



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΚΥΜΑΤΩΝ ΚΑΙ ΟΠΤΙΚΩΝ ΙΝΩΝ

Εντοπισμός Θέσης Χρήστη σε Περιβάλλον Εικονικής Πραγματικότητας με Χρήση του Wii Remote

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Ι. Καραφωτιάς

Επιβλέπων : Νικόλαος Ουζούνογλου
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2009



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΚΥΜΑΤΩΝ ΚΑΙ ΟΠΤΙΚΩΝ ΙΝΩΝ

Εντοπισμός Θέσης Χρήστη σε Περιβάλλον Εικονικής Πραγματικότητας με Χρήση του Wii Remote

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Ι. Καραφωτιάς

Επιβλέπων : Νικόλαος Ουζούνογλου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30^η Οκτωβρίου 2009.

.....
Ν. Ουζούνογλου
Καθηγητής Ε.Μ.Π.

.....
Δ. Κακλαμάνη
Αν. Καθηγήτρια Ε.Μ.Π.

.....
Γ. Ματσόπουλος
Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2009

.....
Γεώργιος Ι. Καραφωτιάς
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γεώργιος Ι. Καραφωτιάς, 2009
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Ο σκοπός της διπλωματικής εργασίας είναι η μελέτη και υλοποίηση λογισμικού ενός συστήματος ανίχνευσης (*tracking*) της θέσης και κατεύθυνσης κάποιου στόχου με χρήση υπέρυθρου φωτός. Οι πληροφορίες για την τυχαία κίνηση του στόχου στο χώρο μεταφέρονται σε πραγματικό χρόνο σε ένα σύστημα Εικονικής Πραγματικότητας όπου ένα συγκεκριμένο αντικείμενο κινείται με τον ίδιο τρόπο μέσα στο Εικονικό Περιβάλλον.

Ο εντοπισμός του χρήστη πραγματοποιείται με τη χρήση διόδων εκπομπής υπέρυθρου φωτός το οποίο ανιχνεύεται από δύο κάμερες ευαίσθητες στο υπέρυθρο φάσμα που περιέχονται μέσα στο Wii Remote. Το υπολογιστικό σύστημα των καμερών στέλνει ασύρματα τα δεδομένα στον ηλεκτρονικό υπολογιστή που χρησιμοποιεί τις βασικές αρχές της επιπολικής γεωμετρίας και στερεοσκοπίας για να απεικονίσει σωστά την τρισδιάστατη κίνηση του ελεγχόμενου αντικειμένου στο εσωτερικό του Εικονικού Κόσμου. Η δημιουργία του Εικονικού Περιβάλλοντος, όλων των αντικειμένων που περιλαμβάνει αλλά και της εφαρμογής έγινε με τη βοήθεια του λογισμικού τρισδιάστατης ανάπτυξης Quest3D και του λογισμικού μοντελισμού 3D Studio Max. Όσον αφορά την εκπομπή του υπέρυθρου φωτός, αυτό γίνεται από μια πλακέτα που κατασκευάσαμε στο εργαστήριο και έχει στα άκρα της τις φωτοδιόδους. Ο χρήστης την κρατά στα χέρια του και έτσι επιτυγχάνεται η ανίχνευση της θέσης και κατεύθυνσής του. Τέλος, η εφαρμογή έχει τη μορφή ενός παιχνιδιού που καλείται να ολοκληρώσει ο χρήστης εντός χρονικού πλαισίου.

Λέξεις Κλειδιά

Εικονική Πραγματικότητα, Εικονικό Περιβάλλον, Ανίχνευση, Εντοπισμός, Ιχνηλάτηση, Υπέρυθρο Φως, Υπέρυθρη Κάμερα, Δίοδοι Εκπομπής Υπέρυθρου Φωτός, Στερεοσκοπία, Επιπολική Γεωμετρία, Quest3D, 3D Studio Max, Wii Remote.

Abstract

The goal of this thesis is the analysis and programming development of a tracking system of a randomly moving target's position and direction, using infrared light. The data about the target's movement in space are transferred in real time to a Virtual Reality system where a particular object is moving in like manner inside the Virtual Environment.

Tracking of user's movement is actualized by two infrared spectrum sensitive cameras, that are contained in the Wii Remote game controller, which trace the infrared light emitted by diodes. The cameras' computational system send wirelessly the data to the computer where algorithms, using the basic principles of epipolar geometry and stereoscopy, correctly replicate the three-dimensional movement of the controlled object inside the Virtual World. The Virtual Environment, including all the objects it contains, as well as the application were created with the help of the visual three-dimensional development software Quest3D and the modeling software 3D Studio Max. Concerning the infrared light emission, it is accomplished by a printed circuit board, with the diodes on its left and right extreme, which we manufactured in the workshop. The user is holding the board with his hands and thus the tracking of his position and direction is achieved. The application is in the form of a game which the user is trying to complete within a time limit.

Key Words

Virtual Reality, Virtual Environment, Tracking, Tracing, Spotting, Infrared Light, Infrared Camera, Infrared Light Emissive Diodes, Stereoscopy, Epipolar Geometry, Quest3D, 3D Studio Max, Wii Remote.

Πρόλογος

Σκοπός της παρούσας εργασίας είναι η μελέτη και η υλοποίηση λογισμικού ενός συστήματος ανίχνευσης της θέσης και κατεύθυνσης κάποιου στόχου με χρήση υπερύθρου φωτός. Ο στόχος στη προκειμένη περίπτωση είναι ο χρήστης που κρατά στα χέρια του μια ράβδο η οποία έχει στα άκρα της διόδους εκπομπής υπερύθρου φωτός. Αυτό μπορεί να εντοπισθεί από μία απλή CCD κάμερα με προσαρμοσμένο ένα φίλτρο που απορρίπτει όλα τα μήκη κύματος εκτός του υπερύθρου. Η κάμερα αυτή περιέχεται μέσα στο Wii Remote και μεταβιβάζει ασύρματα τα δεδομένα σε έναν ηλεκτρονικό υπολογιστή, όπου με τη βοήθεια του λογισμικού Quest3D έχει δημιουργηθεί ένα περιβάλλον εικονικής πραγματικότητας. Έτσι, η θέση και κατεύθυνση της τυχαίας κίνησης του χρήστη απεικονίζεται σε πραγματικό χρόνο στο εικονικό περιβάλλον.

Στο πρώτο κεφάλαιο περιγράφεται η εφαρμογή και αναλύονται τα συστήματα εικονικής πραγματικότητας, με ποιό τρόπο επιτυγχάνεται η διαδραστικότητα του συστήματός μας, δηλαδή η αμφίδρομη επικοινωνία με το χρήστη, όπως και οι βασικές αρχές της στερεοσκοπίας, της τεχνικής μέσω της οποίας γίνεται η τρισδιάστατη αναπαράσταση του χώρου. Τα εργαλεία που χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής εξετάζονται στο επόμενο κεφάλαιο. Αυτά είναι τα λογισμικά πακέτα, Quest3D και 3D Studio Max. Το τρίτο κεφάλαιο είναι και το κεντρικό κεφάλαιο όπου εξηγείται πώς υλοποιήθηκε προγραμματιστικά η εφαρμογή. Πρώτα παρουσιάζονται το γραφικό περιβάλλον και τα αντικείμενα του εικονικού κόσμου. Έπειτα αναφέρεται πώς δημιουργήσαμε τον τρισδιάστατο ήχο για το εικονικό περιβάλλον και τη γραφική διεπαφή με την οποία επικοινωνεί ο χρήστης με την εφαρμογή. Επίσης, αναλύεται η λογική της εφαρμογής και το πώς γίνεται η στερεοσκοπική προβολή. Στο τέλος του κεφαλαίου περιγράφεται η κατασκευή της ηλεκτρονική πλακέτας και οι αλγόριθμοι που αναπαριστούν την κίνηση και περιστροφή στον εικονικό κόσμο απ'την ανίχνευση της κίνησης της πλακέτας στον πραγματικό κόσμο. Για τα πιο σημαντικά σημεία της λογικής του προγράμματος υπάρχουν εικόνες του αντίστοιχου κώδικα συνοδευόμενες από επεξηγηματικά σχόλια. Τα προβλήματα που αντιμετωπίσαμε εκθέτονται στο τέταρτο κεφάλαιο και πώς λύσαμε ορισμένα από αυτά. Στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα και ποιες μελλοντικές εργασίες μπορούν να βελτιώσουν την εφαρμογή.

Η εργασία πραγματοποιήθηκε στο *Εργαστήριο Μικροκυμάτων και Οπτικών Ινών* υπό την επίβλεψη του Καθηγητή Ε.Μ.Π. κ. Ν. Ουζούνογλου και του Ερευνητή Α' ΕΠΙΣΕΥ Δρ. Α. Αμδίτη. Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον κ. Ουζούνογλου, καθώς και τον Δρ. Αμδίτη, για την άριστη συνεργασία μας καθόλη τη διάρκεια της προετοιμασίας της εργασίας. Επίσης για την πολύτιμη βοήθειά τους θα ήθελα να ευχαριστήσω τον Δρ. Ι. Καρασεϊτανίδη, τον υποψήφιο διδάκτορα Π. Ψώνη και τους Ηλεκτρολόγους Μηχανικούς Κ. Λούπο και Ν. Φραγκάκη.

Περιεχόμενα

Περίληψη	5
Abstract	6
Πρόλογος	7
Περιεχόμενα.....	8
Πίνακας Σχημάτων	9
Κεφάλαιο 1 ^ο) ΕΙΣΑΓΩΓΗ	11
1.1) ΕΦΑΡΜΟΓΗ	11
1.2) ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ	12
1.3) ΔΙΑΔΡΑΣΤΙΚΟΤΗΤΑ	18
1.4) ΕΠΙΠΟΛΙΚΗ ΓΕΩΜΕΤΡΙΑ ΚΑΙ ΣΤΕΡΕΟΣΚΟΠΙΑ	20
Κεφάλαιο 2 ^ο) ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ	25
2.1) QUEST3D.....	25
2.1.1) Γενικά.....	25
2.1.2) Τρόπος λειτουργίας.....	25
2.2) 3D STUDIO MAX	28
2.2.1) Γενικά.....	28
2.2.2) Τρόπος λειτουργίας.....	28
Κεφάλαιο 3 ^ο) ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ.....	31
3.1) ΥΛΟΠΟΙΗΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ	32
3.1.1) Γραφικό Περιβάλλον	32
3.1.2) Βοηθητικά Αντικείμενα	40
3.1.3) Ήχος	45
3.2) ΥΛΟΠΟΙΗΣΗ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ.....	47
3.2.1) Γραφική Διεπαφή.....	47
3.2.2) Στερεοσκοπική Προβολή	57
3.2.3) Λογική Εφαρμογής	58
3.3) ΑΝΙΧΝΕΥΣΗ ΡΑΒΔΟΥ.....	64
3.3.1) Ηλεκτρονική Πλακέτα	64
3.3.2) Κίνηση στους άξονες $x'x$, $y'y$, $z'z$	66
3.3.3) Περιστροφή γύρω απ'τους άξονες $x'x$, $y'y$, $z'z$	76
3.4) ΟΡΓΑΝΩΣΗ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗ ΣΥΣΤΗΜΑΤΟΣ	79
Κεφάλαιο 4 ^ο) ΠΡΟΒΛΗΜΑΤΑ	85
Κεφάλαιο 5 ^ο) ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ	89
Βιβλιογραφία	91
Παράρτημα	93
Έντυπο τεχνικών δεδομένων (data sheet) της διόδου εκπομπής φωτός OP293.....	93

Πίνακας Σχημάτων

Κεφάλαιο 1^ο)

ΕΙΚΟΝΑ 1.1 : ΑΠΤΙΚΑ ΓΑΝΤΙΑ	13
ΕΙΚΟΝΑ 1.2 : 3Δ ΠΟΝΤΙΚΙ	13
ΕΙΚΟΝΑ 1.3 : 3Δ ΣΤΥΛΟ.....	13
ΕΙΚΟΝΑ 1.4 : ΠΑΡΑΔΕΙΓΜΑ ΗΜD.....	14
ΕΙΚΟΝΑ 1.5 : ΠΑΡΑΔΕΙΓΜΑ ΒΟΟΜ	15
ΕΙΚΟΝΑ 1.6 : ΠΑΡΑΔΕΙΓΜΑ CAVE	15
ΕΙΚΟΝΑ 1.7 : ΌΨΕΙΣ ΧΕΙΡΙΣΤΗΡΙΟΥ WII REMOTE.....	18
ΕΙΚΟΝΑ 1.8 : ΠΑΡΑΔΕΙΓΜΑ SENSOR BAR.....	19
ΕΙΚΟΝΑ 1.9 : ΜΟΝΤΕΛΟ ΚΑΜΕΡΑΣ ΛΕΠΤΗΣ ΟΠΗΣ ΜΕ ΑΝΤΕΣΤΡΑΜΜΕΝΟ ΕΠΙΠΕΔΟ ΠΡΟΒΟΛΗΣ	20
ΕΙΚΟΝΑ 1.10 : ΜΟΝΤΕΛΟ ΚΑΜΕΡΑΣ ΛΕΠΤΗΣ ΟΠΗΣ ΜΕ ΟΡΘΟ ΕΠΙΠΕΔΟ ΠΡΟΒΟΛΗΣ	20
ΕΙΚΟΝΑ 1.11 : ΕΠΙΠΟΛΙΚΗ ΓΕΩΜΕΤΡΙΑ ΤΥΧΑΙΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ	21
ΕΙΚΟΝΑ 1.12 : ΕΠΙΠΟΛΙΚΗ ΓΕΩΜΕΤΡΙΑ ΚΑΝΟΝΙΚΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ	22
ΕΙΚΟΝΑ 1.13 : ΚΑΤΟΨΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΝΟΝΙΚΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ (ΠΑΡΑΛΛΗΛΕΣ ΚΑΜΕΡΕΣ)	23

Κεφάλαιο 2^ο)

ΕΙΚΟΝΑ 2.1 : ΠΑΡΑΔΕΙΓΜΑ ΤΗΣ ΔΙΕΠΑΦΗΣ (INTERFACE) ΤΟΥ QUEST3D.....	26
ΕΙΚΟΝΑ 2.2 : ΠΑΡΑΔΕΙΓΜΑ PUBLIC CALLER ΚΑΙ PUBLIC CHANNEL	27
ΕΙΚΟΝΑ 2.3 : ΠΑΡΑΔΕΙΓΜΑ ΤΗΣ ΔΙΕΠΑΦΗΣ (INTERFACE) ΤΟΥ 3D STUDIO MAX.....	28
ΕΙΚΟΝΑ 2.4 : ΠΑΡΑΔΕΙΓΜΑ ΠΛΕΓΜΑΤΟΣ ΔΙΠΟΔΟΥ ΚΑΙ ΕΦΑΡΜΟΓΗΣ ΥΦΩΝ ΣΕ ΑΥΤΟ.....	30

Κεφάλαιο 3^ο)

ΕΙΚΟΝΑ 3.1 : ΣΤΙΓΜΙΟΤΥΠΟ ΟΘΟΝΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	31
ΕΙΚΟΝΑ 3.2 : ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΟΥΡΑΝΟΣ.....	32
ΕΙΚΟΝΑ 3.3 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΟΥΡΑΝΟΥ	33
ΕΙΚΟΝΑ 3.4 : ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΕΔΑΦΟΣ	34
ΕΙΚΟΝΑ 3.5 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΕΔΑΦΟΥΣ	34
ΕΙΚΟΝΑ 3.6 : ΚΟΚΚΙΝΟΣ ΣΤΟΧΟΣ ΚΑΙ ΤΟ ΠΛΑΙΣΙΟ ΤΟΥ	35
ΕΙΚΟΝΑ 3.7 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΣΤΟΧΟΥ ΚΑΙ ΠΛΑΙΣΙΟΥ	36
ΕΙΚΟΝΑ 3.8 : ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΚΟΚΚΙΝΗ ΣΦΑΙΡΑ	37
ΕΙΚΟΝΑ 3.9 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΚΟΚΚΙΝΗΣ ΣΦΑΙΡΑΣ.....	37
ΕΙΚΟΝΑ 3.10 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΣΥΣΤΗΜΑΤΟΣ ΣΩΜΑΤΙΔΙΩΝ	38
ΕΙΚΟΝΑ 3.11 : ΙΔΙΟΤΗΤΕΣ ΣΥΣΤΗΜΑΤΟΣ ΣΩΜΑΤΙΔΙΩΝ	38
ΕΙΚΟΝΑ 3.12 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΥΠΟΔΟΧΕΑ	39
ΕΙΚΟΝΑ 3.13 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΦΩΤΕΙΝΗΣ ΠΗΓΗΣ	40
ΕΙΚΟΝΑ 3.14 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΣΚΙΩΝ.....	41
ΕΙΚΟΝΑ 3.15 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΑΝΤΙΚΕΙΜΕΝΟΥ BASKET_SHADOW	42
ΕΙΚΟΝΑ 3.16 : ΠΡΟΟΠΤΙΚΗ ΤΗΣ ΚΙΝΗΤΗΣ ΚΑΜΕΡΑΣ	43
ΕΙΚΟΝΑ 3.17 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΤΩΝ ΔΥΟ ΚΑΜΕΡΩΝ	43
ΕΙΚΟΝΑ 3.18 : RENDERING ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ.....	44
ΕΙΚΟΝΑ 3.19 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΤΩΝ ΗΧΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	45
ΕΙΚΟΝΑ 3.20 : ΙΔΙΟΤΗΤΕΣ ΚΑΝΑΛΙΟΥ SOUND FILE	46
ΕΙΚΟΝΑ 3.21 : ΚΩΔΙΚΑΣ ΤΩΝ ΣΚΗΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	47

ΕΙΚΟΝΑ 3.22 : ΣΚΑΡΙΦΗΜΑ ΤΩΝ ΣΚΗΝΩΝ ΠΟΥ ΕΛΕΓΧΕΙ Η ΜΗΧΑΝΗ ΚΑΤΑΣΤΑΣΕΩΝ	48
ΕΙΚΟΝΑ 3.23 : ΚΩΔΙΚΑΣ ΕΜΦΑΝΙΣΗΣ ΚΕΙΜΕΝΟΥ	49
ΕΙΚΟΝΑ 3.24 : ΙΔΙΟΤΗΤΕΣ ΚΑΝΑΛΙΟΥ ΜΑΘΗΜΑΤΙΚΗΣ ΕΚΦΡΑΣΗΣ	50
ΕΙΚΟΝΑ 3.25 : ΠΑΡΑΔΕΙΓΜΑ ΕΝΟΠΟΙΗΜΕΝΟΥ (MERGED) ΚΕΙΜΕΝΟΥ	51
ΕΙΚΟΝΑ 3.26 : ΠΑΡΑΔΕΙΓΜΑ ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ ΚΑΙ ΚΟΥΜΠΙΟΥ.....	52
ΕΙΚΟΝΑ 3.27 : ΚΩΔΙΚΑΣ ΚΑΤΑΣΚΕΥΗΣ ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ ΚΑΙ ΚΟΥΜΠΙΟΥ	52
ΕΙΚΟΝΑ 3.28 : ΚΕΙΜΕΝΟ ΚΑΙ ΧΡΟΝΟΣ ΣΤΗΝ ΠΡΟΟΠΤΙΚΗ ΘΕΑΣΗΣ ΤΟΥ ΧΡΗΣΤΗ.....	53
ΕΙΚΟΝΑ 3.29 : ΚΩΔΙΚΕΣ ΚΑΤΑΣΚΕΥΗΣ ΚΕΙΜΕΝΟΥ ΚΑΙ ΧΡΟΝΟΥ	54
ΕΙΚΟΝΑ 3.30 : Η ΥΦΗ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ ΠΟΥ ΑΠΕΙΚΟΝΙΖΕΙ ΤΟ ΧΡΟΝΟ.....	55
ΕΙΚΟΝΑ 3.31 : ΚΩΔΙΚΑΣ ΔΗΜΙΟΥΡΓΙΑΣ ΔΥΟ ΕΙΚΟΝΩΝ ΓΙΑ ΣΤΕΡΕΟΣΚΟΠΙΚΗ ΠΡΟΒΟΛΗ	57
ΕΙΚΟΝΑ 3.32 : ΚΩΔΙΚΑΣ ΕΛΕΓΧΟΥ ΤΟΥ ΧΡΟΝΟΥ	58
ΕΙΚΟΝΑ 3.33 : ΚΩΔΙΚΑΣ ΛΟΓΙΚΗΣ ΤΗΣ ΚΥΡΙΑΣ ΣΚΗΝΗΣ.....	59
ΕΙΚΟΝΑ 3.34 : ΚΩΔΙΚΑΣ ΕΛΕΓΧΟΥ ΣΥΓΚΡΟΥΣΕΩΝ.....	60
ΕΙΚΟΝΑ 3.35 : ΚΩΔΙΚΑΣ ΕΝΗΜΕΡΩΣΗΣ ΤΟΥ ΣΚΟΡ.....	61
ΕΙΚΟΝΑ 3.36 : ΙΔΙΟΤΗΤΕΣ ΚΑΝΑΛΙΟΥ TRIGGER.....	61
ΕΙΚΟΝΑ 3.37 : ΚΩΔΙΚΑΣ ΤΩΝ ΣΤΑΔΙΩΝ ΤΗΣ ΛΟΓΙΚΗΣ ΤΗΣ ΚΥΡΙΑΣ ΣΚΗΝΗΣ.....	62
ΕΙΚΟΝΑ 3.38 : ΚΥΚΛΩΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ ΠΛΑΚΕΤΑΣ.....	65
ΕΙΚΟΝΑ 3.39 : ΚΩΔΙΚΑΣ ΣΥΝΔΕΣΗΣ ΤΩΝ ΔΥΟ WII REMOTE	66
ΕΙΚΟΝΑ 3.40 : ΚΩΔΙΚΑΣ ΚΑΝΑΛΙΩΝ ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕ ΤΑ ΔΥΟ WII REMOTE.....	67
ΕΙΚΟΝΑ 3.41 : ΚΩΔΙΚΑΣ ΕΥΡΕΣΗΣ ΤΟΥ ΚΕΝΤΡΟΥ ΤΗΣ ΡΑΒΔΟΥ ΓΙΑ ΚΑΘΕ ΜΙΑ ΑΠ'ΤΙΣ ΔΥΟ ΚΑΜΕΡΕΣ	68
ΕΙΚΟΝΑ 3.42 : ΚΩΔΙΚΑΣ ΕΥΡΕΣΗΣ ΤΟΥ ΚΕΝΤΡΟΥ ΤΗΣ ΡΑΒΔΟΥ	69
ΕΙΚΟΝΑ 3.43 : ΣΥΣΤΗΜΑ ΝΕΩΝ ΣΥΝΤΕΤΑΓΜΕΝΩΝ.....	70
ΕΙΚΟΝΑ 3.44 : ΚΩΔΙΚΑΣ ΥΠΟΛΟΓΙΣΜΟΥ ΑΠΟΣΤΑΣΗΣ ΤΩΝ ΔΥΟ ΑΚΡΩΝ ΤΗΣ ΡΑΒΔΟΥ ΑΠ'ΤΙΣ ΚΑΜΕΡΕΣ	71
ΕΙΚΟΝΑ 3.45 : ΚΩΔΙΚΑΣ ΥΠΟΛΟΓΙΣΜΟΥ ΑΠΟΣΤΑΣΗΣ ΤΟΥ ΚΕΝΤΡΟΥ ΤΗΣ ΡΑΒΔΟΥ ΑΠ'ΤΙΣ ΚΑΜΕΡΕΣ	72
ΕΙΚΟΝΑ 3.46 : ΚΩΔΙΚΑΣ ΥΛΟΠΟΙΗΣΗΣ ΜΕΤΑΚΙΝΗΣΗΣ ΤΗΣ ΚΙΝΗΤΗΣ ΚΑΜΕΡΑΣ	73
ΕΙΚΟΝΑ 3.47 : ΠΑΡΑΔΕΙΓΜΑ ΕΝΟΣ ΚΑΝΑΛΙΟΥ ENVELOPE.....	75
ΕΙΚΟΝΑ 3.48 : ΚΩΔΙΚΑΣ ΥΠΟΛΟΓΙΣΜΟΥ ΓΩΝΙΑΣ ΠΕΡΙΣΤΡΟΦΗΣ ΓΥΡΩ ΑΠΟ ΤΟΝ ΑΞΟΝΑ Z'Z'.....	76
ΕΙΚΟΝΑ 3.49 : ΚΩΔΙΚΑΣ ΥΠΟΛΟΓΙΣΜΟΥ ΓΩΝΙΑΣ ΠΕΡΙΣΤΡΟΦΗΣ ΓΥΡΩ ΑΠΟ ΤΟΝ ΑΞΟΝΑ Y'Y'.....	78
ΕΙΚΟΝΑ 3.50 : ΚΩΔΙΚΑΣ ΥΛΟΠΟΙΗΣΗΣ ΠΕΡΙΣΤΡΟΦΗΣ ΤΗΣ ΚΙΝΗΤΗΣ ΚΑΜΕΡΑΣ.....	78
ΕΙΚΟΝΑ 3.51 : ΚΑΤΟΨΗ ΣΥΣΤΗΜΑΤΟΣ ΤΥΧΑΙΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ (ΣΥΓΚΛΙΝΟΥΣΕΣ ΚΑΜΕΡΕΣ).....	79

Κεφάλαιο 4^ο)

ΕΙΚΟΝΑ 4.1 : ΣΚΑΡΙΦΗΜΑ ΤΩΝ ΠΕΔΙΩΝ ΑΝΙΧΝΕΥΣΗΣ ΠΑΡΑΛΛΗΛΩΝ ΚΑΜΕΡΩΝ.....	86
ΕΙΚΟΝΑ 4.2 : ΣΚΑΡΙΦΗΜΑ ΤΩΝ ΠΕΔΙΩΝ ΑΝΙΧΝΕΥΣΗΣ ΣΥΓΚΛΙΝΟΥΣΩΝ ΚΑΜΕΡΩΝ	87

Κεφάλαιο 1^ο) ΕΙΣΑΓΩΓΗ

1.1) Εφαρμογή

Όπως αναφέρθηκε και στον πρόλογο, ο σκοπός της εργασίας είναι η μελέτη και η υλοποίηση λογισμικού ενός συστήματος εντοπισμού της θέσης και κατεύθυνσης ενός στόχου με χρήση υπερύθρου φωτός. Ως πηγές του υπερύθρου φωτός χρησιμοποιούμε ειδικές διόδους (LED) προσαρμοσμένες στα άκρα μίας ράβδου. Ο χρήστης κρατάει στα χέρια του τη ράβδο και οι θέσεις στο χώρο των άκρων της, όπου βρίσκονται οι φωτοδιόδοι, ανιχνεύονται από την κάμερα που βρίσκεται μέσα στο Wii Remote. Προτιμήσαμε τη χρήση του Wii Remote καθώς είναι μια πολύ φθηνή και εύκολα προσβάσιμη λύση. Η κάμερα του Wii Remote μας δίνει πληροφορίες για τις συντεταγμένες x και y της προβολής του ανιχνευόμενου σημείου στο επίπεδο προβολής της κάμερας. Όμως δεν έχουμε δεδομένα όσον αφορά την απόσταση του σημείου, άρα και του χρήστη, από την κάμερα. Γι' αυτό το λόγο χρησιμοποιούμε δύο κάμερες Wii Remote, ώστε εκμεταλλευόμενοι τις εξισώσεις της στερεοσκοπίας να υπολογίσουμε αυτή την απόσταση.

Τα δεδομένα, που απαιτούνται για να γίνει η μέτρηση της απόστασης, μεταφέρονται ασύρματα απ' τις δύο κάμερες προς το λογισμικό πακέτο Quest3D, το οποίο με τον κατάλληλο κώδικα κάνει τον αντίστοιχο υπολογισμό. Το Quest3D επίσης είναι υπεύθυνο για την δημιουργία του εικονικού περιβάλλοντος και τον έλεγχο όλων των αντικειμένων που περιέχονται σε αυτό. Η υλοποίηση λογισμικού της εργασίας πρόκειται για μια μορφή παιχνιδιού, όπου επιλέγεται τυχαία μία σφαίρα συγκεκριμένου χρώματος και εκτοξεύεται προς το χρήστη μέσα στο εικονικό περιβάλλον. Αυτός κινούμενος στον πραγματικό χώρο και κρατώντας τη ράβδο, προσπαθεί να τοποθετήσει το αντικείμενο που ελέγχει στο εικονικό περιβάλλον σε τέτοια θέση και με τη σωστή γωνία, ώστε να εγκλωβίσει την σφαίρα πριν αυτή ακουμπήσει το εικονικό έδαφος. Έπειτα, πρέπει να μεταφέρει την εγκλωβισμένη σφαίρα και να στρέψει το αντικείμενο με τέτοιο τρόπο για να αποθέσει τη σφαίρα σε ένα πλαίσιο ίδιου χρώματος με τη συγκεκριμένη σφαίρα. Αυτές οι ενέργειες πρέπει να γίνουν μέσα σε ένα χρονικό διάστημα που έχει αρχικά επιλεχθεί απ' το χρήστη και στο πέρας του οποίου εκτοξεύεται μια καινούργια σφαίρα. Όταν τελειώσει ο συνολικός χρόνος του παιχνιδιού, παρουσιάζονται τα αποτελέσματα, δηλαδή πόσες σφαίρες κατάφερε ο χρήστης να αποθέσει σωστά.

Τα αντικείμενα απλού σχήματος, όπως οι σφαίρες που εκτοξεύονται, έχουν δημιουργηθεί στο Quest3D. Όμως, τα πιο πολύπλοκου σχήματος αντικείμενα, όπως αυτό που ελέγχει ο χρήστης, έχουν δημιουργηθεί στο λογισμικό 3D Studio Max και εισάγονται μετέπειτα στο Quest3D για να τα εμφανίσει στο εικονικό περιβάλλον. Επίσης, για να επιτύχουμε μεγαλύτερη γωνία του πεδίου αντίληψης, αντί για την ράβδο με τα LED, του εμπορίου, φτιάξαμε μια ηλεκτρονική πλακέτα και τοποθετήσαμε διόδους υπερύθρου φωτός του ίδιου μήκους κύματος με αυτό που ανιχνεύουν οι κάμερες των Wii Remote.

1.2) Εικονική Πραγματικότητα

Η *Εικονική Πραγματικότητα (Virtual Reality – VR)* ορίζεται [1] ως ένα τρισδιάστατο Εικονικό Περιβάλλον που αναπαράγεται αναδραστικά στις ενέργειες του χρήστη σε πραγματικό χρόνο (real time). Συνήθως έχει ως σκοπό την εκπαίδευση, διευκόλυνση της εργασίας ή ακόμα και τη ψυχαγωγία του χρήστη του.

Λέγοντας *Εικονικό Περιβάλλον*, εννοούμε έναν Εικονικό Κόσμο, ο οποίος δημιουργείται από ηλεκτρονικό υπολογιστή και είτε είναι εξομοίωση πραγματικού περιβάλλοντος, ή είναι ένα φανταστικό περιβάλλον που εξυπηρετεί το σκοπό της συγκεκριμένης Εικονικής Πραγματικότητας. Το Εικονικό Περιβάλλον διέπεται από τρεις θεμελιώδεις έννοιες : [2]

- Εμβύθιση (Immersion) - ψευδαίσθηση της ύπαρξης του χρήστη μέσα σε ένα Εικονικό Περιβάλλον.
- Αλληλεπίδραση (Interaction) - η δυναμική αναπαραγωγή σε πραγματικό χρόνο του Εικονικού Περιβάλλοντος ανάλογα με τις ενέργειες και κινήσεις του χρήστη.
- Πλοήγηση (Navigation) – πρέπει να γίνεται με όσο το δυνατόν πιο διαισθητικό και φυσικό τρόπο.

Τα τελευταία χρόνια, γίνονται πολλά βήματα προόδου της *Επαυξημένης Πραγματικότητας (Augmented Reality – AR)* [3], η οποία είναι μια εξέλιξη της Εικονικής Πραγματικότητας. Στην Επαυξημένη Πραγματικότητα ο χρήστης διατηρεί το κανονικό οπτικό του πεδίο στον πραγματικό κόσμο αλλά επιπροσθέτως λαμβάνει εικόνες και πληροφορίες σε μία οθόνη που βρίσκεται μέσα στο οπτικό του πεδίο και κινείται μαζί του. Χρησιμοποιείται συχνά στην εκπαίδευση πιλότων σε πρωτότυπα αεροσκάφη και στην ιατρική όπου, παραδείγματος χάρη, ο χειρουργός ιατρός παρακολουθεί τον χειρουργούμενο ασθενή στον πραγματικό κόσμο και παράλληλα ενημερώνεται για τις κυριότερες ιατρικές παραμέτρους του απ' την οθόνη.

Η Εικονική Πραγματικότητα βασίζεται στη δημιουργία εικόνων που φαίνονται τρισδιάστατες στο χρήστη. Αυτή η ψευδαίσθηση του βάθους επιτυγχάνεται όταν προβάλλονται στον αριστερό και δεξιό οφθαλμό δύο διαφορετικές προοπτικές της ίδιας εικόνας. Στο παρελθόν, αυτό γινόταν εφικτό χρωματίζοντας μπλε την εικόνα που προορίζεται για το δεξιό οφθαλμό και κόκκινη την εικόνα για τον αριστερό. Τότε, για το χρήστη που φοράει γυαλιά με έναν μπλε και έναν κόκκινο φακό, η εικόνα που είναι χρωματισμένη με το ίδιο χρώμα με έναν φακό είναι ορατή ως προς αυτόν τον φακό και αόρατη ως προς τον άλλο. Το μειονέκτημα αυτής της μεθόδου είναι η έντονη αλλοίωση των χρωμάτων της εικόνας. Σήμερα, με την εφαρμογή της ψηφιακής τεχνολογίας, η ψευδαίσθηση του βάθους επιτυγχάνεται με τις μεθόδους Passive Stereo και Active Stereo : [4]

- Στην Passive Stereo χρησιμοποιούνται δύο προβολικά συστήματα εξοπλισμένα με φίλτρα πόλωσης της εικόνας. Η εικόνα που προβάλλεται απ' το ένα σύστημα είναι πολωμένη καθέτως και η εικόνα που προβάλλεται απ' το άλλο σύστημα είναι πολωμένη οριζοντίως. Ομοίως πολωμένοι είναι και οι φακοί των γυαλιών που φοράει ο χρήστης. Έτσι, ο κάθε φακός επιτρέπει τη διέλευση μόνο της εικόνας με την ίδια πόλωση και τελικά ο χρήστης λαμβάνει δύο διαφορετικές εικόνες. Το μειονέκτημα αυτής της μεθόδου είναι ότι ο χρήστης πρέπει να προσέχει να μη στρέφει το κεφάλι του σε μεγάλη γωνία ώστε να μην επηρεάσει την ευθυγράμμιση των πολωτικών φίλτρων.

- Στην Active Stereo ο χρήστης φοράει ειδικά γυαλιά τα οποία χρησιμοποιούν για φακούς ένα ζεύγος ηλεκτρονικά ελεγχόμενων διαφραγμάτων φωτός τύπου LCD. Τα διαφράγματα αυτά ανοιγοκλείνουν εναλλάξ και είναι συγχρονισμένα με το προβολικό σύστημα. Αυτό τότε προβάλλει διαδοχικά μία εικόνα που προορίζεται για τον αριστερό οφθαλμό και μία που προορίζεται για το δεξί. Τα διαφράγματα είναι συγχρονισμένα με τον ρυθμό προβολής των εικόνων, οπότε όταν προβάλλεται η εικόνα που προορίζεται για τον αριστερό οφθαλμό, το διάφραγμα μπροστά απ' το δεξιό είναι κλειστό. Έπειτα, ανοίγει το διάφραγμα του δεξιού οφθαλμού και κλείνει το διάφραγμα του αριστερού, οπότε η επόμενη εικόνα που προβάλλεται γίνεται αντιληπτή μόνο απ' το δεξιό οφθαλμό. Το μειονέκτημα αυτής της μεθόδου είναι η αγορά ενός προβολικού συστήματος ικανού να προβάλλει εικόνες με ρυθμό 100 Hz ή περισσότερο ώστε να μην παρατηρείται τρέμουλο της εικόνας που αντιλαμβάνεται ο χρήστης.

Οι ηλεκτρονικοί υπολογιστές, δηλαδή το σύνολο hardware και software που δημιουργούν τον Εικονικό Κόσμο, ορίζονται ως το *Εικονικό Σύστημα*. Μέσω αυτού γίνεται και η αλληλεπίδραση του χρήστη με το Εικονικό Περιβάλλον. Οι εξής τεχνολογίες είναι απαραίτητες στο Εικονικό Σύστημα : [1]

- Οπτικές (visual) και ακουστικές (aural) διατάξεις που επιτυγχάνουν την εμπύθιση του χρήστη στον Εικονικό Κόσμο αλλά και εμποδίζουν εξωτερικά ερεθίσματα απ' τον πραγματικό κόσμο.
- Η μηχανή γραφικών (graphics engine) που είναι υπεύθυνη να δημιουργεί και να παρουσιάζει, συνήθως σε πραγματικό χρόνο, τον Εικονικό Κόσμο.
- Η βάση δεδομένων στην οποία αποθηκεύονται τα λεπτομερή και ρεαλιστικά τρισδιάστατα μοντέλα του Εικονικού Κόσμου.
- Υποσύστημα ανίχνευσης (tracking) της θέσης και προσανατολισμού του χρήστη που τροφοδοτεί με δεδομένα τον ηλεκτρονικό υπολογιστή και την μηχανή γραφικών ώστε οι κινήσεις του χρήστη στον πραγματικό κόσμο να αναπαριστώνται σωστά στον Εικονικό Κόσμο.
- Διατάξεις αφής (haptic) [5], όπως τα γάντια δεδομένων ή απτικά γάντια (data gloves) και το τρισδιάστατο ποντίκι και στυλό (3D mouse, pen) με τα οποία ο χρήστης μπορεί να επιδράσει στα εικονικά αντικείμενα.



Εικόνα 1.1
Απτικά Γάντια



Εικόνα 1.2
3Δ Ποντίκι



Εικόνα 1.3
3Δ Στυλό

Για όσο το δυνατόν πιο πιστή και ρεαλιστική εμπύθιση, το κυριότερο από τα παραπάνω είναι οι οπτικοακουστικές διατάξεις με τις πιο σημαντικές από αυτές τις : [6,7]

- ❖ *Οθόνη Κάσκας (Head Mounted Display – HMD)*. Μια συσκευή που μοιάζει με κάσκα, φοριέται στο κεφάλι και έχει στο εσωτερικό της δύο μικρές LCD ή CRT οθόνες. Οι οθόνες βρίσκονται μπροστά από τα μάτια του χρήστη και σε μικρή απόσταση έτσι ώστε ο κάθε οφθαλμός να βλέπει μόνο την οθόνη που του αντιστοιχεί. Ακόμα, οι εικόνες προβάλλονται υπό ελαφρά χωρική απόσταση μεταξύ τους ώστε όταν ο εγκέφαλος του χρήστη συνδιάζει τις δύο εικόνες να αντιλαμβάνεται και την αίσθηση του βάθους. Τις πιο πολλές φορές, η συσκευή HMD έχει και έναν ανιχνευτή θέσης και προσανατολισμού (motion tracker) ο οποίος ενημερώνει συνέχεια το υπολογιστικό σύστημα για την εκάστοτε θέση και προσανατολισμό της κεφαλής. Με τη σειρά του το υπολογιστικό σύστημα μεταβάλλει συνέχεια τις προβαλλόμενες εικόνες σύμφωνα με τα δεδομένα. Συνήθως, οι συσκευές HMD είναι αδιαφανείς και έτσι ο χρήστης δε λαμβάνει καθόλου ερεθίσματα απ'τον πραγματικό κόσμο. Υπάρχουν, όμως και οι διαφανείς HMD, οι οποίες αποτελούνται ουσιαστικά από μία οθόνη που βρίσκεται σε τέτοια απόσταση από τα μάτια ώστε να μην εμποδίζει την παρατήρηση του πραγματικού κόσμου. Αυτές είναι που χρησιμοποιούνται συνήθως στην περίπτωση της Επαυξημένης Πραγματικότητας που αναφέρθηκε προηγουμένως.



