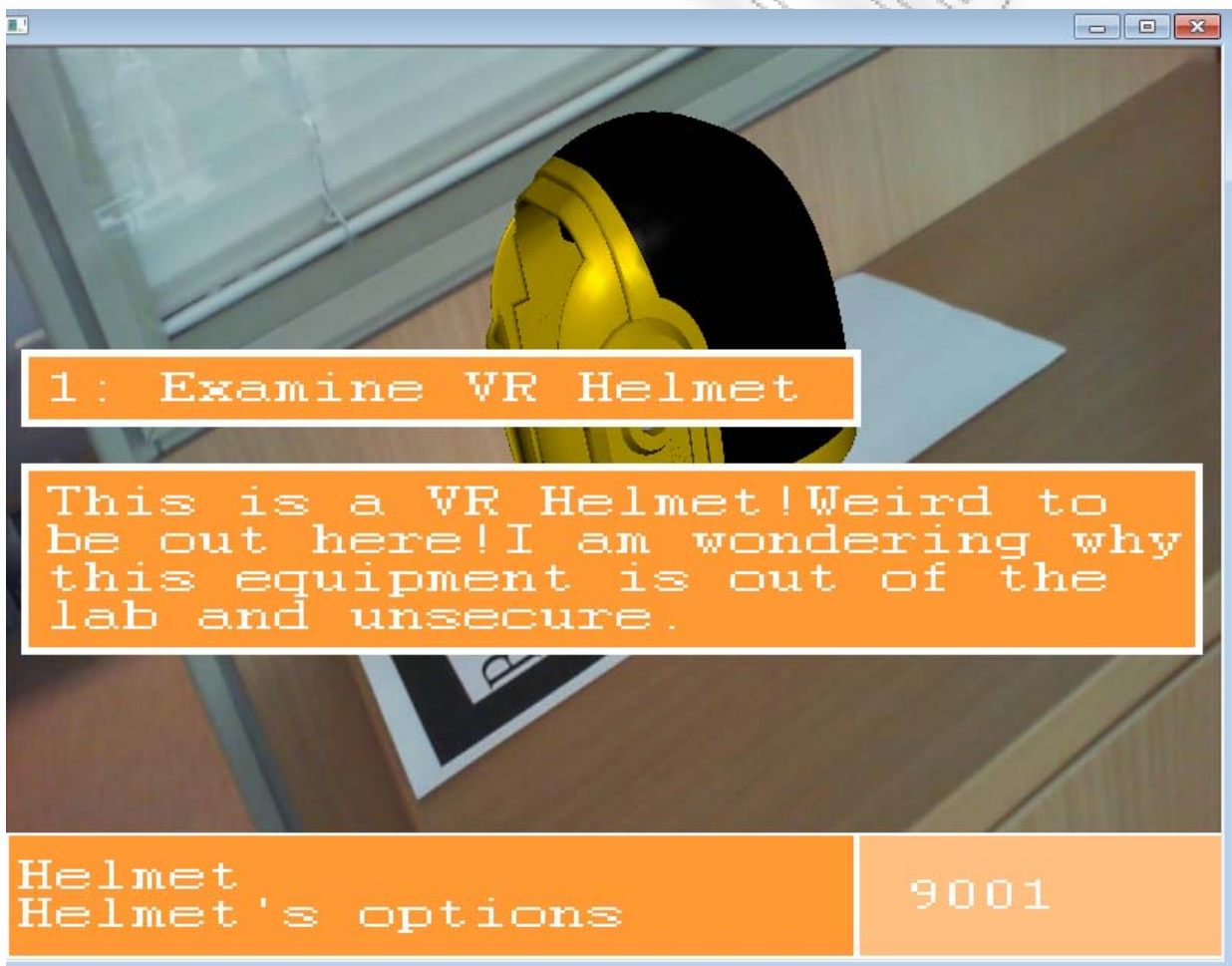
Πατώντας τον αριθμό 1 στο πληκτρολόγιο, ο χαρακτήρας μας εξετάζει αυτή την πλακέτα και μαθαίνει ότι πρόκειται για μια προγραμματιζόμενη πλακέτα που συναντάται στις εφαρμογές εικονικής πραγματικότητας. Ο χαρακτήρας μας αναρωτιέται γιατί αυτή η πλακέτα να είναι αφύλακτη σε έναν εξωτερικό χώρο.



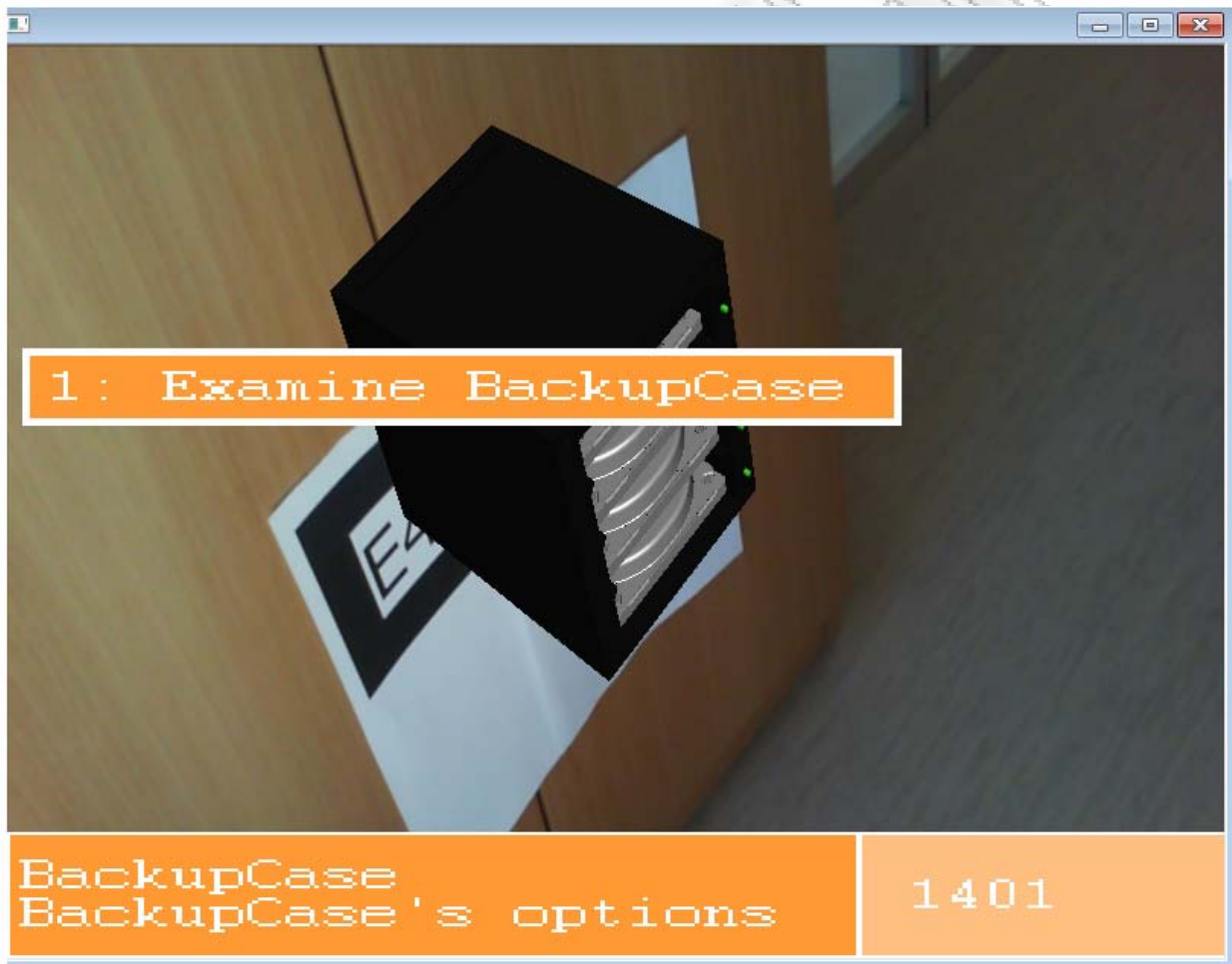
Εφόσον δεν υπάρχουν άλλα αντικείμενα στον χώρο που να μπορούμε να αλληλεπιδράσουμε, προχωρούμε στην τελευταία τοποθεσία που είναι η και αυτή στο εξωτερικό του εργαστηρίου. Εκεί συναντάμε ένα κράνος αρκετά ασυνήθιστο.



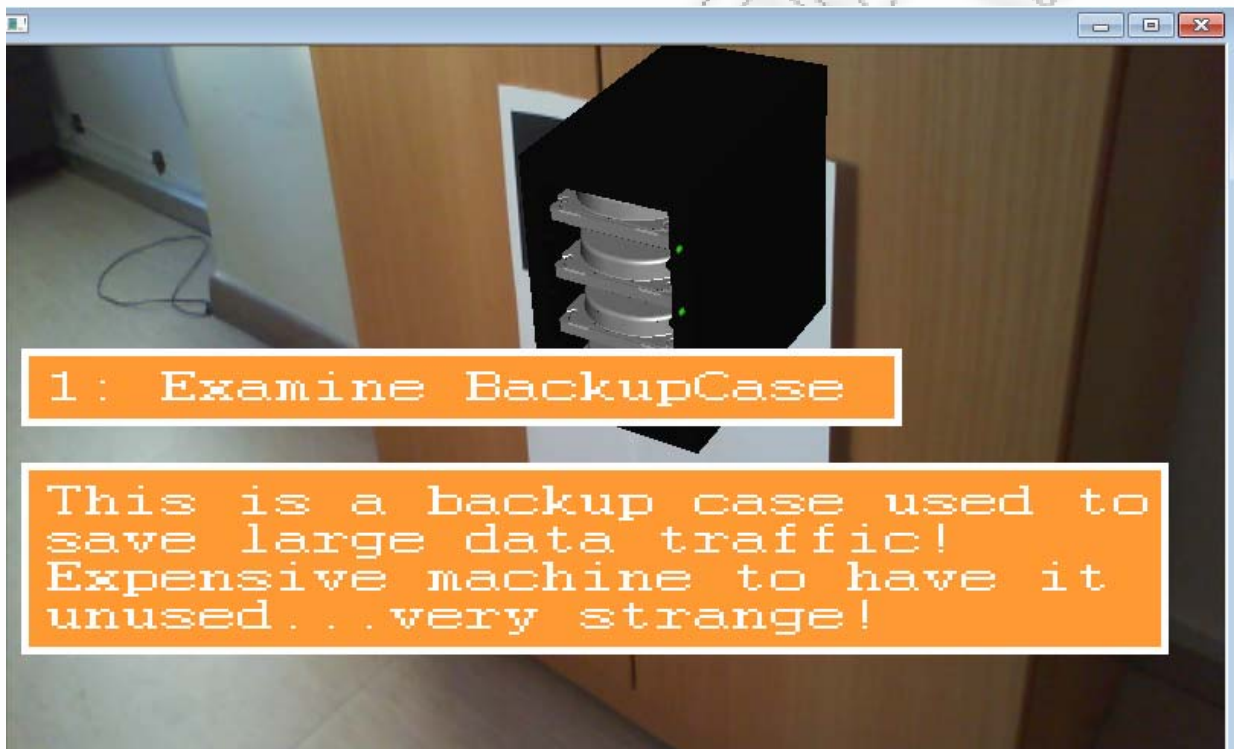
Επιλέγοντας να πάρουμε περισσότερες πληροφορίες για αυτό, μαθαίνουμε ότι είναι ένα προηγμένο κράνος εικονικής πραγματικότητας και φαίνεται ιδιαίτερα περίεργο το γιατί τέτοιος ακριβός και ευπαθής εξοπλισμός βρίσκεται έξω από το εργαστήριο και χωρίς κάποιος να τον προσέχει.