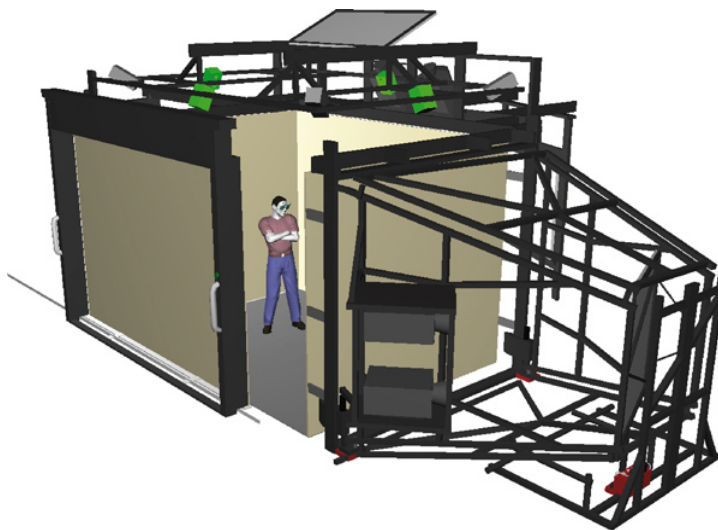
Εικόνα 1.4
Παράδειγμα HMD

- ❖ *Διοπτρική Περισκοπική Οθόνη (Binocular Omni-Orientation Monitor - BOOM).* Η αρχή λειτουργίας της συσκευής είναι ίδια με την HMD. Η διαφορά έγκειται στο ότι το ζεύγος των οθονών δε βρίσκεται στο εσωτερικό της κάσκας που στερεώνεται στο κεφάλι του χρήστη, αλλά βρίσκεται στο άκρο ενός άξονα πολλών συνδέσμων. Ο χρήστης εφαρμόζει το κουτί που περιέχει τις δύο οθόνες μπροστά απ'το πρόσωπό του και μετακινώντας με τα χέρια του ή με ειδικό χειριστήριο (joystick) τον ειδικό άξονα, μπορεί και αλλάζει τη θέση και το προσανατολισμό μέσα στον Εικονικό Κόσμο.



Εικόνα 1.5
Παράδειγμα BOOM

- ❖ *Αυτόματο Εικονικό Περιβάλλον Σπηλαιού (Cave Automatic Virtual Environment – CAVE).* Είναι ένας κύβος σε διαστάσεις κανονικού δωματίου όπου ειδικοί προβολείς προβάλλουν εικόνες πάνω στους τέσσερις τοίχους, στο δάπεδο και στην οροφή του δωματίου. Η ψευδαίσθηση του βάθους επιτυγχάνεται είτε με τη μέθοδο Active Stereo είτε με τη Passive Stereo. Η προβολή των εικόνων γίνεται πίσω απ'τους τοίχους, ώστε να μην εμποδίζεται απ'τη παρουσία του χρήστη στο εσωτερικό του κύβου. Πάλι, το υπολογιστικό σύστημα σε συνδιασμό με έναν ανιχνευτή θέσης και προσανατολισμού αναπροσαρμόζει την όψη που προβάλλεται στερεοσκοπικά στο χρήστη.



Εικόνα 1.6
Παράδειγμα CAVE

- ❖ *Επιτραπέζια συστήματα (desktop)* όταν χρησιμοποιείται απλά μια οθόνη υπολογιστή (CRT, TFT ή LCD). Η αλληλεπίδραση του χρήστη με το σύστημα γίνεται μέσω συσκευών εισόδου του υπολογιστή. Αυτές μπορεί να είναι οι συνηθισμένες συσκευές εισόδου όπως το πληκτρολόγιο και το κοινό ποντίκι αλλά και οι προαναφερθείσες διατάξεις αφής (απτικά γάντια, 3D ποντίκι και στυλό).

Λαμβάνοντας υπ' όψη τις μεγάλες υπολογιστικές δυνατότητες των σημερινών ηλεκτρονικών υπολογιστών, μπορούμε εύκολα να φτιάξουμε οποιοδήποτε Εικονικό Περιβάλλον με αποτέλεσμα οι εφαρμογές της Εικονικής Πραγματικότητας να χρησιμοποιούνται σε πολλαπλά πεδία. Θα αναφέρουμε τα κυριότερα από αυτά : [1,5]

- *Εκπαίδευση.* Είναι ο τομέας που χρησιμοποιείται περισσότερο η Εικονική Πραγματικότητα. Κυρίως γίνεται με την προσομοίωση (simulation) ενός πραγματικού ή φανταστικού περιβάλλοντος, στο οποίο ο χρήστης καλείται να ανταπεξέλθει στις απαιτήσεις του συγκεκριμένου Εικονικού Κόσμου. Το πιο συνηθισμένο παράδειγμα είναι η εκπαίδευση ενός πιλότου πολεμικού αεροσκάφους σε συνθήκες μάχης. Ο χρήστης εισέρχεται στο εικονικό αεροσκάφος και εκπαιδεύεται σε αερομαχίες εναντίον εικονικών εχθρικών αεροσκαφών, χωρίς να υπάρχει κίνδυνος να τραυματιστεί ή να καταστραφεί ένα πραγματικό αεροσκάφος πολύ μεγάλης αξίας. Επίσης, καταλαβαίνουμε τη σημασία της Εικονικής Πραγματικότητας στην εκπαίδευση αν αναλογιστούμε ότι μπορούμε να οπτικοποιήσουμε (visualization) είτε αφηρημένες έννοιες για να καταφέρουμε να τις αντιληφθούμε καλύτερα, παραδείγματος χάρη η οπτικοποίηση του ηλιακού μας συστήματος ώστε ένα παιδί να το κατανοήσει ευκολότερα, είτε καταστάσεις επικίνδυνων συνθηκών, όπως το πρωτόκολλο ενεργειών σε κατάσταση συναγερμού ενός πυρηνικού εργοστασίου.
- *Κατασκευές.* Πάλι βασιζόμαστε στην οπτικοποίηση και φτιάχνουμε ένα τρισδιάστατο μοντέλο της κατασκευής. Ο χρήστης μπορεί να παρατηρήσει το μοντέλο εξωτερικά ή να περιηγηθεί (spatial navigation) στο εσωτερικό του βλέποντας ακόμα πιο πολλές λεπτομέρειες. Χρησιμοποιείται κυρίως στην ναυπηγική, αρχιτεκτονική, γλυπτική, αρχαιολογία, αυτοκινητοβιομηχανία και στο εμπόριο, όπου παρουσιάζεται το μοντέλο του προϊόντος στον μελλοντικό αγοραστή.
- *Ιατρική.* Η ραγδαία εξέλιξη της τηλεϊατρικής και της ρομποτικής καθιστά πλέον δυνατή την ανταλλαγή δεδομένων μεταξύ ιατρών που βρίσκονται σε μεγάλη απόσταση μεταξύ τους και την εκτέλεση χειρουργικών επεμβάσεων εξ αποστάσεως μέσω ενός συστήματος Εικονικής Πραγματικότητας όπου ο απομακρυσμένος ιατρός θα βλέπει τον ασθενή και θα χειρίζεται το χειρουργικό ρομπότ στο Εικονικό Περιβάλλον. Ακόμα, χρησιμοποιείται στη ψυχιατρική, με τον υπεύθυνο ιατρό να τοποθετεί τον ασθενή στο κατάλληλο Εικονικό Περιβάλλον ώστε παραδείγματος χάρη να μπορέσει να ξεπεράσει μερικές φοβίες ή να συμμετάσχει σε ένα ψυχολογικό πείραμα το οποίο ο ασθενής θα μπορεί εύκολα να διακόψει όταν αρχίσει να μην αισθάνεται άνετα. Τέλος, ένας ασθενής με κινητικά προβλήματα, μπορεί να «εισέλθει» σε έναν Εικονικό Κόσμο όπου θα πρέπει να κάνει κάποιες κινήσεις στον πραγματικό κόσμο για να πετύχει τους στόχους της εφαρμογής της Εικονικής Πραγματικότητας, αλλά ταυτόχρονα οι κινήσεις αυτές θα βοηθούν τη σωματική αποκατάστασή του.

- *Ψυχαγωγία.* Η δυνατότητα να δημιουργήσουμε οποιοδήποτε πραγματικό ή φανταστικό περιβάλλον και να αλληλεπιδρούμε με αυτό με πολλούς τρόπους και μέσα είναι μία βέβαιη πηγή ψυχαγωγίας και διασκέδασης. Συνήθως, ο χρήστης απεικονίζεται ως μία οντότητα (avatar) μέσα στον Εικονικό Κόσμο της οποίας ελέγχει τις κινήσεις και ενέργειες με τη βοήθεια του συστήματος αντίχενυσης (tracking) του Εικονικού Συστήματος. Ως παράδειγμα αναφέρουμε το Πολυμήχανο (Polymechanon), το πρώτο τεχνολογικό πάρκο διασκέδασης στην Ελλάδα. Διαθέτει εγκαταστάσεις Εικονικής και Επαυξημένης Πραγματικότητας. Οι εφαρμογές του Πολυμήχανου [8] έχουν σχεδιαστεί από το Εργαστήριο Εκπαιδευτικής Τεχνολογίας του τομέα Παιδαγωγικής της Φιλοσοφικής Σχολής του Πανεπιστημίου Αθηνών, υπό την επίβλεψη του καθ. Χρόνη Κυνηγού.

1.3) Διαδραστικότητα

Διαδραστικότητα είναι η δυνατότητα ενός μέσου για αμφίδρομη επικοινωνία. Στις εφαρμογές Εικονικής Πραγματικότητας, ο χρήστης στέλνει πληροφορίες αναφορικά με τη θέση του και τον προσανατολισμό του στο Εικονικό Σύστημα μέσω του υποσυστήματος ανίχνευσης (tracking). Το Εικονικό Σύστημα, στη συνέχεια, επικοινωνεί με το χρήστη στέλνοντας του καινούργιες εικόνες του Εικονικού Κόσμου αξιοποιώντας τα δεδομένα απ' το υποσύστημα ανίχνευσης. Στη περίπτωση μας, χρησιμοποιούμε ένα σύστημα ανίχνευσης με οπτικούς αισθητήρες [9]. Μία δίοδος εκπομπής φωτός (Light Emitting Diodes - LED) εκπέμπει φωτεινά κύματα που λαμβάνονται από έναν οπτικό αισθητήρα, συγκεκριμένα μια κάμερα CCD (Charged Coupled Device) η οποία μετατρέπει την οπτική φωτεινότητα (optical brightness) σε ηλεκτρικό σήμα ανάλογου πλάτους.

Παλαιότερα, τα συστήματα ανίχνευσης με οπτικούς αισθητήρες δε χρησιμοποιούντουσαν συχνά λόγω του μεγάλου κόστους των CCD καμερών. Όμως, το 2006 η εταιρεία Nintendo κυκλοφόρησε μια καινούργια κονσόλα οικιακής ψυχαγωγίας, τη Wii console, που ήταν η πρώτη που αξιοποιούσε οπτικούς αισθητήρες για την ανίχνευση της θέσης του χρήστη στον πραγματικό χώρο. Έτσι, έγινε διαθέσιμη στο ευρύ κοινό μία εύχρηστη και φθηνή CCD κάμερα. Το χειριστήριο της κονσόλας, το Wii Remote (πολλές φορές αναφέρεται και ως Wiimote για χάρη συντομίας), εκτός από τη CCD κάμερα, περιέχει μία συσκευή παραγωγής δονήσεων (rumble motor) που λειτουργεί στα 3.3 Volt με 35mA για να προσομοιώνει, παραδείγματος χάρι, μία σύγκρουση του avatar του χρήστη με ένα εικονικό αντικείμενο, και ένα επιταχυνσιόμετρο (accelerometer) ικανό να μετράει τις επιταχύνσεις και στους τρεις άξονες X,Y,Z μέχρι $\pm 3g$ με 10% ευαισθησία. Το Wii Remote επικοινωνεί ασύρματα με άλλες συσκευές με την τεχνολογία Bluetooth.



Εικόνα 1.7

Όψεις χειριστηρίου Wii Remote

Θα εξετάσουμε πιο λεπτομερώς την ενσωματωμένη CCD κάμερα του Wii Remote, καθώς αυτήν χρησιμοποιούμε στην εφαρμογή μας. Λειτουργεί [10,11] στο υπέρυθρο (infrared – IR) μήκος κύματος φωτός, περίπου στα 940nm, χάρη σε ένα IR-pass φίλτρο που απορρίπτει τα υπόλοιπα μήκη κύματος. Να σημειώσουμε εδώ ότι προτιμάται γενικά η υπέρυθρη ακτινοβολία στην ανίχνευση, γιατί οι ακτινοβολίες υψηλότερης συχνότητας

(ακτίνες γ , X, υπεριώδεις) μπορούν να προκαλέσουν βλάβες στο ανθρώπινο σώμα, ενώ οι υπέρυθρες είναι ακίνδυνες. Η κάμερα είναι συνδεδεμένη με ένα ολοκληρωμένο κύκλωμα ανάλυσης εικόνας, το οποίο μπορεί να αναγνωρίσει μέχρι τέσσερις διαφορετικές πηγές υπέρυθρου φωτός και να καταγράψει τις θέσεις τους. Η κάμερα είναι μονόχρωμη, με ανάλυση 128x96 εικονοστοιχεία (pixels). Όμως, το ολοκληρωμένο κύκλωμά της χρησιμοποιεί 8x ανάλυση υπο-εικονοστοιχείων (subpixel analysis) και τελικά καταγράφει τα ανιχνεύσιμα σημεία σε ανάλυση 1024x768. Η εστιακή απόσταση (focal length) της CCD κάμερας είναι περίπου 1320 εικονοστοιχεία [12]. Τέλος, το πεδίο θέασης της έχει περίπου μέγιστη οριζόντια γωνία 33°, κάθετη 23° και απόσταση 5m.

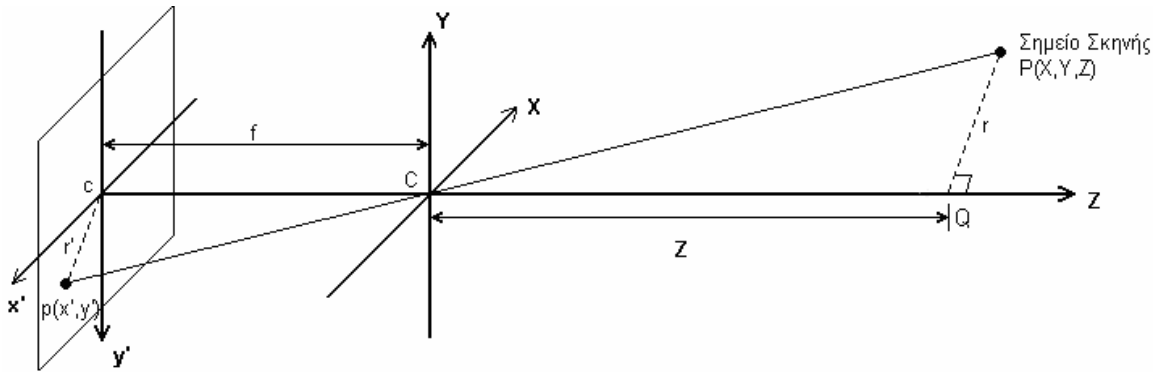
Ως πηγή υπέρυθρου φωτός, συνήθως χρησιμοποιείται μια ράβδος (sensor bar) η οποία έχει στις δύο άκρες της συστοιχίες από υπέρυθρα LED (IRED). Τις περισσότερες φορές είναι ασύρματη για να μην υπάρχει καλώδιο τροφοδοσίας που να εμποδίζει τις κινήσεις του χρήστη. Σε αυτή τη περίπτωση η τροφοδοσία γίνεται από μπαταρίες. Για την εφαρμογή μας, φτιάξαμε μία πλακέτα με υπέρυθρα LED και τροφοδοσία από μία μπαταρία. Αναλύεται περισσότερο στη παράγραφο 3.3.1.



Εικόνα 1.8
Παράδειγμα Sensor Bar

1.4) Επιπολική Γεωμετρία και Στερεοσκοπία

Πριν αναφερθούμε στην στερεοσκοπία, ας κοιτάξουμε το μοντέλο της κάμερας λεπτής οπής (pinhole camera) [13,14], όπως είναι και η CCD κάμερα του Wii Remote, που ακολουθεί τους κανόνες της Προοπτικής Προβολής (*Perspective Projection*) [15,16].



Εικόνα 1.9

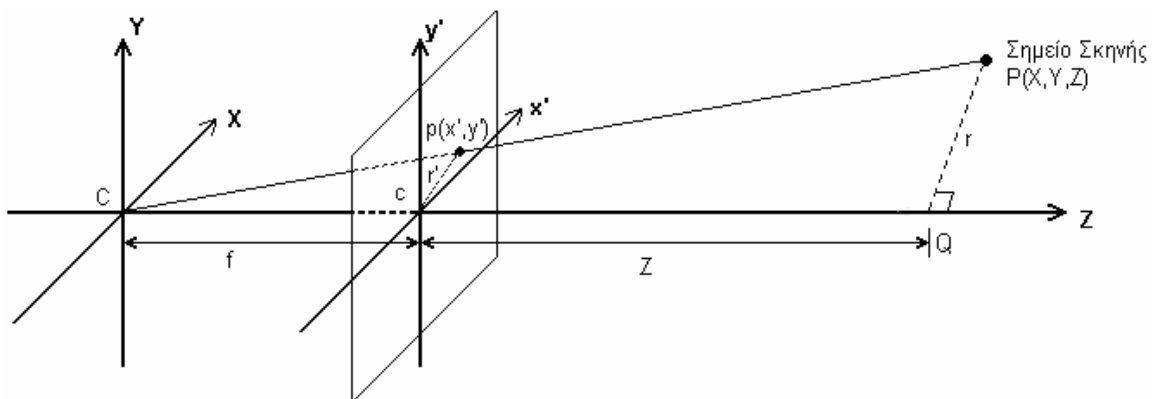
Μοντέλο κάμερας λεπτής οπής με αντεστραμμένο επίπεδο προβολής

Στην εικόνα 1.9, το σημείο C είναι η θέση της κάμερας, λέγεται κέντρο προβολής (center of projection) και ορίζει το σύστημα συντεταγμένων (X,Y,Z) . Σε αυτό το σύστημα συντεταγμένων, το σημείο της σκηνής μας P έχει συντεταγμένες (X,Y,Z) . Το σημείο P προβάλλεται μέσω του C στο σημείο $p(x',y')$. Το p ονομάζεται βασικό σημείο (principal point) και ανήκει στο επίπεδο προβολής (projection plane) με κέντρο το c. Το επίπεδο προβολής έχει το δικό του σύστημα συντεταγμένων (x',y',z') και απέχει απ' το C απόσταση f δηλαδή η εστιακή απόσταση (focal length). Ο άξονας $Z'Z$ ονομάζεται οπτικός άξονας (optical axis).

Από τα όμοια τρίγωνα που σχηματίζονται από τα σημεία P, C, προβολή του P στον $Z'Z$ (σημείο Q) και από τα σημεία p, C, προβολή του p πάνω στον $Z'Z$ (η οποία ταυτίζεται με το σημείο c), προκύπτει ότι :

$$-y' = f \frac{Y}{Z} \quad (1.1)$$

Για να αποφύγουμε το αντεστραμμένο σημείο p, μπορούμε να υποθέσουμε το μοντέλο της εικόνας 1.10, με ορθό επίπεδο προβολής το οποίο είναι ισοδύναμο με το πρώτο :



Εικόνα 1.10

Μοντέλο κάμερας λεπτής οπής με ορθό επίπεδο προβολής

Σε αυτή τη περίπτωση, πάλι από τα όμοια τρίγωνα προκύπτει :

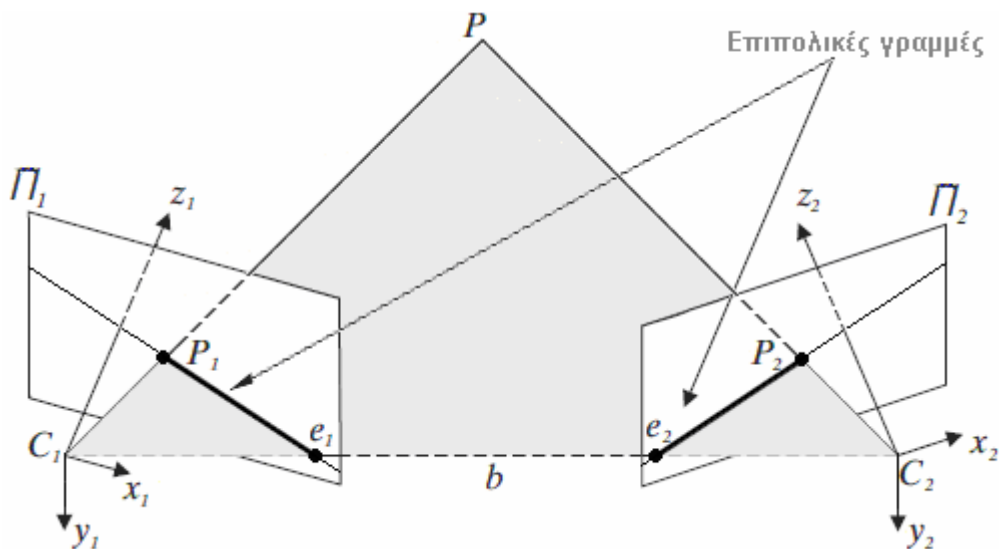
$$y' = f \frac{Y}{Z} \quad (1.2)$$

και :

$$x' = f \frac{X}{Z} \quad (1.3)$$

Στην εφαρμογή μας το Wii Remote επικοινωνεί με το πρόγραμμα Quest3D και του μεταφέρει τις συντεταγμένες x' , y' των προβολών των ανιχνευομένων σημείων πάνω στο επίπεδο προβολής. Για να υπολογίσουμε και την απόσταση Z , την απόσταση του ανιχνευόμενου σημείου από την κάμερα ή αλλιώς το βάθος ενός σημείου της σκηνής, θα καταφύγουμε στην *Στερεοσκοπία (Stereoscopy)* [7], δηλαδή την τεχνική με την οποία δημιουργείται τρισδιάστατη αναπαράσταση του χώρου με τη βοήθεια δύο εικόνων του χώρου από δύο διαφορετικές θέσεις. Χρησιμοποιώντας δύο Wii Remote, επιτυγχάνουμε να έχουμε δύο εικόνες.

Η στερεοσκοπία βασίζεται στην *Επιπολική Γεωμετρία (Epi-polar Geometry)* [14,15] :

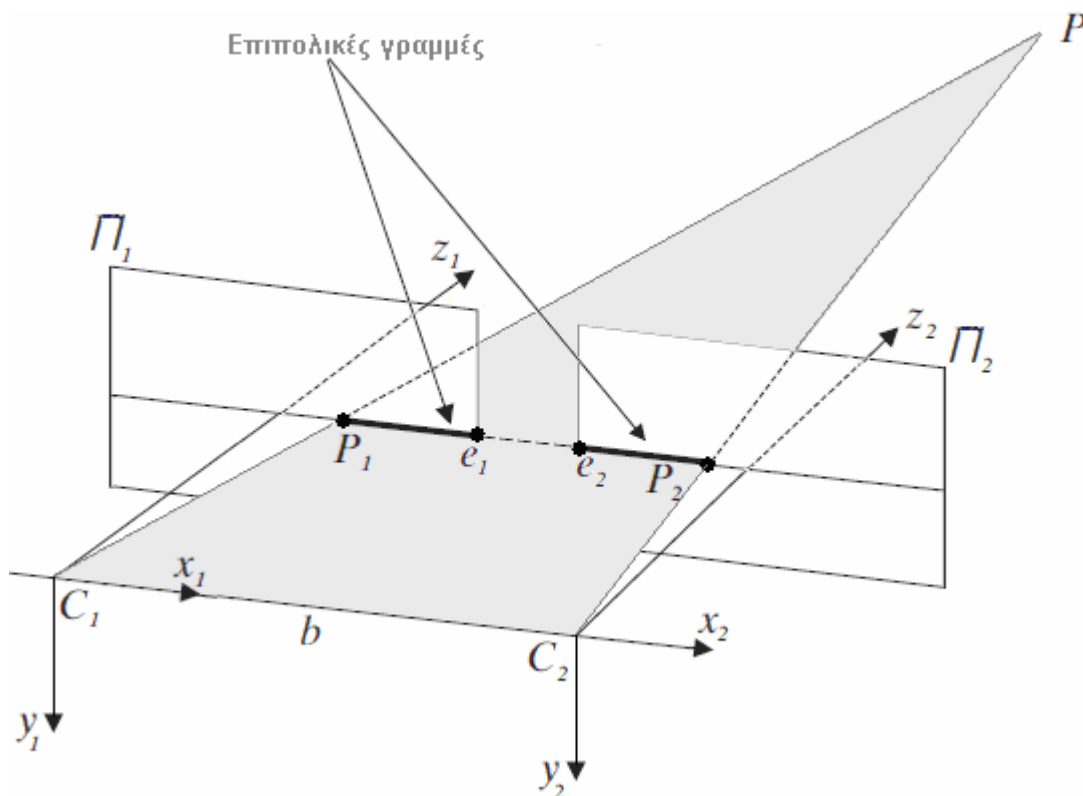


Εικόνα 1.11
Επιπολική Γεωμετρία τυχαίου προσανατολισμού

Στην εικόνα 1.11 βλέπουμε τη γενική περίπτωση τυχαίου προσανατολισμού. Η κάθε μία από τις δύο κάμερες έχει τα δικά της επίπεδα προβολής (Π_1 και Π_2), κέντρα προβολής (C_1 και C_2) και συστήματα συντεταγμένων $[(x_1, y_1, z_1)$ και $(x_2, y_2, z_2)]$. Τα δύο σημεία P_1 και P_2 είναι οι προβολές του σημείου του χώρου P στα δύο επίπεδα προβολής. Το e_1 είναι η προβολή του C_2 πάνω στο επίπεδο Π_1 και αντίστοιχα το e_2 η προβολή του C_1 στο επίπεδο Π_2 . Τα e_1 και e_2 ονομάζονται *επίπολα* ή *επιπολικά σημεία (epipoles)*. Το επίπεδο που ορίζεται από τα σημεία : P , C_1 και C_2 ονομάζεται *επιπολικό επίπεδο*. Τα ευθύγραμμα τμήματα P_1e_1 και P_2e_2 ονομάζονται *επιπολικές γραμμές*.

Για να καταλάβουμε τη χρησιμότητα της επιπολικής γεωμετρίας, ας θεωρήσουμε ότι υπάρχει μόνο η αριστερή κάμερα C_1 . Τότε, παρατηρώντας το σημείο P μέσα από τη κάμερα C_1 βλέπουμε τη προβολή του P_1 αλλά όμως δε μπορούμε να υπολογίσουμε την απόσταση του P από το C_1 . Το P μπορεί να βρίσκεται σε οποιοδήποτε σημείο της ημιευθείας C_1P_1 . Τώρα, αν θεωρήσουμε ότι υπάρχει και η δεξιά κάμερα C_2 , παρατηρούμε ότι η προβολή αυτής της ημιευθείας στο επίπεδο προβολής Π_2 της C_2 είναι η επιπολική γραμμή P_2e_2 . Δηλαδή η εικόνα όλων των πιθανών θέσεων ενός σημείου παρατηρούμενο από τη μία κάμερα είναι η επιπολική γραμμή που σχηματίζεται από τη προβολή του σημείου στο επίπεδο προβολής της άλλης κάμερας και το επιπολικό σημείο της. Το γεγονός αυτό είναι γνωστό και ως *επιπολικός περιορισμός*.

Στην *υπολογιστική όραση (computer vision)* [13] έχουμε δύο κάμερες που κοιτούν την ίδια σκηνή από διαφορετικές θέσεις. Για να υπολογίσουμε το βάθος κάθε σημείου της σκηνής, θα πρέπει πρώτα να ταυτίσουμε κάθε προβαλλόμενο σημείο στο Π_1 στο αντίστοιχό του στο Π_2 . Ο επιπολικός περιορισμός βοηθάει πολύ τον αλγόριθμο ταύτισης καθώς πλέον το αντίστοιχο σημείο δεν είναι κάποιο από τα σημεία του επιπέδου Π_2 (διδιάστατη αναζήτηση) αλλά κάποιο από τα σημεία της επιπολικής γραμμής του Π_2 (μονοδιάστατη αναζήτηση) [17].

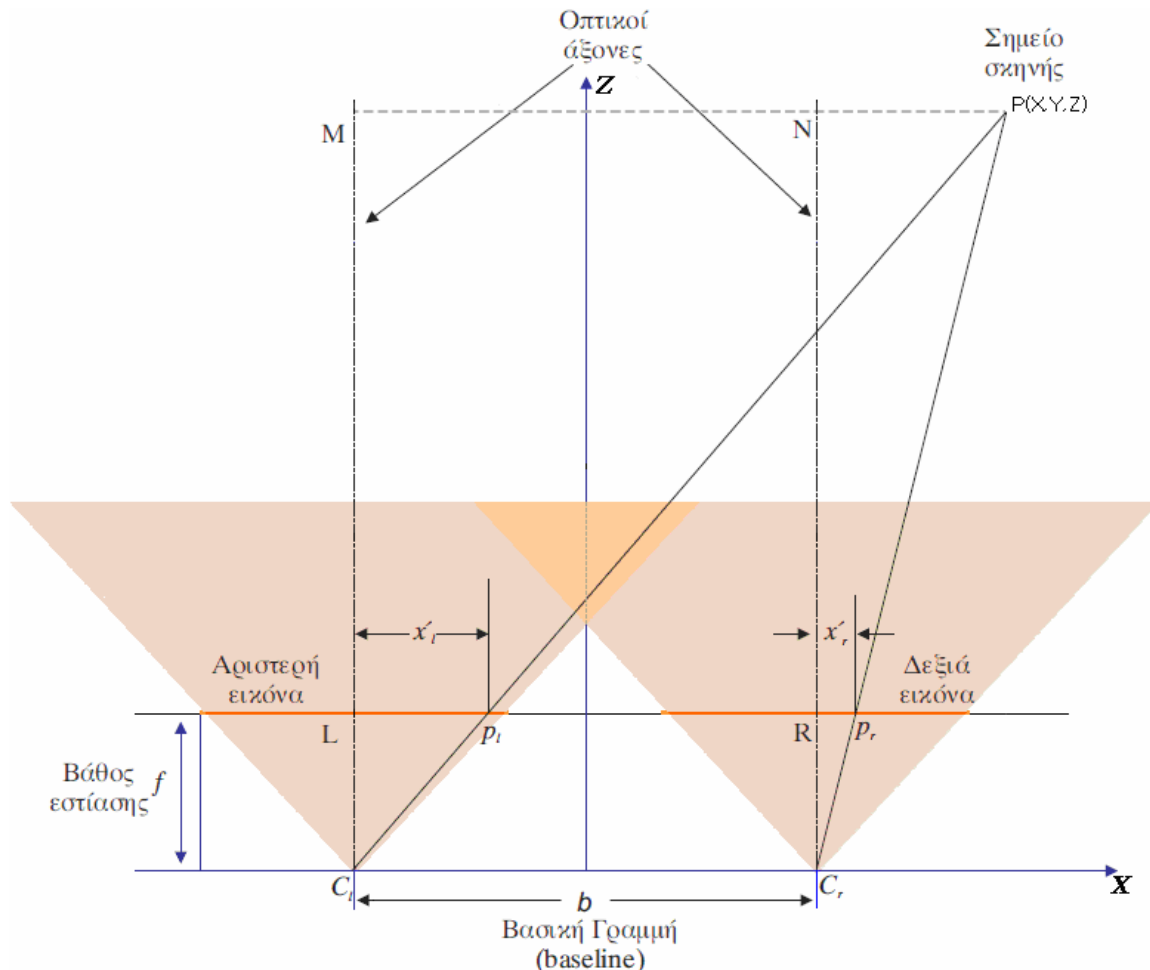


Εικόνα 1.12

Επιπολική Γεωμετρία κανονικού προσανατολισμού

Για ελαχιστοποίηση των αριθμητικών πράξεων, χρησιμοποιούμε το μοντέλο κανονικού προσανατολισμού που φαίνεται στην εικόνα 1.12. Σύστημα λήψης κανονικού προσανατολισμού ονομάζουμε το σύστημα στο οποίο οι δύο οπτικοί άξονες C_1z_1 και C_2z_2 είναι παράλληλοι και οι δύο οριζόντιοι άξονες των εικόνων C_1x_1 και C_2x_2 ταυτίζονται. Σε αυτή τη περίπτωση τα προβαλλόμενα σημεία P_1 και P_2 βρίσκονται πάνω στην ίδια επιπολική γραμμή που είναι πάντα παράλληλη προς τον οριζόντιο άξονα, άρα τα P_1 και P_2 έχουν το ίδιο y , γεγονός που μας διευκολύνει να βρούμε το βάθος του σημείου μόνο από τη διαφορά των x των σημείων P_1 και P_2 . Αυτή η διαφορά (σε

εικονοστοιχεία) του σημείου P_1 στο πρώτο επίπεδο προβολής από το αντίστοιχο σημείο P_2 στο άλλο επίπεδο προβολής, όταν τοποθετήσουμε το ένα επίπεδο πάνω στο άλλο, ονομάζεται *στερεοσκοπική ανομοιότητα (stereo disparity)* του σημείου P και συμβολίζεται με το λατινικό d . Ο υπολογισμός της ανομοιότητας και του βάθους του σημείου P φαίνεται καλύτερα στην εικόνα 1.13 αν εξετάσουμε την κάτοψη του συστήματος κανονικού προσανατολισμού [15,18,19] που προκύπτει από την εικόνα 1.12 και την εικόνα 1.10.



Εικόνα 1.13

Κάτοψη συστήματος κανονικού προσανατολισμού (παράλληλες κάμερες)

Οι δείκτες l και r είναι για τα μεγέθη της αριστερής και δεξιάς κάμερας αντίστοιχα. Η απόσταση ανάμεσα στις δύο κάμερες ονομάζεται *βασική γραμμή (baseline)* και συμβολίζεται με το λατινικό b . Η αρχή των αξόνων του συστήματος συντεταγμένων του πραγματικού κόσμου (X, Y, Z) βρίσκεται στο κέντρο της βασικής γραμμής. Όπως είπαμε και προηγουμένως οι δύο κάμερες βρίσκονται στο ίδιο επίπεδο με το P , οπότε τα y'_l και y'_r είναι ίσα. Από τα όμοια τρίγωνα C_lMP και C_lP_l , και σύμφωνα με την εξίσωση (1.3) της προοπτικής προβολής, προκύπτει :

$$\frac{Z}{X + \frac{b}{2}} = \frac{f}{x'_l} \quad (1.4)$$

Από τα όμοια τρίγωνα C_rNP και C_rRp_r , προκύπτει :

$$\frac{Z}{X - \frac{b}{2}} = \frac{f}{x'_r} \quad (1.5)$$

Λύνοντας την (1.5) ως προς X και αντικαθιστώντας στην (1.4), έχουμε :

$$\begin{aligned} Z * x'_r &= f * \left[\frac{Z * x'_l}{f} - \frac{b}{2} - \frac{b}{2} \right] \Rightarrow \\ Z * x'_r &= Z * x'_l - b * f \Rightarrow \\ Z * x'_r - Z * x'_l &= -b * f \end{aligned}$$

και τελικά προκύπτει το βάθος Z του σημείου P :

$$Z = f \frac{b}{(x'_l - x'_r)} = f \frac{b}{d} \quad (1.6)$$

Δηλαδή, βρίσκουμε το βάθος ενός σημείου P της σκηνής από την ανομοιοτήτά του $d = (x'_l - x'_r)$, την εστιακή απόσταση f (κοινή αν οι δύο κάμερες είναι ίδιου τύπου) και την βασική γραμμή b . Ας προσέξουμε ότι όλα τα μεγέθη μετρώνται με βάση το σύστημα συντεταγμένων της αριστερής κάμερας (κατά σύμβαση) εκτός από το x'_r που υπολογίζεται στο σύστημα συντεταγμένων της δεξιάς κάμερας. Επίσης, η δική μας περίπτωση είναι πιο απλοποιημένη απ'τη γενική περίπτωση της επιπολικής γεωμετρίας που περιγράψαμε προηγουμένως, καθώς δε χρειάζεται να υλοποιήσουμε τον αλγόριθμο αντιστοίχισης [15] ενός σημείου της μίας εικόνας σε κάποιο άλλο της δεύτερης εικόνας, αφού η κάθε μία απ'τις δύο κάμερες που χρησιμοποιούμε, ανιχνεύει μόνο δύο σημεία, το αριστερό και το δεξί άκρο της ράβδου με τα LED. Έτσι, με τη βοήθεια της εξίσωσης (1.6), υπολογίζουμε το βάθος των δύο άκρων της ράβδου.

Για την υλοποίηση της εφαρμογής θα χρειαστεί να υπολογίσουμε και τη τιμή της συντεταγμένης X του σημείου της σκηνής. Λύνοντας τις (1.4) και (1.5) ως προς x'_l και x'_r αντίστοιχα και διαιρώντας τες, έχουμε :

$$\begin{aligned} \frac{x'_l}{x'_r} &= \frac{(X + \frac{b}{2}) * \frac{f}{Z}}{(X - \frac{b}{2}) * \frac{f}{Z}} \Rightarrow \frac{x'_l}{x'_r} = \frac{2X + b}{2X - b} \Rightarrow \\ X &= \frac{b * (x'_l + x'_r)}{2 * (x'_l - x'_r)} \quad (1.7) \end{aligned}$$

Κεφάλαιο 2^ο) ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ

2.1) QUEST3D

2.1.1) Γενικά

Το Quest3D της εταιρείας Act-3D είναι [20] το λογισμικό πακέτο που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής. Είναι ιδιαίτερος γνωστό για την ανάπτυξη τρισδιάστατων εφαρμογών σε πραγματικό χρόνο. Το εύρος των εφαρμογών είναι ποικίλο με κύρια παραδείγματα την δημιουργία εκπαιδευτικών προγραμμάτων, οπτικοποιημένων προϊόντων και αρχιτεκτονικών, λογισμικών ψηφιακής ψυχαγωγίας, προσομοιωτικών διατάξεων (simulators), τρισδιάστατων παρουσιάσεων, ιστοσελίδων και πολυμέσων, υλοποιήσεων εικονικής πραγματικότητας.

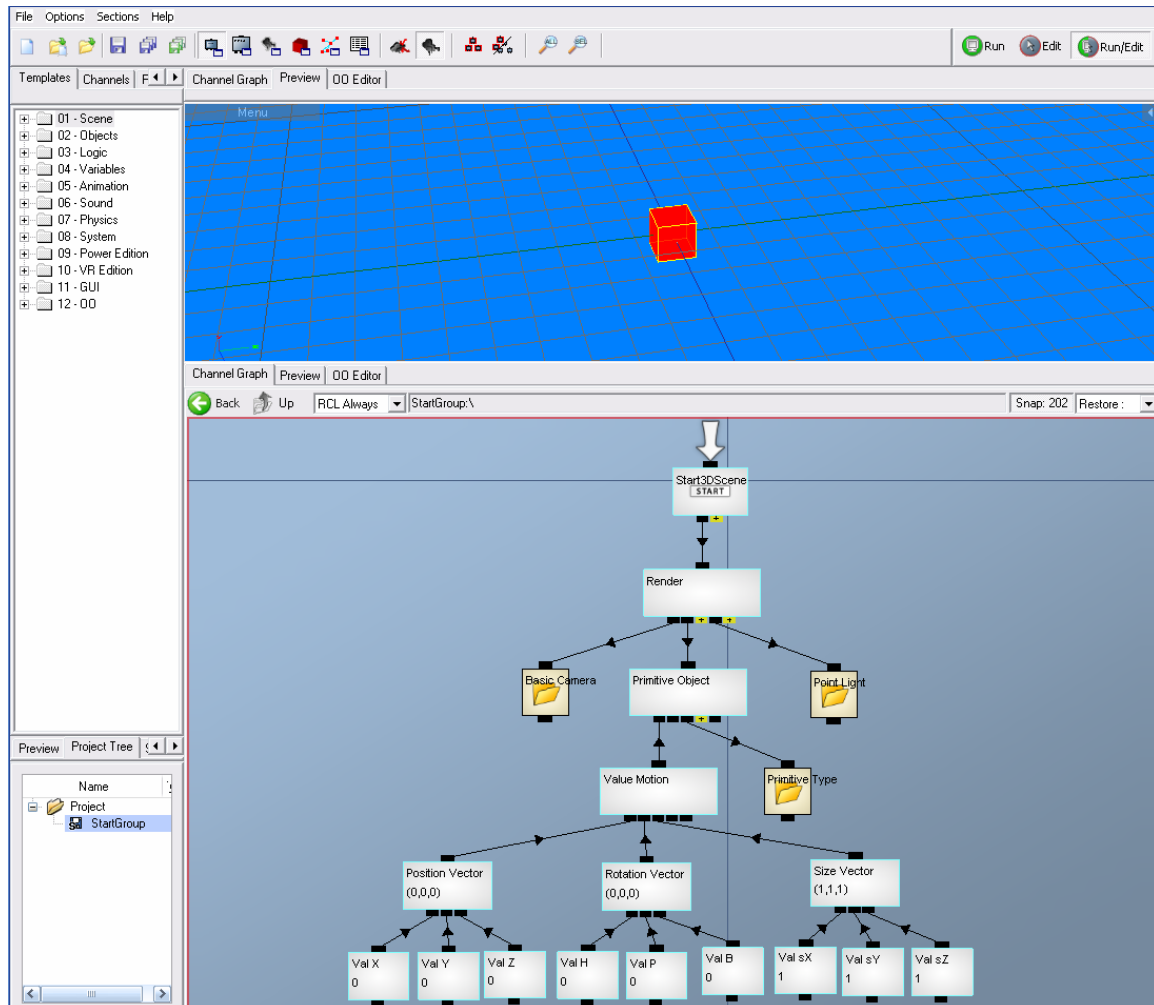
Μπορεί πολύ εύκολα να εισάγει δεδομένα από άλλα προγράμματα, όπως τρισδιάστατα μοντέλα ακίνητα ή κινούμενα (3D models), υφές (textures), πλέγματα (meshes), φωτεινές πηγές, κάμερες, ήχους και σε συνδυασμό με τις πολλές λειτουργίες που διαθέτει, όπως μηχανή φυσικής (physics and dynamic engine), εύρεση μονοπατιού (path finding), συγχρονισμός δικτύου, συνδεσιμότητα βάσεων δεδομένων, High Dynamic Range Rendering (HDR Rendering), High Level Shading Language (HLSL) και υποστήριξη τρισδιάστατων οθονών, συστημάτων CAVE, απτικών συσκευών και ανιχνευτικών διατάξεων, καταφέρνει τελικά να παράξει αληθοφανείς τρισδιάστατες σκηνές. Η σκηνή γίνεται rendered σε πραγματικό χρόνο και μόλις ολοκληρώνεται ένας κύκλος rendering, ξεκινάει αμέσως ο επόμενος. Η διάρκεια του κύκλου, άρα και ο αριθμός των κύκλων σε ένα δευτερόλεπτο (συχνότητα rendering) εξαρτάται από την πολυπλοκότητα της σκηνής. Με αυτό το τρόπο, ο χρήστης βλέπει άμεσα τα αποτελέσματα των αλλαγών που κάνει στο πρόγραμμα.

2.1.2) Τρόπος λειτουργίας

Στο Quest3D, οι εφαρμογές δομούνται από προγραμματιστικές μονάδες που λέγονται κανάλια (channels). Ακολουθώντας το μοντέλο του αντικειμενοστραφούς προγραμματισμού, το κάθε κανάλι είναι ένα αντικείμενο (object) και μία καινούργια υπόσταση (instance) δημιουργείται όταν το κανάλι χρησιμοποιείται. Όλα τα κανάλια προέρχονται από το βασικό κανάλι A3d_channel. Συνήθως, το κάθε κανάλι αποτελείται από ένα κομμάτι προ-μεταγλωττισμένου κώδικα C++ και κώδικα DirectX. Το κάθε κανάλι είναι ικανό να εκτελεί κάποια ενέργεια, να αποθηκεύει δεδομένα ή και τα δύο. Παραδείγματος χάρη, κοιτώντας την εικόνα 2.1, ξεχωρίζουμε το κανάλι render που εκτελεί το rendering της σκηνής και το κανάλι Position Vector που είναι το διάνυσμα της θέσης (X,Y,Z) του εικονιζόμενου αντικειμένου στο χώρο της σκηνής. Φαίνεται και το αρχικό κανάλι που είναι το πρώτο κανάλι που εκτελείται και ορίζει τις διαστάσεις του παραθύρου της εφαρμογής. Ακόμα, βλέπουμε τη μορφή δένδρου του προγράμματος. Τα κανάλια γονείς συνδέονται με συνδέσμους με τα κανάλια παιδιά τους (τα κανάλια παιδιά εκτελούνται αρχίζοντας απ'το πιο αριστερό και τελειώνοντας στο πιο δεξί). Οι σύνδεσμοι μπορεί να έχουν :

- Βέλος προς τα πάνω, όταν ο γονέας ζητάει και χρησιμοποιεί τα δεδομένα από το παιδί.
- Βέλος προς τα κάτω, όταν η λογική και η ροή δεδομένων κατεβαίνει προς το παιδί ή όταν το παιδί τίθεται σε καινούργια τιμή.
- Χωρίς βέλος, όταν ο γονέας και τα παιδιά δεν έχουν καθορισμένες λειτουργίες ή όταν μπορούν να τροποποιούν τη λειτουργία τους.

Κάθε κανάλι έχει και έναν τύπο, π.χ. 3D αντικείμενο ή διάνυσμα, ο οποίος ορίζει και με τι άλλου τύπου κανάλια μπορεί να συνδεθεί.

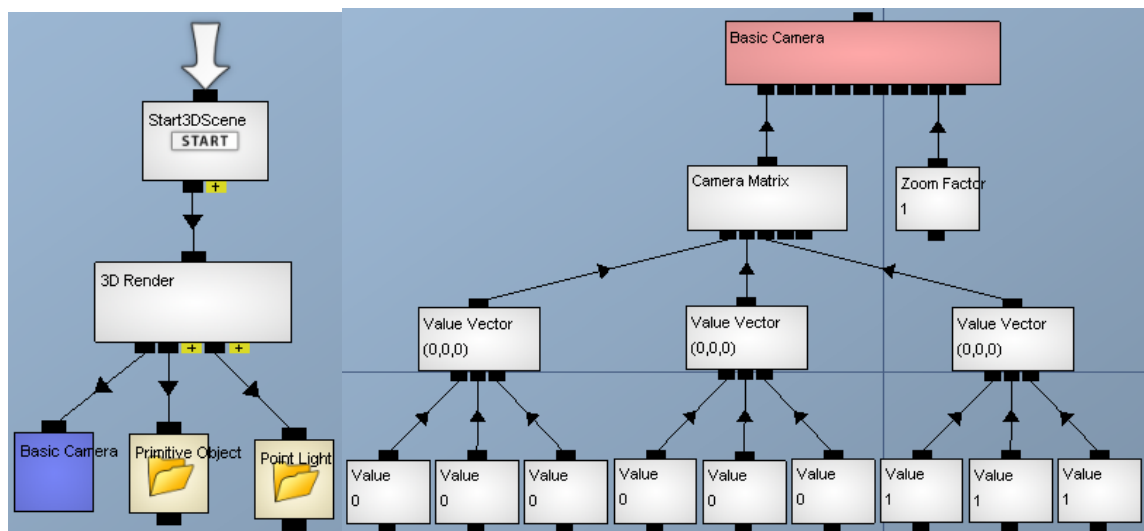


Εικόνα 2.1
Παράδειγμα της διεπαφής (interface) του Quest3D

Παρατηρούμε στην εικόνα 2.1 τη βασική διεπαφή του προγράμματος. Αποτελείται από τέσσερα μέρη:

- Το πάνω αριστερά έχει συγκεντρωμένα όλα τα κανάλια που υποστηρίζει η συγκεκριμένη έκδοση του Quest3D.
- Το πάνω δεξιά παράθυρο μας δείχνει την σκηνή μας σε πραγματικό χρόνο.
- Το κάτω δεξιά παράθυρο είναι εκεί που δημιουργούμε το πρόγραμμά μας συνδέοντας τα κανάλια.

- Το κάτω αριστερά μέρος περιέχει τα σύνολα (groups) στα οποία μπορεί να διαιρείται το πρόγραμμα. Τα σύνολα είναι ανεξάρτητες ομάδες καναλιών που συνήθως είναι υπεύθυνες για την εκτέλεση κάποια συγκεκριμένης εργασίας. Όπως βλέπουμε στην εικόνα 2.2, το κανάλι Render στο αρχικό σύνολο καλεί το κανάλι Basic Camera το οποίο ουσιαστικά είναι απλά ένας δείκτης (public caller) που καλεί το κομμάτι του προγράμματος για τη βασική κάμερα που βρίσκεται σε διαφορετικό σύνολο. Ο δείκτης είναι πάντα χρώματος μπλε και το κανάλι (ονομάζεται public channel) στο υποσύνολο που καλείται από το κυρίως σύνολο είναι κόκκινο. Το κανάλι με το κίτρινο εσωτερικό είναι απλά ένας φάκελος (directory) που περιέχει κανάλια, παραδείγματος χάρη ο φάκελος Point Light περιέχει τα κανάλια που δημιουργούν τη φωτεινή πηγή της σκηνής.



Εικόνα 2.2

Παράδειγμα public caller και public channel

Στη διεπαφή υπάρχουν και άλλες επιλογές, με τις πιο σημαντικές να είναι οι :

- Η object, όπου παραμετροποιούμε τα αντικείμενα της σκηνής.
- Η animation, όπου μπορούμε να προκαθορίσουμε την κίνηση κάποιου αντικειμένου.
- Η path finding, όπου καταγράφουμε τις επιλογές κίνησης και τους περιορισμούς που έχει ένα αντικείμενο ώστε να κινείται σύμφωνα με αυτούς.

Ακόμα, να επισημάνουμε τη δυνατότητα που έχει ο χρήστης να δημιουργήσει δικά του κανάλια που να εκτελούν διάφορες λειτουργίες σύμφωνα με τις ανάγκες του. Αυτό γίνεται με τον editor που περιέχεται στο Quest3D Software Development Kit (SDK) στη γλώσσα Visual C++. Επίσης, μπορούμε να φτιάξουμε και σενάρια εντολών (script) χρησιμοποιώντας την ελεύθερη και ανοικτή γλώσσα για scripts Lua. Τέλος, το Quest3D μπορεί να κατασκευάσει ένα εκτέλεσιμο αρχείο από το πρόγραμμα που έχουμε φτιάξει, έτσι ώστε να μπορούμε να το τρέξουμε και σε έναν υπολογιστή που δεν έχει εγκατεστημένο το Quest3D (προφανώς δε μπορούμε να κάνουμε περαιτέρω αλλαγές στο εκτελέσιμο αρχείο).

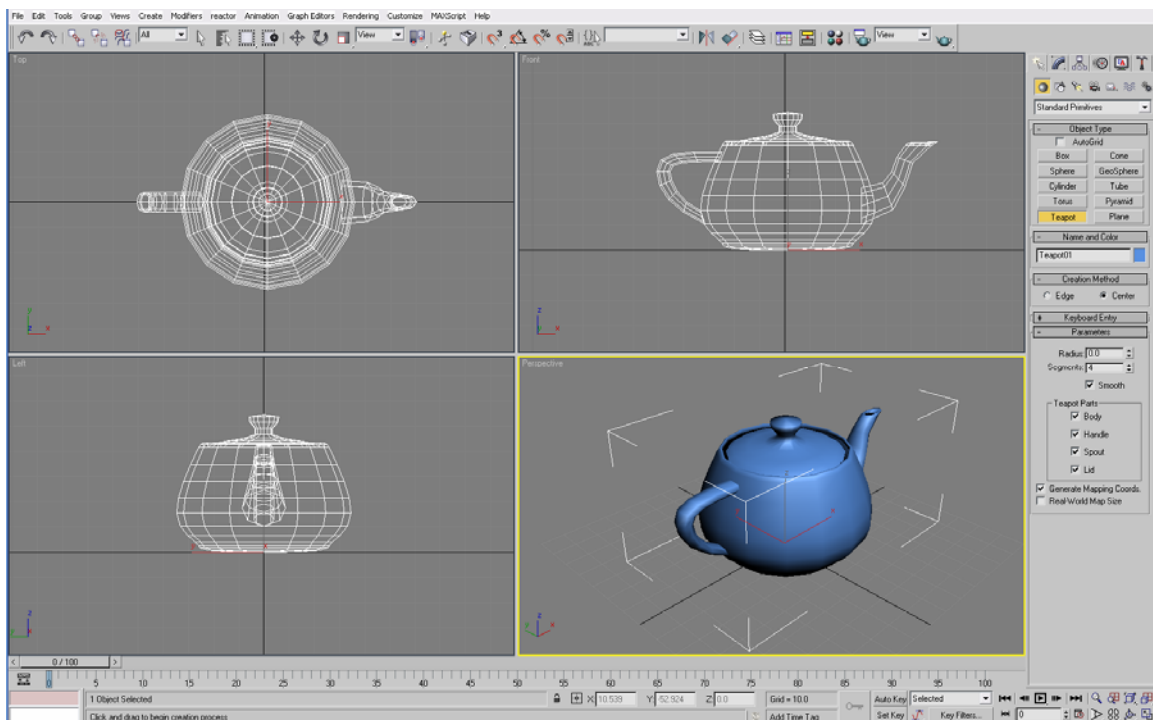
2.2) 3D Studio Max

2.2.1) Γενικά

Το Quest3D μπορεί να κατασκευάζει μόνο στοιχειώδη (primitive) αντικείμενα όπως κώνους, σφαίρες, σπείρες, τετράγωνα, κύβους, πολύγωνα. Για πιο πολύπλοκα αντικείμενα, όμως, θα χρησιμοποιήσουμε το 3D Studio Max της εταιρείας Autodesk. Το 3D Studio ειδικεύεται [21] στη δημιουργία τρισδιάστατων αντικειμένων, χαρακτήρων, περιβαλλόντων και γενικά λεπτομερών τρισδιάστατων σκηνών συμπεριλαμβανομένων και των φωτεινών πηγών, καμερών και φωτοσκιάσεων τους. Ακόμα, μπορούμε να εφαρμόσουμε υφές (textures) στα αντικείμενα και να προσδιορίζουμε τον τρόπο κίνησης τους. Το πιο σημαντικό χαρακτηριστικό του 3D Studio είναι η δημιουργία πολύπλοκων αντικειμένων και η εφαρμογή ποικίλων μετασχηματισμών σε αυτά.

2.2.2) Τρόπος λειτουργίας

Στην εικόνα 2.3 φαίνεται η διεπαφή του 3D Studio Max :



Εικόνα 2.3

Παράδειγμα της διεπαφής (interface) του 3D Studio Max

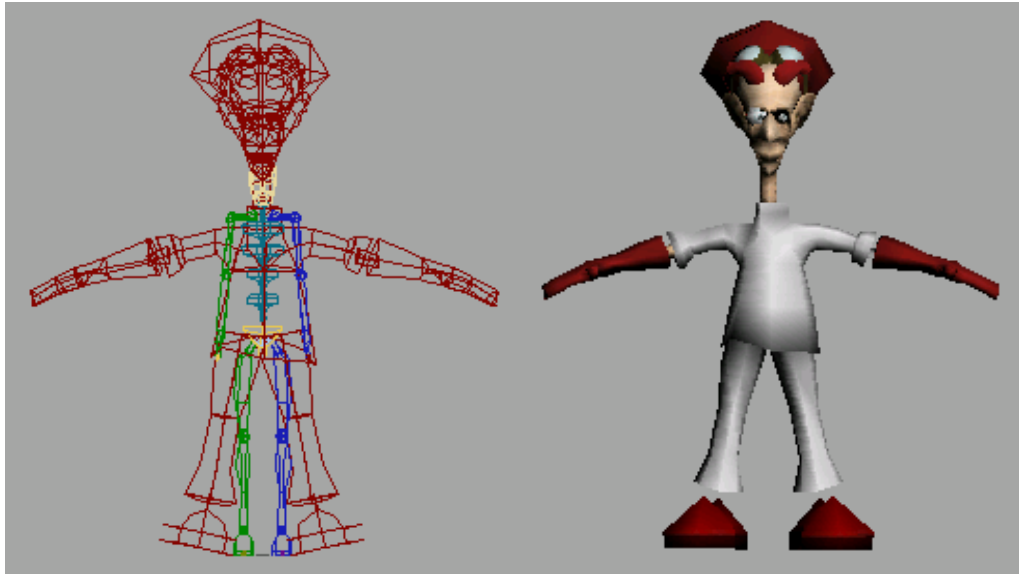
Βλέπουμε την κάτωψη, πλάγια όψη, πρόσοψη και την προοπτική όψη του υπό κατασκευή αντικειμένου, έχοντας έτσι καλύτερη τρισδιάστατη αντίληψη της σκηνής. Στο πάνω μέρος, βρίσκονται τα εργαλεία που χρησιμοποιούνται πιο συχνά, όπως επιλογή αντικειμένων ή ομάδα από αυτά, μετασχηματισμοί θέσης, περιστροφής, μεγέθους, αναστροφής και αντιστροφής, διορθωτές (editors) ονόματος, επιπέδων (layers), καμπύλων (curves) και υλικών (material). Στο κάτω μέρος υπάρχει η κλίμακα των καρτέ (frames) με την οποία αλλάζουμε τα χαρακτηριστικά του αντικειμένου σε κάθε καρτέ,

δίνοντας του έτσι την ψευδαίσθηση της κίνησης. Στο δεξί μέρος βρίσκονται οι πιο σημαντικές λειτουργίες δημιουργίας και μορφοποίησης των αντικειμένων. Παραδείγματος χάρη, μπορούμε να φτιάξουμε πολλά στοιχειώδη τρισδιάστατα αντικείμενα (σφαίρες, κύβοι, κύλινδροι, πυραμίδες, σπείρες, κώνοι, σωλήνες), εκτεταμένα αντικείμενα (άξονες τροχών, άτρακτοι, κύβοι και κύλινδροι στρογγυλεμένων άκρων, πρίσματα, κάψουλες, δεσμοί σπειρών, πολύεδρα) και να εφάρμοσουμε πράξεις συνόλων σε αυτά (ένωση, τομή, αφαίρεση A-B και B-A) και μετασχηματισμούς για να δημιουργούμε καινούργια πιο πολύπλοκα αντικείμενα. Οι μετασχηματισμοί εφάρμοζονται σε ολόκληρα τα αντικείμενα ή μόνο σε μέρη τους και είναι τριών ειδών :

- Πλέγματος (mesh)
- Καμπύλης (spline)
- Πολυγώνων (polygon)

Οι πιο σημαντικοί είναι : λύγισης (bend), κλίσης μη κάθετης επιφάνειας (bevel), διατομής (cross section), προεξοχών (extrude), κάμψης (flex), τήξης (melt), καθρεφτισμού (mirror), θορύβου (noise), κανονικοποίησης (normalize), προβολής (projection), ρυτίδων (ripple), κελύφους (shell), ασυμμετρίας (skew), μορφοποίησης του δέρματος (skin morph), τεμαχισμού (slice), σφαιρικοποίησης (spherify), συμπίεσης (squeeze), έκτασης (stretch), υποδιαίρεσης (subdivide), αντικατάστασης (substitute), σάρωσης (sweep), συμμετρίας (symmetry), σταδιακής μείωσης (taper), διακόσμησης με μωσαϊκό (tessellate), αφαίρεση ανεπιθύμητων ακρών (trim), συστροφής (twist), κυματισμού (wave). Μπορούμε να εφαρμόσουμε σχεδόν άπειρο πλήθος μετασχηματισμών σε ένα αντικείμενο και αυτοί τοποθετούνται με μορφή στοίβας, αλλά μπορούμε επίσης να αλλάζουμε τη σειρά τους στη στοίβα.

Επίσης, στο δεξί μέρος βρίσκονται και οι λειτουργίες επιλογής χρωμάτων των αντικειμένων, δημιουργίας συστημάτων σωματιδίων (particle system) για να εξομοιώσουμε π.χ. ένα σύστημα αντικειμένων φωτιάς-καπνού, δυναμικών αντικειμένων (dynamic object) που κινούνται σαν να επιδρούν πάνω τους οι φυσικές δυνάμεις του πραγματικού κόσμου (π.χ. η βαρύτητα), φωτεινών πηγών και καμερών αλλά και τοποθέτησης των σκιών τους. Αξίζει να αναφερθούμε και στη δυνατότητα κατασκευής λεπτομερέστατων διπόδων (biped), όπως φαίνεται στην εικόνα 2.4. Αρχικά φτιάχνουμε το πλέγμα, αριστερό μέρος της εικόνας 2.4, το οποίο αποτελείται από αυτόνομα μέρη που αντιστοιχούν στα διάφορα μέλη του σώματος, έτσι ώστε να είναι δυνατή η μετακίνηση π.χ. μόνο του ενός χεριού αλλά και η γενικότερη κίνηση όλου του σώματος να είναι πιο φυσική. Έπειτα εφαρμόζουμε διαφορετικές υφές στα μέλη του σώματος και εμφανίζεται το αποτέλεσμα του δεξιού μέρους της εικόνας 2.4. Ακόμα, μπορούμε να τοποθετήσουμε κάποιο υλικό, το οποίο προσδιορίζει πως η επιφάνεια αντανάκλα το φως της φωτεινής πηγής της σκηνής ή αν και πως η ίδια η επιφάνεια εκπέμπει φως.



Εικόνα 2.4

Παράδειγμα πλέγματος διπόδου και εφαρμογής υφών σε αυτό

Το 3D Studio μπορεί και εξάγει τη σκηνή που έχουμε φτιάξει σε διαφορετικού είδους αρχεία, οπότε μπορούμε ύστερα να την εισάγουμε και χρησιμοποιήσουμε σε άλλα προγράμματα όπως το Quest3D. Μία ομοιότητα με το Quest3D είναι και η δυνατότητα εξαγωγής της σκηνής σε εκτελέσιμο αρχείο το οποίο μπορούμε να τρέξουμε για να δούμε τη σκηνή σε κάποιον ηλεκτρονικό υπολογιστή που δεν έχει εγκατεστημένο το 3D Studio Max. Τέλος, το 3D Studio, όπως και άλλα προγράμματα της εταιρείας Autodesk, μπορεί και χρησιμοποιεί το σύστημα Autodesk Backburner. Πρόκειται για ένα *σύστημα κατανεμημένης ουράς* (Distributed Queueing System – DQS) στο οποίο συμμετέχουν υπολογιστές διαφόρων λειτουργικών συστημάτων (Linux, Windows, Irix) και εργάζονται συλλογικά στο ίδιο δίκτυο. Επειδή η πολυπλοκότητα μιας σκηνής στο 3D Studio μπορεί είναι πολύ μεγάλη λόγω των πολλών αντικειμένων και λεπτομεριών που μπορεί να περιέχει, το rendering της σκηνής από ένα μόνο υπολογιστή μπορεί να χρειαστεί πολύ ώρα. Έτσι, χρησιμοποιώντας το DQS, πολλοί υπολογιστές στο ίδιο δίκτυο συνδράμουν στην επίτευξη μιας κοινής εργασίας, π.χ. το rendering, σε λιγότερο χρόνο από όσο θα έκανε ένας μόνο υπολογιστής. Το DQS απαρτίζεται από :

- Έναν υπολογιστή (ονομάζεται Render Client) του οποίου κάποια εφαρμογή στέλνει τη ζητούμενη εργασία (rendering) στους render nodes.
- Render Nodes είναι ένας ή περισσότεροι υπολογιστές που αναλαμβάνουν την πραγματοποίηση της εργασίας.
- Ένας υπολογιστής (Backburner Manager) που είναι υπεύθυνος για την διανομή και διαχείριση των εργασιών που εκτελούν οι Render Nodes.
- Ένας ή περισσότεροι υπολογιστές (Backburner Monitor) οι οποίοι παρακολουθούν τις εκτελούμενες εργασίες στους Render Nodes.

Τα τέσσερα μέρη επικοινωνούν μεταξύ τους με το πρωτόκολλο TCP/IP.

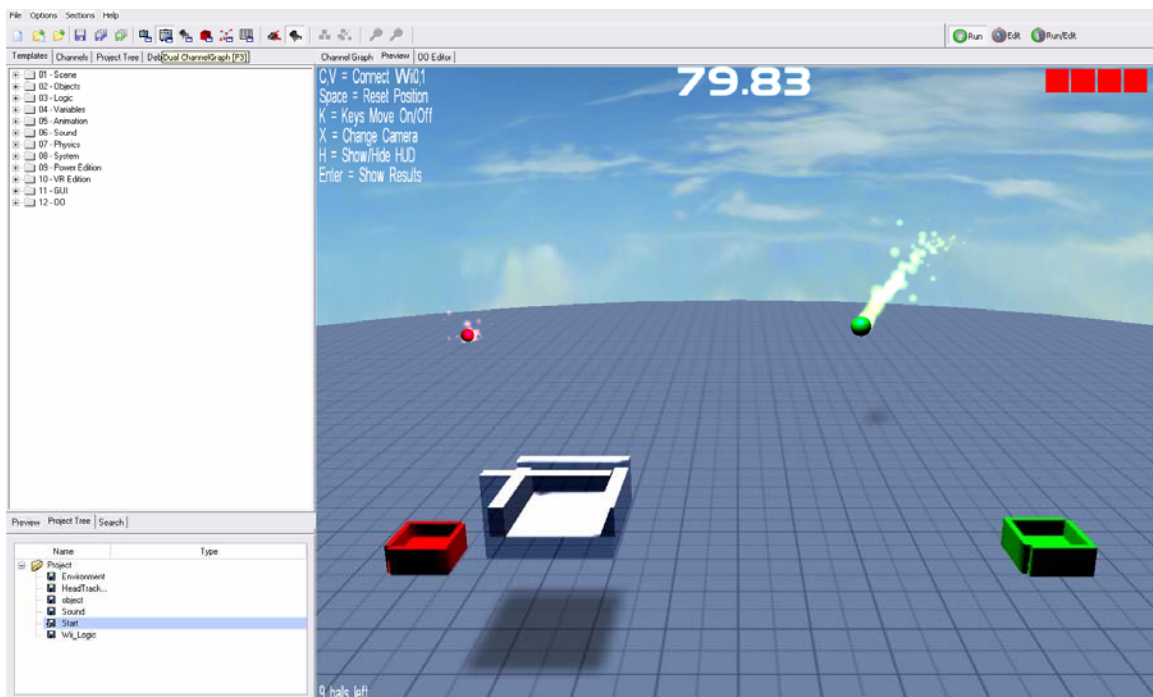
Κεφάλαιο 3^ο) ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

Σε αυτό το κεφάλαιο θα αναλύσουμε το πρόγραμμα στο Quest3D της εφαρμογής μας. Στο πρώτο υποκεφάλαιο θα πούμε για το γραφικό περιβάλλον, τα βοηθητικά αντικείμενα της κύριας σκηνής και τον ήχο. Στο δεύτερο θα εξηγήσουμε για τη διεπαφή του χρήστη, τη στερεοσκοπική προβολή και τη λογική της εφαρμογής, ενώ στο τρίτο για την κατασκευή της ηλεκτρονική πλακέτας που λειτουργεί ως sensor bar και πως επιτυγχάνεται η προσομοίωση της κίνησης και περιστροφής της.

Στην εικόνα 3.1 βλέπουμε ένα στιγμιότυπο οθόνης (screenshot) της εφαρμογής. Ο χρήστης κρατώντας την ράβδο με τα LED με τα χέρια του και κουνώντας την, μεταβάλλει τη θέση και τη κλίση του αντικειμένου-υποδοχέα στο παιχνίδι. Σκοπός του παιχνιδιού είναι να καταφέρει ο χρήστης να εγκλωβίσει τη σφαίρα που πετάγεται προς αυτόν με τον υποδοχέα και έπειτα να την αποθέσει στον στόχο του ίδιου χρώματος με την μπάλα. Αρχικά ο χρήστης ερωτάται πόσες σφαίρες θα αποπειραθεί να πιάσει και το επίπεδο δυσκολίας του παιχνιδιού. Το επίπεδο δυσκολίας ορίζει το χρόνο που έχει στη διάθεσή του για να πιάσει και εναποθέσει την κάθε σφαίρα. Επίσης, ο χρήστης μπορεί να επιλέξει η εφαρμογή να προβληθεί στερεοσκοπικά. Οι σφαίρες εκτοξεύονται με τυχαία σειρά και το μέτρο και η κατεύθυνση της δύναμης, που εφαρμόζεται στιγμιαία στη σφαίρα, είναι επίσης τυχαίες τιμές μέσα σε ένα τέτοιο εύρος ώστε η σφαίρα να μην βγει εκτός πεδίου όρασης του παίκτη. Από τον πολλαπλασιασμό :

(αριθμός σφαιρών) * (χρόνος για κάθε σφαίρα)

προκύπτει ο ολικός χρόνος που θα παίξει ο χρήστης και φαίνεται στο πάνω άκρο της εικόνας, ενώ ο αριθμός των υπολειπομένων σφαιρών διακρίνεται κάτω αριστερά. Όταν ο χρόνος τελειώσει, παρουσιάζονται τα αποτελέσματα, δηλαδή πόσες σφαίρες απ' τις συνολικές εναπόθεσε επιτυχώς ο χρήστης, και ένα μενού με επιλογές αν θέλει να ξαναρχίσει ή να τερματίσει την εφαρμογή.



Εικόνα 3.1

Στιγμιότυπο οθόνης της εφαρμογής

3.1) Υλοποίηση Περιβάλλοντος

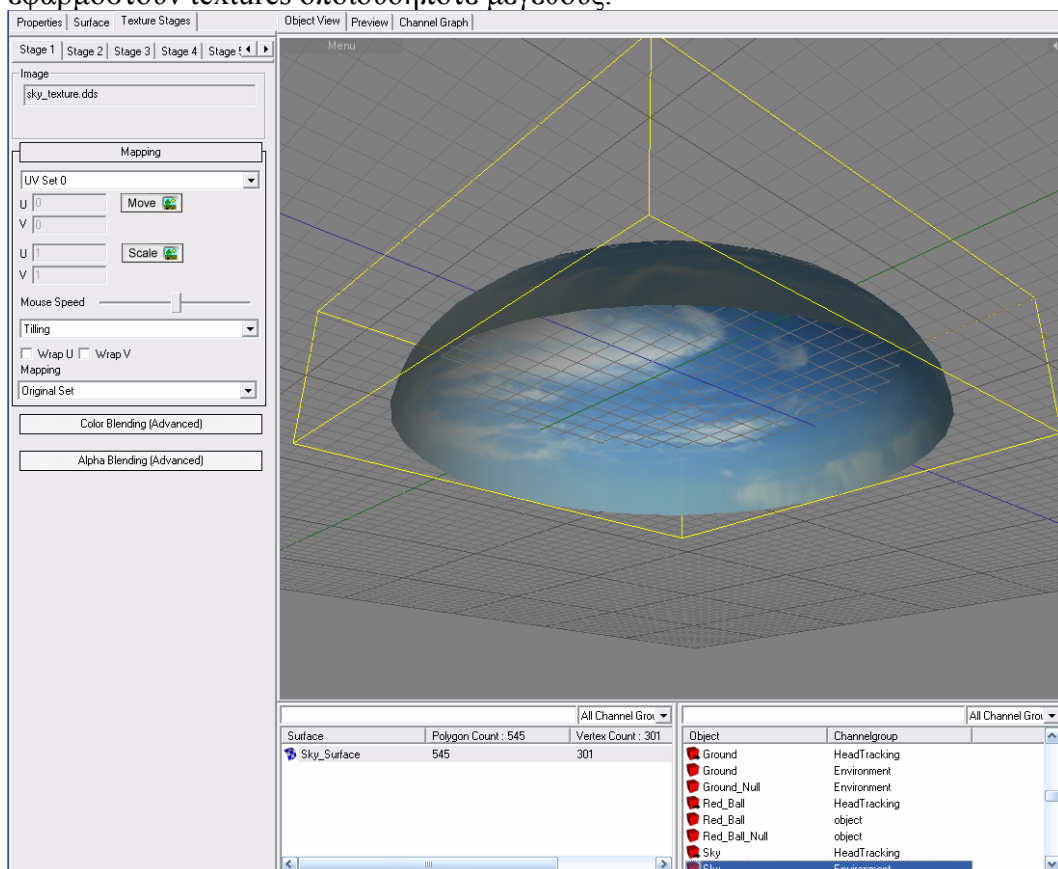
3.1.1) Γραφικό Περιβάλλον

Το γραφικό περιβάλλον αποτελείται από τα εξής τρισδιάστατα (3Δ) αντικείμενα :

- Ουρανός. Πρόκειται για ένα κομμάτι φλοιού σφαίρας στο οποίο έχει εφαρμοστεί ένα texture (το sky_texture.dds) για να μοιάζει με ουρανό. Το αρχείο dds περιέχει mipmap πληροφορίες, δηλαδή το αρχικό texture αποθηκεύεται μαζί με διαφορετικού μεγέθους αντίγραφα του έτσι ώστε όταν το texture εφαρμόζεται σε ένα 3Δ αντικείμενο, οι περιοχές κοντά στην κάμερα χρησιμοποιούν τα μεγαλύτερου μεγέθους αντίγραφα και οι πιο απομακρυσμένες χρησιμοποιούν τα πιο μικρού μεγέθους. Το αποτέλεσμα είναι ότι ο χρόνος δημιουργίας του 3Δ αντικειμένου μειώνεται αλλά αυξάνεται το μέγεθος του αρχείου texture. Επίσης, το Quest3D για την εφαρμογή του texture χρησιμοποιεί την τεχνική UV mapping (επίπεδη και κυβική). Είναι ένας τρόπος για να «τυλιχθεί» μία διδιάστατη εικόνα γύρω από ένα 3Δ αντικείμενο. Δηλαδή ένα σημείο του 3Δ αντικειμένου με συντεταγμένες (x,y,z) αντιστοιχεί σε ένα σημείο της διδιάστατης εικόνας με συντεταγμένες (U,V). Οι σχέσεις που συνδέουν τις δύο συντεταγμένες είναι :

$$U = \frac{x}{\sqrt{x^2 + y^2 + z^2}} \quad \text{και} \quad V = \frac{y}{\sqrt{x^2 + y^2 + z^2}}$$

Οι συντεταγμένες (U,V) είναι σχετικές, οπότε σε ένα αντικείμενο μπορούν να εφαρμοστούν textures οποιουδήποτε μεγέθους.

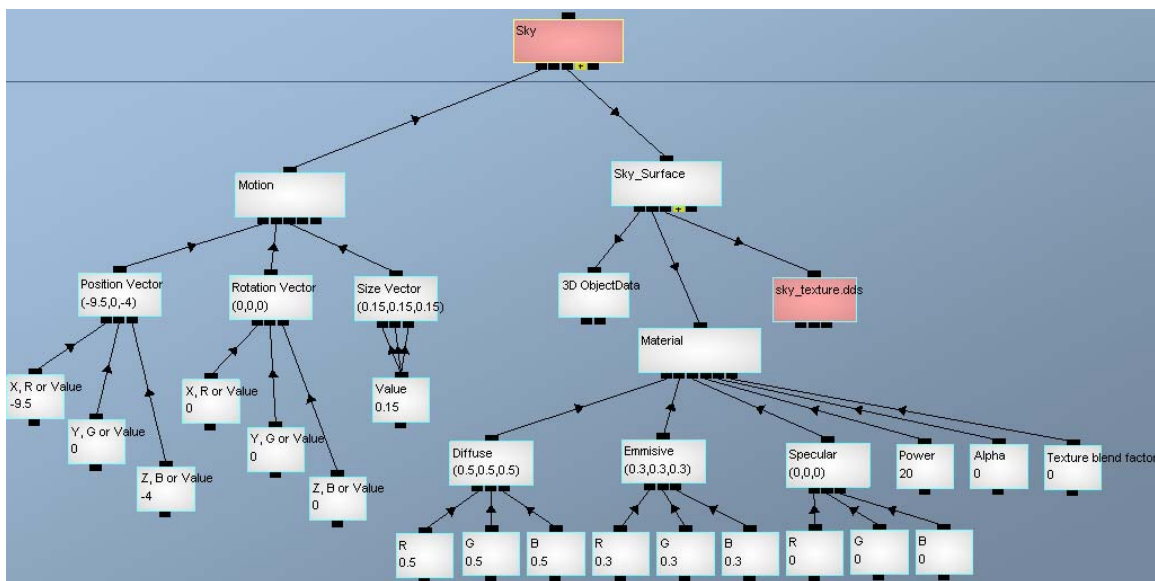


Εικόνα 3.2
Το αντικείμενο ουρανός

Ο κώδικας για την κατασκευή του ουρανού φαίνεται στην εικόνα 3.3. Το κανάλι Motion (Κίνηση) είναι μία μήτρα που περιέχει τα διανύσματα θέσης, περιστροφής και μεγέθους. Το κανάλι Sky_Surface είναι υπεύθυνο για την επιφάνεια του αντικειμένου. Το κανάλι 3D Object Data περιέχει τη γεωμετρία του σχήματος του αντικειμένου. Το sky_texture.dds είναι το texture που εφαρμόζεται. Το Material προσδιορίζει τι είδους υλικό είναι η επιφάνεια αναφορικά με τις χρωματικές της ιδιότητες.

- Diffuse (διαχεόμενο) : είναι το χρώμα που υπολογίζεται με βάση την αντίστοιχη τιμή της φωτεινής πηγής της σκηνής.
- Emmisive (εκπεμπόμενο) : είναι το χρώμα του αντικειμένου αν θεωρηθεί φωτοβόλο.
- Specular (ανακλώμενο) : είναι το χρώμα της λάμψης που εξαρτάται από την θέση και χρώμα της φωτεινής πηγής της σκηνής.
- Power (ισχύς) : η ισχύς της λάμψης.
- Alpha Diffuse : αν δεν υπάρχει εφαρμοζόμενο texture, τότε χρησιμοποιείται αυτή η τιμή για να προσομοιώσουμε βαθμιαία εξαφάνιση του αντικειμένου.
- Texture blend factor : η τιμή που χρησιμοποιείται για την εξαφάνιση όταν υπάρχει texture ή για τη μετάβαση από ένα texture σε ένα άλλο.

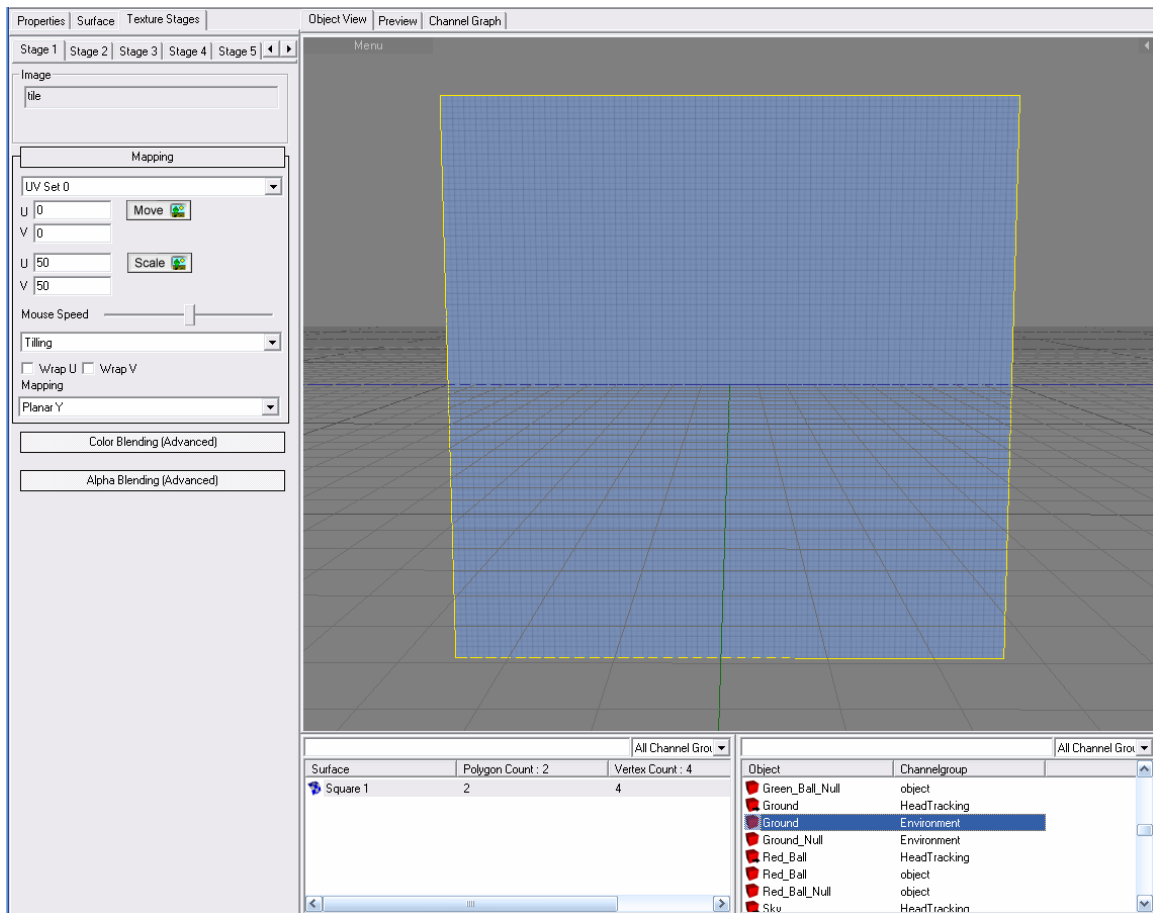
Τα χρώματα είναι όλα σε τιμές Κόκκινο, Πράσινο, Μπλε (RGB).