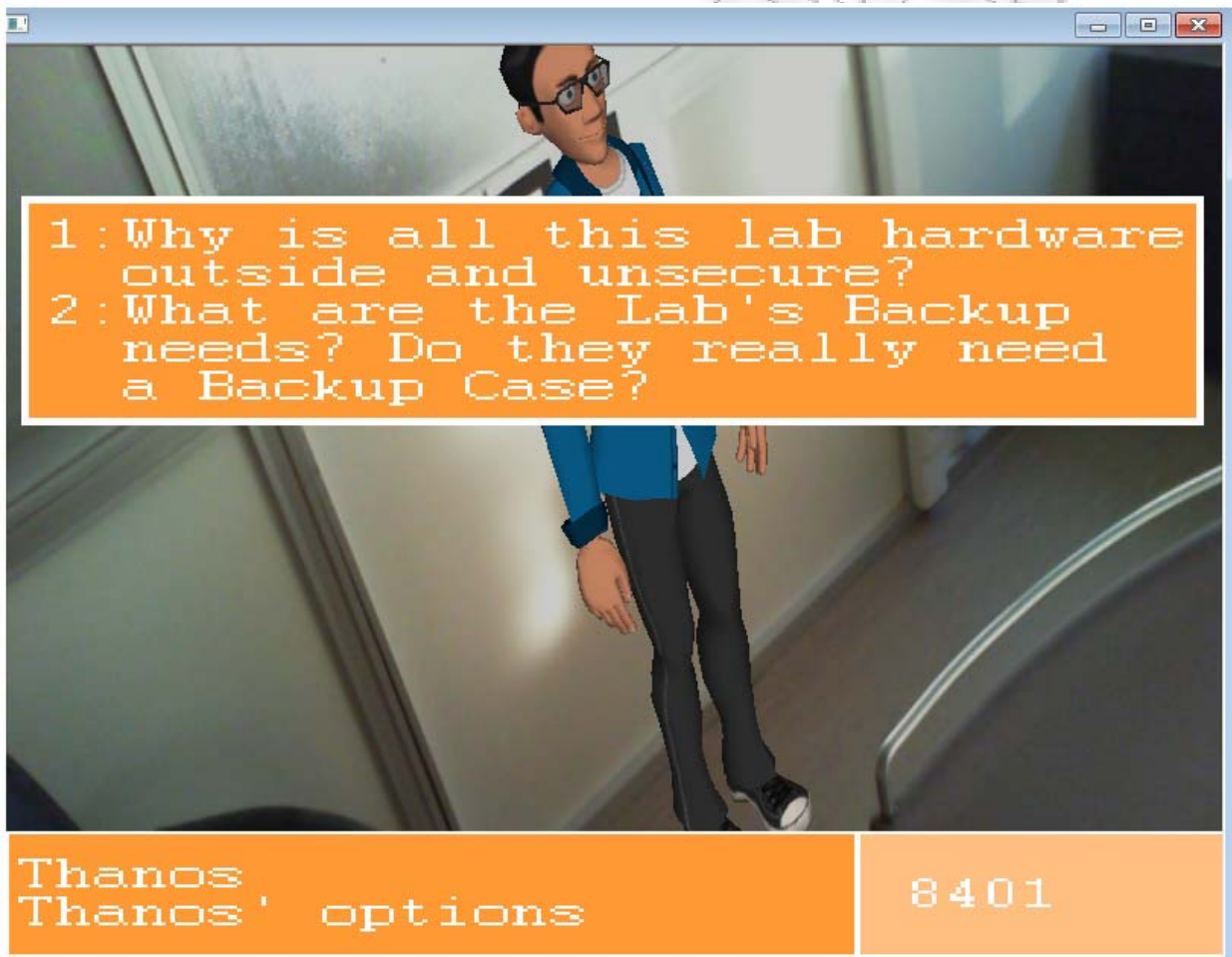
Στον ίδιο χώρο και σε μερικά μόνο μέτρα απόσταση εντοπίζουμε μια άλλη παράξενη και ογκώδη συσκευή.



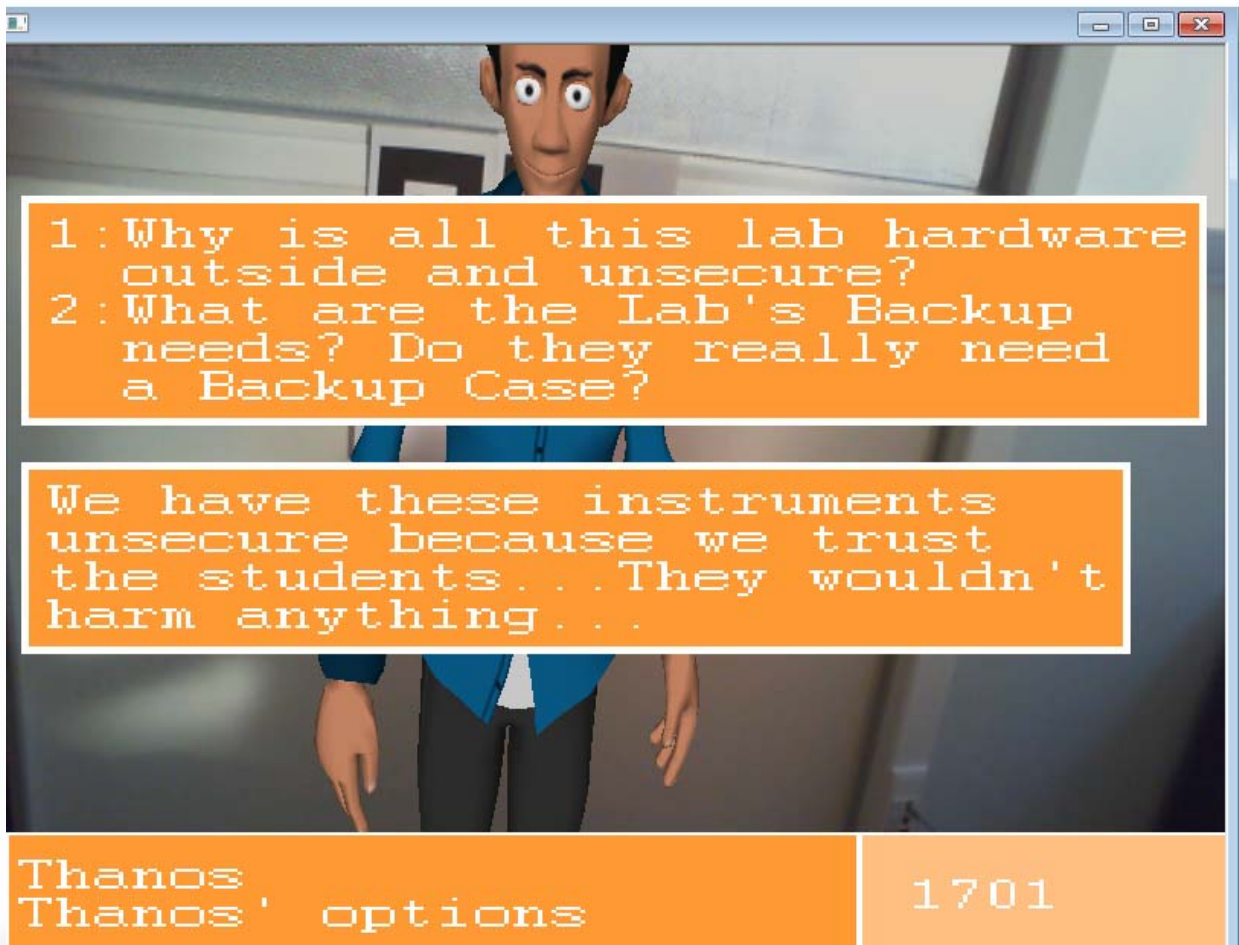
Εξετάζοντας τη συσκευή αυτή μαθαίνουμε ότι είναι μια συστοιχία σκληρών δίσκων που χρησιμοποιείται για την αποθήκευση backup μεγάλου όγκου δεδομένων. Ο χαρακτήρας σχολιάζει ότι είναι πολύ ακριβό μηχάνημα για να μένει αχρησιμοποίητο.