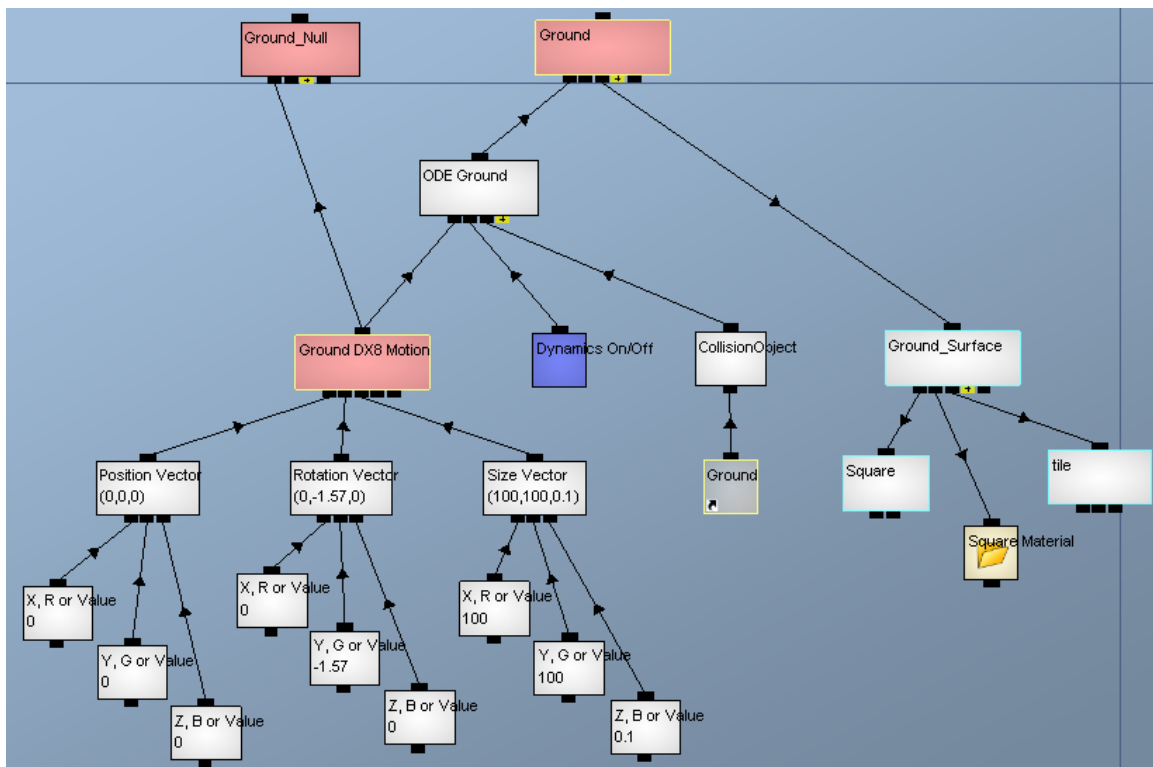
Εικόνα 3.3

Κώδικας κατασκευής ουρανού

- Έδαφος. Είναι ένα τετράγωνο στο οποίο έχει εφαρμοστεί το texture tile. Στην εικόνα 3.4 φαίνεται το αντικείμενο και στην εικόνα 3.5 ο κώδικας για την κατασκευή του. Όπως σε όλα τα αντικείμενα υπάρχουν τα κανάλια Motion και Surface που εξηγήθηκαν παραπάνω. Βλέπουμε το κανάλι ODE Ground το οποίο είναι υπεύθυνο για την λειτουργία του εδάφους ως φυσικό σώμα (Open Dynamics Engine – ODE σώμα).



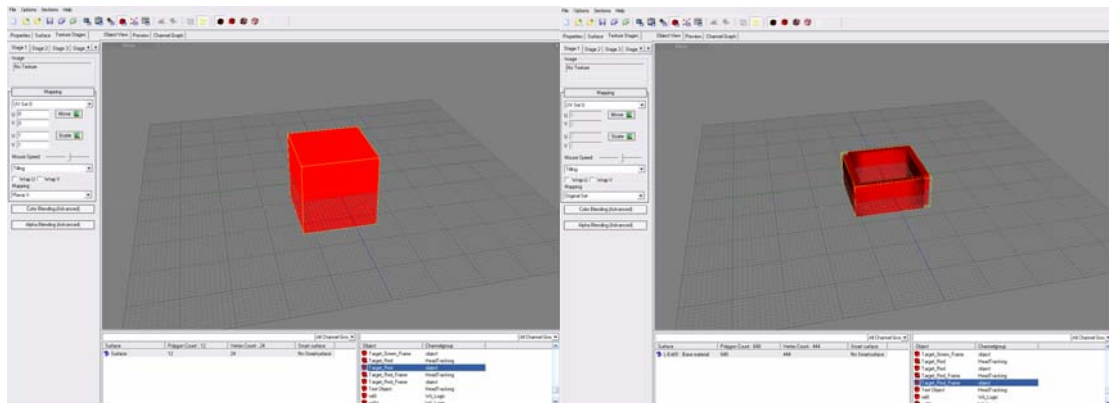
Εικόνα 3.4
Το αντικείμενο έδαφος



Εικόνα 3.5
Κώδικας κατασκευής εδάφους

Παρατηρούμε όμως ότι το κανάλι ODE Ground παρεμβαίνει ανάμεσα στο κανάλι Ground και το κανάλι Motion του. Αυτό έχει ως αποτέλεσμα να μην μπορούμε αρχικά, δηλαδή πριν ενεργοποιηθεί το σύστημα φυσικής, να μεταβάλλουμε τη θέση, τη περιστροφή και το μέγεθός του. Γι' αυτό καταφεύγουμε σε ένα τέχνασμα φτιάχνοντας ένα κενό αόρατο (καθώς δε συνδέεται με κανάλι επιφάνειας) αντίγραφο του αντικειμένου, το Ground_Null, που όμως αυτό δεν είναι ODE σώμα καθώς δε συνδέεται με ODE κανάλι. Έτσι, τώρα μπορούμε να μεταβάλλουμε τις ιδιότητες του αντιγράφου και όταν ενεργοποιηθεί το σύστημα φυσικής μεταβάλλονται οι ιδιότητες του ODE καναλιού. Το κανάλι Dynamics On/Off είναι απλά μια μεταβλητή που ενεργοποιεί και απενεργοποιεί το σύστημα φυσικής. Το ODE Ground έχει οριστεί ως ένα στατικό σώμα, δηλαδή δε μεταβάλλεται από φυσικές δυνάμεις, ώστε όταν παραδείγματος χάρη μία σφαίρα συγκρουστεί με αυτό, το έδαφος παραμένει ακίνητο. Επίσης, δημιουργεί ένα αόρατο αντικείμενο συγκρούσεων (collision object) που αντιστοιχεί στο φυσικό σώμα. Σε αυτή τη περίπτωση το collision object είναι μορφής πολυγώνου. Το σχήμα του collision object ορίζεται απ' το κανάλι παιδί CollisionObject το οποίο συνδέεται με την συντόμευση (γκρι κανάλι) του καναλιού Ground και δημιουργεί το δένδρο συγκρούσεων (collision tree) από τη γεωμετρία του.

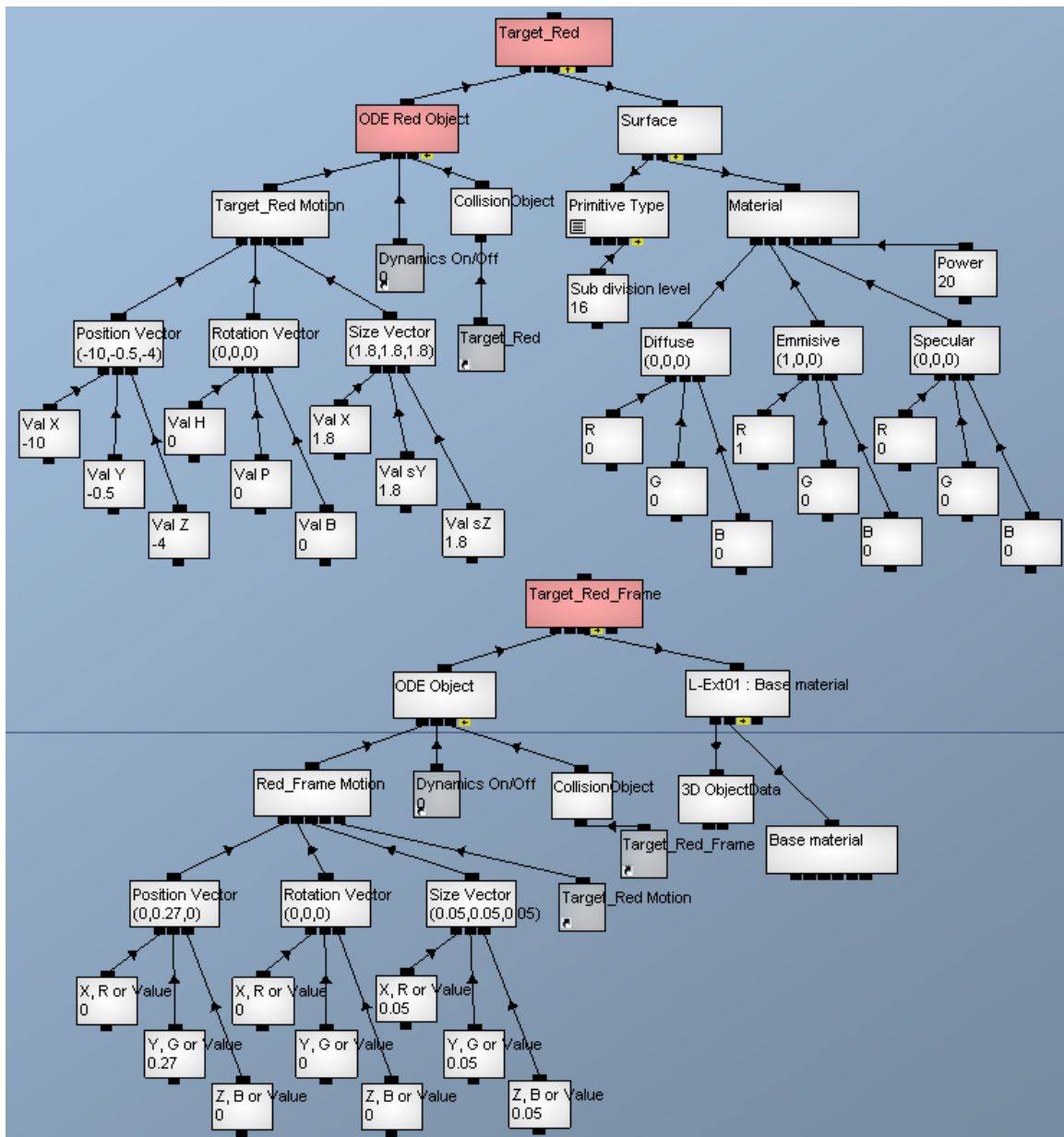
- Οι στόχοι και τα πλαίσια τους. Όπως βλέπουμε στην εικόνα 3.1, οι δύο στόχοι με τα δύο πλαίσια είναι τελείως όμοια αντικείμενα εκτός από το χρώμα τους. Άρα αρκεί να εξετάσουμε τον στόχο και το πλαίσιο ενός χρώματος. Ο στόχος είναι ένας απλός κύβος, ενώ το πλαίσιο που τον περιβάλλει είναι ένα λίγο πιο πολύπλοκο αντικείμενο που δημιουργήθηκε στο πρόγραμμα 3D Studio Max (κανάλι L-Ext01 : Base Material της εικόνας 3.7 και τα παιδιά του). Στην εικόνα 3.6 φαίνονται τα δύο αντικείμενα.



Εικόνα 3.6

Κόκκινος στόχος και το πλαίσιό του

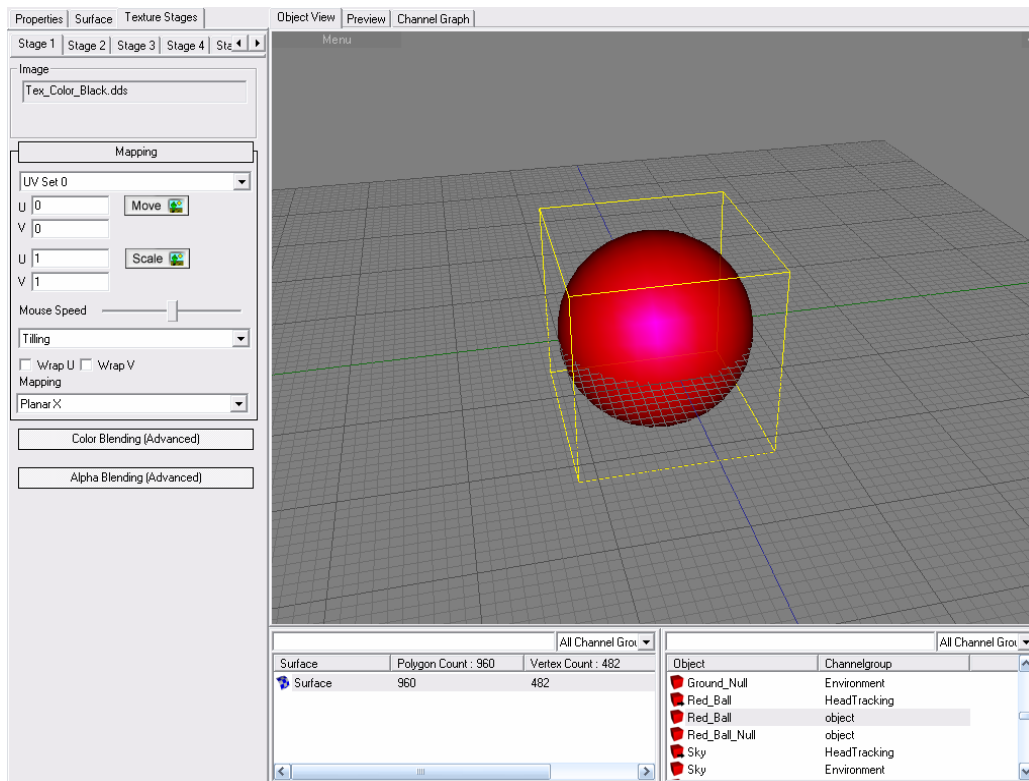
Ο στόχος είναι χαμηλότερα από το έδαφος και προεξέχει λίγο μόνο. Το πλαίσιο ακουμπάει πάνω στο έδαφος και περικλείει το στόχο. Ο παίκτης προσπαθεί να ρίξει την σφαίρα ίδιου χρώματος πάνω στο στόχο μέσα στο πλαίσιο. Στην εικόνα 3.7 φαίνεται ο κώδικας κατασκευής των δύο αντικειμένων. Ο στόχος βλέπουμε ότι είναι ένας στοιχειώδης κύβος (κανάλι Primitive Type) και το κανάλι Sub division level ορίζει την πολυπλοκότητα του σχήματος. Όπως και το έδαφος, είναι και αυτά στατικά σώματα στα οποία δε μπορούν να ασκηθούν δυνάμεις. Αυτό που μας ενδιαφέρει είναι να ανιχνεύσουμε συγκρούσεις μεταξύ των στόχων και σφαιρών ίδιου χρώματος για να αυξάνουμε το σκορ του παίκτη.



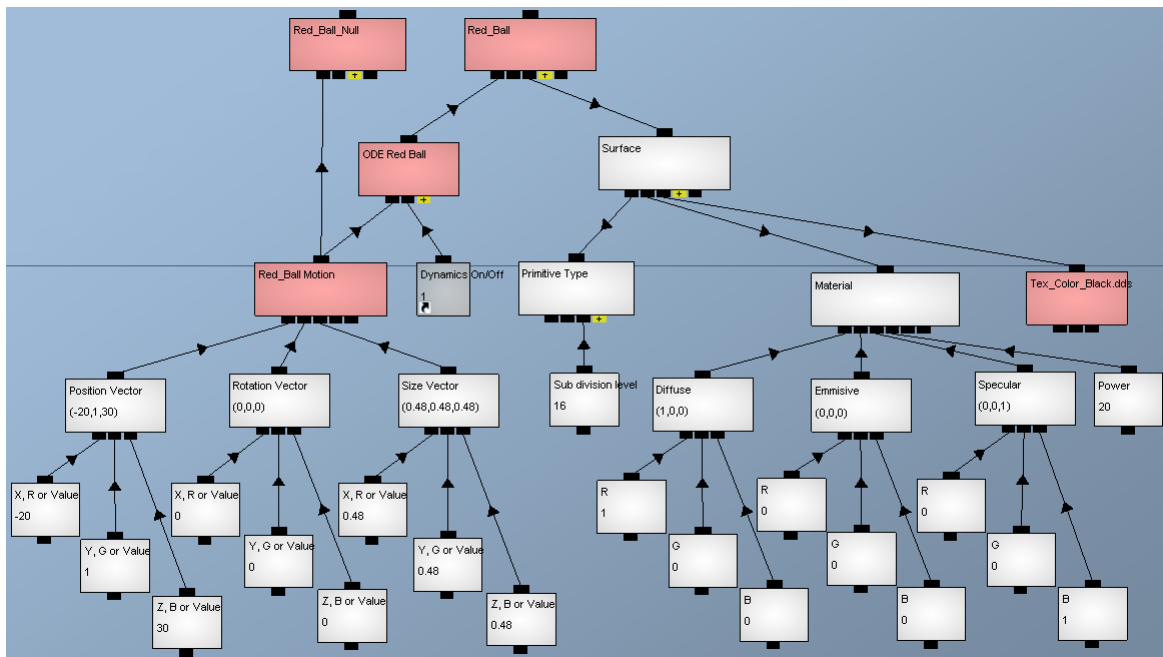
Εικόνα 3.7
Κώδικας κατασκευής στόχου και πλαισίου

Στην εικόνα 3.7 παρατηρούμε την συντόμευση του καναλιού Target_Red Motion που συνδέεται με το κανάλι Red_Frame Motion. Η σύνδεση γίνεται στη θέση Parent Matrix του καναλιού Red_Frame Motion. Αυτό σημαίνει ότι η μήτρα κίνησης του αντικειμένου κόκκινου πλαισίου είναι πλέον σχετικός με την μήτρα κίνησης του αντικειμένου κόκκινου στόχου. Οπότε η θέση, περιστροφή και το μέγεθος του κόκκινου πλαισίου εξαρτώνται από τα αντίστοιχα διανύσματα του κόκκινου στόχου. Αυτός είναι ένας εύκολος τρόπος να φτιάχνουμε σύνολα από αντικείμενα με εξαρτημένες μήτρες κίνησης μεταξύ τους. Έτσι, μεταβάλλοντας την μήτρα κίνησης του γονικού (parent) αντικειμένου, μεταβάλλονται και τα υπόλοιπα αντικείμενα του συνόλου.

- Οι δύο σφαίρες. Πάλι τα δύο αντικείμενα είναι ακριβώς ίδια εκτός απ' το χρώμα, άρα αρκεί να εξετάσουμε το ένα από αυτά.

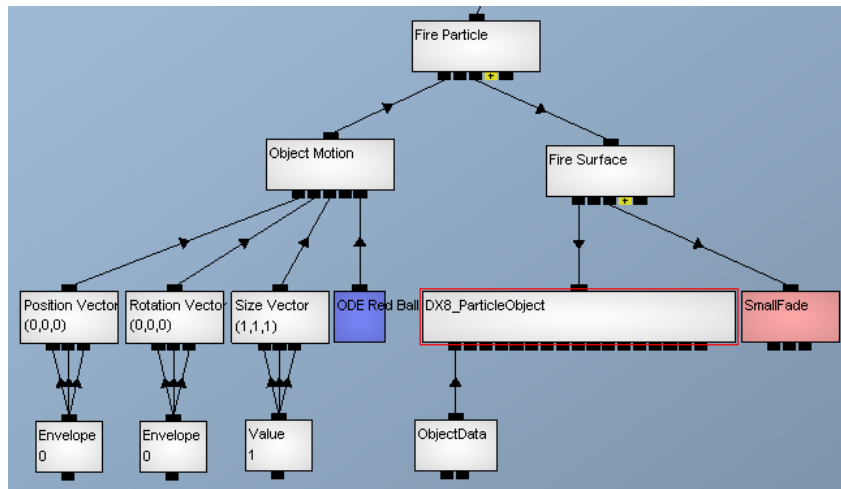


Εικόνα 3.8
Το αντικείμενο κόκκινη σφαίρα



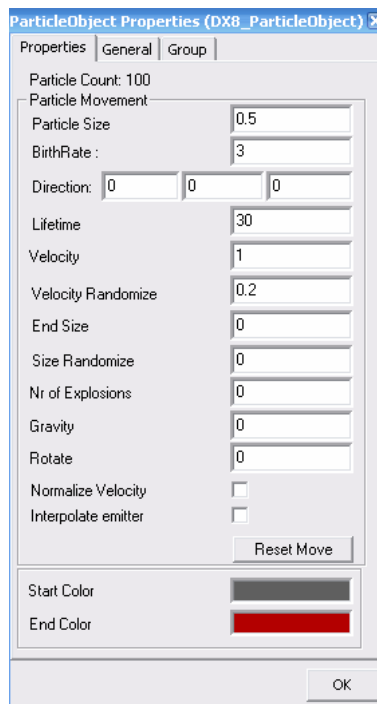
Εικόνα 3.9
Κώδικας κατασκευής κόκκινης σφαίρας

Στην εικόνα 3.8 βλέπουμε την κόκκινη σφαίρα και στην εικόνα 3.9 βλέπουμε τον κώδικα κατασκευής της. Παρατηρούμε πάλι την ύπαρξη ενός κενού αντικείμενου Red_Ball_Null που μας βοηθάει να αλλάζουμε την αρχική μήτρα κίνησης του Red_Ball και το κανάλι-texture Tex_Color_Black.dds που εφαρμόζεται στην επιφάνεια της σφαίρας. Διακρίνουμε στην εικόνα 3.1 και το σύστημα σωματιδίων (particle system) που συνοδεύει τη σφαίρα. Πρόκειται για ένα 3Δ αντικείμενο που έχει τα ίδια χαρακτηριστικά με ένα οποιοδήποτε άλλο αντικείμενο. Η μοναδική του διαφορά είναι ότι αποτελείται από μικρά σωματίδια τα οποία έχουν διάρκεια ζωής (lifetime) οπότε εξαφανίζονται και έπειτα ξαναδημιουργούνται με ένα συγκεκριμένο ρυθμό (birthrate). Ο κώδικας κατασκευής του συστήματος σωματιδίων φαίνεται στην εικόνα 3.10 και στην εικόνα 3.11 οι ιδιότητες του συστήματος που ορίζονται στο κανάλι DX8_ParticleObject.



Εικόνα 3.10

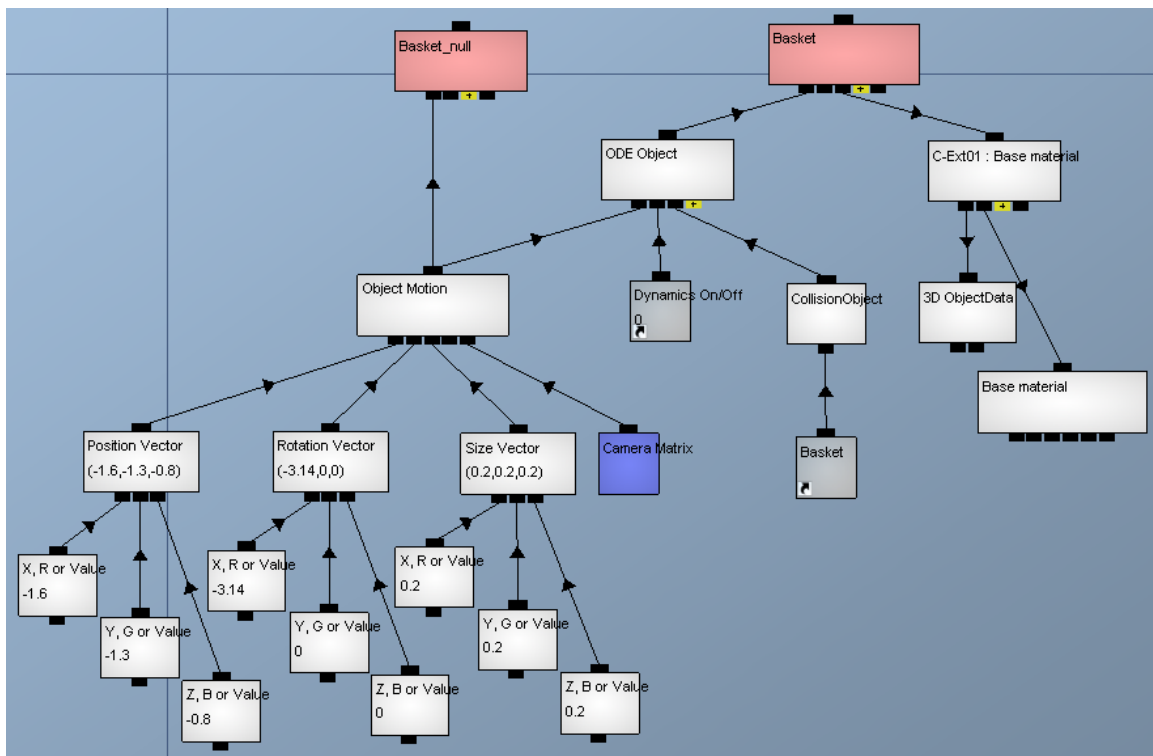
Κώδικας κατασκευής συστήματος σωματιδίων



Εικόνα 3.11

Ιδιότητες συστήματος σωματιδίων

- Το αντικείμενο υποδοχέας. Είναι το αντικείμενο που ελέγχει ο χρήστης μετακινώντας την sensor bar στο πραγματικό κόσμο. Ουσιαστικά, η κίνηση του υποδοχέα γίνεται έμμεσα καθώς αυτός είναι συνδεδεμένος με την μήτρα κίνησης της κινητής κάμερας, οπότε ακολουθεί την πορεία της. Την κάμερα θα εξετάσουμε στη παράγραφο 3.1.2. Ο υποδοχέας φτιάχθηκε στο πρόγραμμα 3D Studio Max (κανάλι C-Ext01 : Base material και τα παιδιά του) και εισήχθη έπειτα στο Quest3D. Στην εικόνα 3.12 βλέπουμε τον κώδικα κατασκευής του και παρατηρούμε ότι είναι και αυτό ένα ODE σώμα που έχει χαρακτηριστεί ως στατικό, οπότε δε μετακινείται από φυσικές δυνάμεις όπως η βαρύτητα ή από τη σύγκρουση με ένα άλλο σώμα. Επειδή είναι ένα πολύπλοκο σώμα, το αντικείμενο συγκρούσεων δίνεται απ' το κανάλι CollisionObject το οποίο τροφοδοτείται με την γεωμετρία του σχήματος από την συντόμευση του υποδοχέα. Και εδώ χρησιμοποιούμε το τέχνασμα της δημιουργίας ενός Null αντικειμένου για να μπορεί να αλλάζει η μήτρα κίνησης του υποδοχέα. Το κανάλι Camera Matrix είναι η μήτρα κίνησης της κινητής κάμερας και λειτουργεί ως γονική μήτρα ώστε ο υποδοχέας να την ακολουθεί.



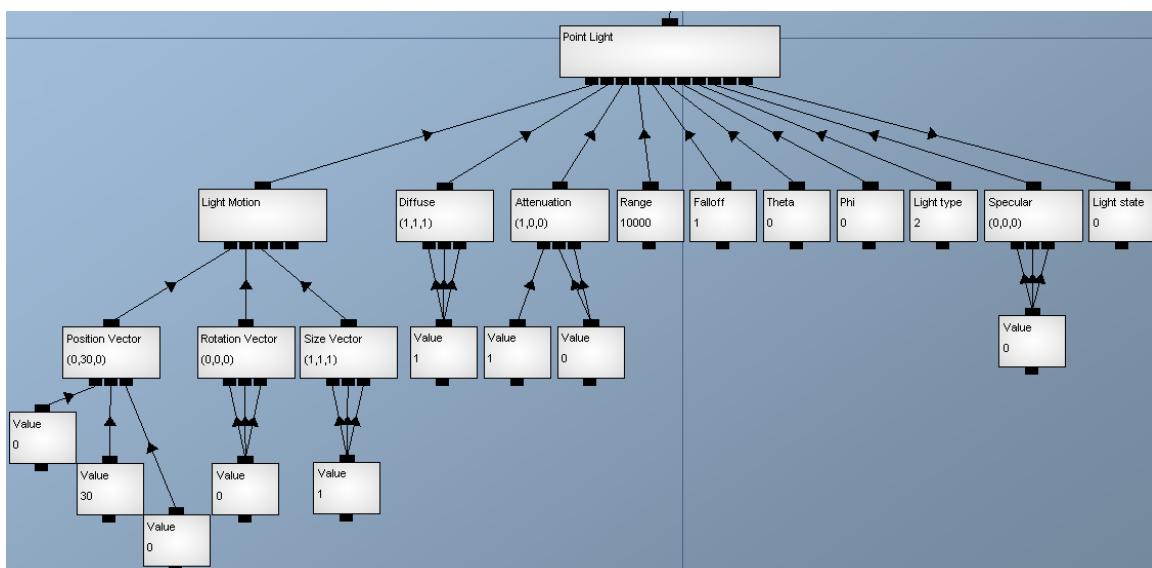
Εικόνα 3.12
Κώδικας κατασκευής υποδοχέα

3.1.2) Βοηθητικά Αντικείμενα

Είναι τα αντικείμενα που βοηθούν στην καλύτερη οπτικοποίηση της σκηνής. Πρόκειται για τα εξής :

- Φωτεινή Πηγή. Το Quest3D υποστηρίζει τρεις τύπους φωτεινών πηγών : κατευθυντική (directional light), προβολική (spot light) και σημειακή (point light). Εδώ χρησιμοποιούμε σημειακή φωτεινή πηγή. Στην εικόνα 3.13 βλέπουμε τα στοιχεία που την απαρτίζουν :

- Μήτρα κίνησης (Light Motion) που ορίζει τη θέση, το προσανατολισμό και το μέγεθος της φωτεινής πηγής.
- Το χρώμα του διαχεόμενου (diffuse) φωτός.
- Η εξασθένηση (attenuation) της εντάσεως του φωτός που είναι αντιστρόφως ανάλογη της απόστασης.
- Μέγιστη απόσταση (range), πέραν της οποίας δεν έχει επίδραση.
- Μείωση της φωτεινότητας (illumination) ανάμεσα στον εσωτερικό κώνο και στην εξωτερική πλευρά του εξωτερικού κώνου της προβολικής πηγής.
- Θήτα (Theta) είναι η γωνία σε ακτίνια του εσωτερικού κώνου προβολικής πηγής. Η τιμή της κυμαίνεται από 0 μέχρι την τιμή Φι ακτίνια.
- Φι (Phi) είναι η γωνία σε ακτίνια που ορίζει την εξωτερική πλευρά του εξωτερικού κώνου προβολικής πηγής. Σημεία εκτός του εξωτερικού κώνου δε φωτίζονται. Η τιμή της κυμαίνεται από 0 μέχρι π ακτίνια.
- Ο τύπος φωτεινής πηγής (Light type).
- Το χρώμα ανάκλασης (Specular) που χρησιμοποιούν μερικά αντικείμενα για τη δημιουργία λάμψης.
- Κατάσταση φωτεινής πηγής : ενεργή/ανενεργή.

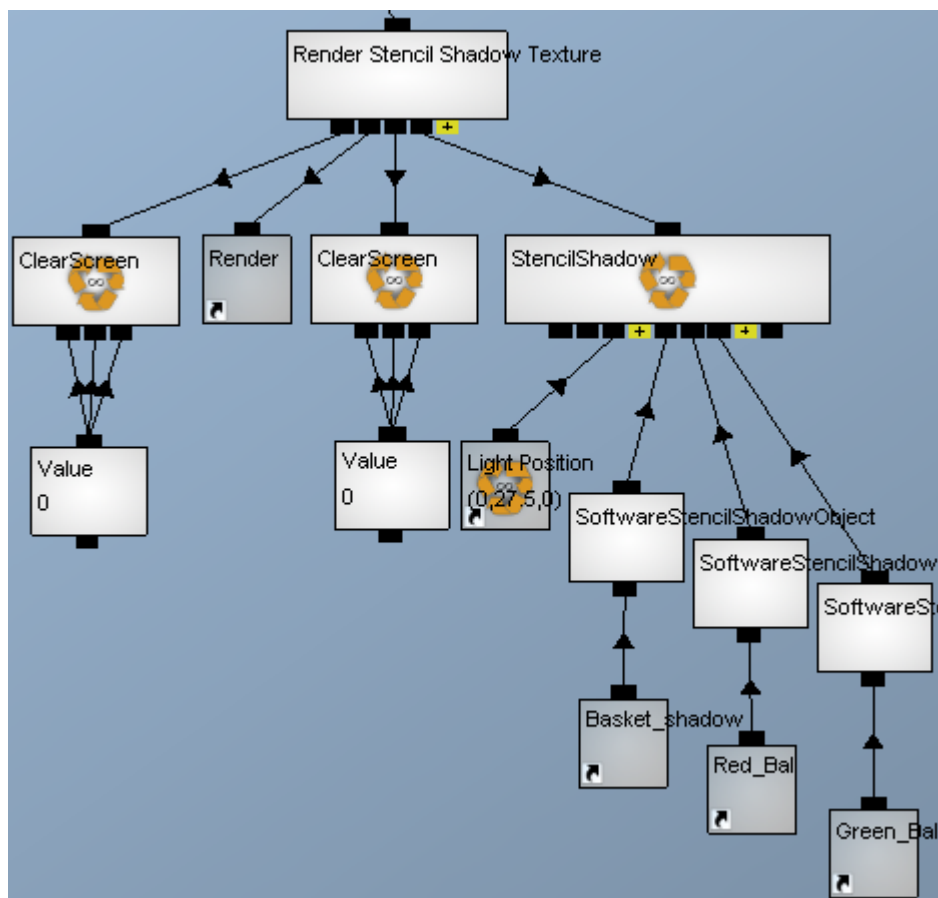


Εικόνα 3.13

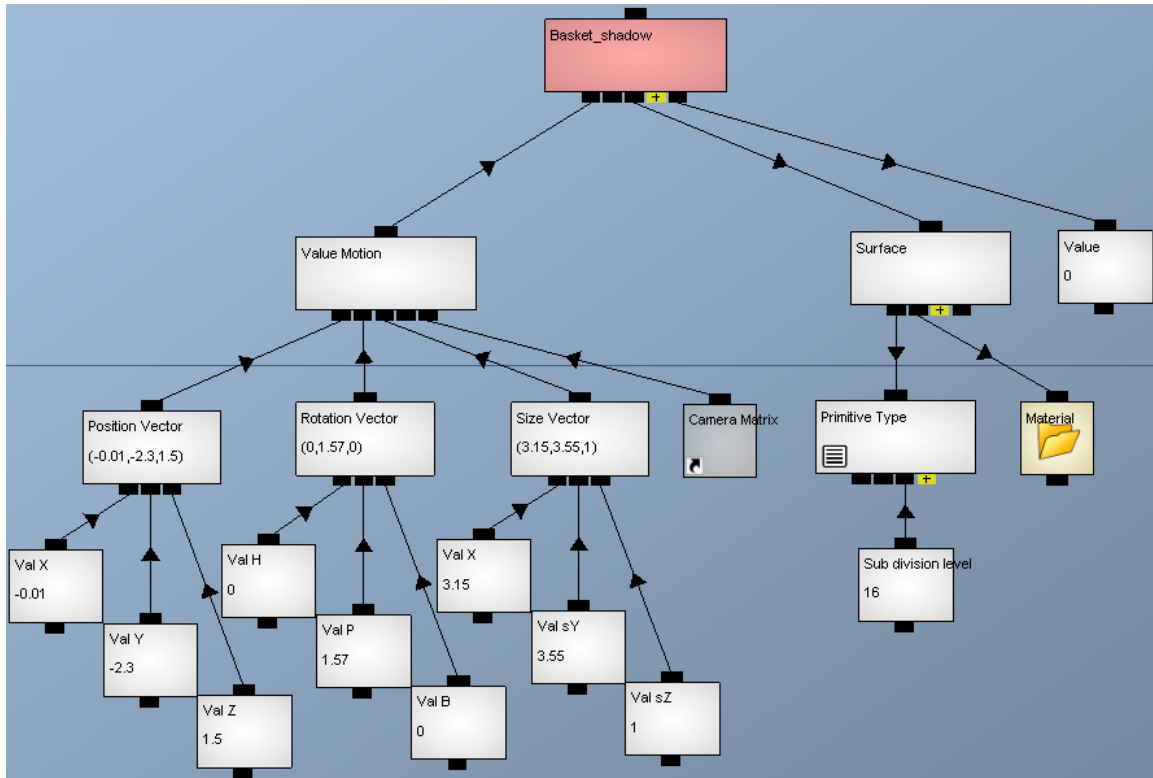
Κώδικας κατασκευής φωτεινής πηγής

Το Quest3D μπορεί να αξιοποιήσει μέχρι 50 φωτεινές πηγές. Μια ενδιαφέρουσα λεπτομέρεια είναι η ικανότητα να χρησιμοποιήσουμε μία φωτεινή πηγή με αρνητικές τιμές χρωμάτων, ώστε να αφαιρέσουμε φωτεινότητα από μια σκηνή (παραδείγματος χάρη από αυτοφωτιζόμενα αντικείμενα) κάνοντάς την πιο σκοτεινή.

- Σκιές. Οι δυο σφαίρες και ο υποδοχέας είναι τα αντικείμενα που έχουν σκιές για να μπορεί ο χρήστης να υπολογίζει καλύτερα τις θέσεις τους στον εικονικό κόσμο. Στην εικόνα 3.14 φαίνεται ο κώδικας κατασκευής των σκιών των τριών αντικειμένων. Το κανάλι StencilShadow δημιουργεί τις φωτοσκιάσεις των shadow αντικειμένων που έχει ως παιδιά. Η πρώτη συντόμευση του είναι το διάνυσμα θέσης της φωτεινής πηγής. Το κανάλι Software Stencil Shadow Object δημιουργεί το shadow αντικείμενο του παιδιού του. Μπορεί να δημιουργήσει σκιά οποιουδήποτε αντικειμένου σε πραγματικό χρόνο. Όμως παρατηρούμε ότι η σκιά του πρώτου σώματος είναι του Basket_shadow και όχι του αντικειμένου Basket (εικόνα 3.12) που είναι ο υποδοχέας. Το Basket_shadow είναι ένα τετράγωνο ίδιων διαστάσεων με τον υποδοχέα και χρησιμοποιεί την μήτρα κίνησης του ώστε να βρίσκεται ακριβώς από κάτω του και ακολουθεί την κίνησή του. Έτσι, στον εικονικό κόσμο φαίνεται η σκιά του Basket_shadow. Αυτό έγινε για να ελαχιστοποιηθεί ο υπολογιστικός φόρτος, καθώς η σκιά ενός απλού τετραγώνου είναι πολύ ευκολότερο να γίνει render από τη σκιά ενός πολύπλοκου αντικειμένου. Στην εικόνα 3.15 βλέπουμε τον κώδικα κατασκευής του Basket_shadow.



Εικόνα 3.14
Κώδικας κατασκευής σκιών

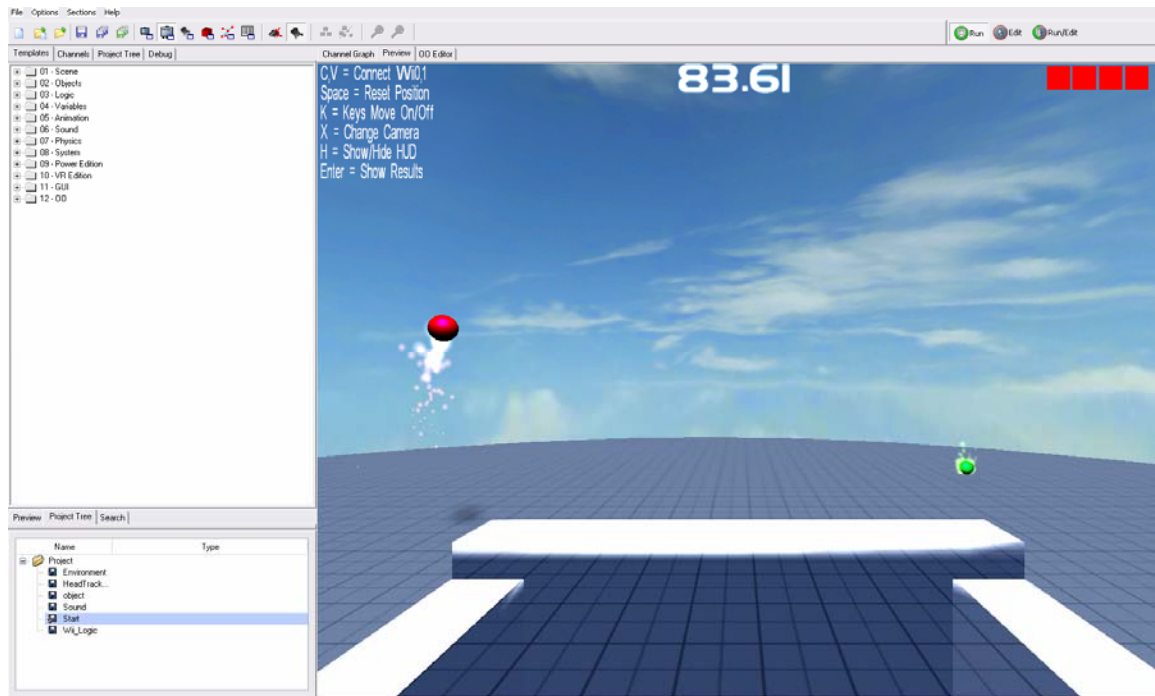


Εικόνα 3.15

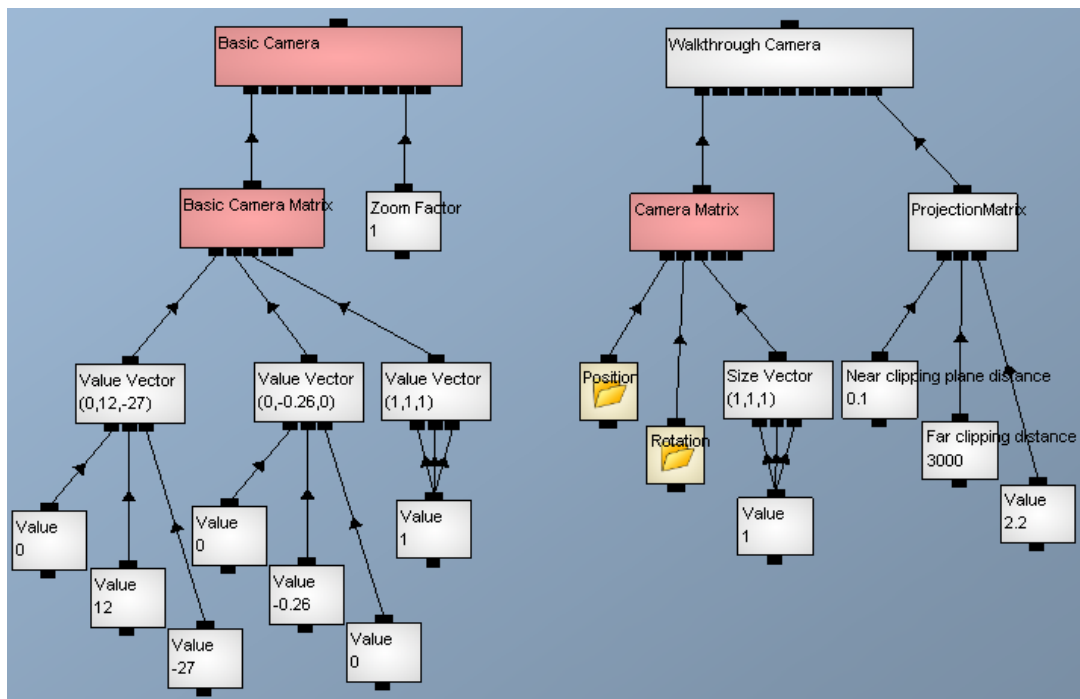
Κώδικας κατασκευής αντικειμένου Basket_shadow

Παρατηρούμε στην εικόνα 3.15 το δεξιότερο παιδί του καναλιού Basket_shadow. Είναι μια μεταβλητή με τιμή 0 που ορίζει ότι το συγκεκριμένο αντικείμενο δε θα γίνει rendered. Οπότε, γίνεται rendered μόνο η σκιά του και φαίνεται σαν να είναι η σκιά του υποδοχέα ενώ το ίδιο το αντικείμενο δεν είναι ορατό αφού δε γίνεται rendered.

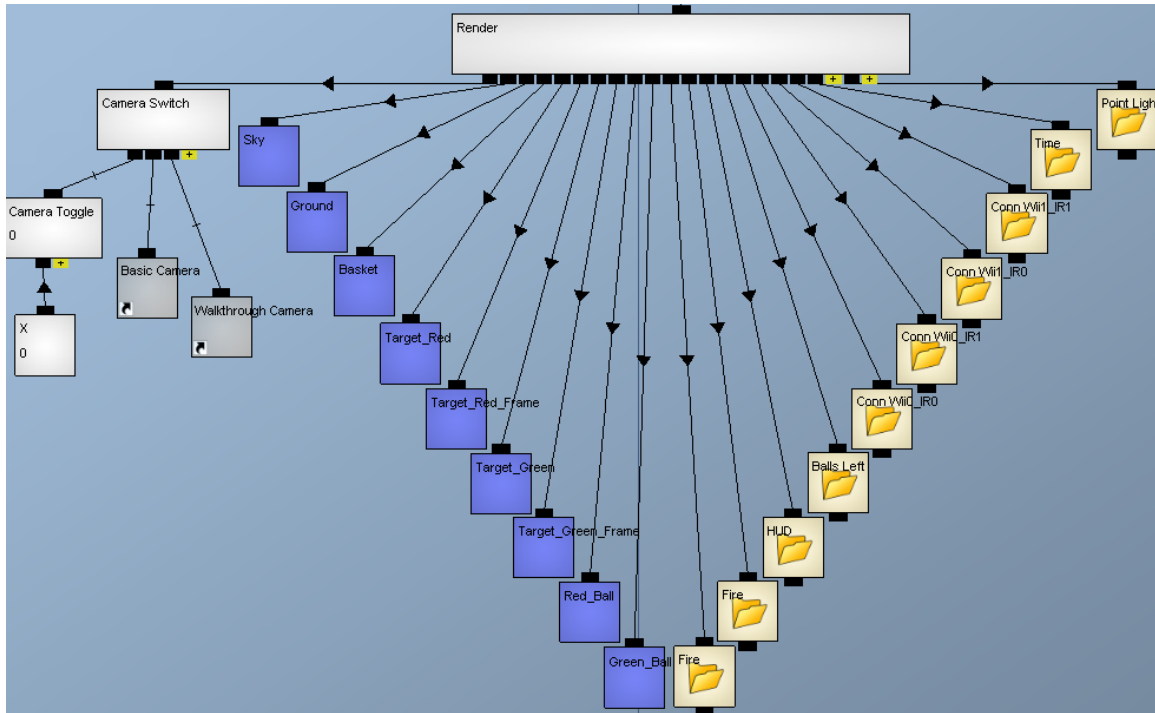
- **Κάμερες.** Υπάρχουν δύο κάμερες και ο χρήστης με το πάτημα ενός πλήκτρου επιλέγει ποια από τις δύο θα χρησιμοποιεί. Η πρώτη είναι μια ακίνητη κάμερα με το οπτικό πεδίο της να φαίνεται στην εικόνα 3.1 και η δεύτερη είναι μια κινητή κάμερα που βρίσκεται πάνω από τον υποδοχέα και το οπτικό της πεδίο φαίνεται στην εικόνα 3.16. Όπως εξηγήσαμε και στη παράγραφο 3.1.1, ο χρήστης κρατώντας στα χέρια του τη sensor bar και μετακινώντας την, επηρεάζει την πορεία της κινητής κάμερας και αναλόγως μετατοπίζεται και ο υποδοχέας καθώς είναι συνδεδεμένος με την κάμερα μέσω της μήτρας κίνησής της. Είναι ευκολότερο να μετατοπίζουμε στον εικονικό κόσμο την κινητή κάμερα γιατί δεν είναι ένα φυσικό σώμα, όπως είναι ο υποδοχέας, και άρα δεν ασκούνται φυσικές δυνάμεις πάνω της. Ο κώδικας κατασκευής των δύο καμερών φαίνεται στην εικόνα 3.17. Οι δύο φάκελοι Position και Rotation που προσδιορίζουν την θέση και περιστροφή της κινητής (Walkthrough) κάμερας θα αναλυθούν περισσότερο στις παραγράφους 3.3.2 και 3.3.3. Τέλος, στην εικόνα 3.18 φαίνεται το κανάλι που κάνει το rendering όλων των αντικειμένων του εικονικού κόσμου και η λειτουργία της επιλογής των καμερών (πρώτο παιδί). Το τελευταίο παιδί είναι η πηγή φωτός της σκηνής (εξηγείται στην εικόνα 3.13). Οι φάκελοι HUD, Balls left, Conn Wii0_IR0, Conn Wii0_IR1, Conn Wii1_IR0, Conn Wii1_IR1, Time δίνουν πληροφορίες στο χρήστη πάνω στην οθόνη και είναι παρόμοιου τύπου κανάλια. Ως παράδειγμα εξηγείται το αντικείμενο Time, που δείχνει τον χρόνο που έχει στη διάθεσή του ο παίκτης, στη παράγραφο 3.2.1.



Εικόνα 3.16
Προοπτική της κινητής κάμερας



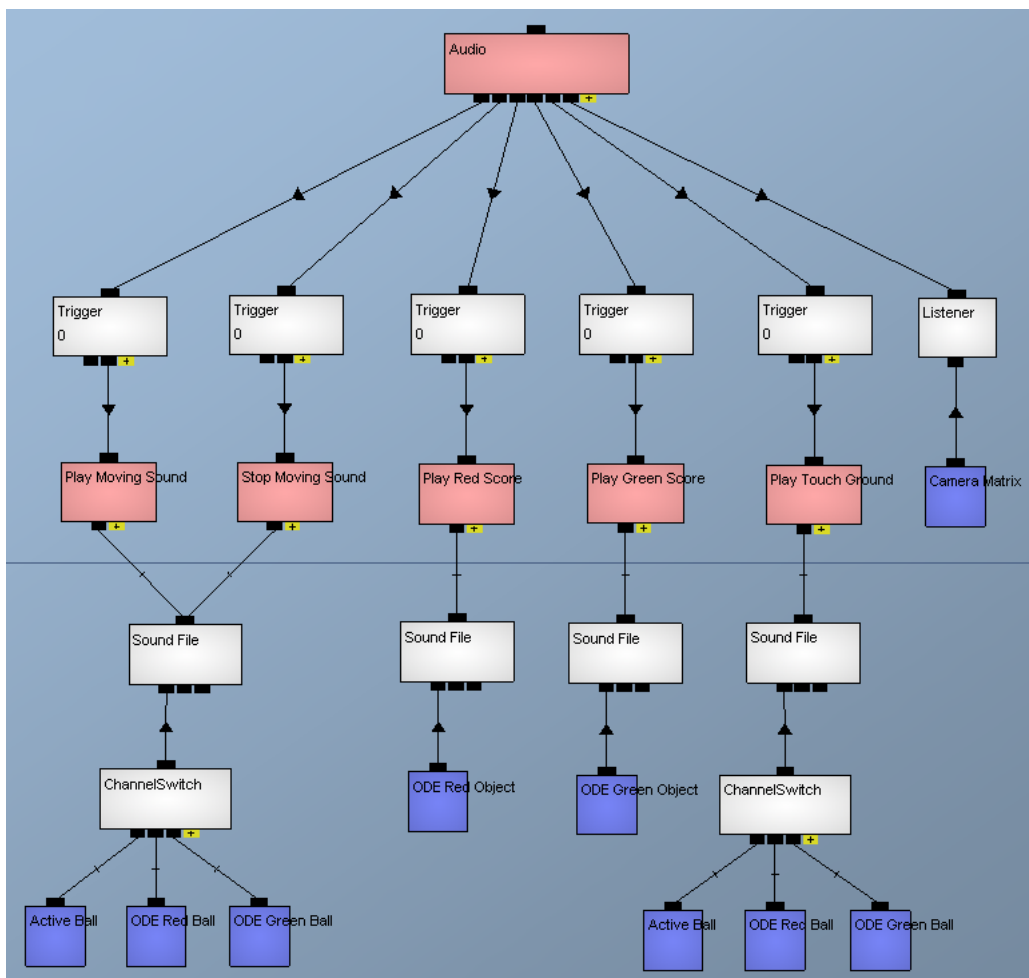
Εικόνα 3.17
Κώδικας κατασκευής των δύο καμερών



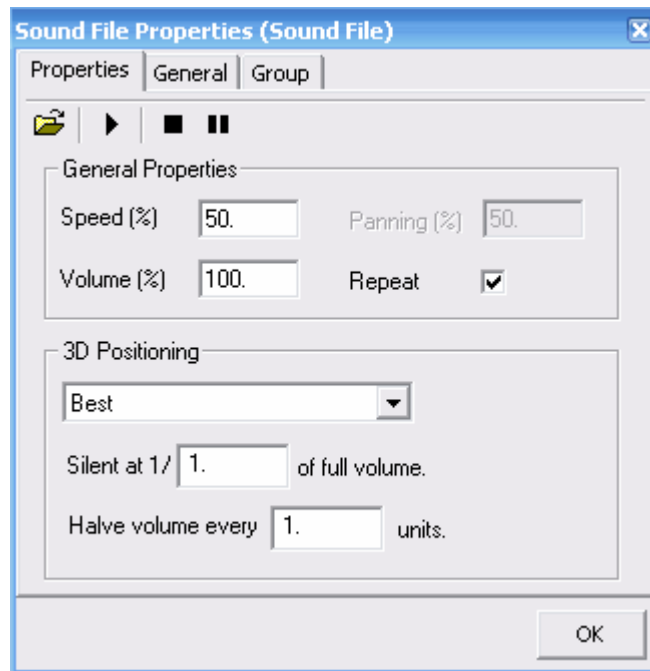
Εικόνα 3.18
 Rendering των αντικειμένων

3.1.3) Ήχος

Το Quest3D χρησιμοποιεί wav αρχεία και μπορεί να προσομοιώσει τρισδιάστατο ήχο αν υπάρχει το κατάλληλο surround ηχητικό σύστημα. Στην εικόνα 3.19 βλέπουμε τον κώδικα υπεύθυνο για τους ήχους της εφαρμογής. Παρατηρούμε ότι πριν όλες τις εντολές έναρξης/παύσης υπάρχει ένα κανάλι trigger το οποίο εμποδίζει την εντολή που το ακολουθεί να εκτελείται σε κάθε frame. Η εντολή καλείται μόνο όταν πρέπει από την λογική του προγράμματος που θα δούμε πιο αναλυτικά στη παράγραφο 3.2.3. Από αριστερά προς τα δεξιά, βλέπουμε το κανάλι που ξεκινάει τον ήχο που κάνει η σφαίρα όταν είναι στον αέρα, το κανάλι που σταματάει αυτόν τον ήχο, το κανάλι που παίζει τον ήχο όταν ο παίκτης καταφέρει να βάλει την κόκκινη σφαίρα πάνω απ'τον κόκκινο στόχο, ομοίως για την πράσινη σφαίρα και το κανάλι που παίζει τον ήχο όταν ο παίκτης δε καταφέρει να πιάσει την σφαίρα με τον υποδοχέα οπότε αυτή ακουμπάει το έδαφος.



Εικόνα 3.19
Κώδικας κατασκευής των ήχων της εφαρμογής



Εικόνα 3.20
Ιδιότητες καναλιού Sound File

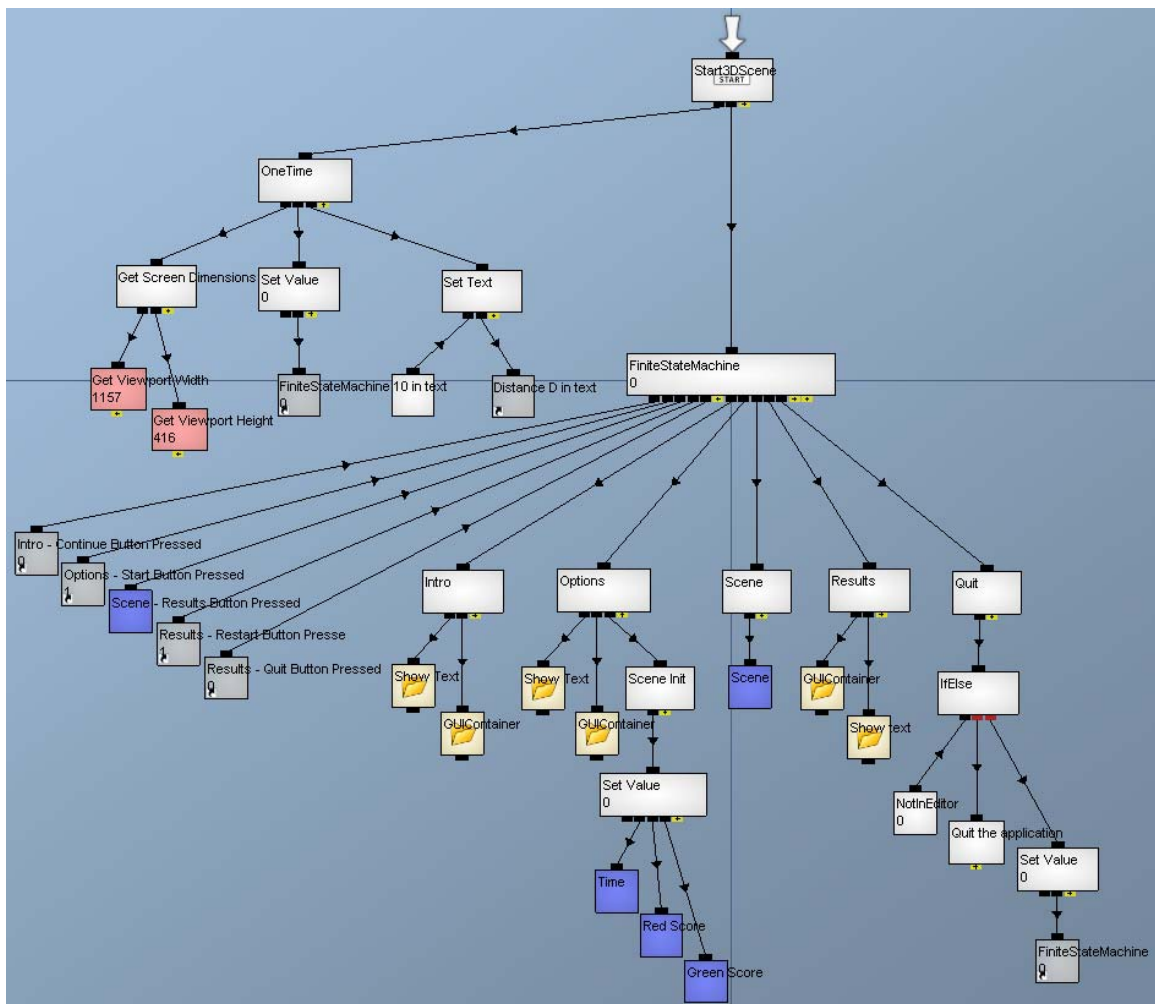
Βλέπουμε στην εικόνα 3.20 τις ιδιότητες του καναλιού Sound File : τα κουμπιά αναπαραγωγής, την ταχύτητα αναπαραγωγής (speed), την ένταση του ήχου (volume), απενεργοποίηση των δεξιών ή των αριστερών ηχείων (panning), επανάληψη (repeat), τρισδιάστατο ηχητικό σύστημα (3D Positioning), σε ποιο ποσοστό της μέγιστης εντάσεως σταματάει τελείως ένας ήχος (silent at % of full volume, αυτή την επιλογή τη χρησιμοποιούμε για να ελαττώσουμε τον υπολογιστικό φόρτο όταν τα αντικείμενα που παράγουν τον ήχο βρίσκονται μακριά απ'τον ακροατή, οπότε μας συμφέρει να τα σιγήσουμε εντελώς), σε πόσες μονάδες απόστασης μειώνεται κατά 50% η ένταση του ήχου (Halve volume every units). Το συγκεκριμένο κανάλι Sound File της εικόνας 3.20 είναι το υπεύθυνο για τον ήχο της σφαίρας όταν αυτή ίπταται και γι'αυτό είναι ενεργοποιημένη η επιλογή επανάληψη, ώστε ο ήχος να επαναλαμβάνεται μέχρι να δοθεί η εντολή παύσης.

Για τον τρισδιάστατο ήχο, πρέπει να οριστεί η θέση της πηγής του ήχου και η θέση του δέκτη-ακροατή. Η θέση της πηγής είναι το πρώτο παιδί του καναλιού Sound File. Παραδείγματος χάρη στον ήχο της επαφής της σφαίρας με το έδαφος, η πηγή είναι είτε το φυσικό σώμα κόκκινη σφαίρα ή το φυσικό σώμα πράσινη σφαίρα, αναλόγως ποιά από τις δύο είναι η ενεργή σφαίρα κάθε φορά που εκτελείται το κανάλι Sound File. Η ενεργή σφαίρα προσδιορίζεται από μία μεταβλητή κατάστασης (Active Ball) που μεταβάλλει την τιμή του διακόπτη (ChannelSwitch). Η θέση του δέκτη ορίζεται απ'το κανάλι Listener που έχει ως παιδί τη μήτρα κίνησης της κινητής κάμερας. Επομένως, η εξομοίωση του τρισδιάστατου ήχου γίνεται σαν να βρίσκεται ο χρήστης στη θέση της κινητής κάμερας (εικόνα 3.16) άρα και του υποδοχέα, όπως εξηγήσαμε στο προηγούμενο υποκεφάλαιο.

3.2) Υλοποίηση Διεπαφής Χρήστη

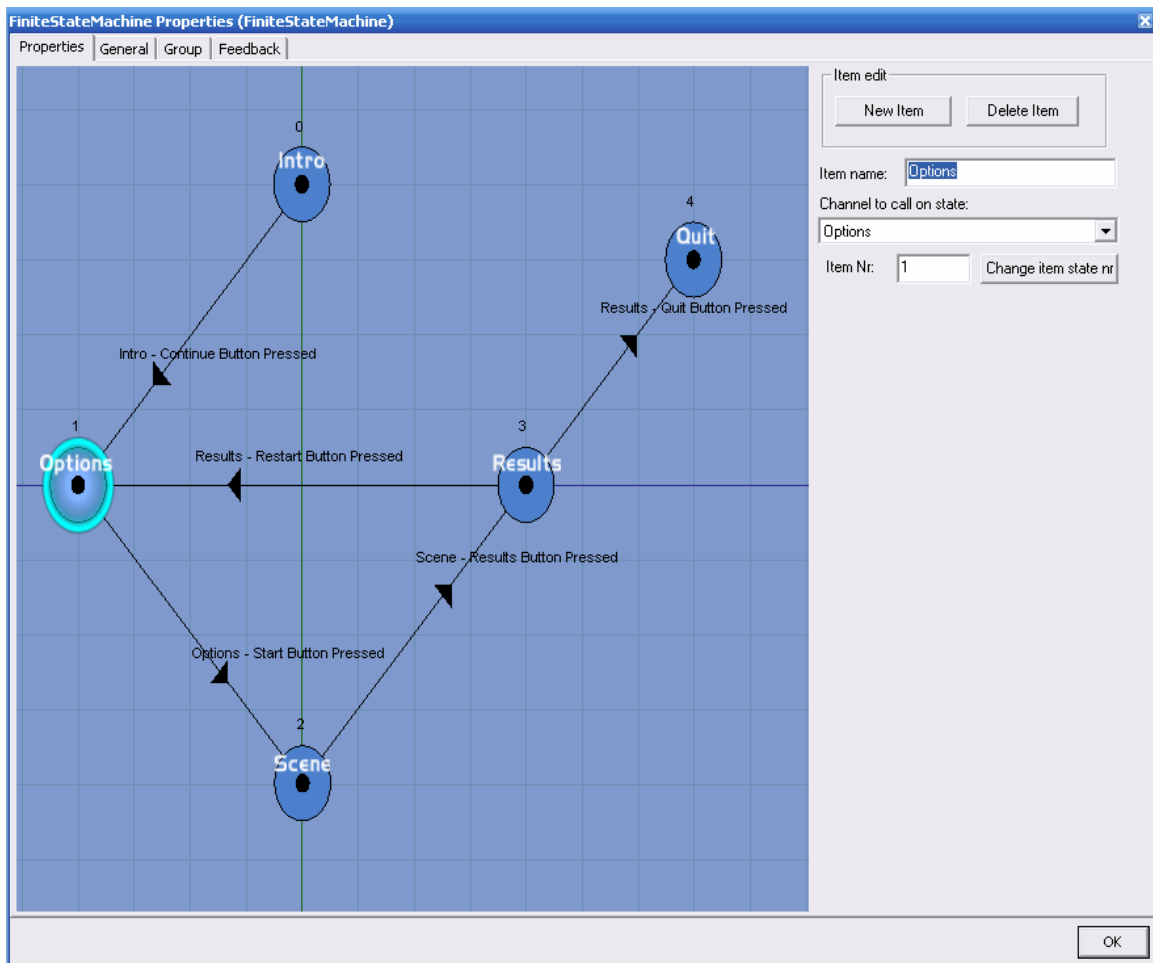
3.2.1) Γραφική Διεπαφή

Ας δούμε πρώτα την λογική διαδοχή των σκηνών της εφαρμογής. Βλέπουμε στην εικόνα 3.21 ότι το αρχικό κανάλι είναι το Start3DScene. Αυτό, όταν ξεκινήσει η εφαρμογή, ρωτάει το χρήστη για τις διαστάσεις του παραθύρου που θα χρησιμοποιήσει, και καθορίζει το χρώμα του φόντου της σκηνής και με ποιο πλήκτρο θα τερματίζει το πρόγραμμα. Έπειτα καλεί το κανάλι OneTime, του οποίου η λειτουργία είναι ότι τα παιδιά του καλούνται μόνο μία φορά. Το πρώτο παιδί, Get Screen Dimensions, μετράει το πλάτος και το ύψος του παραθύρου για να μπορούν να τα χρησιμοποιούν ως δεδομένα τα υπόλοιπα κανάλια. Το δεύτερο παιδί δίνει στη μηχανή καταστάσεων (FiniteStateMachine) την τιμή 0, ώστε να ξεκινήσει από την πρώτη της κατάσταση. Το τρίτο παιδί θέτει ως προκαθορισμένη τιμή της απόστασης μεταξύ των δύο Wii Remote τα 10 cm. Ο χρήστης μπορεί να αλλάξει αυτή την τιμή στην δεύτερη σκηνή.



Εικόνα 3.21

Κώδικας των σκηνών της εφαρμογής



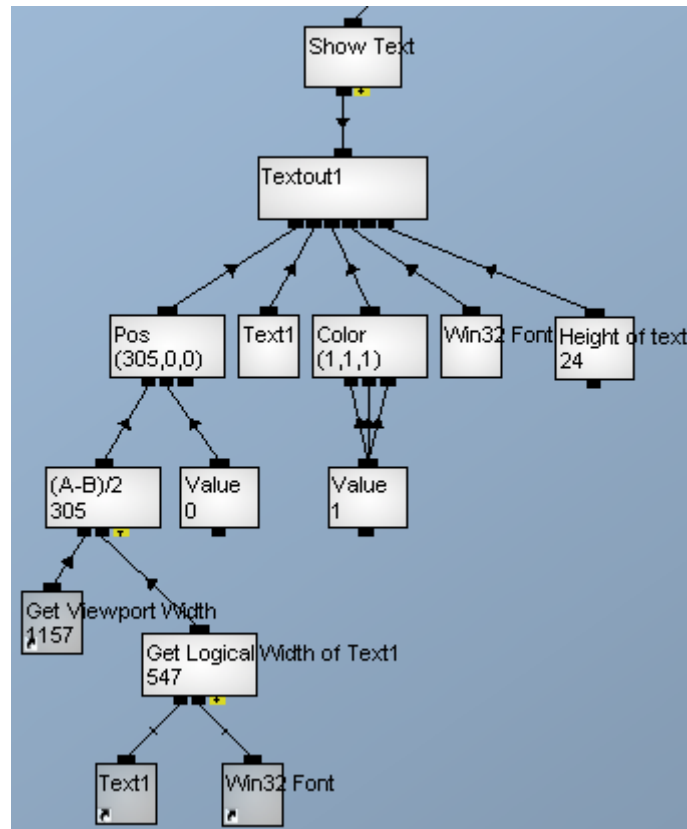
Εικόνα 3.22

Σκαρίφημα των σκηνών που ελέγχει η μηχανή καταστάσεων

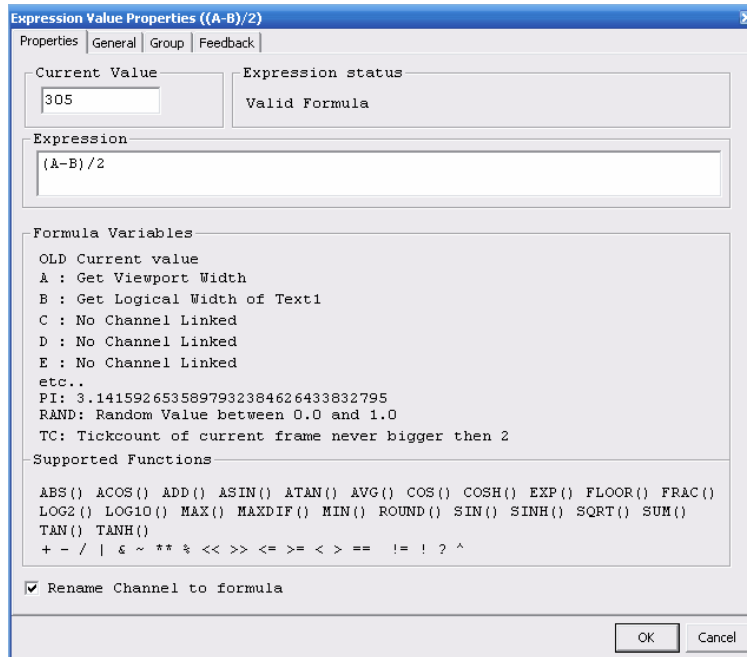
Μετά, καλείται η μηχανή καταστάσεων, που μας βοηθάει να δημιουργούμε και να χειριζόμαστε σύνθετες δομές. Στην εικόνα 3.22 φαίνεται πως συνδέονται μεταξύ τους οι καταστάσεις-σκηνές (μπλε κύκλοι) και με ποιές μεταβλητές (τα βέλη που συνδέουν τους κύκλους) γίνεται η μετάβαση από την μία στην άλλη. Οι μεταβλητές αυτές είναι τα πέντε πρώτα παιδιά της μηχανής καταστάσεων και αλλάζουν τιμή όταν ενεργοποιούνται τα κουμπιά των σκηνών, οπότε δίνουν εντολή στην μηχανή να μεταφερθεί η ροή προγράμματος στην επόμενη σκηνή.

Τα επόμενα παιδιά είναι οι τέσσερις σκηνές της εφαρμογής και το τελευταίο παιδί είναι το κανάλι υπεύθυνο για τον τερματισμό της. Η πρώτη σκηνή (κανάλι Intro) περιέχει το εισαγωγικό κείμενο και το κουμπί για να μεταφερθούμε στην επόμενη σκηνή. Σε αυτήν (κανάλι Options), ο χρήστης εισαγάγει την απόσταση μεταξύ των δύο Wii Remote, την γωνία που σχηματίζουν τα δύο Wii Remote με το επίπεδο της οθόνης, τον αριθμό των σφαιρών με τις οποίες θα παίξει και το βαθμό δυσκολίας. Ακόμα, μηδενίζονται οι τιμές των μεταβλητών του χρόνου και των δύο σκορ και όταν ενεργοποιηθεί το κουμπί αυτής της σκηνής ξεκινάει η επόμενη σκηνή που είναι και η κύρια σκηνή (κανάλι Scene). Στην κύρια σκηνή αρχίζει η αντίστροφη χρονική μέτρηση και είτε όταν αυτή μηδενιστεί ή όταν πατηθεί το πλήκτρο Enter, μεταφερόμαστε στην τελική σκηνή (κανάλι Results). Εκεί, πρώτα σταματάει ο ήχος και ύστερα παρουσιάζονται τα σκορ του παίκτη. Επίσης εμφανίζονται δύο κουμπιά, ένα για επανεκκίνηση της εφαρμογής (προς το κανάλι Options) και ένα για τον τερματισμό της (κανάλι Quit).

Ας δούμε ένα παράδειγμα δημιουργίας κειμένου. Στην εικόνα 3.23 φαίνεται το κανάλι, Textout1, που εμφανίζει το κείμενο. Τα παιδιά του ορίζουν με αυτή τη σειρά : τη θέση του κειμένου στην οθόνη, το κείμενο, το χρώμα του κειμένου, τη γραμματοσειρά του κειμένου και το ύψος του κειμένου. Επειδή το παράθυρο θέασης του χρήστη μπορεί να μεταβάλλεται, παίρνουμε τις διαστάσεις του τρέχοντος παραθύρου από τα κανάλια Get Viewport Width και Get Viewport Height (εικόνα 3.21) και τις χρησιμοποιούμε σε μαθηματικές εκφράσεις ώστε το κείμενο να βρίσκεται στις ίδιες σχετικές πάντα θέσεις ανεξάρτητα από το μέγεθος του παραθύρου.



Εικόνα 3.23
Κώδικας εμφάνισης κειμένου



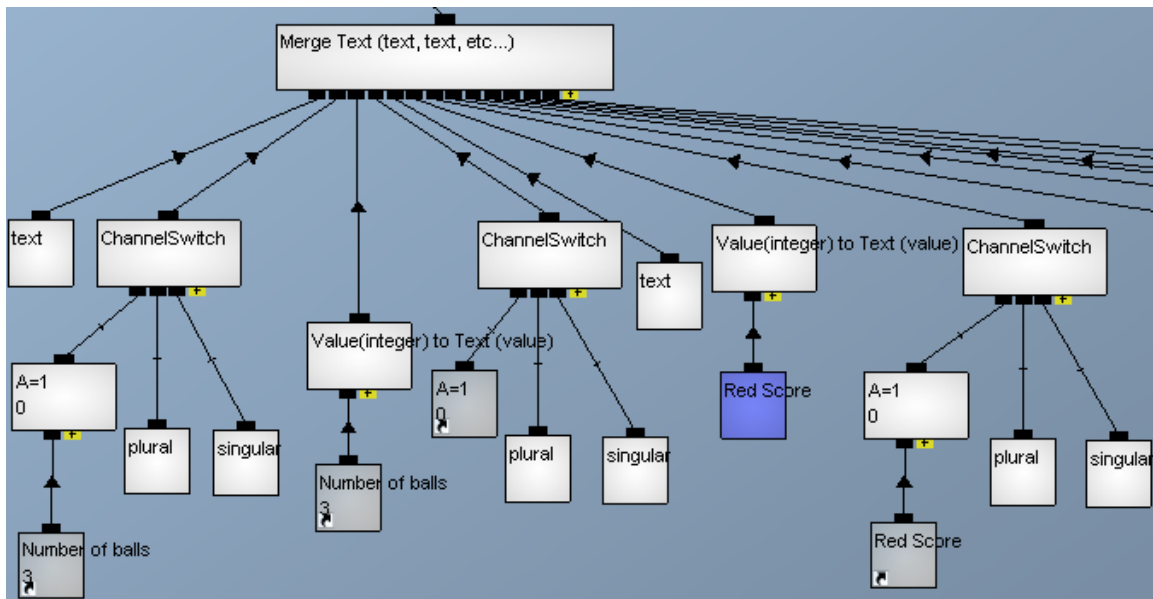
Εικόνα 3.24

Ιδιότητες καναλιού μαθηματικής έκφρασης

Στην εικόνα 3.24 φαίνονται οι ιδιότητες του καναλιού $(A-B)/2$ της εικόνας 3.23. Είναι ένα κανάλι με το οποίο μπορούμε πολύ εύκολα να διατυπώσουμε μαθηματικές εκφράσεις χρησιμοποιώντας ως μεταβλητές τα παιδιά του. Το πρώτο παιδί ονομάζεται ως A, το δεύτερο ως B, κτλ. Στη συγκεκριμένη περίπτωση τα δύο παιδιά είναι το πλάτος του παραθύρου και το συνολικό πλάτος του κειμένου. Έπειτα, χρησιμοποιώντας την έκφραση $(A-B)/2$ καταφέρνουμε να τοποθετούμε το κείμενο ακριβώς στη μέση του πλάτους του παραθύρου. Ομοίως μπορούμε να πράξουμε για να ορίσουμε σε ποιο ύψος του παραθύρου θα εμφανισθεί το κείμενο. Το πλάτος του κειμένου, μας το πληροφορεί το κανάλι Get Logical Width of Text1 που έχει ως παιδιά το κείμενο και τη γραμματοσειρά του.

Το κείμενο μπορεί να αποτελείται από διάφορους τελεστές ώστε το τελικό αποτέλεσμα να εξαρτάται από λογικές εκφράσεις και μεταβλητές. Ένα παράδειγμα φαίνεται στην εικόνα 3.25 όπου χρησιμοποιούμε το κανάλι Merge Text για να ενώσουμε όλα τα παιδιά του σε μορφή κειμένου και να το εμφανίσουμε στην οθόνη. Το παράδειγμα είναι απ' την τελική σκηνή που παρουσιάζονται τα αποτελέσματα. Το δεύτερο παιδί του Merge Text ελέγχει με μία μαθηματική έκφραση αν ο αριθμός των σφαιρών είναι η μονάδα και αν είναι ψευδής παίρνει την τιμή 0 προωθώντας στο κανάλι ChannelSwitch το παιδί plural αλλιώς αν είναι αληθής παίρνει την τιμή 1 και προωθεί το παιδί singular, ώστε το κείμενο να γίνει «τις» ή «την». Το τρίτο παιδί μετατρέπει μια αριθμητική τιμή, τον αριθμό των σφαιρών, σε κείμενο. Το τέταρτο λειτουργεί όπως το δεύτερο για να γράψει είτε «σφαίρες» είτε «σφαίρα». Το έκτο γράφει σε μορφή κειμένου τον αριθμό των κόκκινων σφαιρών που βρήκαν στόχο και το έβδομο ελέγχει αυτό τον αριθμό για να γράψει είτε «κόκκινες» είτε «κόκκινη». Το τελικό αποτέλεσμα, θεωρώντας παραδείγματος χάρη 5 σφαίρες και από αυτές 1 κόκκινη βρήκε στόχο, είναι :

Από τις 5 σφαίρες, βάλανε 1 κόκκινη.



Εικόνα 3.25

Παράδειγμα ενοποιημένου (merged) κειμένου

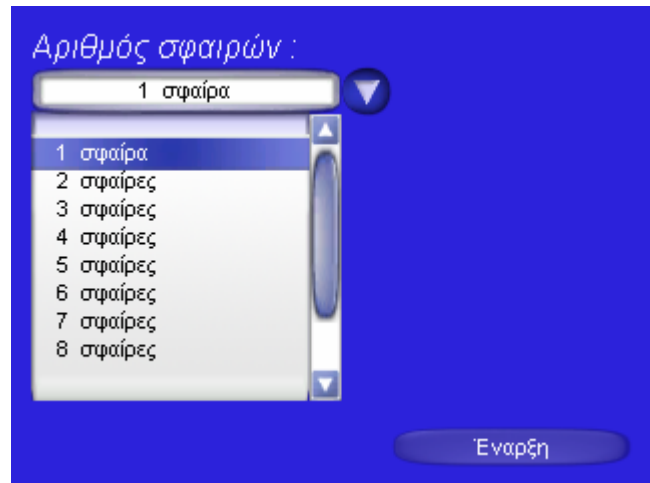
Βλέπουμε στην εικόνα 3.26 ένα παράδειγμα μενού επιλογών (συγκεκριμένα ένα combobox) και κουμπιού (button) και στην εικόνα 3.27 τον κώδικα κατασκευής τους. Στο κανάλι GUIContainer μπορούμε να συνδέουμε πολλά αντικείμενα GUI (Graphical User Interface - Γραφική Διεπαφή Χρήστη) και να τα χειριζόμαστε ως σύνολο, παραδείγματος χάρη αλλάζοντας τη θέση X του GUIContainer, αλλάζουν κατά το ίδιο X και όλα τα GUI παιδιά του. Το Combobox είναι ένα μενού επιλογών με αυτά τα χαρακτηριστικά :

- X pos : η X συντεταγμένη του μενού στην οθόνη.
- Y pos : η Y συντεταγμένη του μενού στην οθόνη.
- Width : το πλάτος του μενού.
- Height : το ύψος του μενού.
- Enabled : αν είναι ενεργό ή όχι.
- Visible : αν είναι ορατό ή όχι.
- Drop height : πόσο ύψος έχει το μενού όταν φαίνονται οι επιλογές.
- Items : Οι επιλογές του μενού.
- Selected item as text : Η επιλογή μας ως κείμενο.
- Selected item as value : Η επιλογή μας ως τιμή.