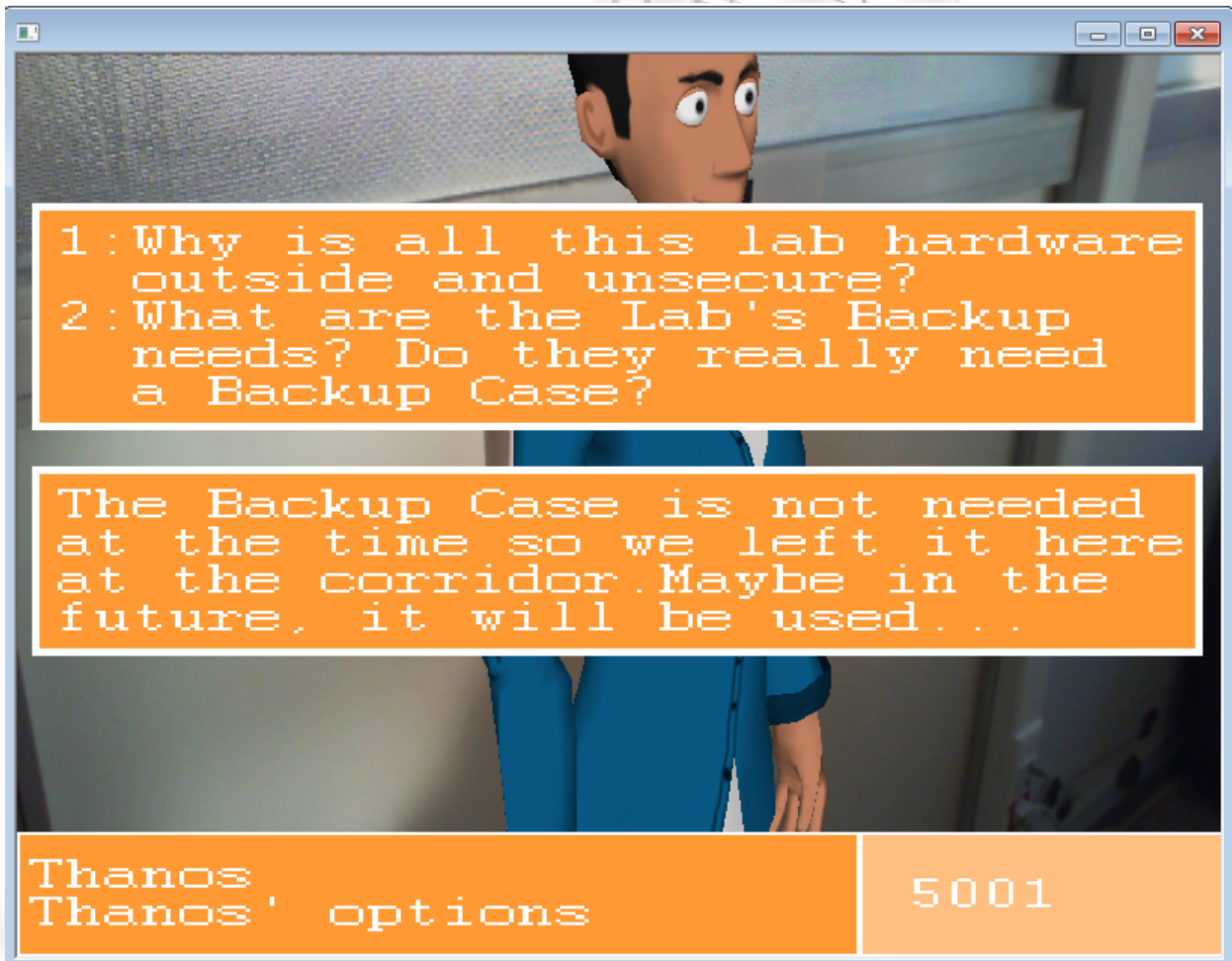
Στρέφοντας την προσοχή μας προς την αντίθετη κατεύθυνση συναντάμε έναν νέο χαρακτήρα. Πλησιάζοντά τον εμφανίζονται οι επιλογές αλληλεπίδρασης.



Ο χαρακτήρας είναι ένας ακόμα φοιτητής του Πανεπιστημίου και ονομάζεται Θάνος. Πατώντας την πρώτη επιλογή του κάνουμε την ερώτηση που μας έχει δημιουργηθεί, δηλαδή το πως είναι δυνατόν όλος αυτός ο εξοπλισμός να βρίσκεται εκτός εργαστηρίου και να είναι αφύλακτος. Μετα από τα αρκετά αντικείμενα που έχουμε συναντήσει, αυτή είναι μια απορία που όλοι θα είχαμε.



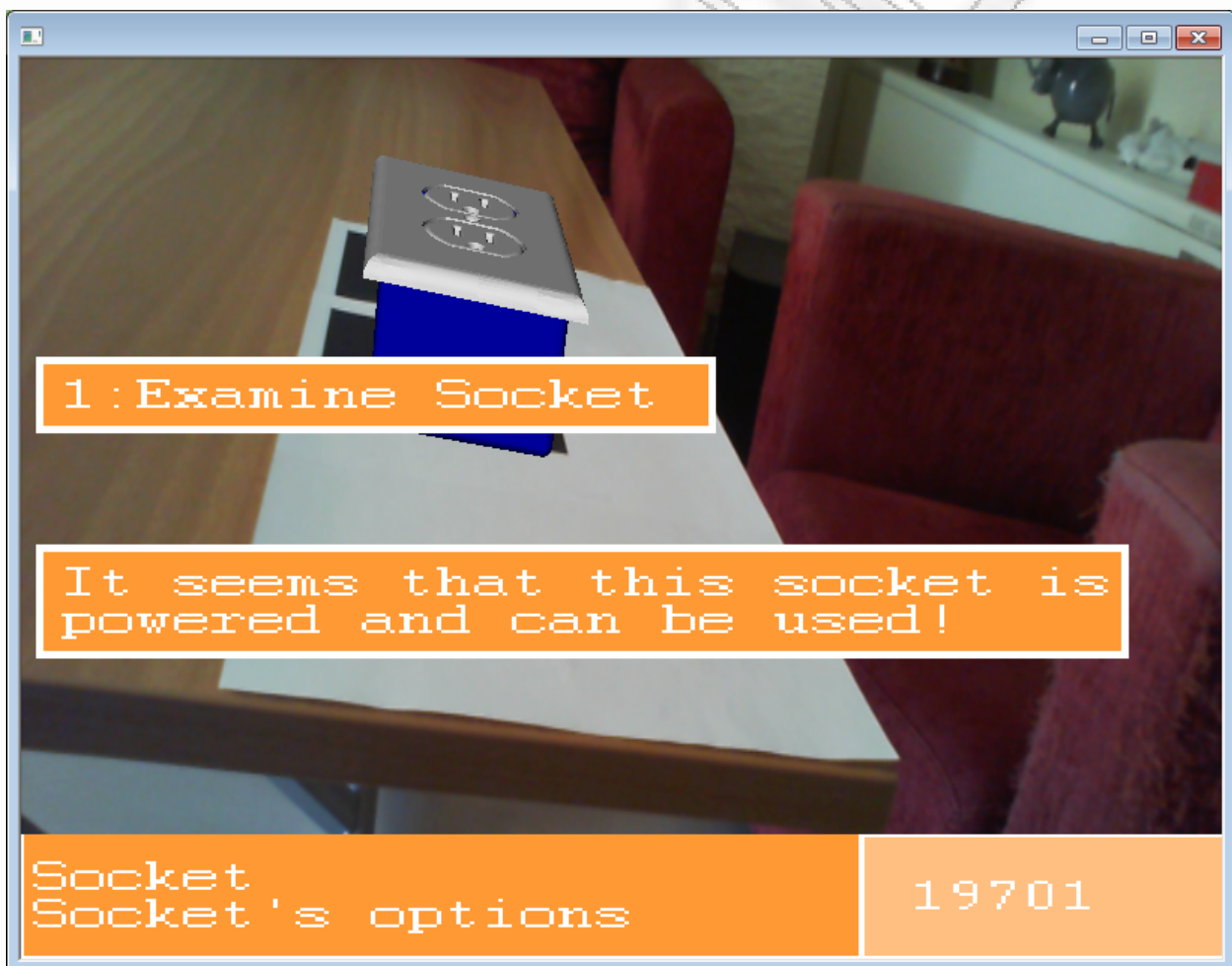
Ο Θάνος μας δίνει μια ιδιαίτερα περίεργη απάντηση...λέει πως όλα αυτά τα όργανα παραμένουν εκτός εργαστηρίου και αφύλακτα διότι το Πανεπιστήμιο εμπιστεύεται τους φοιτητές και ότι ο κόσμος που περιφέρεται δεν θα τα έβλαπτε. Παρόλο που μας φαίνεται ανεξήγητη η απάντηση τον ρωτάμε και για το δεύτερο θέμα που είναι το γιατί το εργαστήριο έχει αγοράσει μια τόσο μεγάλη συσκευή για backup και εάν είναι όντως τόσο μεγάλες οι τρέχουσες ανάγκες του.



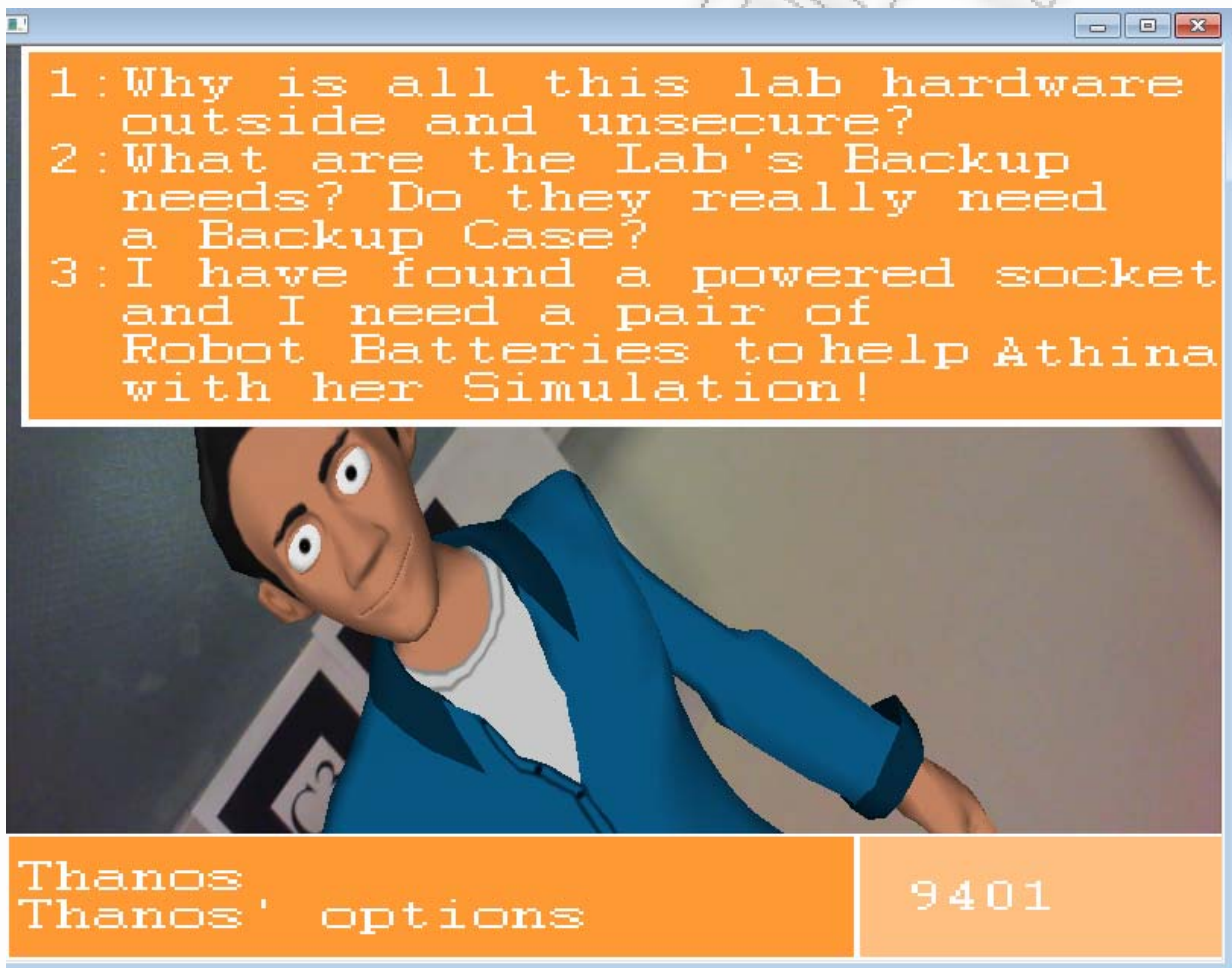
Ο φοιτητής απαντά ότι δεν χρειάζεται η συσκευή προς το παρόν, ότι ίσως χρειαστεί στο μέλλον και ότι την άφησαν προσωρινά στον διάδρομο. Οι απάντησή μας βάζει σε υποψίες γενικά και εφόσον δεν υπάρχουν άλλες δράσεις για εμάς γυρίζουμε προς την προηγούμενη τοποθεσία σε περίπτωση που κάτι μας έχει ξεφύγει. Κοιτάζοντας πιο προσεκτικά τον χώρο στην τοποθεσία Out1 που είναι το lobby έξω από το εργαστήριο παρατηρούμε ότι υπάρχει ένα ακόμα αντικείμενο μια διπλή πρίζα.