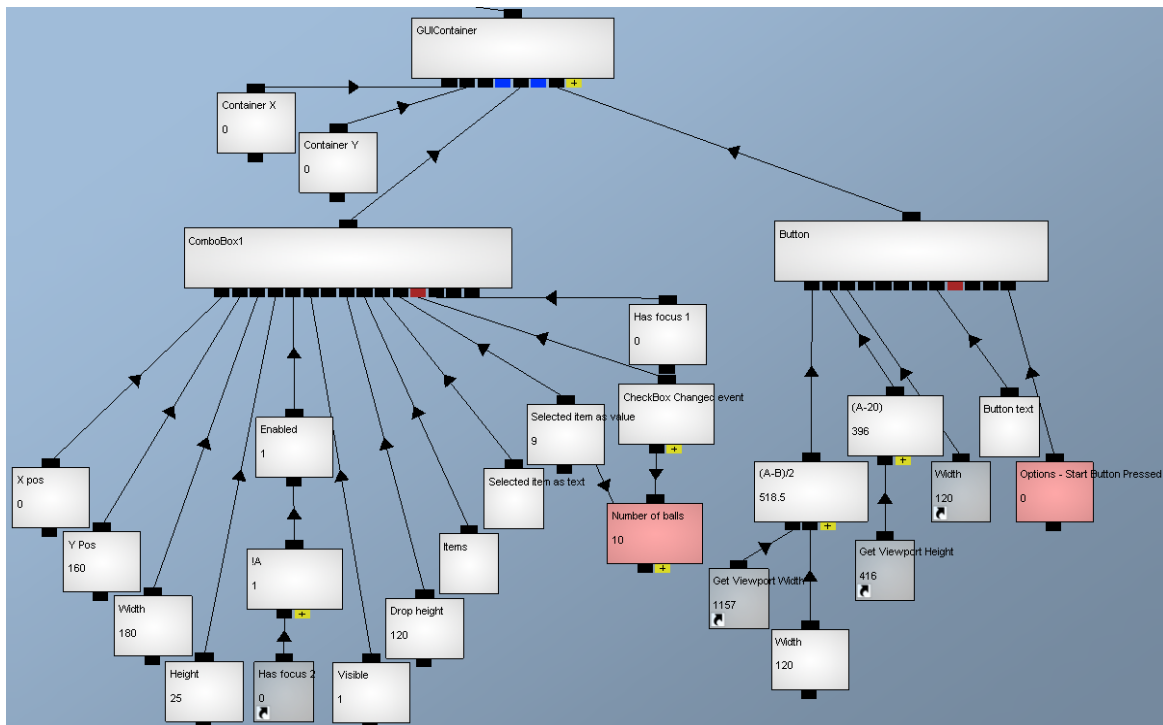
Πολύ σημαντικά είναι και τα τέσσερα τελευταία παιδιά του καναλιού Combobox. Αυτά είναι τα : Clicked event (ενέργεια που εκτελείται όταν χρησιμοποιούμε το GUI), Has focus (ο χρήστης έχει ενεργοποιήσει το συγκεκριμένο GUI), Mouse over (ο χρήστης μεταφέρει το ποντίκι πάνω απ'το GUI), Pressed (ο χρήστης κάνει click στο GUI). Στο combobox του παραδείγματος, χρησιμοποιούμε το Clicked event και το Has focus. Το Has focus παίρνει την τιμή 1 όταν ο χρήστης το επιλέξει με το ποντίκι. Τότε στα υπόλοιπα GUI η μεταβλητή Enable γίνεται False, καθώς είναι το λογικό αντίστροφο του Has focus του συγκεκριμένου GUI, οπότε απενεργοποιούνται για να μην μπερδεύονται μεταξύ τους τα GUI. Όταν κάνει ο χρήστης την επιλογή του, εκτελείται το κανάλι Checkbox Changed event που με τη σειρά του καλεί το κανάλι Number of balls που είναι η μαθηματική έκφραση :

$$A = 0 ? 1 : (A+1).$$

Το A είναι το πρώτο παιδί της έκφρασης, δηλαδή το κανάλι Selected item as value που μας δίνει τον αριθμό της επιλογής. Η πρώτη επιλογή έχει την τιμή 0, επειδή όμως δε θέλουμε να υπάρχει το 0 σαν επιλογή αριθμού των σφαιρών χρησιμοποιούμε την παραπάνω έκφραση, ώστε για την πρώτη επιλογή (A=0) η έκφραση παίρνει την τιμή 1, αλλιώς για τις υπόλοιπες επιλογές η μαθηματική έκφραση που δίνει τον αριθμό των σφαιρών είναι ο αριθμός της επιλογής + 1.



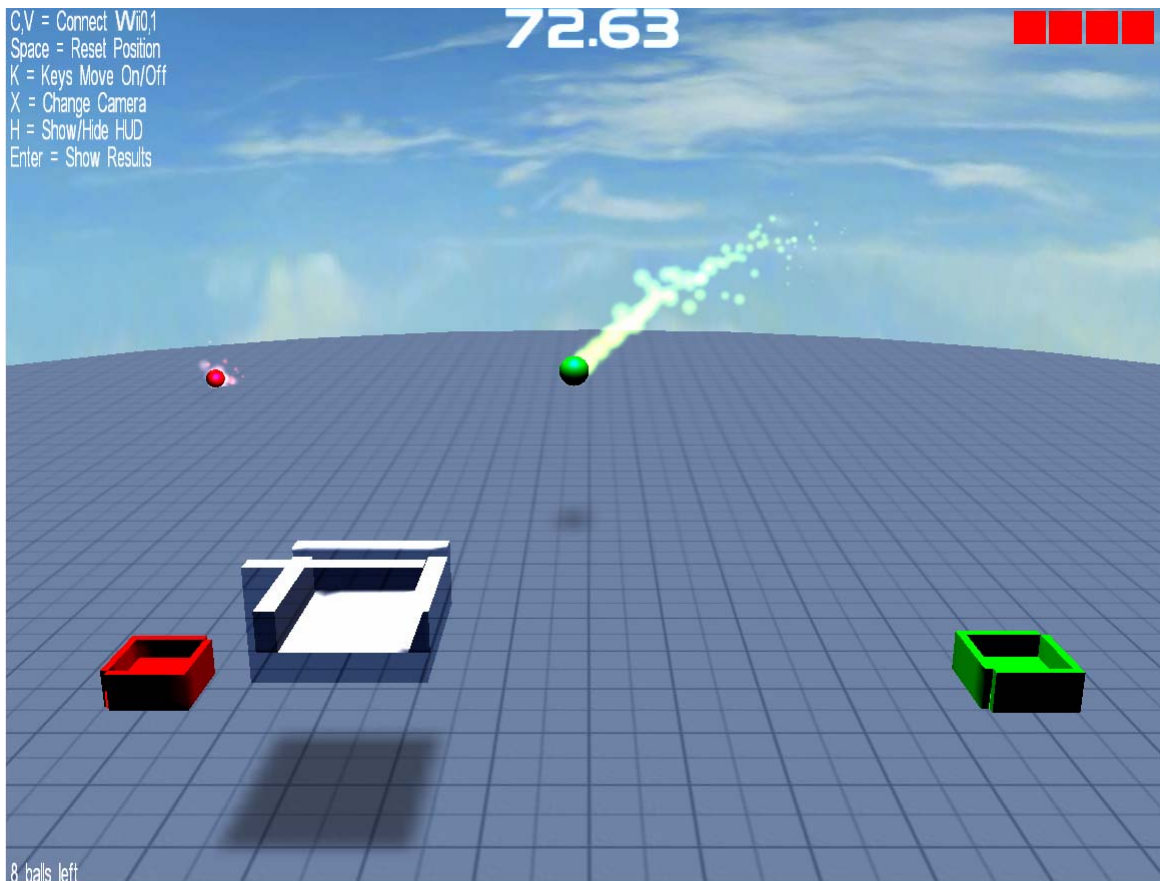
Εικόνα 3.26
Παράδειγμα μενού επιλογών και κουμπιού



Εικόνα 3.27
Κώδικας κατασκευής μενού επιλογών και κουμπιού

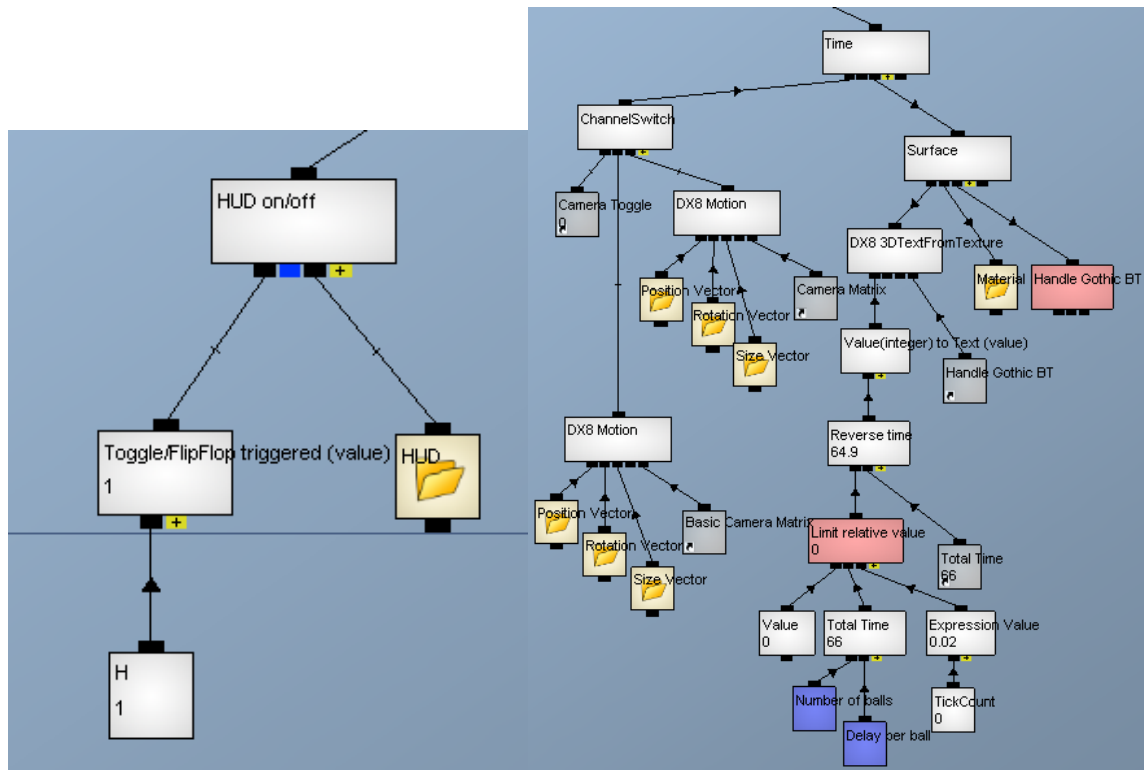
Το GUI Button λειτουργεί με τον ίδιο τρόπο με το Combobox, αλλά σε αυτό χρησιμοποιούμε την μεταβλητή Options – Start Button Pressed που όταν ενεργοποιούμε το κουμπί Έναρξη, παίρνει την τιμή 1 και δίνει εντολή στην μηχανή καταστάσεων να προχωρήσει στην επόμενη σκηνή που είναι και η κύρια.

Στην κύρια σκηνή το πάτημα του πλήκτρου Enter λειτουργεί με τον ίδιο τρόπο με τα buttons και θέτει στην μεταβλητή Scene – Results Button Pressed την τιμή 1 ώστε η μηχανή καταστάσεων να μας μεταφέρει στην τελική σκηνή προβολής των αποτελεσμάτων. Ακόμα, υπάρχει το αντικείμενο κειμένου HUD που εξηγεί τις λειτουργίες ορισμένων πλήκτρων, το αντικείμενο Time που δείχνει πόσο χρόνο έχει ακόμα στη διάθεσή του ο παίκτης, το αντικείμενο Balls left που πληροφορεί το χρήστη πόσες σφαίρες θα εκτοξευθούν ακόμα μέχρι το τέλος του χρόνου και τα αντικείμενα Conn Wii0_IR0, Conn Wii0_IR1, Conn Wii1_IR0, Conn Wii1_IR1 πάνω δεξιά που ενημερώνουν το χρήστη αν κάποιο απ'τα δύο Wii Remote δεν ανιχνεύει κάποιο απ'τα δύο άκρα της sensor bar (κόκκινο όταν δεν ανιχνεύεται, πράσινο όταν ανιχνεύεται). Στην εικόνα 3.28 βλέπουμε αυτά τα αντικείμενα γραφικής διεπαφής χρήστη. Το πάτημα του πλήκτρου H εμφανίζει και εξαφανίζει απ'την οθόνη τα αντικείμενα HUD και Balls left και στην εικόνα 3.29 φαίνεται πως γίνεται αυτό και δίνεται ως παράδειγμα ο κώδικας κατασκευής του αντικειμένου Time.



Εικόνα 3.28

Κείμενο και χρόνος στην προοπτική θέασης του χρήστη

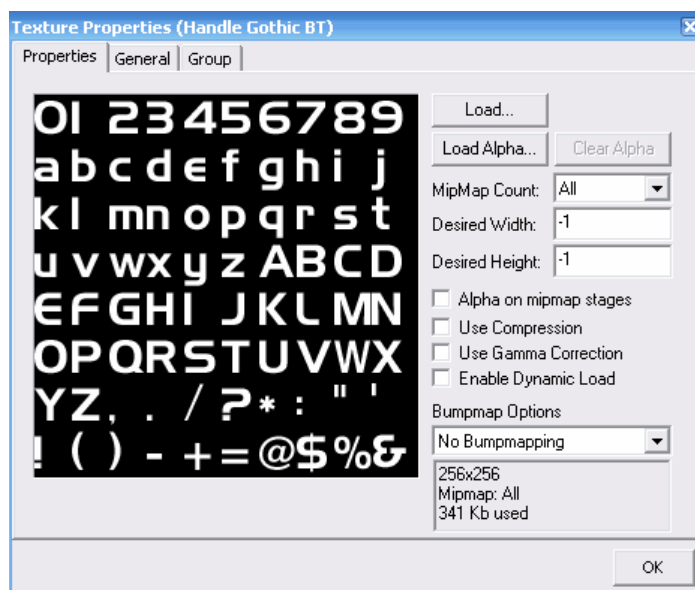


Εικόνα 3.29

Κώδικες κατασκευής κειμένου και χρόνου

Το κανάλι HUD on/off είναι ένα κανάλι ChannelSwitch το οποίο χρησιμοποιεί ως λογική σημαία το κανάλι Toggle/FlipFlop που είναι ένας διακόπτης που εναλλακτικά παίρνει τις τιμές 0 και 1 όταν πατάμε το πλήκτρο H. Έτσι, πατώντας το πλήκτρο, εμφανίζεται και εξαφανίζεται στην οθόνη το αντικείμενο HUD. Το ίδιο κανάλι HUD on/off χρησιμοποιεί και το αντικείμενο Balls left οπότε εμφανίζονται και εξαφανίζονται απ' την οθόνη ταυτόχρονα με το HUD.

Ο χρόνος είναι ένα διδιάστατο αντικείμενο που αποτελείται από την μήτρα κίνησης και την επιφάνειά (Surface) του. Η μήτρα κίνησης είναι το κανάλι ChannelSwitch που μετατρέπεται είτε στη μήτρα του δεύτερου παιδιού είτε στη μήτρα του τρίτου παιδιού, αναλόγως την τιμή του πρώτου παιδιού που είναι η συντόμευση της μεταβλητής Camera Toggle. Η μεταβλητή αυτή αλλάζει τιμή όταν εναλλάσσουμε την κάμερα που χρησιμοποιούμε για να παρατηρούμε τη σκηνή. Έτσι, καταφέρνουμε να εμφανίζεται ο χρόνος και στις δύο κάμερες στο ίδιο σημείο. Οι δύο μήτρες διαφέρουν μεταξύ τους γιατί η μία κάμερα είναι ακίνητη (Basic Camera Matrix) ενώ η άλλη κινητή (Camera Matrix) και γιατί έχουν διαφορετικές τιμές μεγέθυνσης (zoom). Η επιφάνεια δημιουργείται από τα παιδιά της : DX8 3DTextFromTexture, Material και Handle Gothic BT. Το τελευταίο είναι η χρησιμοποιούμενη υφή (texture) και η απεικόνισή της φαίνεται στην εικόνα 3.30. Το Material είναι φάκελος και περιέχει τα κανάλια που ρυθμίζουν το χρώμα της επιφάνειας. Το κανάλι DX8 3DTextFromTexture λαμβάνει ως παραμέτρους ένα κείμενο και μία υφή και δημιουργεί τα γεωμετρικά δεδομένα (Object Data) του αντικειμένου Time που χρησιμοποιεί η επιφάνειά του. Δηλαδή, με μία εικόνα όπως της 3.30, μπορούμε να φτιάξουμε τη γραμματοσειρά που θέλουμε και έπειτα να εμφανίσουμε διδιάστατο κείμενο, αυτής της γραμματοσειράς, στην οθόνη με τις ιδιότητες ενός κανονικού αντικειμένου.



Εικόνα 3.30

Η υφή της επιφάνειας του αντικειμένου που απεικονίζει το χρόνο

Το κείμενο του DX8 3DTextFromTexture προέρχεται απ'το κανάλι Value to Text. Αυτό παίρνει μία αριθμητική τιμή (value) και τη μετατρέπει σε κείμενο (text). Η αριθμητική αυτή τιμή είναι η αντίστροφη μέτρηση του χρόνου που βλέπει ο χρήστης στην οθόνη του και προκύπτει απ'τη μαθηματική έκφραση B-A (κανάλι Reverse time) που αφαιρεί το πρώτο της παιδί απ'το δεύτερό της. Το δεύτερο παιδί είναι η συντόμευση του καναλιού Total Time που και αυτό είναι μία μαθηματική έκφραση με τύπο :

$$(A+1)*B.$$

Εδώ τα δύο παιδιά είναι ο αριθμός των σφαιρών που έχει επιλέξει ο χρήστης στη δεύτερη σκηνή και ο χρόνος που αντιστοιχεί για μία σφαίρα που προκύπτει απ'το επίπεδο δυσκολίας που επίσης έχει επιλέξει ο χρήστης. Χρησιμοποιούμε (A+1), όπου A είναι ο αριθμός των σφαιρών, γιατί η πρώτη σφαίρα δεν εκτοξεύεται αμέσως μόλις ξεκινήσει η σκηνή αλλά σε τόσα δευτερόλεπτα όσα αντιστοιχούν σε μία σφαίρα, ώστε ο χρήστης να προλάβει να πατήσει τα πλήκτρα σύνδεσης των δύο Wii Remote την πρώτη φορά εκτέλεσης και να συνηθίσει την κίνηση του υποδοχέα όταν μετακινεί τη sensor bar στα χέρια του. Επομένως, ο χρήστης χρειάζεται (A+1) χρονικά διαστήματα για να έχει και χρόνο, αφού εκτοξευθεί και η τελευταία σφαίρα, να την βάλει πάνω στο στόχο, πριν εμφανιστούν τα τελικά αποτελέσματα.

Το πρώτο παιδί του καναλιού Reverse time είναι το κανάλι Limit Relative Value. Αυτό προσθέτει στην αρχική του τιμή (πρώτο του παιδί) την τιμή αύξησης (τρίτο παιδί). Έπειτα το κανάλι Limit Relative Value λαμβάνει την τιμή αυτού του αθροίσματος. Στο επόμενο πλαίσιο (frame) που ξανακαλείται το κανάλι, προσθέτει στο άθροισμα την τιμή της αύξησης και λαμβάνει την τιμή του καινούργιου αθροίσματος. Αυτή η διαδικασία συνεχίζεται σε κάθε frame, μέχρι το άθροισμα να φτάσει την τελική τιμή (δεύτερο παιδί). Στη συγκεκριμένη περίπτωση, η αρχική τιμή είναι το 0, η τελική τιμή είναι η τιμή του συνολικού χρόνου (κανάλι Total Time) που είδαμε παραπάνω πως υπολογίζεται και η

τιμή αύξησης είναι η μαθηματική έκφραση : $\frac{A}{25}$

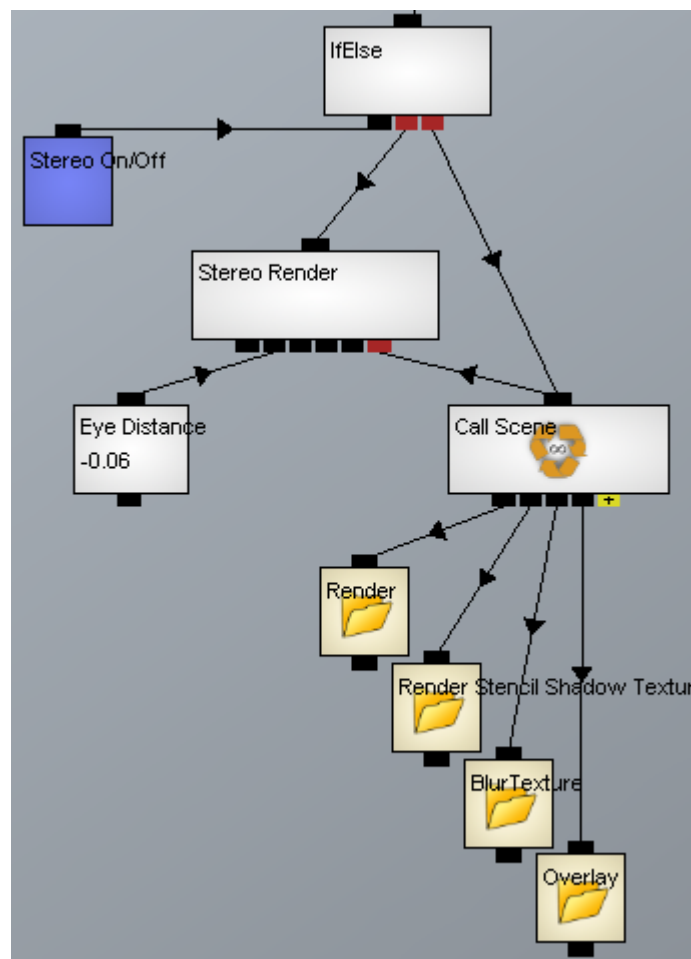
όπου A είναι η μεταβλητή TickCount. Η μεταβλητή TickCount μετράει το χρόνο που πέρασε απ'το τελευταίο κάλεσμα της. Εξ ορισμού της, δίνεται ως μία τιμή σχετική με τα

25 frames/sec. Δηλαδή όταν η TickCount έχει τιμή 1, ο χρόνος μεταξύ δύο διαδοχικών καλεσμάτων του καναλιού είναι $\frac{1}{25} = 0.04$ δευτερόλεπτα. Επόμεως, όταν έχουμε x TickCount, η χρονική απόσταση σε sec είναι $\frac{x}{25}$. Άρα το κανάλι με τη μαθηματική έκφραση $\frac{A}{25}$ υπολογίζει το χρόνο σε sec και με τη σειρά του το κανάλι Limit Relative Value υπολογίζει το χρόνο μέχρι την τιμή Total Time. Βλέπουμε τη χρησιμότητα της μεταβλητής TickCount αφού πολλαπλασιάζοντας μία οποιαδήποτε τιμή με την TickCount, κάνουμε την τιμή αυτή να είναι ανεξάρτητη απ'το ρυθμό των πλαισίων (frame rate) και άρα και απ'την υπολογιστική δύναμη του ηλεκτρονικού υπολογιστή που εκτελεί την εφαρμογή.

3.2.2) Στερεοσκοπική Προβολή

Ο χρήστης μπορεί να επιλέξει αν θέλει η προβολή της κύριας σκηνής να γίνει στερεοσκοπικά. Στην εφαρμογή που έγινε στο εργαστήριο χρησιμοποιήθηκε η οριζόντια τεχνική διαχωρισμού. Σε αυτήν, η ίδια σκηνή σε κάθε frame, γίνεται rendered δύο φορές. Οι δύο σκηνές που στέλνονται στο ζευγάρι των προβολέων είναι σα να έχουν δημιουργηθεί από δύο κάμερες που απέχουν τόση απόσταση όση είναι η απόσταση των οφθαλμών, περίπου 6 cm, έτσι ώστε με τη μέθοδο της passive stereo (παράγραφος 1.2) η σκηνή να γίνει αντιληπτή απ'το χρήστη ως τρισδιάστατη. Υπενθυμίζεται ότι η μία εικόνα πολώνεται καθέτως ενώ η άλλη οριζοντίως και ο χρήστης φοράει ειδικά γυαλιά με έναν φακό πολωμένο καθέτως και τον άλλο πολωμένο οριζοντίως, με αποτέλεσμα ο κάθε οφθαλμός να βλέπει μόνο τη μία εικόνα απ' τις δύο που προβάλλονται στην οθόνη.

Στην εικόνα 3.31 φαίνεται ο κώδικας που δημιουργεί τις δύο διαφορετικές εικόνες. Είναι μια If – Else λειτουργία. Το If ελέγχει αν ο χρήστης στη σκηνή επιλογών έχει επιλέξει το checkbox για στερεοσκοπική προβολή. Αν ναι, τότε η μεταβλητή Stereo On/Off έχει την τιμή 1 (true) οπότε εκτελείται το κανάλι Stereo Render το οποίο αναπαραγάγει τη σκηνή που δημιουργείται απ'το τελευταίο παιδί του δύο φορές με τέτοια οριζόντια μετατόπιση όση ορίζεται στο δεύτερο παιδί του που είναι η μεταβλητή Eye Distance. Αν ο χρήστης δεν έχει επιλέξει στερεοσκοπική προβολή, τότε η μεταβλητή Stereo On/Off έχει την τιμή 0 (false) και έτσι το κανάλι IfElse εκτελεί κατευθείαν το κανάλι Call Scene που κάνει render τη σκηνή μία μόνο φορά αυτή τη περίπτωση.

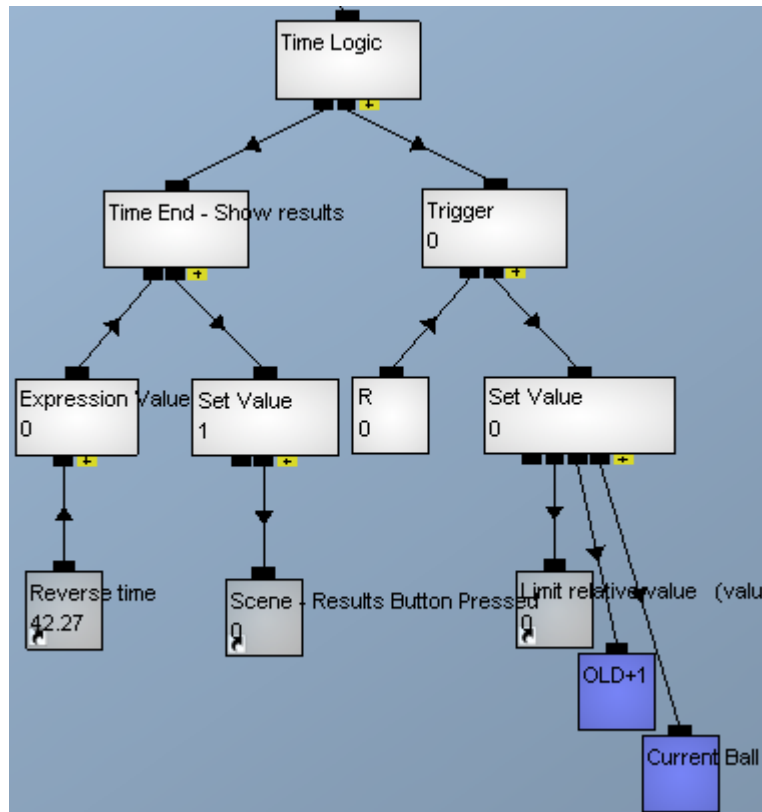


Εικόνα 3.31

Κώδικας δημιουργίας δύο εικόνων για στερεοσκοπική προβολή

3.2.3) Λογική Εφαρμογής

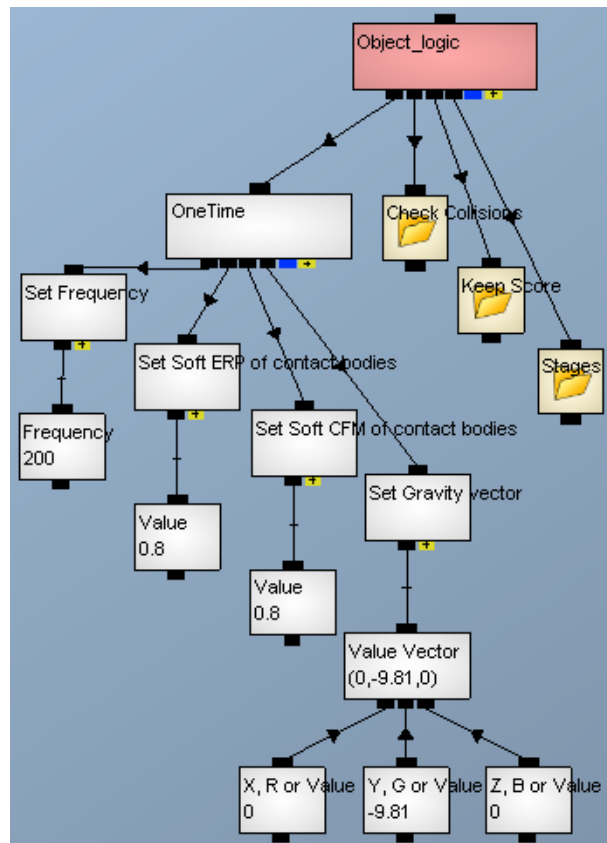
Θα εξετάσουμε τη λογική της κύριας σκηνής της εφαρμογής. Ο σχετικός κώδικας, όσον αφορά το χρόνο, φαίνεται στην εικόνα 3.32. Το πρώτο παιδί ελέγχει αν η αντίστροφη μέτρηση γίνει 0, τότε θέτει την τιμή 1 στην μεταβλητή της μηχανής καταστάσεων Scene – Results Button Pressed, με αποτέλεσμα να προωθείται η ροή προγράμματος στην επόμενη σκηνή που είναι η παρουσίαση των αποτελεσμάτων. Το δεύτερο παιδί ελέγχει αν πατηθεί το πλήκτρο R ώστε να ξαναρχίσει η αντίστροφη χρονική μέτρηση απ' την αρχική της τιμή και ο αριθμός των σφαιρών να ξαναγίνει ο αρχικός, ώστε η διαδικασία εκτόξευσης των σφαιρών να γίνει απ' την αρχή.



Εικόνα 3.32

Κώδικας ελέγχου του χρόνου

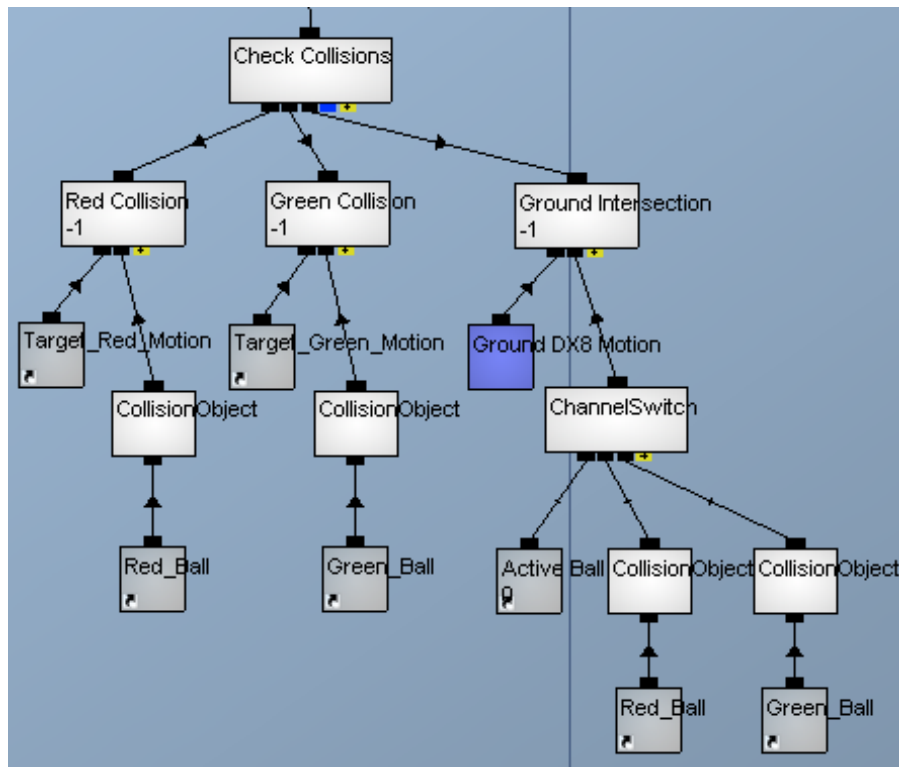
Στην εικόνα 3.33 βλέπουμε τον κώδικα για τη γενική λογική. Το πρώτο παιδί είναι ένα κανάλι One Time το οποίο σημαίνει ότι όλα τα παιδιά του θα εκτελεστούν μία μόνο φορά στο πρόγραμμα και όχι σε κάθε frame όπως τα υπόλοιπα κανάλια. Έτσι θέτουμε το ρυθμό ενημέρωσης (update rate) της φυσικής προσομοίωσης στα 200 Hz. Το επόμενο παιδί θέτει την τιμή 0.8 στη παράμετρο ERP (Error Reduction Parameter) που ορίζει πόσο προσπαθεί η μηχανή φυσικής να διορθώνει τις θέσεις των φυσικών σωμάτων όταν αυτά συγκρούονται ή όταν ένα σώμα διεισδύει μέσα σε ένα άλλο.. Ομοίως το τρίτο παιδί δίνει την τιμή 0.8 στη παράμετρο CFM (Constraint Force Mixing) που ορίζει πόσο δυνατές είναι οι δυνάμεις που απομακρύνουν ένα σώμα όταν ο όγκος του επικαλύπτεται με τον όγκο κάποιου άλλου σώματος. Το τελευταίο παιδί του καναλιού One Time καθορίζει το διάλυμα της βαρύτητας.



Εικόνα 3.33

Κώδικας λογικής της κύριας σκηνής

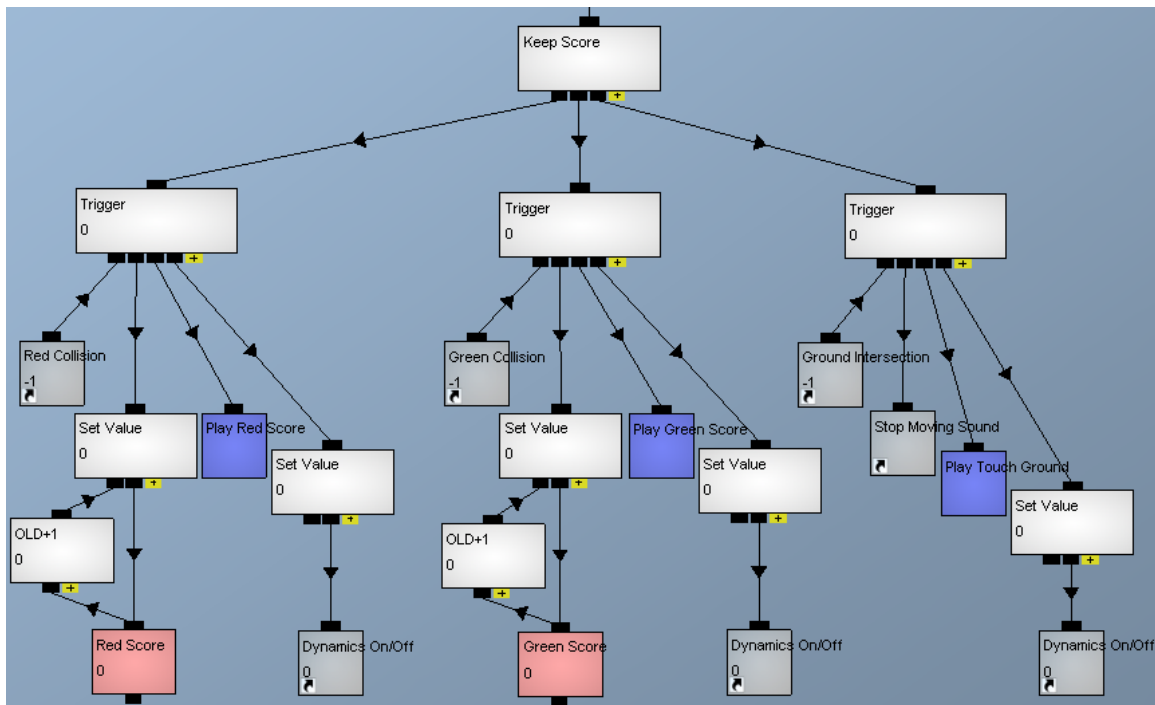
Στην εικόνα 3.34 βλέπουμε πιο λεπτομερώς το φάκελο Check Collisions της εικόνας 3.33. Σε αυτό το κομμάτι, γίνεται ο έλεγχος των συγκρούσεων μεταξύ των αντικειμένων. Το κανάλι CollisionBoxCheck, όπως είναι παραδείγματος χάρη το κανάλι Red Collision, δημιουργεί ένα αόρατο κύβο με σχετικές συντεταγμένες ως προς τη μήτρα κίνησης που έχει ως πρώτο παιδί, από (-0.5, -0.5, -0.5) μέχρι (0.5, 0.5, 0.5). Έπειτα μετασχηματίζει τον κύβο αυτό σύμφωνα με την μήτρα κίνησης ώστε να δημιουργήσει τον τελικό κύβο συγκρούσεων που να ταυτίζεται σχεδόν με το αντικείμενο της μήτρας. Το δεύτερο παιδί είναι ένα κανάλι CollisionObject το οποίο δημιουργεί ένα αντικείμενο συγκρούσεων από τη γεωμετρία του σώματος που έχει ως παιδί. Μπορούμε να συνδέσουμε πολλά CollisionObjects στο CollisionBoxCheck για να ελέγχουμε τις συγκρούσεις πολλών αντικειμένων με τον τελικό κύβο. Έτσι το κανάλι CollisionBoxCheck έχει την τιμή -1 όταν δεν ανιχνεύει συγκρούσεις μεταξύ του κύβου και των αντικειμένων συγκρούσεων που έχει ως παιδιά του και λαμβάνει την τιμή του αύξοντος αριθμού του παιδιού αντικειμένου (αρχίζοντας απ'την τιμή 1 για το πρώτο CollisionObject) όταν ανιχνεύει συγκρούσεις μεταξύ αυτού του παιδιού και του κύβου. Άρα το κανάλι Red Collision ανιχνεύει τις συγκρούσεις μεταξύ του κόκκινου στόχου και της κόκκινης σφαίρας. Ομοίως το κανάλι Green Collision για το πράσινο στόχο και σφαίρα. Για τις συγκρούσεις με το έδαφος, χρησιμοποιούμε το κανάλι ChannelSwitch ώστε ο έλεγχος να γίνεται μεταξύ του εδάφους και της ενεργής, κάθε φορά, σφαίρας. Το αποτέλεσμα είναι ότι δεν ανιχνεύεται η σύγκρουση μεταξύ του εδάφους και της ανενεργής σφαίρας καθώς αυτή δε μας ενδιαφέρει.



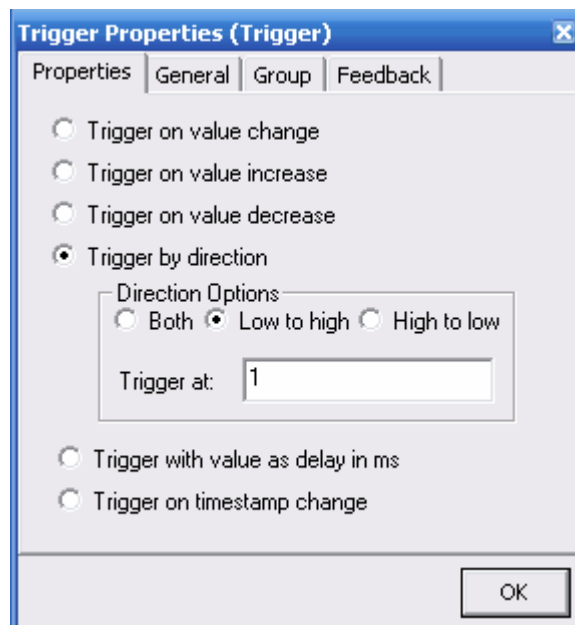
Εικόνα 3.34

Κώδικας ελέγχου συγκρούσεων

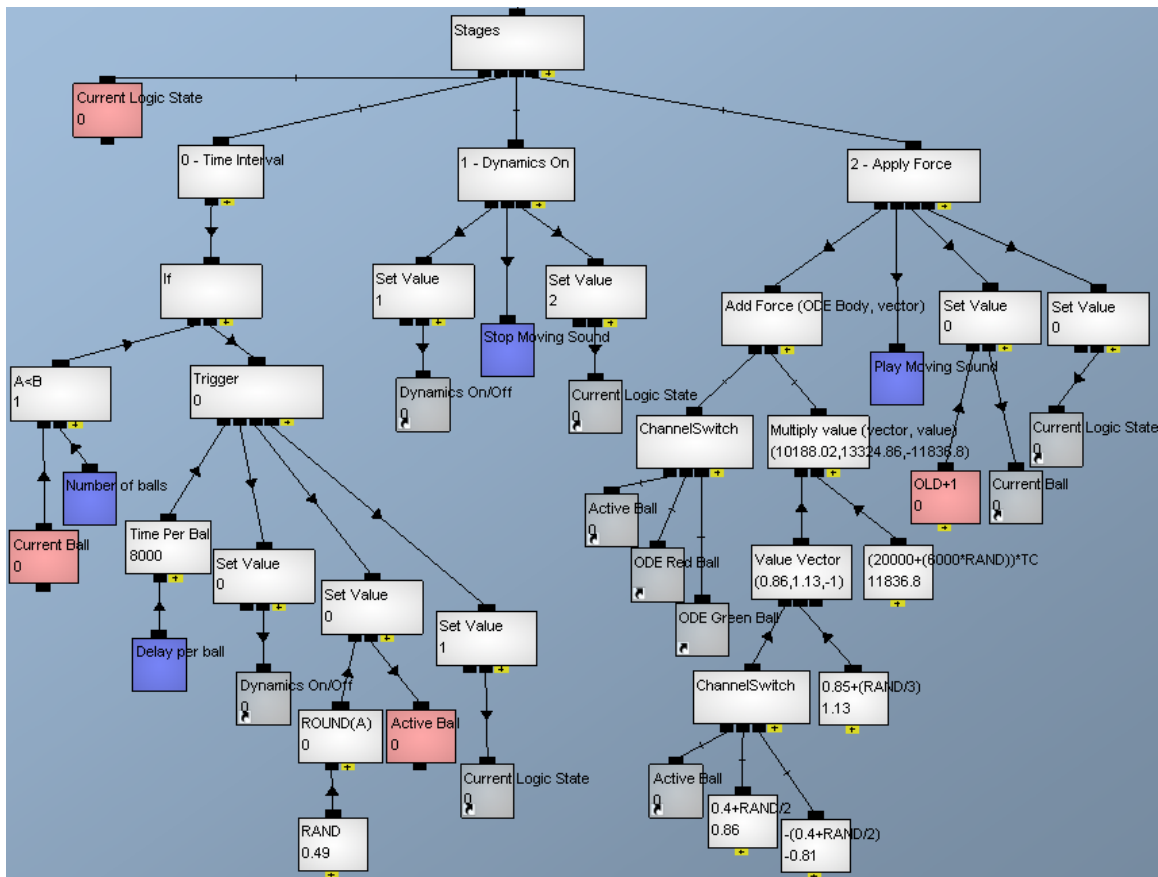
Ας δούμε πιο διεξοδικά το φάκελο Keep Score της εικόνας 3.33. Βλέπουμε στην εικόνα 3.35 τα κανάλια Trigger με τα παιδιά τους. Το Trigger ενεργοποιείται από την μεταβλητή που έχει ως πρώτο παιδί και τότε εκτελεί με τη σειρά τα υπόλοιπα παιδιά του. Οι συνθήκες ενεργοποίησης είναι ποικίλες και φαίνονται στις ιδιότητες του καναλιού Trigger (εικόνα 3.36). Στη δική μας περίπτωση, για παράδειγμα στο πρώτο παιδί που ενημερώνει το σκορ για τις κόκκινες σφαίρες, το κανάλι Trigger ενεργοποιείται όταν η μεταβλητή ελέγχου, που είναι η τιμή του καναλιού Red Collision, πάρει την τιμή 1 και προηγουμένως είχε χαμηλότερη τιμή. Αυτό συμβαίνει όταν έχουμε σύγκρουση μεταξύ του κόκκινου στόχου και της κόκκινης σφαίρας, οπότε το κανάλι Red Collision από -1 γίνεται 1. Άρα, τότε το Trigger εκτελεί το δεύτερο παιδί του θέτει στη μεταβλητή Red Score την τιμή της μαθηματικής έκφρασης : $OLD+1$, δηλαδή την αυξάνει κατά ένα. Έπειτα, εκτελείται το κανάλι Play Red Score που εκκινεί την αναπαραγωγή ενός ήχου. Τέλος, η μεταβλητή Dynamics On/Off γίνεται 0, οπότε απενεργοποιείται το σύστημα φυσικής και όλα τα φυσικά σώματα γυρνάνε στις αρχικές θέσεις τους για να είναι έτοιμα για την επόμενη φάση (οι φάσεις εξηγούνται στην εικόνα 3.37). Τα ίδια ισχύουν και για το μεσαίο παιδί που ενημερώνει το σκορ για τις πράσινες σφαίρες. Το τρίτο παιδί εκτελείται όταν έχουμε σύγκρουση μεταξύ της ενεργής σφαίρας και του εδάφους. Τότε, σταματάει ο ήχος που κάνει η σφαίρα όταν κινείται στον αέρα (κανάλι Stop Moving Sound) και ξεκινά η αναπαραγωγή του ήχου που μας ενημερώνει ότι η σφαίρα ακούμπησε το έδαφος (κανάλι Play Touch Ground). Εδώ δε μεταβάλλεται ούτε το κόκκινο ούτε το πράσινο σκορ, απλά πάλι απενεργοποιείται το σύστημα φυσικής.



Εικόνα 3.35
Κώδικας ενημέρωσης του σκορ



Εικόνα 3.36
Ιδιότητες καναλιού Trigger



Εικόνα 3.37

Κώδικας των σταδίων της λογικής της κύριας σκηνής

Στην εικόνα 3.37 βλέπουμε την κεντρική λογική (φάκελος Stages στην εικόνα 3.33) της κύριας σκηνής. Το κανάλι Stages είναι ένα κανάλι ChannelSwitch που εκτελεί τον κώδικα κάποιου απ'τα παιδιά του αναλόγως την τιμή της μεταβλητής Current Logic State. Όταν αυτή είναι 0, τότε εκτελείται το παιδί 0 – Time Interval. Αυτό ελέγχει με το κανάλι If αν ο αριθμός της σφαίρας που ελέγχεται τώρα (κανάλι Current Ball) είναι μικρότερος απ'τον συνολικό αριθμό σφαιρών (κανάλι Number of balls) που έχει επιλέξει αρχικά ο χρήστης. Αν δεν είναι μικρότερος, τότε αυτό σημαίνει ότι έχει εκτοξευθεί και η τελευταία σφαίρα οπότε δε γίνεται κάτι άλλο παρά μόνο ο έλεγχος πότε θα τελειώσει ο χρόνος (εικόνα 3.32) ώστε να παρουσιαστούν τα αποτελέσματα. Αν είναι μικρότερος, τότε χρησιμοποιείται ένα κανάλι Trigger που ενεργοποιείται κάθε τόσα δευτερόλεπτα όσα έχει επιλέξει ο χρήστης με το επίπεδο δυσκολίας (κανάλι Delay per ball). Όταν ενεργοποιηθεί το Trigger, καλεί τα επόμενα τρία παιδιά του. Δηλαδή, πρώτον, απενεργοποιεί το σύστημα φυσικής ώστε οι σφαίρες να γυρίσουν στις αρχικές τους θέσεις. Έπειτα, γίνεται η τυχαία επιλογή ποια από την κόκκινη ή την πράσινη σφαίρα θα είναι η ενεργή σφαίρα για αυτά τα δευτερόλεπτα. Η μαθηματική έκφραση RAND δημιουργεί έναν τυχαίο αριθμό, με ακρίβεια πέμπτου δεκαδικού, που να ανήκει στο σύνολο [0,1]. Ύστερα, η μαθηματική έκφραση ROUND(A) στρογγυλοποιεί τον τυχαίο αριθμό στον πλησιέστερο ακέραιο, άρα έχουμε 50% πιθανότητα να γίνει 0 και 50% να γίνει 1. Αυτή η ακέραια τιμή εκχωρείται στην μεταβλητή Active Ball που ορίζει ποιά απ'τις δύο σφαίρες είναι η ενεργή (με 0 είναι η κόκκινη και με 1 η πράσινη). Το τελευταίο παιδί του Trigger δίνει στην μεταβλητή Current Logic State την τιμή 1, οπότε το κανάλι Stages αρχίζει και εκτελεί το παιδί του 1 – Dynamics On. Σε αυτό,

ενεργοποιείται το σύστημα φυσικής, σταματάει η αναπαραγωγή του ήχου της κινούμενης ενεργούς (όποια και να είναι αυτή) σφαίρας (κανάλι Stop Moving Sound) και η μεταβλητή Current Logic State γίνεται ίση με 2, δίνοντας έτσι εντολή στο Stages να εκτελέσει το τελευταίο του παιδί.

Στο τελευταίο παιδί, κανάλι 2 – Apply Force, ασκείται δύναμη τυχαίου μέτρου και διεύθυνσης στην ενεργή σφαίρα ικανή να την εκτοξεύσει. Το κανάλι Add Force ασκεί στο σώμα που προσδιορίζεται στο πρώτο του παιδί την δύναμη που ορίζεται στο δεύτερό του. Η ενεργή σφαίρα επιλέγεται με βάση την τιμή της μεταβλητής Active ball, όπως εξηγήθηκε προηγουμένως. Το διάνυσμα της δύναμης προκύπτει απ'τον πολλαπλασιασμό (κανάλι Multiply value) του διανύσματος διεύθυνσης επί μία τιμή που είναι το μέτρο της δύναμης. Το διάνυσμα διεύθυνσης, όπως βλέπουμε προκύπτει από μαθηματικές εκφράσεις που περιέχουν τη παράμετρο RAND, ώστε σε κάθε εκτόξευση να είναι διαφορετικό. Παραδείγματος χάρη η τιμή y υπολογίζεται απ'τη μαθηματική έκφραση :

$$0.85 + (\text{RAND}/3)$$

άρα ανήκει στο σύνολο $[0.85, 1.18]$. Η μεταβλητή x παίρνει είτε θετική τιμή όταν ενεργή σφαίρα είναι η κόκκινη καθώς η αρχική της θέση βρίσκεται στα αρνητικά x είτε αρνητική τιμή όταν ενεργή είναι η πράσινη αφού αυτή ξεκινάει από θετικά x . Ομοίως και η μαθηματική έκφραση του μέτρου της δύναμης περιέχει τη παράμετρο RAND για να μας δίνει τυχαίες τιμές. Οι τιμές των άκρων των συνόλων που προκύπτουν απ'τη παράμετρο RAND έχουν υπολογιστεί έτσι ώστε να μην ασκείται δύναμη στην ενεργή σφαίρα τέτοια ώστε να τη βγάλει εκτός πεδίου όρασης του χρήστη.

Αφού εφαρμοστεί η δύναμη, έπειτα το κανάλι Play Moving Sound ξεκινά την αναπαραγωγή του ήχου που κάνει η ενεργή σφαίρα όταν κινείται. Ύστερα, η τιμή της μεταβλητής Current Ball αυξάνεται κατά μία μονάδα για να χρησιμοποιηθεί απ'το κανάλι 0 – Time Interval για τον έλεγχο αν οι σφαίρες που έχουν εκτοξευθεί μέχρι τη συγκεκριμένη χρονική στιγμή είναι λιγότερες απ'τον συνολικό αριθμό των σφαιρών. Τέλος, η μεταβλητή ελέγχου της λογικής, Current Logic State, λαμβάνει την τιμή 0 και επομένως η ροή προγράμματος ξαναγυρίζει στο παιδί 0 – Time Interval, κλείνοντας έτσι μια εκτέλεση του βρόχου.

Δε χρησιμοποιήσαμε μια μηχανή πεπερασμένων καταστάσεων (Finite State Machine) για να υλοποιήσουμε τα στάδια της κεντρικής σκηνής όπως κάναμε για τη σύνδεση των σκηνών της εφαρμογής (παράγραφος 3.2.1), καθώς αυτή μπορεί να λειτουργήσει μόνο με προκαθορισμένα χρονικά διαστήματα. Όμως, ο χρήστης επιλέγει το επίπεδο δυσκολίας, οπότε σε κάθε εκκίνηση της εφαρμογής μπορεί να είναι διαφορετικό το χρονικό διάστημα που μεσολαβεί μεταξύ δύο εκτοξεύσεων σφαιρών. Έτσι, χρησιμοποιήσαμε ένα κανάλι ChannelSwitch που αλλάζει κατάσταση εξαιτίας του καναλιού Trigger που ενεργοποιείται σε κάθε χρονικό διάστημα που έχει επιλέξει ο χρήστης.

3.3) Ανίχνευση Ράβδου

Σε αυτή τη παράγραφο θα δείξουμε πως επιτυγχάνεται η συσχέτιση μεταξύ του αντικειμένου υποδοχέα του εικονικού κόσμου και της ράβδου με τα LED (ή της ηλεκτρονικής πλακέτας), δηλαδή πως καταφέρνει ο χρήστης κουνώντας τη ράβδο στα χέρια του να ελέγχει την κίνηση και περιστροφή του υποδοχέα. Θυμίζουμε ότι τα δύο Wii Remote είναι είτε παράλληλα μεταξύ τους είτε σχηματίζουν γωνία με το επίπεδο της οθόνης που παρατηρεί ο χρήστης, απέχουν μεταξύ τους απόσταση b και βρίσκονται στο ίδιο ύψος με αποτέλεσμα να μην υπάρχει κάθετη ανομοιότητα αλλά μόνο οριζόντια. Ας εξετάσουμε πρώτα την ηλεκτρονική πλακέτα που κατασκευάσαμε και λειτουργεί όπως η ράβδος με τα LED (εικόνα 1.8).

3.3.1) Ηλεκτρονική Πλακέτα

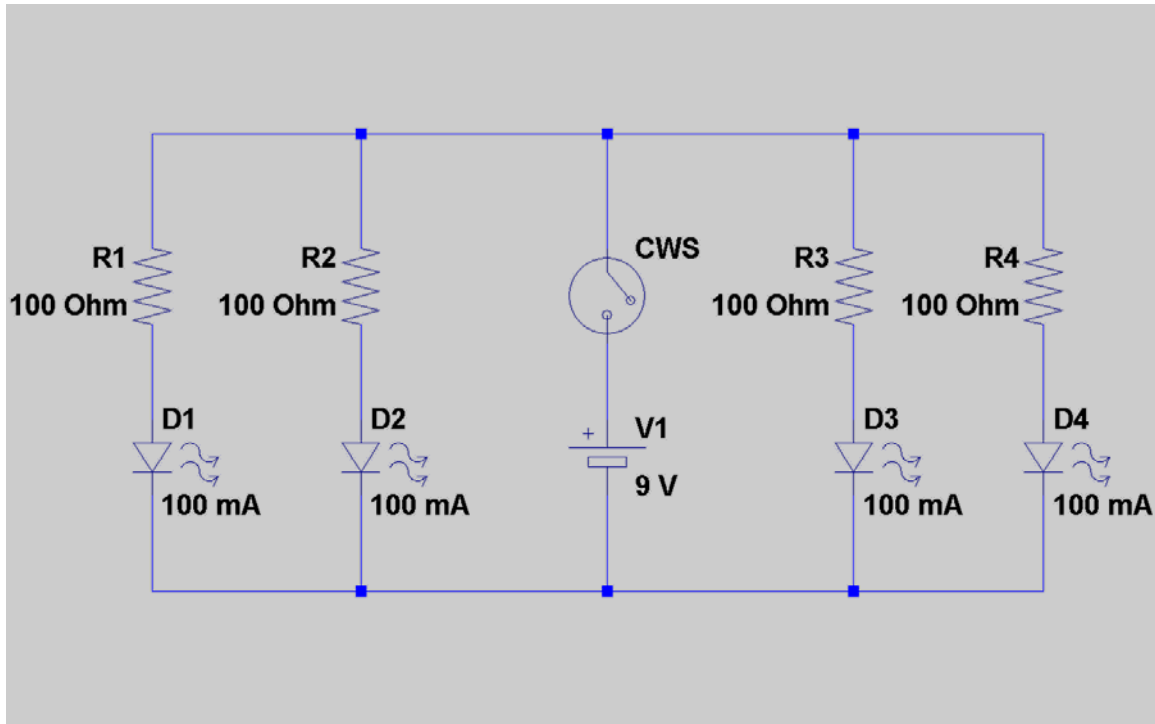
Ο χρήστης για να μεταβάλλει την κίνηση του αντικειμένου που ελέγχει στον εικονικό κόσμο, κρατάει με τα χέρια του και μετακινεί μια ράβδο με LED (sensor bar). Η κίνηση των LED στο πραγματικό χώρο ανιχνεύεται απ'την κάμερα του Wii Remote και οι πληροφορίες της θέσης των προβολών των LED στο επίπεδο προβολής διοχετεύονται στο λογισμικό πρόγραμμα Quest3D για να εξομοιωθεί η κίνηση του ελεγχόμενου αντικειμένου στο εικονικό περιβάλλον.

Στο εργαστήριο είχαμε στη διάθεσή μας μια sensor bar της εταιρείας NYKO. Επειδή όμως δε γνωρίζαμε τα τεχνικά χαρακτηριστικά των LED της και προσπαθώντας να επιτύχουμε όσο το δυνατόν καλύτερη ανίχνευση απ'τις κάμερες των Wii Remote, φτιάξαμε μια πλακέτα με LED γνωστών χαρακτηριστικών. Πρόκειται [22] για πλαστικές διόδους με κρύσταλλο Γαλλίου-Αργιλίου-Αρσενικού (GaAlAs) εκπομπής υπερύθρου φωτός, τύπου OP293 της εταιρείας OPTEK. Το φως έχει μήκος κύματος 890nm και εκπέμπεται σε γωνία 60° . Το μέγιστο ρεύμα λειτουργίας της φωτοδιόδου είναι 100mA και η εξαγόμενη ισχύς είναι περίπου 18mW. Χρησιμοποιήσαμε σε παράλληλη σύνδεση δύο φωτοδιόδους στο αριστερό και άλλες δύο στο δεξί άκρο της πλακέτας. Τα δύο LED στο κάθε άκρο δε τοποθετήθηκαν να εκπέμπουν υπό την ίδια γωνία, αλλά το ένα είναι στραμμένο προς το εσωτερικό της πλακέτας ενώ το άλλο προς το εξωτερικό, αυξάνοντας έτσι τη συνολική γωνία εκπομπής του κάθε άκρου της πλακέτας. Η τροφοδοσία τους, όταν ο διακόπτης CSW είναι κλειστός, γίνεται από μία μπαταρία (συνεχές ρεύμα) τάσης 9V.

Όλοι οι κλάδοι είναι παράλληλοι μεταξύ τους, οπότε όλοι έχουν στα άκρα τους την τάση της πηγής δηλαδή 9V. Με ρεύμα 100mA, στο έντυπο τεχνικών δεδομένων (data sheet) της φωτοδιόδου, που βρίσκεται στο παράρτημα, βλέπουμε ότι η μέγιστη πτώση τάσης είναι 2V. Οπότε ως μια τυπική πτώση τάσης μπορούμε να θεωρήσουμε το 50% της μέγιστης τιμής, άρα η τάση στα άκρα της κάθε φωτοδιόδου είναι 1V. Αν V_s είναι η τάση στα άκρα της πηγής, V_f η πτώση στα άκρα της φωτοδιόδου, I_f η ένταση του ρεύματος λειτουργίας της φωτοδιόδου και R η αντίσταση που βάζουμε εν σειρά προς τη φωτοδίοδο και της οποίας ψάχνουμε την τιμή, τότε θα ισχύει :

$$V_s = V_f + I_f R \Rightarrow R = \frac{V_s - V_f}{I_f} = \frac{9V - 1V}{100mA} = 80\Omega$$

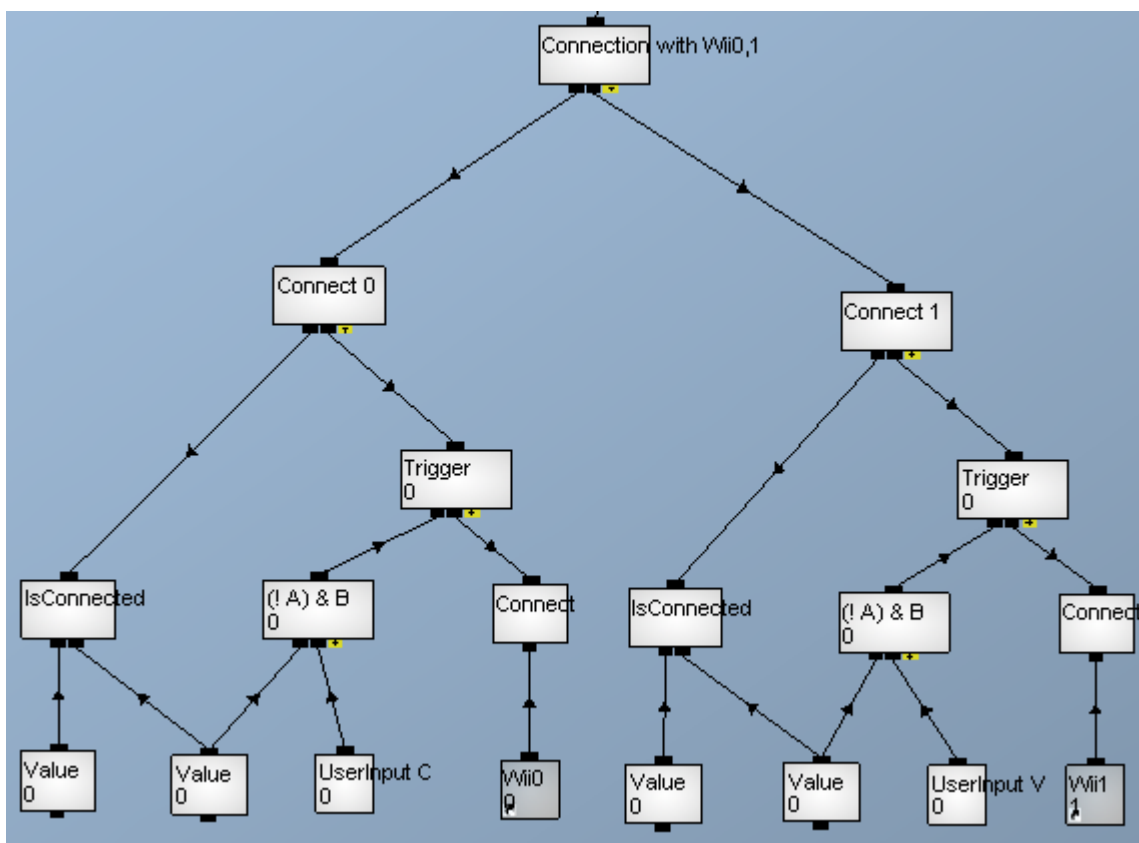
Η κοντινότερη και λίγο μεγαλύτερη αντίσταση που βρήκαμε και τελικά χρησιμοποιήσαμε σε κάθε κλάδο ήταν 100 Ohm. Στην εικόνα 3.38 βλέπουμε το ηλεκτρονικό διάγραμμα της πλακέτας το οποίο δημιουργήσαμε στο ελεύθερο λογισμικό LTspice IV. Το έντυπο τεχνικών δεδομένων των φωτοдиодων βρίσκεται στο παράρτημα.



Εικόνα 3.38
Κύκλωμα ηλεκτρονικής πλακέτας

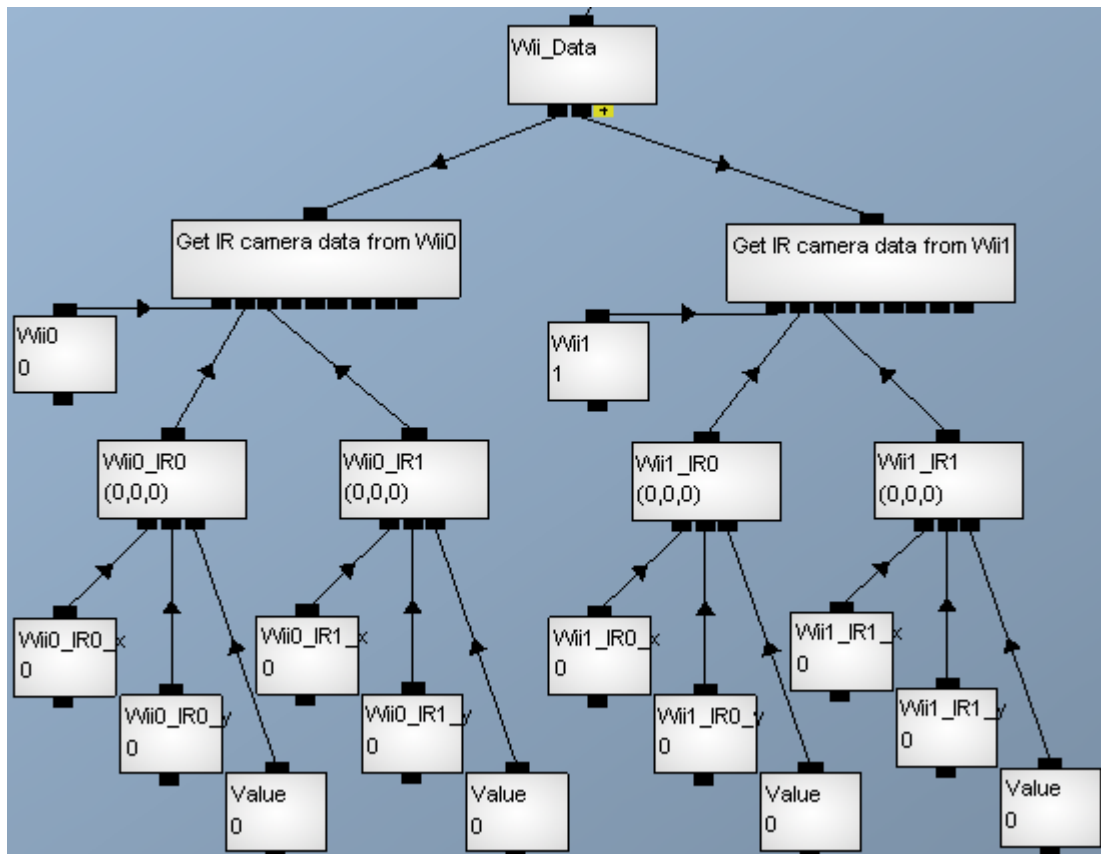
3.3.2) Κίνηση στους άξονες x'x, y'y, z'z

Επισημαίνουμε ότι οι άξονες x'x, y'y, z'z είναι του συστήματος συντεταγμένων του εικονικού κόσμου. Αρχικά, εκτελείται ο κώδικας της εικόνας 3.39 για να συνδεθούν τα δύο Wii Remote με το πρόγραμμα Quest3D και να είναι έτοιμα να του δίνουν δεδομένα. Το πρώτο Wii Remote που συνδέσαμε στο λειτουργικό σύστημα του υπολογιστή μέσω της ασύρματης τεχνολογίας Bluetooth ονομάζεται Wii0 και το δεύτερο Wii1. Ας δούμε το αριστερό κομμάτι με το οποίο γίνεται η σύνδεση του Wii0 και τα ίδια ισχύουν και στο δεξί για το Wii1. Πρώτα, με το κανάλι IsConnected ελέγχεται αν το Wii0 έχει συνδεθεί σωστά με το λειτουργικό σύστημα. Αν είναι συνδεδεμένο, τότε το δεύτερο παιδί του γίνεται 0. Έτσι, η μαθηματική έκφραση $(!A) \& B$, που μπορεί να διαβαστεί και ως $(NOT A) \text{ AND } B$, γίνεται αληθής όταν το Wii Remote είναι συνδεδεμένο και ο χρήστης πατήσει ένα συγκεκριμένο πλήκτρο. Μόλις η μαθηματική έκφραση γίνει αληθής, το κανάλι της παίρνει την τιμή 1 οπότε και ενεργοποιεί το γονικό της κανάλι Trigger που εκείνη τη στιγμή ολοκληρώνει τη σύνδεση (κανάλι Connect) του πρώτου Wii Remote με το λογισμικό Quest3D.



Εικόνα 3.39
Κώδικας σύνδεσης των δύο Wii Remote

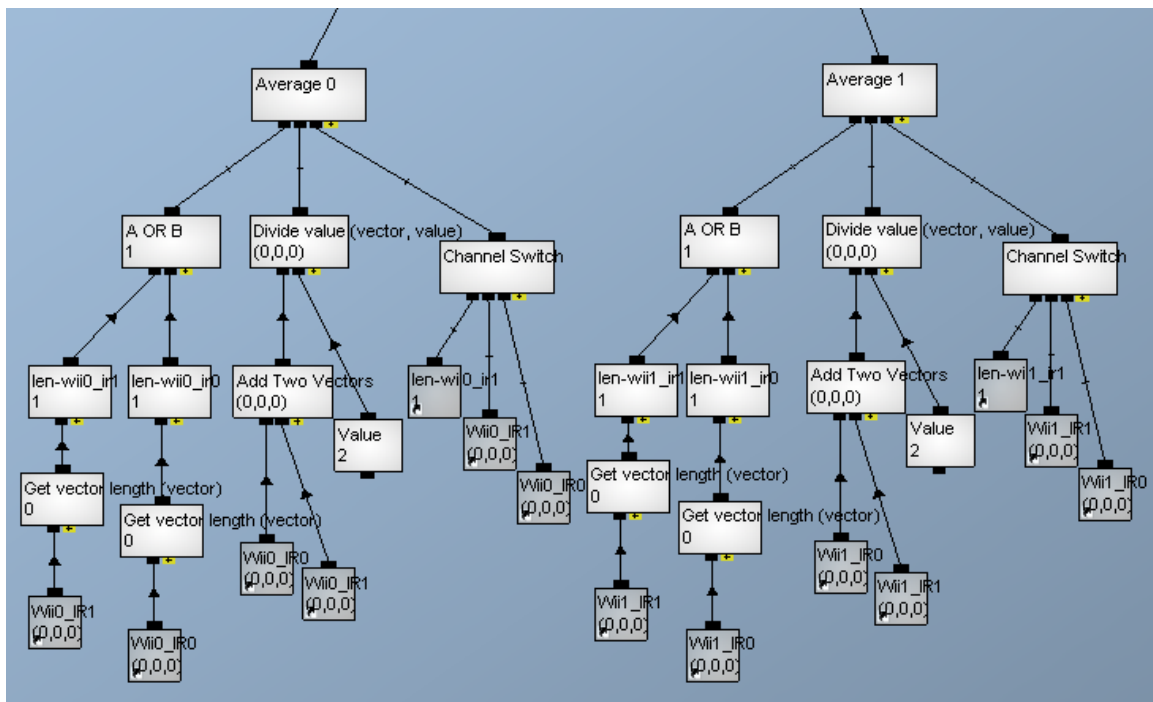
Αφού συνδεθούν επιτυχώς τα δύο Wii Remote, μπορούμε τώρα να αρχίσουμε να λαμβάνουμε δεδομένα απ'αυτά. Αυτό επιτυγχάνουν τα δύο κανάλια Get IR camera data της εικόνας 3.40. Το πρώτο παιδί είναι ο δείκτης του Wii Remote, το 0 για το Wii0 και το 1 για το Wii1. Η υπέρυθη CCD κάμερα του Wii Remote μπορεί να ανιχνεύσει μέχρι τέσσερεις στόχους, όμως το κανάλι Get IR camera data μας δίνει πληροφορίες για μόνο μέχρι δύο στόχους. Έτσι, όταν ο χρήστης κρατάει τη ράβδο με τις συστοιχίες από LED στις άκρες της, οι προβολές των LED της μίας άκρης (IR0) πάνω στο επίπεδο xy του συστήματος συντεταγμένων της κάμερας καταγράφονται στις τιμές Wii0_IR0_x και Wii0_IR0_y του διανύσματος Wii0_IR0 (για το Wii0). Όμως δεν έχουμε πληροφορία για την απόσταση των LED απ'τη κάμερα, δηλαδή την τιμή Z (τα κεφαλαία X,Y,Z είναι συντεταγμένες στο σύστημα συντεταγμένων του πραγματικού κόσμου). Γι'αυτό και χρησιμοποιούμε δύο Wii Remote ώστε μέσω της στερεοσκοπίας να μπορέσουμε να υπολογίσουμε και το Z των LED. Η άλλη άκρη με τα LED έχει τον δείκτη IR1. Τελικά το κανάλι Get IR camera data του Wii0 δίνει δύο διανύσματα Wii0_IR0 (Wii0_IR0_x, Wii0_IR0_y, 0) και Wii0_IR1 (Wii0_IR1_x, Wii0_IR1_y, 0). Τα ίδια ισχύουν και για το δεύτερο Wii Remote, Wii1, όπου τα διανύσματα και οι συντεταγμένες έχουν δείκτη Wii1. Να τονίσουμε ότι ως IR0 ονοματίζει η κάμερα τα πρώτα LED που ανιχνεύει και ως IR1 τα δεύτερα.



Εικόνα 3.40

Κώδικας καναλιών επικοινωνίας με τα δύο Wii Remote

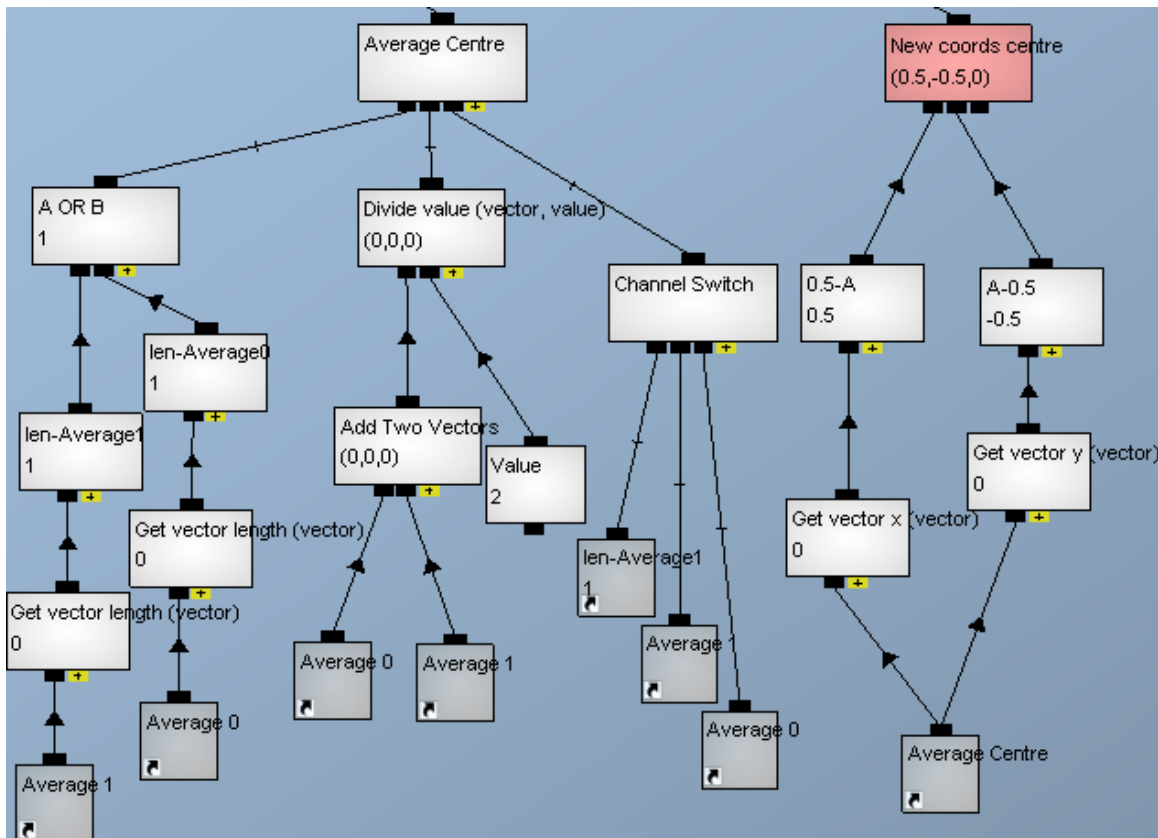
Για να επιτύχουμε τη σωστή κίνηση του υποδοχέα κατά τους άξονες x'x και y'y (του εικονικού κόσμου) χρησιμοποιούμε τα δύο κανάλια Average 0 και Average 1 της εικόνας 3.41 και το κανάλι Average Centre της εικόνας 3.42 που προκύπτει απ'τα Average 0 και Average 1. Το Average 0 είναι το θεωρητικό κέντρο της ράβδου όπως αυτό προκύπτει απ'τα LED στις άκρες της που ανιχνεύει η πρώτη κάμερα Wii0. Ομοίως, το Average 1 είναι το κέντρο που προκύπτει απ'τις μετρήσεις της κάμερας Wii1. Ας δούμε τον κώδικα για το Average 0 και τα ίδια ισχύουν και για το Average 1. Πρώτα ελέγχεται αν κάποια απ'τις δύο συστοιχίες LED βρίσκεται εκτός του πεδίου ανίχνευσης της Wii0. Σε αυτή τη περίπτωση το αντίστοιχο διάνυσμα Wii0_IR έχει την τιμή (0,0,0). Έτσι με το κανάλι Get vector length υπολογίζουμε το μήκος του διανύσματος Wii0_IR1 και αν είναι 0, σημαίνει ότι αυτή η άκρη είναι εκτός του πεδίου ανίχνευσης. Αυτή την λειτουργία εκτελεί το κανάλι len-wii0_ir1 και παίρνει την τιμή 1 όταν το μήκος του διανύσματος είναι 0, αλλιώς για οποιοδήποτε μη μηδενικό μήκος παίρνει την τιμή 0. Εάν και οι δύο άκρες είναι εντός του πεδίου, τότε και τα δύο κανάλια len-wii0_ir0 και len-wii0_ir1 έχουν την τιμή 0, οπότε και η μαθηματική έκφραση : $A \text{ OR } B$, που έχουν ως γονικό κανάλι, παίρνει την τιμή 0 δίνοντας εντολή στο Average 0 που είναι ένα ChannelSwitch να εκτελέσει το μεσαίο παιδί του. Σε αυτό, αρχικά προστίθενται (κανάλι Add Two Vectors) τα δύο διανύσματα Wii0_IR0 και Wii0_IR1 και έπειτα το αποτέλεσμα διαιρείται (κανάλι Divide value) με την τιμή 2 για να μας δώσει το κέντρο της ράβδου. Αν κάποιο απ'τα διανύσματα έχει μήκος 0, τότε το Average 0 εκτελεί το τελευταίο του παιδί που επίσης είναι ένα ChannelSwitch που ελέγχει αν το μήκος του Wii0_IR1 είναι 0. Αν δεν είναι 0, τότε το ChannelSwitch και συνεπώς και το Average 0 γίνεται ίσο με το διάνυσμα Wii0_IR1, αλλιώς αν το μήκος του Wii0_IR1 είναι 0, τότε το ChannelSwitch και το Average 0 ταυτίζονται με το διάνυσμα Wii0_IR0 που μπορεί να είναι και μηδενικό και μη μηδενικό. Αν και τα δύο διανύσματα έχουν μήκος 0, τότε πάλι το Average 0 ισούται με το Wii0_IR0 που είναι τώρα το (0,0,0).



Εικόνα 3.41

Κώδικας εύρεσης του κέντρου της ράβδου για κάθε μία απ'τις δύο κάμερες

Η ίδια λογική εφαρμόζεται και στο κανάλι Average Centre της εικόνας 3.42. Τη θέση των διανυσμάτων Wii0_IR0 και Wii0_IR1 παίρνουν εδώ τα διανύσματα Average 0 και Average 1. Ελέγχονται πρώτα τα μήκη τους και αν και τα δύο είναι διαφορετικά του μηδενός, τότε το Average Centre προκύπτει απ' το άθροισμά τους και έπειτα τη διαίρεση του αθροίσματος με το 2. Αν κάποιο απ' τα δύο έχει μήκος 0, τότε το Average Centre ταυτίζεται με το άλλο που ίσως δεν είναι μηδενικό διάνυσμα. Αν και τα δύο έχουν μήκος 0, τότε το Average Centre γίνεται ίσο με το Average 0 που είναι το (0,0,0).



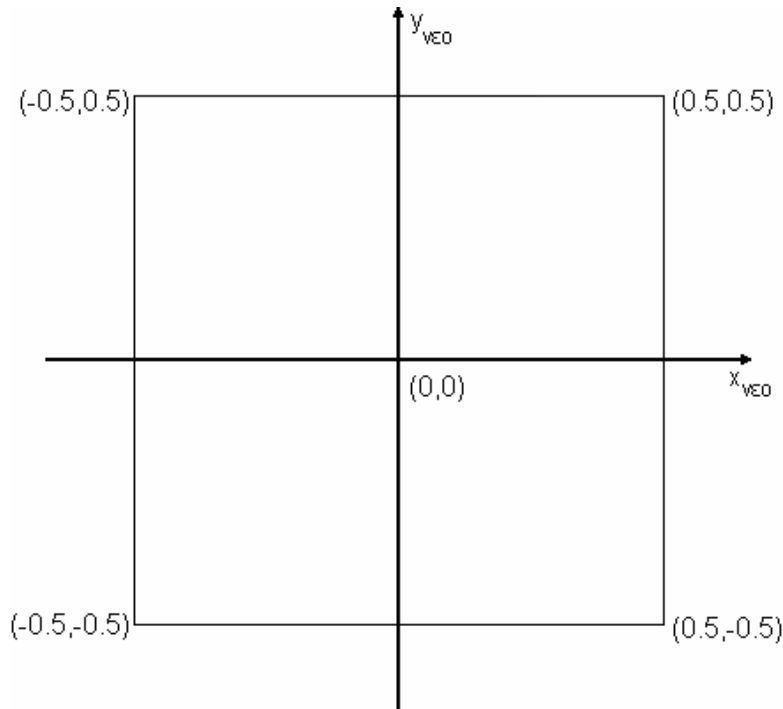
Εικόνα 3.42

Κώδικας εύρεσης του κέντρου της ράβδου

Στο σύστημα συντεταγμένων της κάμερας του Wii Remote, μετακινώντας τη ράβδο στον πραγματικό χώρο προς τα δεξιά το μετρούμενο x μειώνεται απ' την τιμή 1 μέχρι την τιμή 0. Μετακινώντας τη ράβδο προς τα πάνω, το μετρούμενο y αυξάνεται απ' την τιμή 0 μέχρι την τιμή 1. Έτσι, για να έχουμε καλύτερη αντίληψη της συσχέτισης μεταξύ των μετρούμενων x και y και της κίνησης του υποδοχέα στον εικονικό κόσμο, κάνουμε την εξής αλλαγή συστήματος συντεταγμένων (εικόνα 3.40) :

$$\{ x_{veo} = 0.5 - x_{παλιο} \text{ και } y_{veo} = y_{παλιο} - 0.5 \},$$

ώστε όταν, παραδείγματος χάρη, κουνάμε τη ράβδο προς τα δεξιά στον πραγματικό κόσμο το μετρούμενο x αυξάνεται και ο υποδοχέας μετακινείται προς τα δεξιά στον εικονικό κόσμο. Δηλαδή τελικά, το κέντρο της ράβδου (κανάλι New coords centre) χρησιμοποιεί το σύστημα συντεταγμένων της εικόνας 3.43 και εμείς μπορούμε και προσομοιώνουμε την κίνηση κατά τους άξονες x'x και y'y.