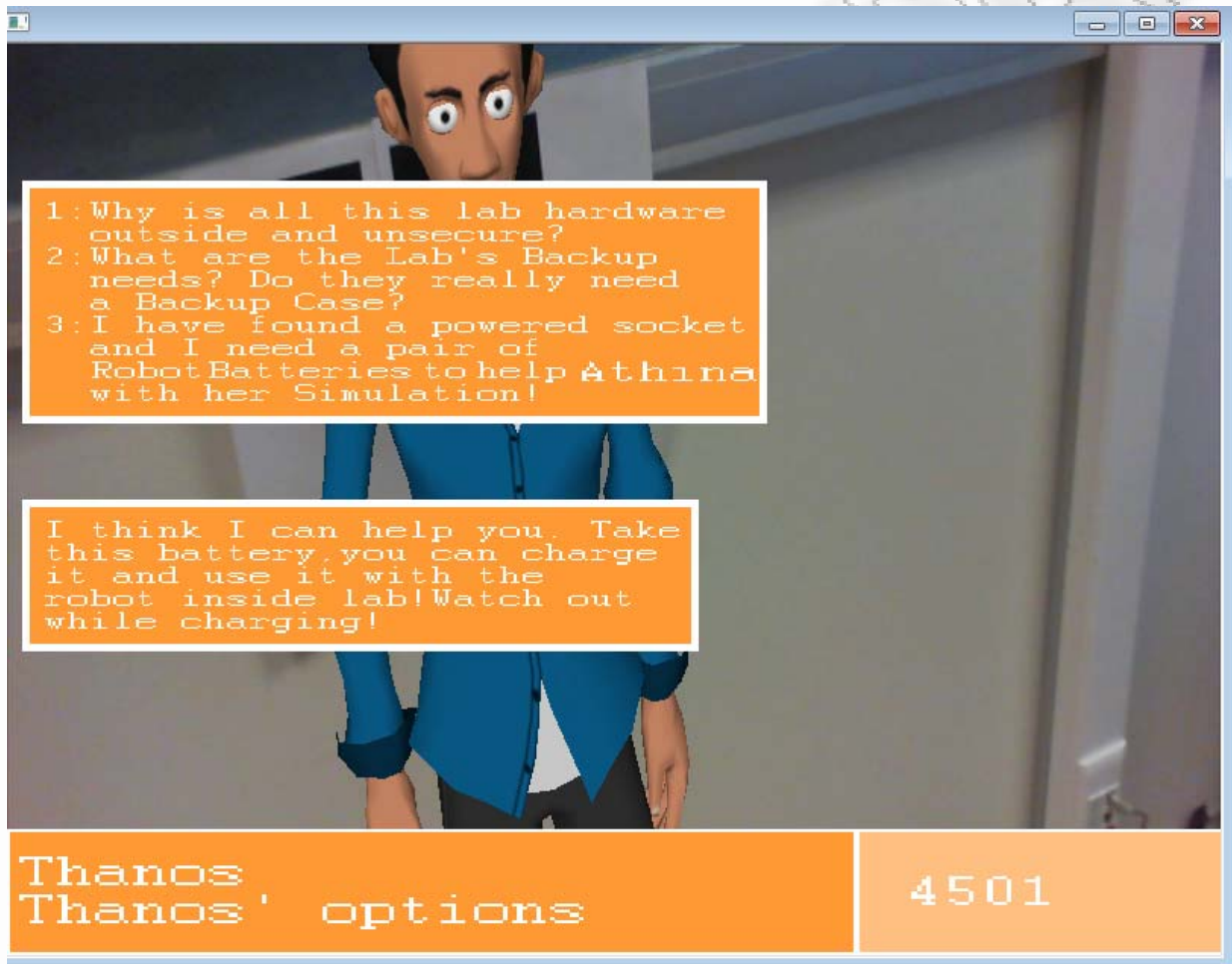
Εξετάζοντας την διπλή πρίζα μαθαίνουμε ότι είναι ενεργή και ότι μπορεί να χρησιμοποιηθεί σαν πηγή ενέργειας.



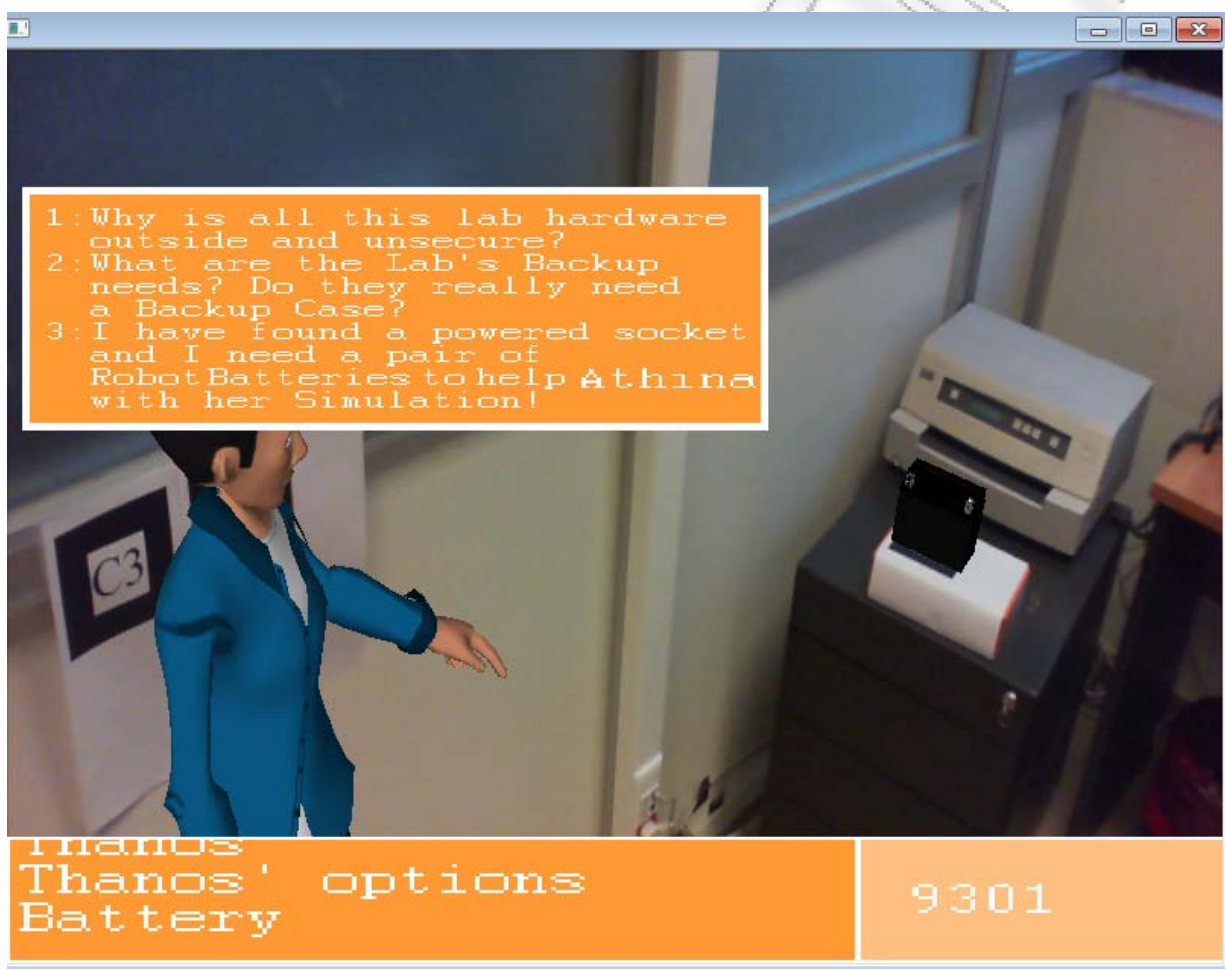
Γυρίζοντας στην εξωτερική τοποθεσία Out2 στον φοιτητή Θάνο για περισσότερη συζήτηση, βλέπουμε ότι πλέον έχουμε μια ακόμη επιλογή. Μπορούμε να του πούμε ότι θέλουμε να βοηθήσουμε την Αθηνά (επιλογή 3), ότι εέχουμε βρεί μια πηγή ενέργειας –την πρίζα- και ότι χρειαζόμαστε μια μπαταρία.



Ο Θάνος ανταποκρίνεται στην έκκλησή μας για βοήθεια και λέει ότι μπορεί να μας βοηθήσει.



Με ένα animation μας υποδεικνύει το που είναι η μπαταρία που ψάχνουμε έτσι ώστε να την πάρουμε και να την μεταφέρουμε στην πρίζα για φόρτιση.



Σε αυτό το σημείο υπάρχουν δυο περιπτώσεις για την συνέχεια του σεναρίου. Εάν έχουμε βρεί όλα τα αντικείμενα που υπάρχουν στους χώρους και εφόσον πλέον αναμφισβήτητα θεωρούμε ότι η ύπαρξή τους είναι αρκετά ασυνήθιστη, ο Θάνος θεωρεί ότι πλέον δεν μπορεί να υπεκφύγει και αποφασίζει να μας βοηθήσει προειδοποιώντας μας για την αλήθεια...Μας λέει ότι στην πραγματικότητα αυτό που ζούμε είναι μια εξομοίωση και ότι είμαστε μέλη ενός τεχνητού κόσμου.



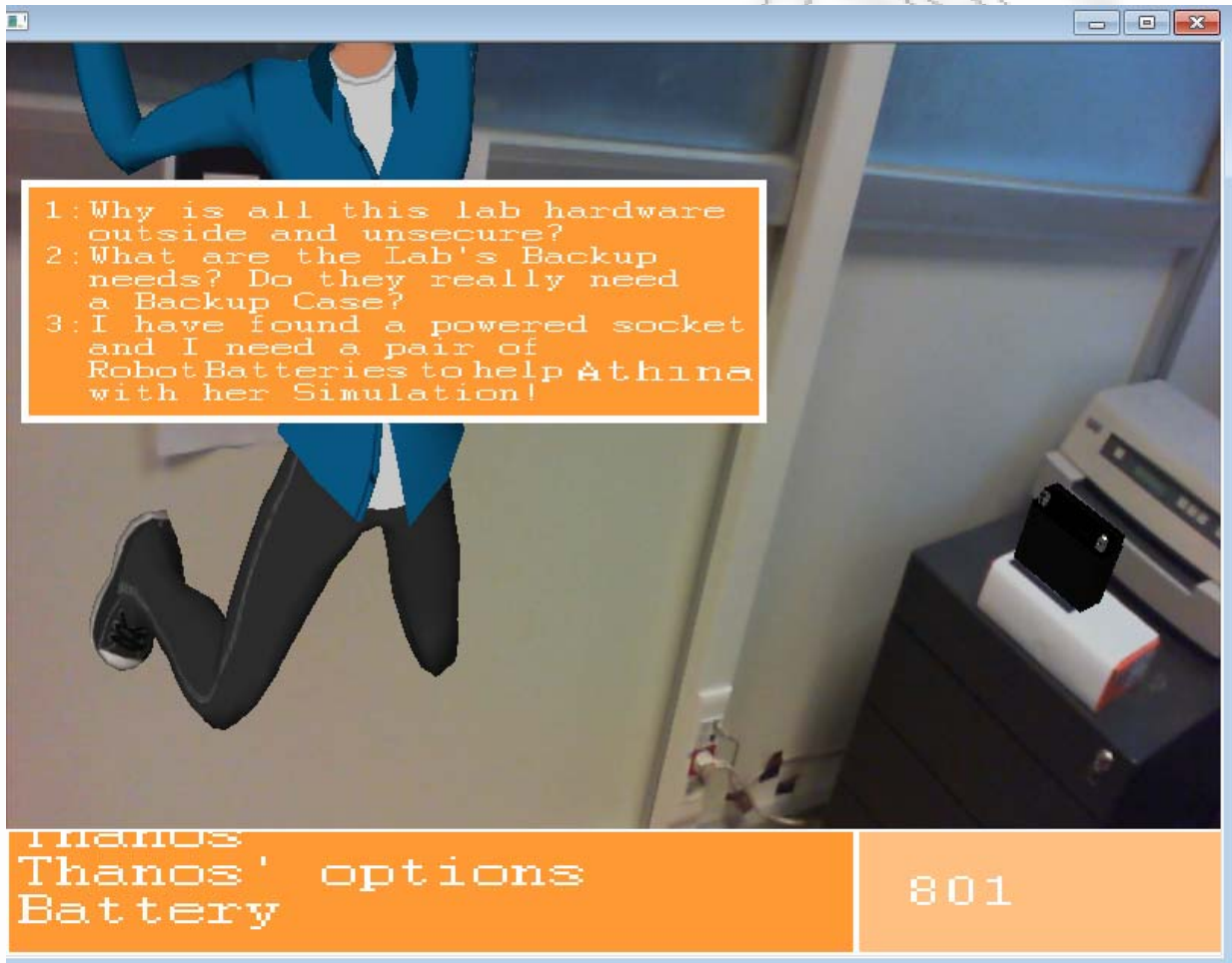
Κοιτάζοντας τον Θάνο τον βλέπουμε πλέον να μας χαιρετά καθώς ετοιμαζόμαστε να πάμε προς την Αθηνά και να μάθουμε την αλήθεια...



Στην περίπτωση που δεν είμαστε παρατηρητικοί και δεν έχουμε μαζέψει αρκετές πληροφορίες από το περιβάλλον μας, δεν μας βοηθά και δεν μας λείει την αλήθεια πίσω από την τρέχουσα εξομοίωση. Αντιθέτως, εμφανίζεται ένα περίεργο μήνυμα λάθους μπροστά μας...

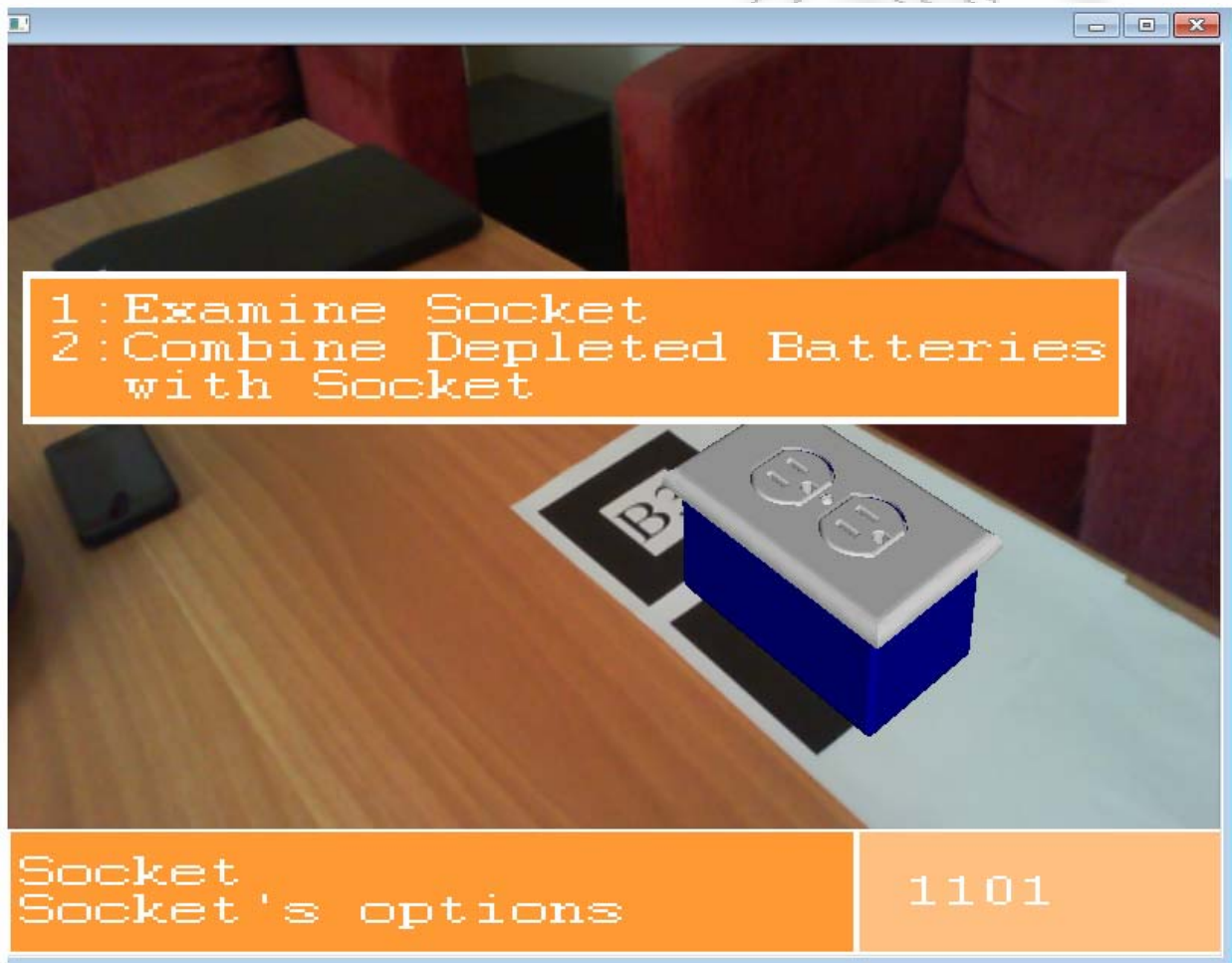


Στο μήνυμα λάθους ο φοιτητής Θάνος αντιδρά και κάνει ένα άλμα για να μας τραβήξει την προσοχή και να μην δώσουμε σημασία σε αυτό που είδαμε λίγο νωρίτερα.

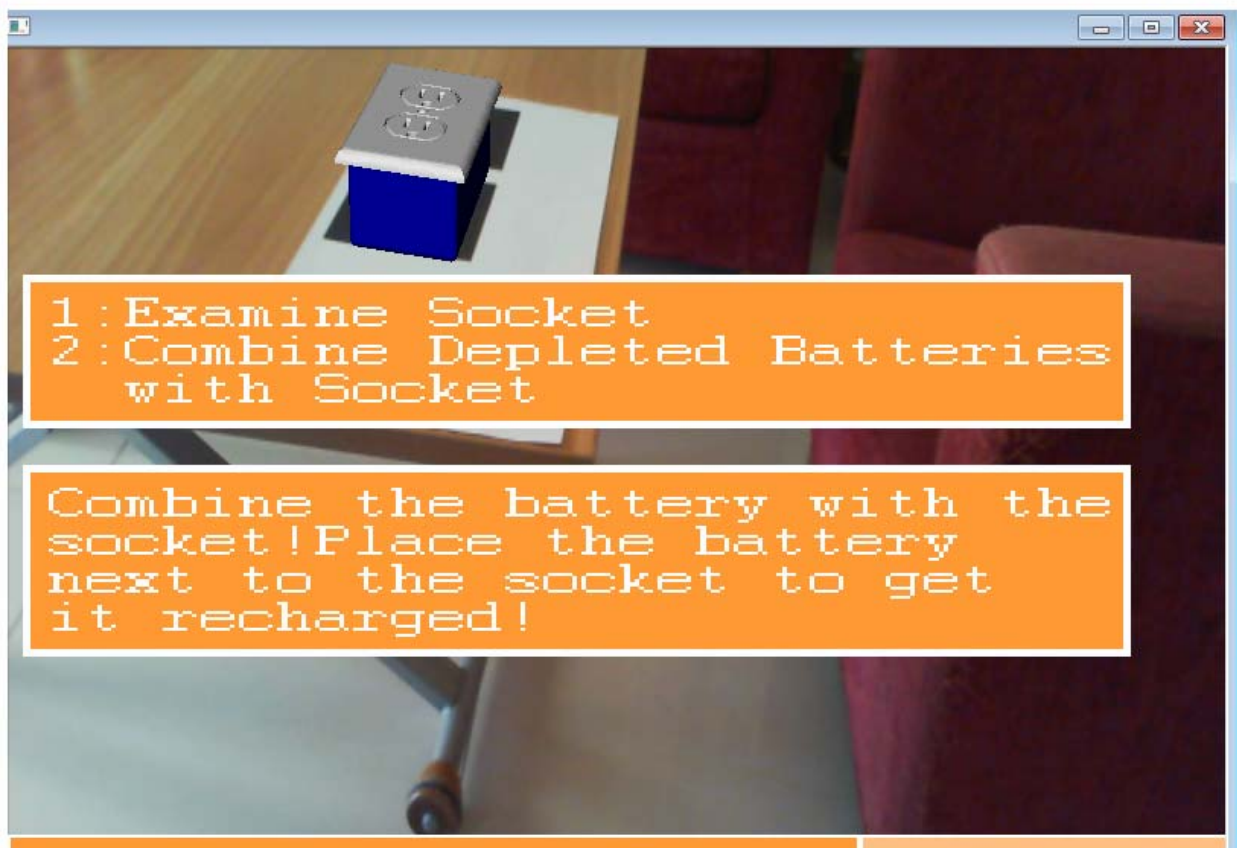


Σε όποια από τις δυο περιπτώσεις και αν είμασταν παίρνουμε την μπαταρία και κατευθυνόμαστε στον πλωτο εξωτερικό χώρο, εκεί που συναντήσαμε την πρίζα για να κάνουμε την φόρτιση.

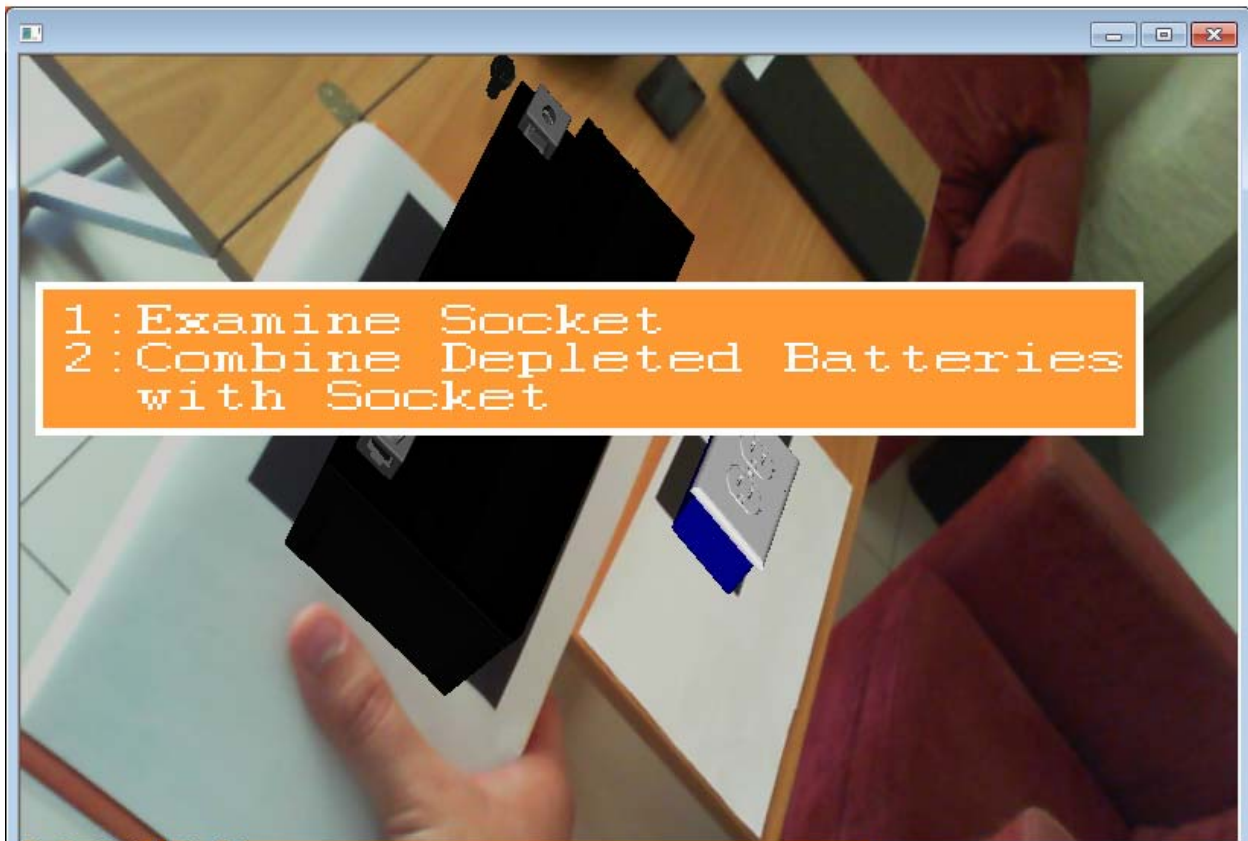
Πλησιάζοντας την πρίζα εμφανίζεται μια ακόμα επιλογή που λέει να συνδυάσουμε την μπαταρία με την πρίζα.



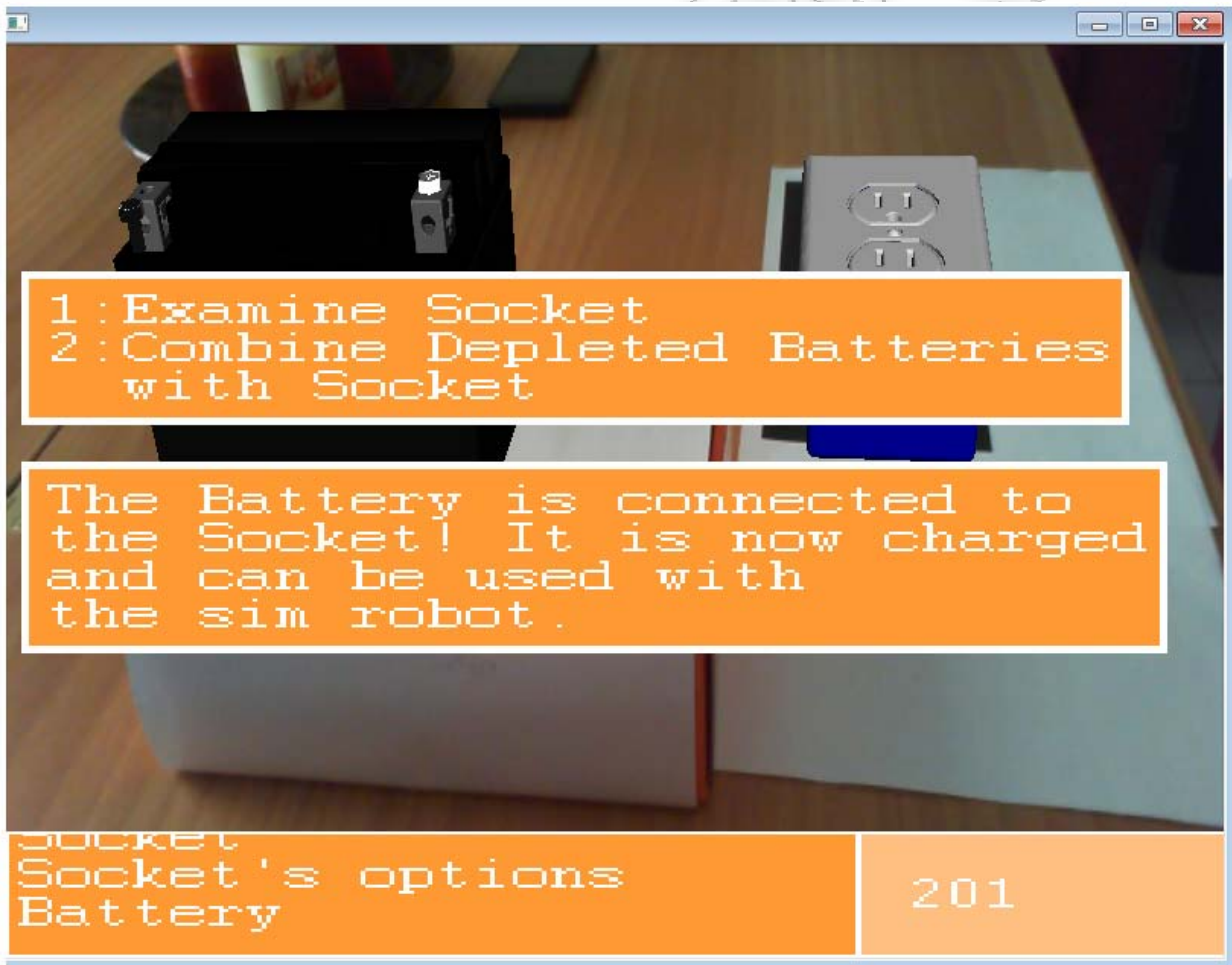
Πατώντας τη νέα επιλογή που εμφανίζεται στις δράσεις με το αντικείμενο πρίζα, εμφανίζεται το μήνυμα του χαρακτήρα μας που λέει ότι πρέπει να τοποθετήσουμε την μπαταρία κοντά στην πρίζα για να φορτιστεί.



Βάζοντας την μπαταρία δίπλα από την πρίζα...



αυτή φορτίζεται και είναι πλέον έτοιμη να χρησιμοποιηθεί από την Αθηνά.



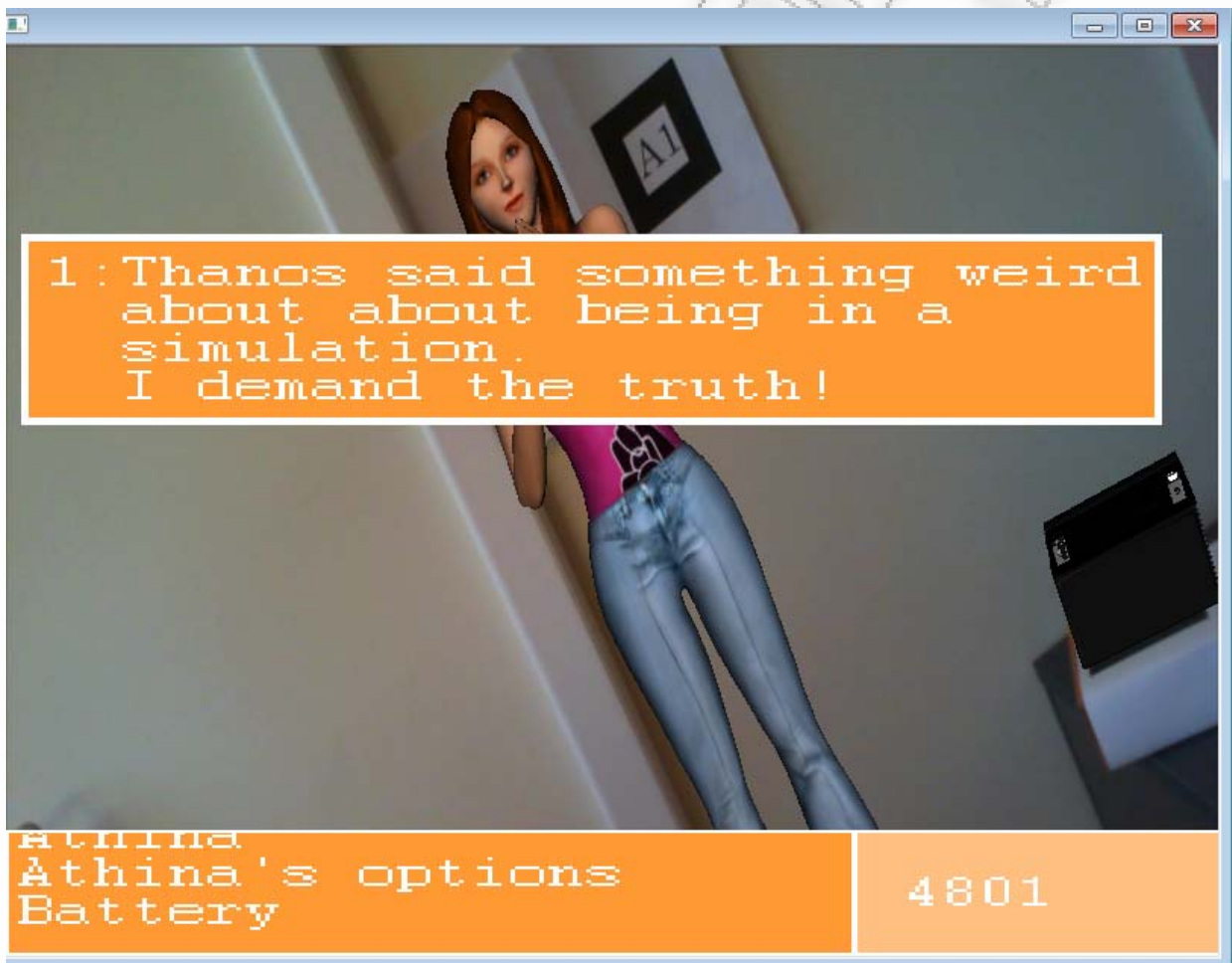
Γυρίζουμε στο εργαστήριο δηλαδή στην τοποθεσία Lab και αφήνουμε την φορτισμένη μπαταρία στην Αθηνά.



Η Αθηνά πλέον μπορεί να συνεχίσει την εργασία της στο ρομπότ και με ένα animation χειροκροτεί για την βοήθεια που της δώσαμε και η ιστορία τελειώνει.

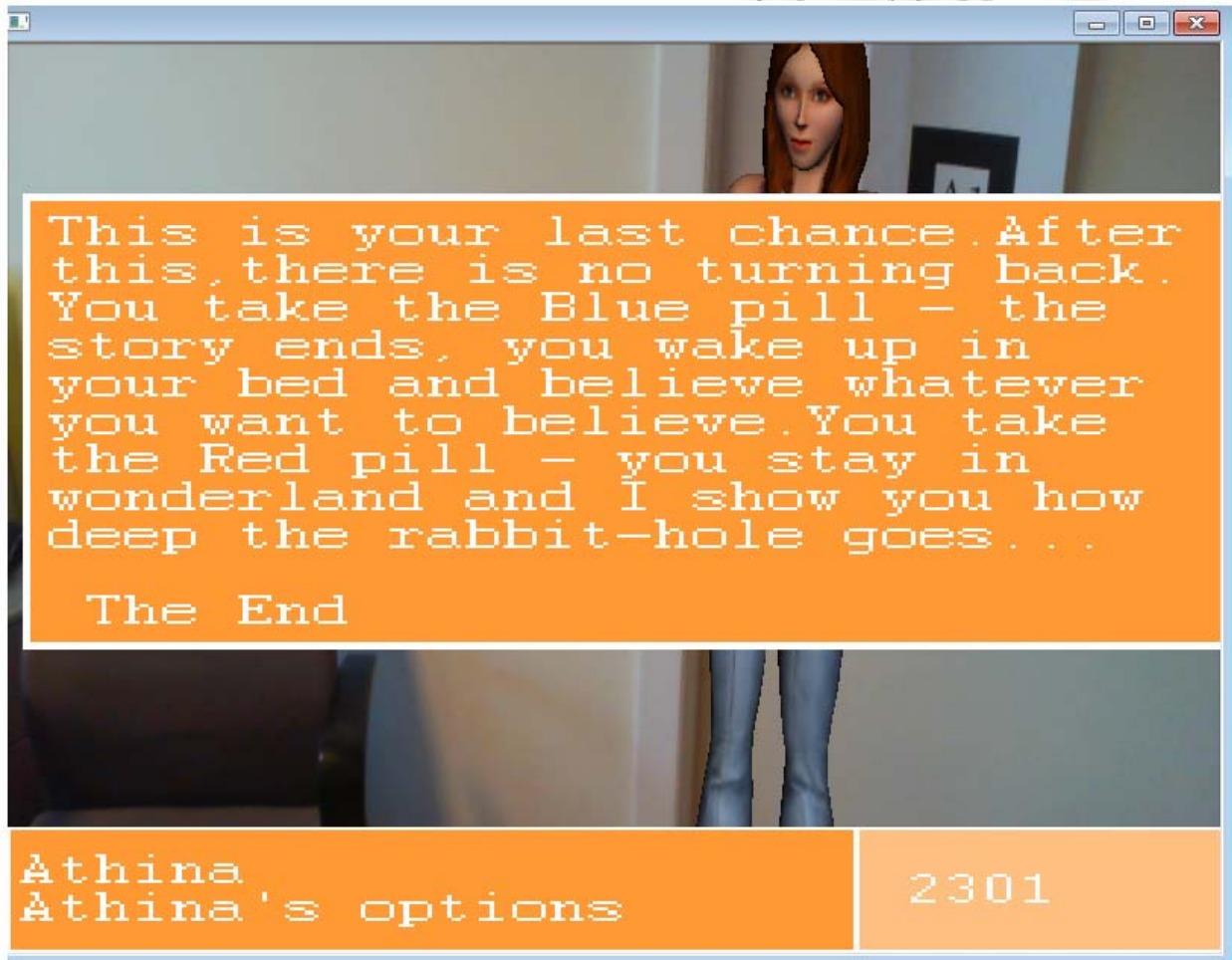


Στην περίπτωση που έχουμε πάρει την βοήθεια του Θάνου και μας έχει μιλήσει για την αλήθεια πίσω από αυτή την ιστορία, μετά το animation της Αθηνάς έχουμε την επιλογή να ζητήσουμε εξηγήσεις για την περίεργη συμπεριφορά, αλλά και για τα λεγόμενα του Θάνου περί της συμμετοχής μας στη εξομοίωση.



Η Αθηνά πλέον δεν μπορεί παρά να πεί την αλήθεια...μας αποκαλύπτει ότι ζούμε σε έναν τεχνητό κόσμο και ότι είμαστε μέρος του.

Σε αυτό το σημείο η ιστορία ολοκληρώνεται.



Το σενάριο που περιγράφηκε ήταν μια πολύ απλή υλοποίηση και έχει στόχο την επίδειξη των δυνατοτήτων που μπορούν να έχουν τέτοια συστήματα. Συμπεριλήφθηκαν δράσεις του χρήστη από το πληκτρολόγιο, events που στηρίζονταν στην απόσταση μεταξύ συγκεκριμένων markers, ο έλεγχος και η πυροδότηση animations, ο έλεγχος ροής εμφάνισης μνημάτων, η εξερεύνηση διαφορετικών τοποθεσιών και η αλληλεπίδραση με απτά αντικείμενα. Στην παρούσα υλοποίηση παρατηρούμε ότι υπάρχουν προβλήματα στην απεικόνιση των μνημάτων καθώς πολλές φορές καλύπτονται τα 3D μοντέλα. Τα προβλήματα αυτά όμως οφείλονται κατά κύριο λόγο στην ανάλυση των γυαλιών video που είναι 640x480. Λόγω αυτής της μικρής ανάλυσης, έπρεπε τα μνημάτα να κασκευάζονται με μεγάλες γραμματοσειρές για να είναι ορατά μέσω των γυαλιών.

6: Πλεονεκτήματα – Συγκρίσεις

Μέχρι την τρέχουσα φάση δεν έχει εμφανιστεί κάποιο σύστημα για αφήγηση μεικτής πραγματικότητας που να βάζει τον ίδιο τον χρήστη μέσα στο περιβάλλον και να του επιτρέπει ελευθερία κίνησης. Για τον λόγο αυτό, το προτεινόμενο σύστημα μπορεί να συγκριθεί με άλλα συστήματα που υπάρχουν στο γενικότερο οικοσύστημα της αφήγησης ιστοριών με χρήση της επαυξημένης πραγματικότητας. Κατά τον κ. R. Grasset και το "Mixed Reality Book" [11], η ιστορία εκτυλίσσεται στατικά, κατασκευασμένη μέσα στις σελίδες ενός βιβλίου και προχωρά αλληλουχικά μέσα από την σειρά των σελίδων του βιβλίου. Αντίθετα, σε ιστορίες που εκτυλίσσονται στο παρόν σύστημα ο ίδιος ο χρήστης παίρνει μέρος στην ιστορία και δεν είναι αναγκασμένος να ακολουθήσει κάποια συγκεκριμένη σειρά στις κινήσεις του, καθώς εξερευνά το περιβάλλον του.

Στο προσέγγιση του συστήματος Geist [10] του κ. N. Braun ο χρήστης είναι ελεύθερος να κινηθεί στο περιβάλλον, όμως οι επαυξήσεις που εμφανίζονται μέσα από την προκαθορισμένη συσκευή δεν είναι απευθείας δεμένες με χαρακτηριστικά του περιβάλλοντος. Οι φιγούρες που εμφανίζονται ως φαντάσματα εμφανίζονται όταν ο χρήστης φτάσει σε συγκεκριμένες τοποθεσίες του περιβάλλοντος. Σύμφωνα με παρόν σύστημα τα τρισδιάστατα γραφικά και οι υφές είναι ακριβώς καταχωρημένα σε συγκεκριμένα σημεία. Με τον τρόπο αυτό ο χρήστης δύναται να περπατήσει γύρω από τις επαυξήσεις και να τις παρατηρήσει από διαφορετικές γωνίες και αποστάσεις. Τέλος τα παραδοσιακά συστήματα εικονικής πραγματικότητας έχουν μειονεκτήματα όπως ότι ο χρήστης πρέπει να παραμένει όρθιος και ακίνητος, είτε ότι πρέπει να μπει σε συγκεκριμένο χώρο, είτε ότι πρέπει να συνδεθεί σε κάποια μεγάλη στατική κατασκευή όπως το προαναφερθέν Cyberwalk. Στην παρούσα προσέγγιση αυτό δεν είναι απαραίτητο και ο χρήστης μπορεί να περπατά, να τρέχει, να πηδά ή να σκαρφαλώνει χωρίς να έχει περιορισμούς και αυτό διότι το σύστημα πάντα μεταφέρεται μαζί του σε όλη τη διάρκεια του σεναρίου.

7: Επόμενα Βήματα

Αυτή η αρχιτεκτονική σχεδιάστηκε με στόχο την αντιμετώπιση δυσκολιών και προβλημάτων που συναντώνται στα ευφυή εικονικά περιβάλλοντα. Επί της παρούσης προτείνεται μια προσέγγιση όπου συνδυάζει υπάρχουσες τεχνολογίες και έχει απώτερο σκοπό την βελτίωση της εμπειρίας του χρήστη σε διαδραστικά σενάρια. Αυτή είναι η πρώτη προσπάθεια για δημιουργία ενός απολύτως ελεύθερου περιβάλλοντος αφήγησης, οπότε αρκετά ζητήματα είναι ακόμα ανοιχτά και πρέπει να υλοποιηθούν. Στη συγκεκριμένη φάση είναι διαθέσιμο ένα σενάριο των 10 λεπτών έτσι ώστε να φανούν κάποιες από τις δυνατότητες του συστήματος. Το πρώτο από τα μελλοντικά βήματα είναι η αντικατάσταση των μενού επιλογών με κάποιο module που θα αντιλαμβάνεται διαλογική επικοινωνία του χρήστη με τους βοηθητικούς χαρακτήρες. Το δε δεύτερο βήμα είναι η ενσωμάτωση ενός module για αλληλεπίδραση με τα τρισδιάστατα γραφικά και τις υφές που θα δέχεται ως είσοδο απλές χειρονομίες από τον χρήστη. Μια άλλη προσθήκη που πρέπει οπωσδήποτε να γίνει είναι η ενσωμάτωση ηχητικών εφέ τριών διαστάσεων αλλά και ένας τρόπος για έλεγχο των γεγονότων κεντρικά από την εφαρμογή για κάθε τοποθεσία/δωμάτιο ξεχωριστά. Τέλος, στα μελλοντικά σχέδια είναι η ανάπτυξη ενός μη ελεγχόμενου ρόλου στην εφαρμογή, που θα έχει τις αρμοδιότητες του αφηγητή/σκηνοθέτη και σύμφωνα με κανόνες, θα ελέγχει τα δρώμενα του σεναρίου ανά χρήστη, κάνοντας την εμπειρία του χρήστη πιο ενδιαφέρουσα και ομαλή χωρίς άσκοπες διακοπές ή περιπλανήσεις.

8: Επίλογος

Το συγκεκριμένο project ξείνησε με σκοπό την ενασχόληση με τα μικτά περιβάλλοντα και την διερεύνηση των δυνατοτήτων των διαφόρων υφιστάμενων τεχνολογιών. Μετά από αρκετή έρευνα σχετικά με το πως υλοποιούνται τέτοια συστήματα και μετά από αρκετές δοκιμές σε σχετικά λογισμικά, συνθέθηκε η εφαρμογή που περιγράφηκε παραπάνω. Επιπλέον, αναλύθηκε βήμα προς βήμα, όλη η διαδικασία που πρέπει κανείς να ακολουθήσει για να δημιουργήσει εφαρμογές μεικτής πραγματικότητας. Σίγουρα η εφαρμογή έχει αρκετές ελλείψεις για να αποτελέσει ένα προϊόν αγοράς, όμως, είναι ένα ολοκληρωμένο παράδειγμα και αποτελεί μια απόδειξη του πως είναι οι εφαρμογές μεικτής πραγματικότητας, ποιός είναι ο βασικός εξοπλισμός, ποιά είναι τα προβλήματα και ποιόι οι περιορισμοί των συγκεκριμένων υλοποιήσεων.

Τα εικονικά περιβάλλοντα έχουν εισαχθεί αρκετά στη ζωή μας και πλέον είναι η εποχή που αρχίζουν να γίνονται πραγματικά μέρος της. Όλα δείχνουν ότι χρειαζόμαστε περισσότερη πληροφορία και διάδραση με διάφορα συστήματα στη ζωή μας. Αυτό σημαίνει ότι η επαυξημένη πραγματικότητα θα έρχεται όλο και πιο κοντά μας με διάφορους τρόπους και με πολλές προσεγγίσεις. Το νεότερο παιδί της επαυξημένης πραγματικότητας είναι τα γυαλιά της εταιρείας google που αν και σε πειραματικό στάδιο, έδειξαν πόσο χρήσιμα μπορούν να είναι στον κόσμο όπως τον αντιλαμβανόμαστε σήμερα.

Στην εξιστόρηση οι περισσότερες εξελίξεις γίνονται σε ό,τι έχει να κάνει με τους τομείς του εκπαιδευτικού λογισμικού και του τομέα του video gaming. Βλέπουμε τους μεγαλύτερους κατασκευαστές παιχνιδιών Microsoft, Sony να εκμεταλλεύονται τα συστήματα επαυξημένης πραγματικότητας μέσω του playstation Move και του Kinect και να φτιάχνουν παιχνίδια με διαδραστικές και παράλληλα διασκεδαστικές ιστορίες. Η εποχή των θεατών ως τρίτα πρόσωπα στις αφηγήσεις έχει αρχίσει και ξεπερνιέται. Με τα συστήματα επαυξημένης πραγματικότητας η αλληλεπίδραση προχωρά ένα βήμα παρακάτω. Μελλοντικά, ίσως δούμε συστήματα που θα υποστηρίζουν την αφήγηση και αναπαράσταση μεγάλων ιστορικών γεγονότων που θα παρακολουθούν και θα συμμετάσχουν πολλοί χρήστες ταυτόχρονα και όντας τελείως ανεξάρτητοι μεταξύ τους. Οι δυνατότητες αυτών των συστημάτων φαίνονται πολλές και αξίζει να τις εξερευνίσουμε.

9: Βιβλιογραφία

1. Junya Fujisawa, Hideaki Touyama and Michitaka Hirose, EEG-based navigation of immersing virtual environment using common spatial patterns, IEEE Virtual Reality Conference 2008.
2. C. Guger, C. Holzner, C. Grönegress, G. Edlinger, M. Slater, "Brain-Computer Interface for Virtual Reality Control", In proceedings of European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009.
3. A. De Luca, R. Mattone, P.R.Giordano, H.H. Bulthoff, "Control design and experimental evaluation of the 2D CyberWalk platform", International Conference on Intelligent Robots and Systems, 2009, IROS 2009.
4. J. Blake, H.B. Gurocak, "Haptic glove with MR brakes for virtual reality", Transactions on Mechatronics, IEEE/ASME, Oct 2009.
5. Blair MacIntyre, "Augmenting Intelligent Environments: Augmented Reality as an Interface to Intelligent Environments". In AAAI 1998 Spring Symposia, Intelligent Environments Symposium.
6. M. Assad, D. J. Carmichael, D. Cutting, A. Hudson, "AR phone: Accessible Augmented Reality in the Intelligent Environment", In OZCHI2003, 2003.
7. P. Mistry, P. Maes, SixthSense – A Wearable Gestural Interface. In the Proceedings of ACM SIGGRAPH ASIA 2009 Sketches, Yokohama, Japan, 2009.
8. W. Piekarski, B. Gunther, and B. Thomas, "Integrating Virtual and Augmented Realities in an Outdoor Application", Proc. 2nd Int'l Workshop on Augmented Reality (IWAR 09), IEEE CS Press, 1999, pp. 45–54.
9. R.T. Azuma. A survey of augmented reality. In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), pp. 355-385.
10. N. Braun, "Storytelling in Collaborative Augmented Reality Environments", In proceedings of the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2003.
11. R. Grasset, A. Dünser, M. Billinghurst, Edutainment with a Mixed Reality Book: A visually augmented illustrative childrens' book. ACM International Conference Proceeding Series, vol. 352, 2008.
12. Yoseph Bar-Cohena, Constantinos Mavroidis, Mourad Bouzit, Benjamin Dolgin, Deborah L. Harm, George E. Kopchok and Rodney White, "Virtual reality robotic telesurgery simulations using MEMICA haptic system". In the Proceedings of the SPIE Smart Structures Conference Newport Beach, CA, 2001.
13. Sharlin, E., & Watson, B. (2004). On tangible user interfaces, human and spatiality. Personal and Ubiquitous Computing, 8(5), 338-346.
14. M. Cavazza, F. Charles, S.J. Mead, "Interacting with virtual characters in interactive storytelling". In Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1 (2002), pp. 318-325.
15. T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. W. Picard, A. Pentland, "Augmented Reality Through Wearable Computing", Presence: Teleoperators and Virtual Environments, Vo. 6, Num. 4, August 1997, pp. 386-398.

16. Marc Cavazza, Olivier Martin, Fred Charles, Steven J. Mead, Xavier Marichal, "Interacting with Virtual Agents in Mixed Reality Interactive Storytelling", IVA 2003, pp. 231-235.
17. AR Toolkit, <http://www.hitl.washington.edu/artoolkit/>
18. GLUT - The OpenGL Utility Toolkit, <http://www.opengl.org/resources/libraries/glut/>