Εικόνα 3.43
Σύστημα νέων συντεταγμένων

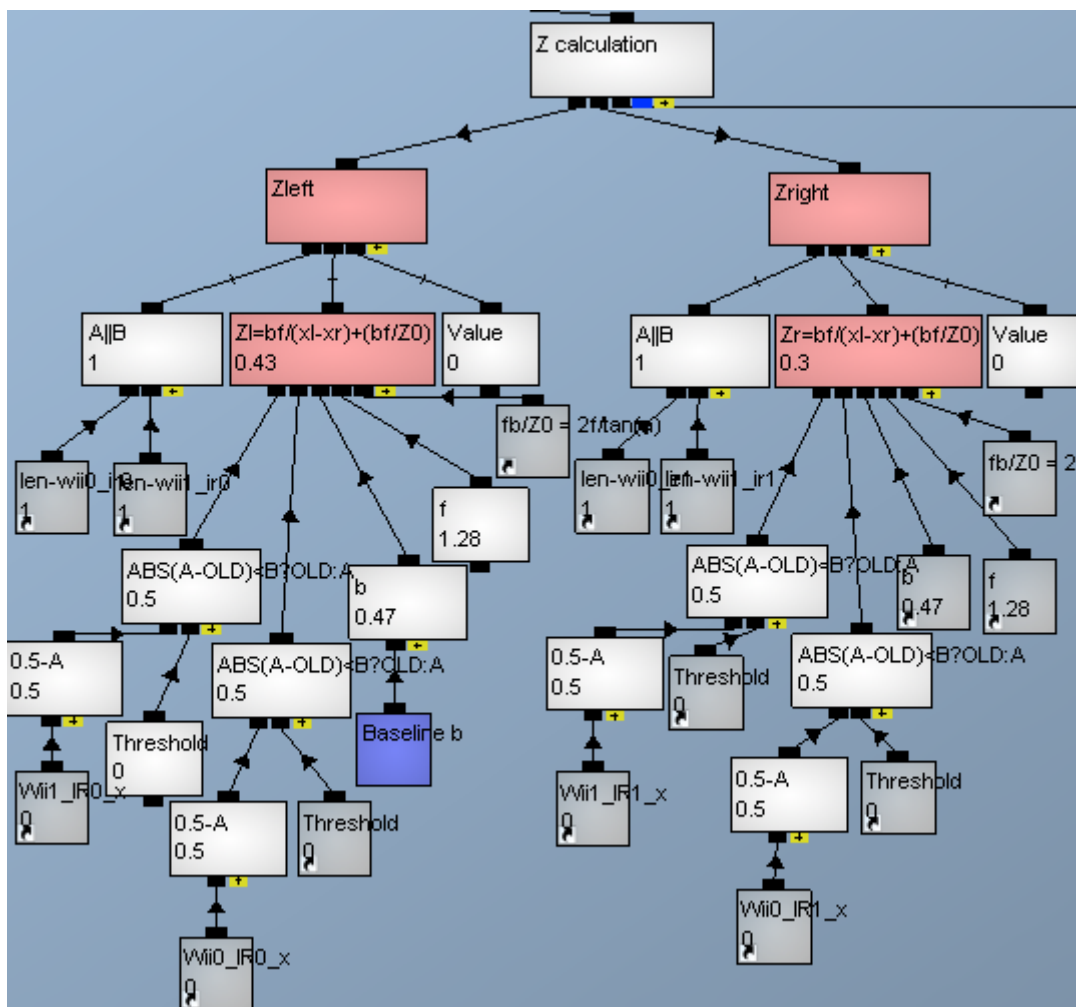
Για να προσομοιώσουμε και την κίνηση του υποδοχέα κατά τον άξονα $z'z$ του εικονικού κόσμου (που ταυτίζεται με τον $Z'Z$ του πραγματικού), εφαρμόζουμε την παρακάτω εξίσωση (που αποδεικνύεται στην παραγράφου 3.4) :

$$Z = \frac{fb}{(x'_l - x'_r) + fb/Z_0} \quad (3.1)$$

στα κανάλια της εικόνας 3.44 (στα κανάλια γράφουμε το Z με κεφαλαίο για να φαίνεται ότι ανήκει στο σύστημα συντεταγμένων του πραγματικού κόσμου). Εδώ υπολογίζουμε τις αποστάσεις της αριστερής (κανάλι Z_{left}) και δεξιάς (Z_{right}) συστοιχίας LED της ράβδου απ'τα δύο Wii Remote και βασισμένοι σε αυτές βρίσκουμε την απόσταση του κέντρου (Z_{aver}) της ράβδου (εικόνα 3.45). Θα αναλύσουμε τη δομή του καναλιού Z_{left} της εικόνας 3.44 και τα ίδια ισχύουν και για το Z_{right} . Το πρώτο του παιδί $A||B$, ή αλλιώς $A \text{ OR } B$, ελέγχει αν το αριστερό άκρο της ράβδου είναι μέσα στο πεδίο αντίχενωσης και των δύο καμερών υπολογίζοντας το μήκος των διανυσμάτων $Wii0_IR0$ και $Wii1_IR0$. Αν κάποιο απ'τα δύο ή και τα δύο είναι μηδενικά, τότε το Z_{left} παίρνει την τιμή 0 (τρίτο παιδί) καθώς δε μπορούμε να υπολογίσουμε το Z_{left} με μηδενικές τιμές τα $Wii0_IR0_x (x'_l)$ και $Wii1_IR0_x (x'_r)$.

Αν και τα δύο διανύσματα είναι μη μηδενικά, τότε το Z_{left} ισούται με το δεύτερο παιδί του που είναι η μαθηματική έκφραση της παραπάνω εξίσωσης. Σε αυτήν, το b είναι η τιμή της βασικής γραμμής, δηλαδή της απόστασης μεταξύ των δύο καμερών, που έχει εισάγει ο χρήστης στη δεύτερη σκηνή της εφαρμογής. Ο χρήστης, όμως, έχει γράψει την απόσταση σε cm, οπότε η τιμή b προκύπτει σε m διαιρώντας με το 100 την τιμή του καναλιού Baseline b . Η μεταβλητή f είναι η εστιακή απόσταση που είναι κοινή για τις δύο όμοιες κάμερες. Όταν η μέτρηση της θέσης της προβολής ενός LED γίνεται σε ανάλυση 1024x768 pixels x pixels, η εστιακή απόσταση είναι 1320 pixels, τότε όταν η μέτρηση που λαμβάνει το Quest3D απ'την κάμερα είναι το σύνολο $[0,1]$, αναλογικά προκύπτει ότι η εστιακή απόσταση που χρησιμοποιούμε τώρα στο Quest3D είναι 1.289. Το σημείο σύγκλισης των οπτικών αξόνων των δύο καμερών, Z_0 , υπολογίζεται απ'την

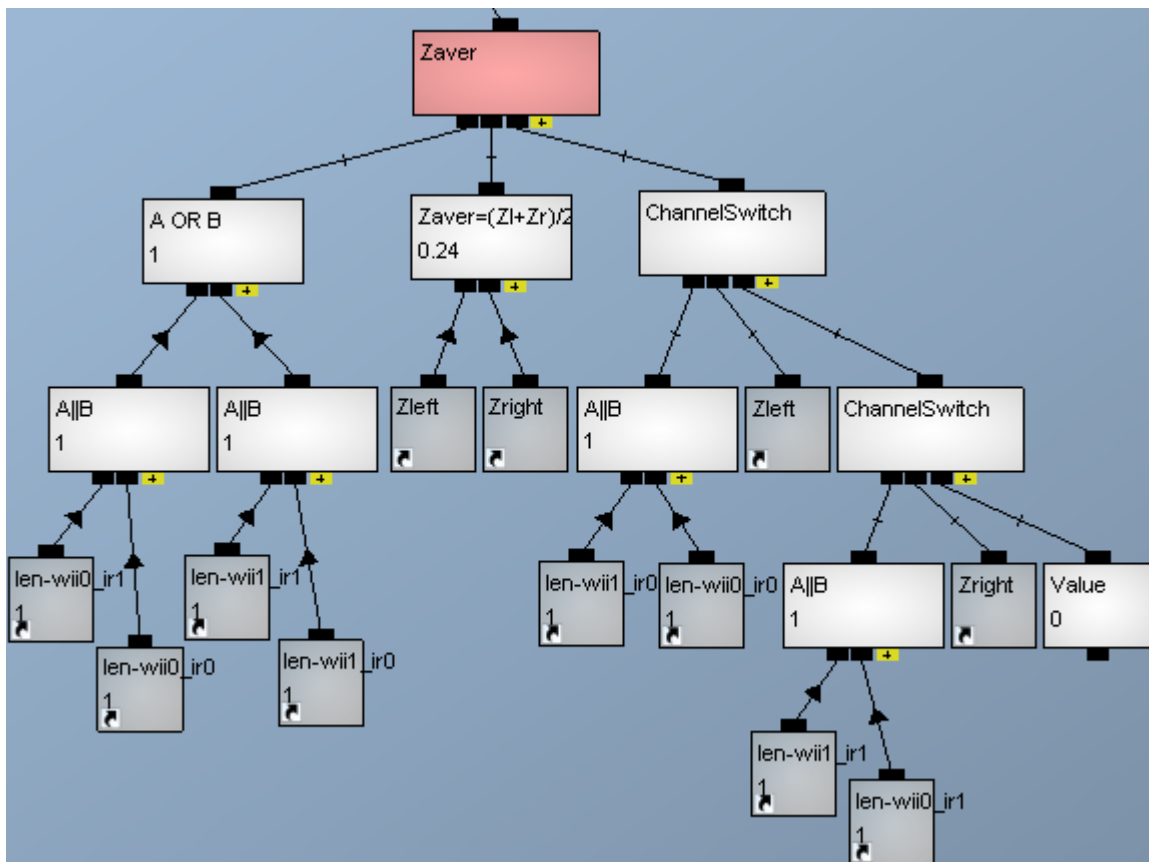
εξίσωση : $Z_0 = \tan a * b/2$ (εξηγείται στην παράγραφο 3.4), όπου a είναι η γωνία που σχηματίζουν τα Wii Remote με το επίπεδο της οθόνης. Οι μετρήσεις των $Wii0_IR0_x$ και $Wii1_IR0_x$ γίνονται με ακρίβεια έκτου δεκαδικού ψηφίου. Αυτό σημαίνει ότι μια πάρα πολύ μικρή κίνηση των χεριών μας γίνεται ανιχνεύσιμη και άρα μεταβάλλονται αυτές οι δύο τιμές. Όμως, έτσι, υπάρχει μια συνεχής κίνηση του υποδοχέα κατά τον άξονα $z'z$, η οποία είναι ενοχλητική για το χρήστη. Γι'αυτό δε συνδέουμε απευθείας τις τιμές $Wii0_IR0_x$ και $Wii1_IR0_x$ στο κανάλι : $Z_l = bf/(x_l - x_r) + (bf/Z_0)$ αλλά μία έκφραση με τύπο : $ABS(A-OLD) < B ? OLD : A$, η οποία προγραμματιστικά μεταφράζεται ως εξής : Αν η απόλυτη τιμή της διαφοράς του πρώτου παιδιού (π.χ. η $Wii0_IR0_x$) πλην της προηγούμενης τιμής της έκφρασης (OLD) είναι μικρότερη της τιμής κατωφλίου (το δεύτερο παιδί της έκφρασης, εδώ είναι ίση με 0.001 αν και στην εικόνα φαίνεται ότι έχει τη στρογγυλοποιημένη τιμή 0), τότε η έκφραση παίρνει ξανά την προηγούμενη τιμή της (δηλαδή η έκφραση δεν αλλάζει τιμή), αλλιώς η έκφραση γίνεται ίση με το πρώτο της παιδί (π.χ. την τιμή $Wii0_IR0_x$). Έτσι, ενώ κανονικά μεταβολές της τάξης του 10^{-6} θα προκαλούσαν αλλαγή του Z_{left} άρα και μετακίνηση του αντικειμένου στον άξονα $z'z$, με τη χρήση της έκφρασης καταφέρνουμε να περιορίσουμε την κίνηση όταν οι μεταβολές είναι μεγαλύτερες από την τιμή κατωφλίου 10^{-3} . Το αποτέλεσμα είναι ότι η κίνηση του υποδοχέα φαίνεται πιο φυσική και δεν ενοχλεί τον χρήστη. Οι μετρήσεις απ'τα Wii Remote και οι υπολογισμοί των εκφράσεων γίνονται σε κάθε frame.



Εικόνα 3.44

Κώδικας υπολογισμού απόστασης των δύο άκρων της ράβδου απ' τις κάμερες

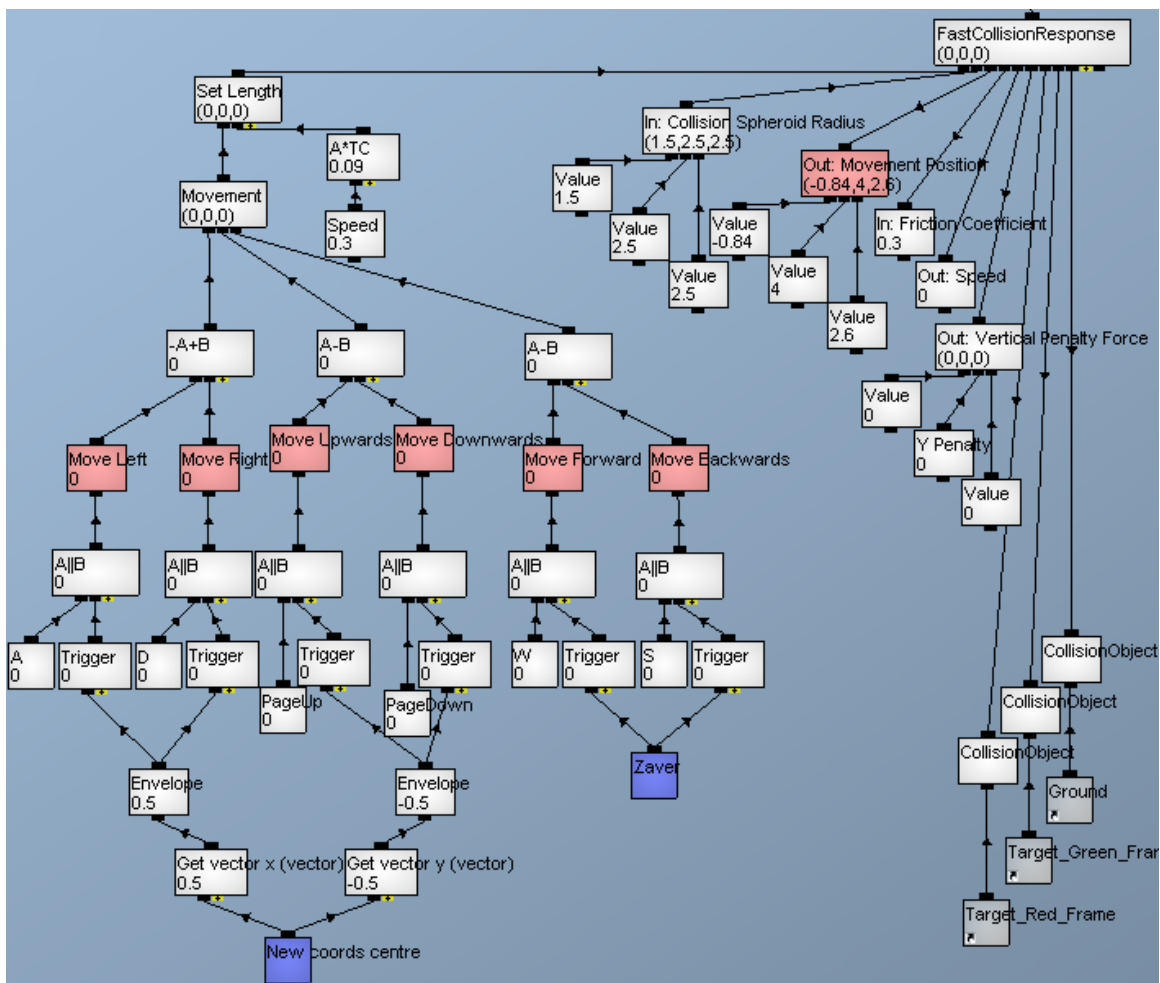
Η ίδια διεργασία γίνεται και για το Zright και τελικά, όπως βλέπουμε στην εικόνα 3.45, υπολογίζουμε την απόσταση του κέντρου της ράβδου, Zaverage. Το πώς υπολογίζεται εξαρτάται απ'το αν και τα δύο άκρα της ράβδου είναι εντός και των δύο πεδίων ανίχνευσης των δύο καμερών. Οπότε το πρώτο βήμα είναι να ελέγξουμε αν είναι μη μηδενικά και τα τέσσερα διανύσματα : Wii0_IR0, Wii0_IR1, Wii1_IR0, Wii1_IR1. Αν ισχύει αυτό, τότε το κανάλι : A OR B παίρνει την τιμή 0 και άρα το Zaver που είναι ένα κανάλι ChannelSwitch εκτελεί το μεσαίο του παιδί και υπολογίζει το Zaverage απ'τον μέσο όρο των Zleft και Zright. Αν κάποιο απ'τα τέσσερα διανύσματα είναι μηδενικό, άρα κάποιο απ'τα Zleft και Zright ή και τα δύο δε θα μπορούν να υπολογισθούν, τότε το Zaver εκτελεί το τελευταίο του παιδί. Να θυμήσουμε εδώ ότι το Wii Remote δίνει την ονομασία IR0 στη πηγή υπερύθρου φωτός που ανιχνεύει πρώτη και IR1 στη δεύτερη. Κατά σύμβαση, στην αρχή της εφαρμογής κρύβουμε τη δεξιά άκρη της ράβδου, οπότε ως IR0 και στις δύο κάμερες χαρακτηρίζεται το αριστερό άκρο της ράβδου. Έτσι, ελέγχοντας τα μήκη των διανυσμάτων Wii0_IR0 και Wii1_IR0 διαπιστώνουμε αν το αριστερό άκρο ανιχνεύεται και απ'τις δύο κάμερες. Αν και τα δύο είναι μη μηδενικά, τότε το Zaver ταυτίζεται με το Zleft. Αν ένα τουλάχιστον είναι μηδενικό, τότε εκτελείται ένα ακόμα ChannelSwitch που ελέγχει αυτή τη φορά τα διανύσματα Wii0_IR1 και Wii1_IR1 του δεξιού άκρου. Αν είναι μη μηδενικά, τότε το Zaver ταυτίζεται με το Zright, ενώ αν ένα τουλάχιστον είναι μηδενικό, το Zaver γίνεται ίσο με το 0. Όλοι αυτοί οι έλεγχοι γίνονται ώστε ακόμα και αν κάποιο άκρο βγει απ'το πεδίο ανίχνευσης μίας κάμερας, το Zaver να υπολογίζεται απ'τα υπόλοιπα μη μηδενικά διανύσματα.



Εικόνα 3.45

Κώδικας υπολογισμού απόστασης του κέντρου της ράβδου απ'τις κάμερες

Απ'την εξίσωση (1.6) που είναι για σύστημα κανονικού προσανατολισμού και την (3.1) που είναι για σύστημα τυχαίου προσανατολισμού προκύπτει ότι το Zaverage μετριέται στις ίδιες μονάδες με τη βασική γραμμή b. Στην εξίσωση το b είναι σε μέτρα άρα το ίδιο και το Zaverage. Βέβαια, εμάς δε μας ενδιαφέρει τόσο πολύ η απόλυτη τιμή του Zaverage, αλλά οι αυξομειώσεις του με τις οποίες προσομοιώνουμε την κίνηση του υποδοχέα στον Z άξονα, όπως θα δούμε στη συνέχεια. Η εικόνα 3.46 είναι ο φάκελος Position της κινητής κάμερας της εικόνας 3.17 και με τον οποίο πραγματοποιείται η μετακίνηση της κινητής κάμερας και επομένως και του υποδοχέα. Βλέπουμε ότι πρόκειται για ένα κανάλι FastCollisionResponse που στην ουσία είναι ένα διάλυμα το οποίο βασικά προκύπτει απ'το πρώτο του παιδί αλλά εξαρτάται και απ'τα υπόλοιπα παιδιά του. Ας δούμε πρώτα τα υπόλοιπα παιδιά :



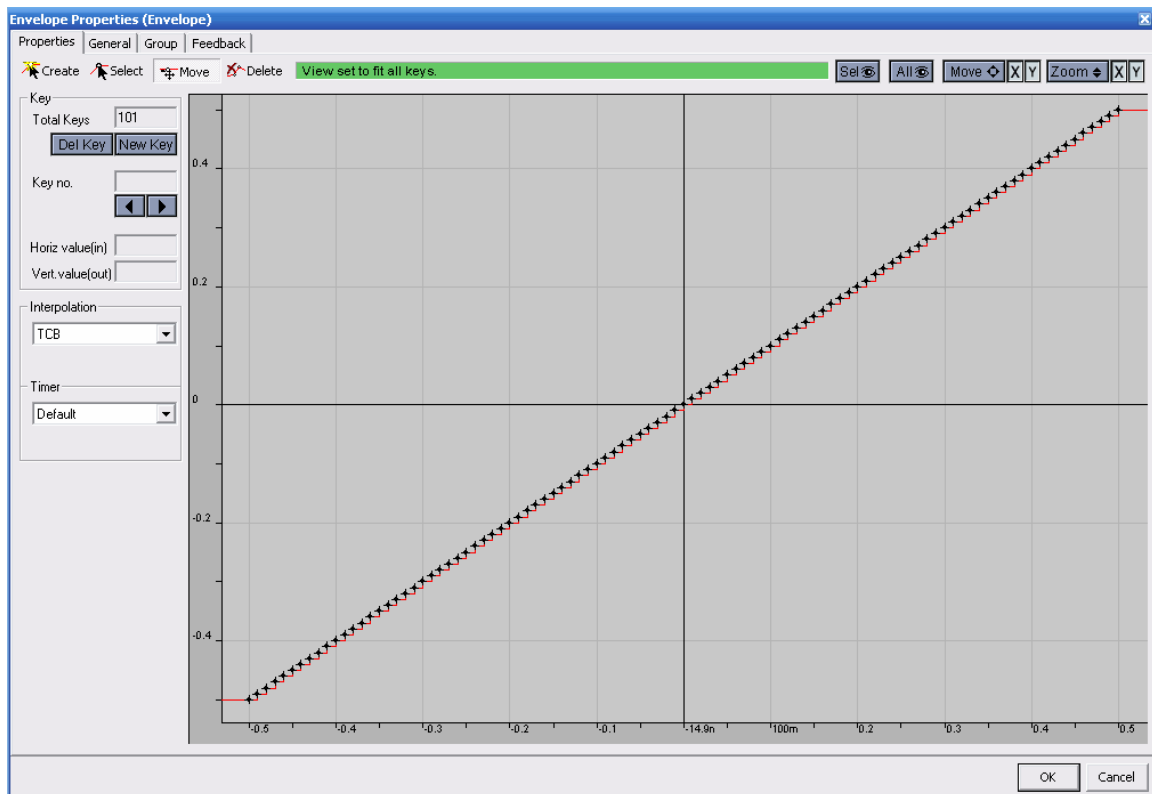
Εικόνα 3.46

Κώδικας υλοποίησης μετακίνησης της κινητής κάμερας

- Το δεύτερο (κανάλι In: Collision Spheroid Radius) είναι ένα διάνυσμα που δίνει τις διαστάσεις ενός σφαιροειδούς που περικλείει το αντικείμενο που έχει ως διάνυσμα κίνησης το FastCollisionResponse. Το σφαιροειδές αυτό είναι αόρατο και χρησιμεύει ως ο όγκος του αντικείμενου όσον αφορά τις συγκρούσεις. Μια χρήση του είναι να δώσουμε σε μια κάμερα ένα σώμα συγκρούσεων, ώστε να μην περνάει μέσα από τοίχους ή να πέφτει κάτω απ' το δάπεδο.
- Το τρίτο (κανάλι Out: Movement Position) είναι το διάνυσμα θέσης που προκύπτει απ' την πρόσθεση του διανύσματος του πρώτου παιδιού στο διάνυσμα θέσης του αντικείμενου του προηγούμενου frame αφού ελεγχθούν πρώτα τυχούσες συγκρούσεις.
- Το τέταρτο (κανάλι In: Friction Coefficient) είναι ο συντελεστής τριβής που αναπτύσσεται όταν το αντικείμενο εφάπτεται με κάποιο άλλο collision αντικείμενο.
- Το πέμπτο (κανάλι Out: Speed) είναι η τιμή της ταχύτητας του αντικείμενου.
- Το έκτο (κανάλι Out: Vertical Penalty Force) είναι ένα διάνυσμα που μπορεί να προστίθεται στην κίνηση του αντικείμενου, π.χ. το διάνυσμα της βαρύτητας.
- Τα επόμενα παιδιά είναι τα collision αντικείμενα με τα οποία μπορούν να υπάρξουν συγκρούσεις.

Το πρώτο παιδί είναι ένα κανάλι Set Length που απλά πολλαπλασιάζει το διάνυσμα που έχει ως πρώτο παιδί του επί την τιμή που έχει ως δεύτερο. Η τιμή αυτή προκύπτει απ' την τιμή της μεταβλητής Speed που είναι η ταχύτητα με την οποία θέλουμε να κινείται το αντικείμενο. Τα κανάλια που αντιστοιχούν στις συντεταγμένες του διανύσματος Movement μεταβάλλουν τις τιμές τους και άρα προκαλούν την κίνηση. Η x προκύπτει απ' την έκφραση $-A+B$, καθώς το πρώτο παιδί της είναι η κίνηση προς τα αριστερά άρα μειώνεται το x . Η κίνηση προς τα αριστερά προκαλείται όταν ο χρήστης πατάει το πλήκτρο A ή (κανάλι A||B) όταν ενεργοποιείται το κανάλι Trigger. Το Trigger ενεργοποιείται όταν μειώνεται η τιμή του καναλιού που έχει ως παιδί. Αυτό είναι ένα κανάλι Envelope το οποίο ουσιαστικά είναι ένα γράφημα με μία τιμή εξόδου για κάθε τιμή εισόδου. Στην εικόνα 3.47 βλέπουμε τον Envelope της εικόνας 3.46 που έχει ως παιδί το κανάλι Get vector y . Για την κίνηση στον άξονα των x , το envelope παίρνει τις τιμές εισόδου απ' τα x (κανάλι Get vector x) του διανύσματος New coords centre. Επειδή οι μετρούμενες τιμές x και y των Wii0_IR0, Wii0_IR1, Wii1_IR0, Wii1_IR1 είναι με ακρίβεια έκτου δεκαδικού ψηφίου, έπεται ότι και οι τιμές x και y του διανύσματος New coords centre θα είναι της ίδια ακρίβειας αφού υπολογίζονται από αυτές. Έτσι, όπως και στην περίπτωση του z , πρέπει να απορριφθούν οι τιμές πολλών δεκαδικών των τιμών x και y για να μην γίνεται ενοχλητική η κίνηση. Η απόρριψη γίνεται με το κανάλι envelope και μοιάζει με τη διαδικασία δειγματοληψίας του πλάτους ενός συνεχούς στο πεδίο του χρόνου σήματος, σε διακριτές χρονικές τιμές. Έτσι, οι τιμές εξόδου του γραφήματος είναι πεπερασμένες σε πλήθος, οπότε τα δύο κανάλια Trigger που προκαλούν είτε την κίνηση προς τα αριστερά (μείωση των τιμών εξόδου του envelope) είτε προς τα δεξιά (αύξηση του envelope) δεν ενεργοποιούνται σε μεταβολές τιμών έκτων δεκαδικών ψηφίων αλλά μόνο στις μεταβολές των τιμών εξόδου του γραφήματος.

Η κίνηση στον άξονα των y ακολουθεί την ίδια αρχή. Με το κανάλι Get vector y λαμβάνουμε μόνο την τιμή y του διανύσματος New coords centre. Έπειτα φτιάχνουμε πάλι ένα envelope και οι αυξήσεις των τιμών εξόδου προκαλούν την ενεργοποίηση του Trigger για την κίνηση προς τα πάνω. Οι μειώσεις προκαλούν την κίνηση προς τα κάτω.



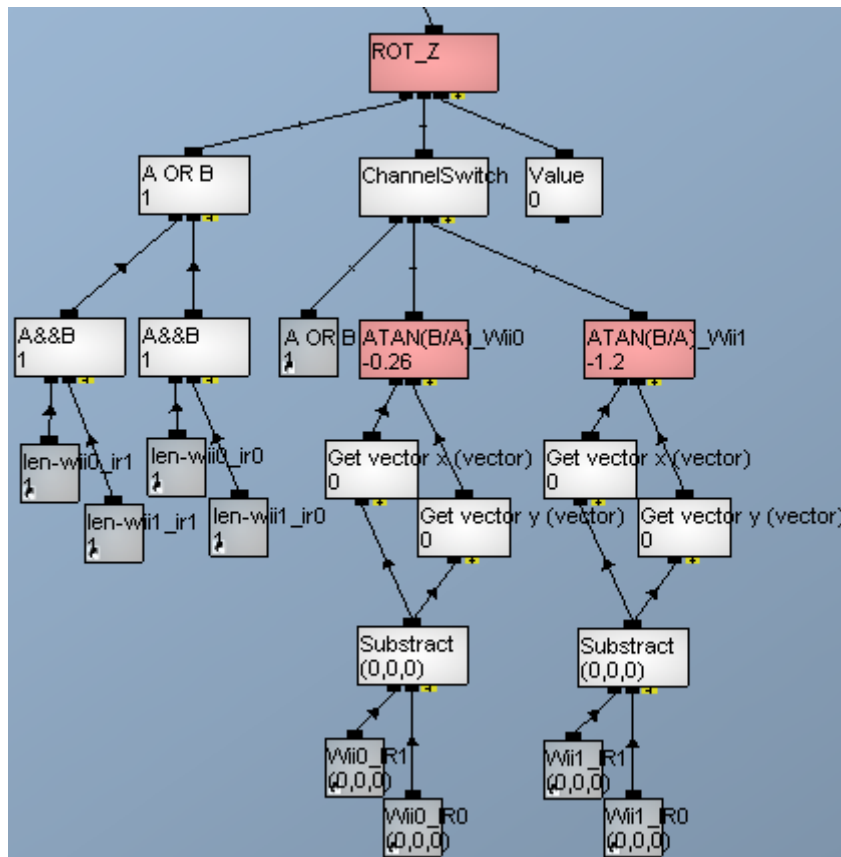
Εικόνα 3.47
 Παράδειγμα ενός καναλιού envelope

Όσον αφορά την κίνηση στον άξονα $z'z$, η ενεργοποίηση των καναλιών Trigger προκαλείται απ' τις μεταβολές της τιμής $Z_{average}$. Σε αυτήν, όπως έχουμε προηγουμένως πει (κώδικας εικόνας 3.44 για τα Z_{left} και Z_{right} απ' τα οποία υπολογίζεται το $Z_{average}$), οι πολύ μικρές μεταβολές της τιμής της απορρίπτονται. Στην τιμή του $Z_{average}$ δε χρησιμοποιήσαμε envelope γιατί τα δύο κανάλια envelope για τις τιμές των x και y είναι προκατασκευασμένα με γνωστές τη μέγιστη (0.5) και ελάχιστη τιμή (-0.5) που μπορούν να πάρουν οι δύο συντεταγμένες x και y του New coords centre, ενώ για το $Z_{average}$ δε γνωρίζουμε το μέγιστο και ελάχιστό του καθώς πρόκειται για την απόσταση του χρήστη απ' τις δύο κάμερες που μεταβάλλεται σε κάθε διαφορετικό χώρο και επίσης εξαρτάται και απ' την απόσταση μεταξύ των δύο καμερών (εξίσωση (1.6) και (3.1)).

3.3.3) Περιστροφή γύρω απ'τους άξονες x'x, y'y, z'z

Η περιστροφή γύρω απ'τον άξονα των x (που συμπίπτει με τον άξονα της ράβδου όταν την κρατάμε στα χέρια μας παράλληλη προς το δάπεδο και την οθόνη που εμφανίζεται ο φανταστικός κόσμος) δε γίνεται να υπολογιστεί λόγω αδυναμίας του software, δηλαδή του προγράμματος Quest3D. Ενώ η Wii Remote κάμερα μπορεί να ανιχνεύει ταυτόχρονα μέχρι τέσσερεις στόχους, το Quest3D μπορεί να λαμβάνει δεδομένα μόνο για μέχρι δύο. Έτσι, έχοντας δύο το πολύ ανιχνεύσιμα σημεία δεν έχουμε αρκετές πληροφορίες για να υπολογίσουμε αυτή τη περιστροφή.

Η γωνία της περιστροφής γύρω απ'τον άξονα των z (συμπίπτει με τον άξονα ανάμεσα στο χρήστη και στην οθόνη, που είναι παράλληλος προς το δάπεδο και κάθετος στην οθόνη) υπολογίζεται απ'τον κώδικα της εικόνας 3.48. Για τον υπολογισμό της αρκεί να βρίσκεται η ράβδος μέσα στο πεδίο ανίχνευσης μίας απ'τις δύο κάμερες, δηλαδή μας αρκούν τα διανύσματα Wii0_IR0 και Wii0_IR1 ή τα Wii1_IR0 και Wii1_IR1. Έτσι το ROT_Z που είναι ένα ChannelSwitch κανάλι ελέγχει πρώτα αν ισχύει ο παραπάνω περιορισμός με το κανάλι A OR B που έχει ως παιδιά τα δύο A AND B. Το κάθε AND ελέγχει ένα διάνυσμα απ'τη μία κάμερα και ένα άλλο απ'την άλλη. Άρα, αν αυτά τα δύο διανύσματα είναι μηδενικά τότε το κανάλι AND παίρνει την τιμή 1 που με τη σειρά του κάνει και το OR κανάλι 1. Σε αυτή τη περίπτωση το ROT_Z παίρνει την τιμή 0. Αν ισχύει ο περιορισμός, τότε το ROT_Z παίρνει την τιμή του μεσαίου παιδιού του που και αυτό είναι κανάλι ChannelSwitch. Ως μεταβλητή ελέγχου έχει τη συντόμευση : A OR B του καναλιού Average0 της εικόνας 3.41. Με αυτή ελέγχει αν κάποιο απ'τα διανύσματα της κάμερας Wii0 είναι μηδενικό.



Εικόνα 3.48

Κώδικας υπολογισμού γωνίας περιστροφής γύρω από τον άξονα z'z

Αν και τα δύο είναι μη μηδενικά, τότε η γωνία περιστροφής προκύπτει απ'το τόξο εφαπτομένης (κανάλι ATAN(B/A)_Wii0) της διαφοράς (κανάλι Subtract) των y (κανάλι Get vector y) διά της διαφοράς (κανάλι Subtract) των x (κανάλι Get vector x) των δύο διανυσμάτων Wii0_IR0 και Wii0_IR1 της κάμερας Wii0. Αν κάποιο είναι μηδενικό, τότε αναγκαστικά προκύπτει (λόγω του προηγούμενου ελέγχου A OR B) ότι τα διανύσματα Wii1_IR0 και Wii1_IR1 της κάμερας Wii1 είναι μη μηδενικά, οπότε και η γωνία υπολογίζεται απ'το τόξο εφαπτομένης (κανάλι ATAN(B/A)_Wii1) της διαφοράς (κανάλι Subtract) των y (κανάλι Get vector y) διά της διαφοράς (κανάλι Subtract) των x (κανάλι Get vector x) των Wii1_IR0 και Wii1_IR1.

Για την γωνία περιστροφής γύρω απ'τον άξονα των y (που συμπίπτει με τον άξονα που διέρχεται απ'το κέντρο της ράβδου στην αρχική θέση της και είναι κάθετος προς το δάπεδο), θα χρειαστεί να υπολογίσουμε πρώτα τη συντεταγμένη X των δύο άκρων της ράβδου στο σύστημα συντεταγμένων του πραγματικού κόσμου και έπειτα να πάρουμε το τόξο εφαπτομένης της διαφοράς των Z διά της διαφοράς των X των δύο άκρων. Η συντεταγμένη X ενός σημείου της σκηνής στον πραγματικό κόσμο δίνεται απ'την εξίσωση :

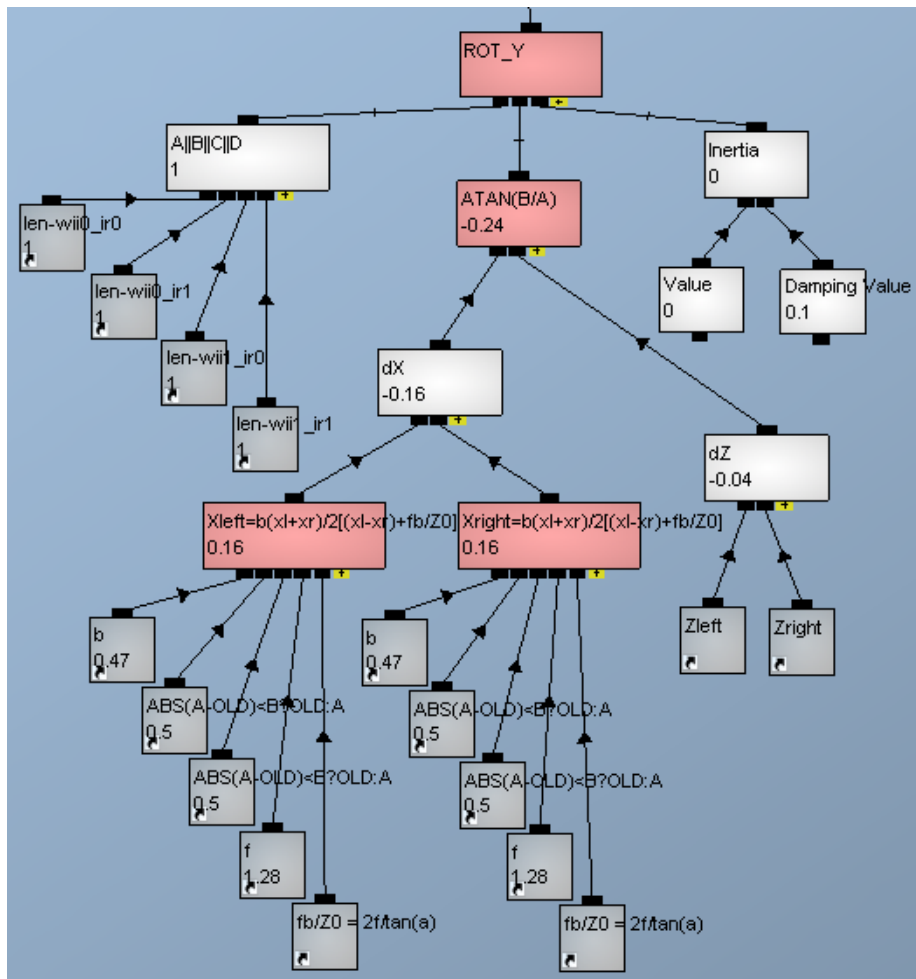
$$X = \frac{b * (x_l' + x_r')}{2 * \left[(x_l' - x_r') + \frac{fb}{Z_0} \right]} \quad (3.2)$$

Η απόδειξη της (3.2) γίνεται στη παράγραφο 3.4. Το b είναι η βασική γραμμή, f είναι η εστιακή απόσταση και το Z₀ είναι το σημείο σύγκλισης των οπτικών αξόνων των δύο καμερών και υπολογίζεται απ'την εξίσωση : $Z_0 = \tan a * \frac{b}{2}$ (εξηγείται στην παράγραφο 3.4), όπου a είναι η γωνία που σχηματίζουν τα Wii Remote με το επίπεδο της οθόνης.

Την εξίσωση (3.2) χρησιμοποιούμε στα δύο κανάλια Xleft και Xright της εικόνας 3.49 για να υπολογίσουμε το X του αριστερού και του δεξιού άκρου της ράβδου, το οποίο όμως σημαίνει ότι πρέπει και τα δύο άκρα να βρίσκονται στο πεδίο αντίληψης και των δύο καμερών. Έτσι, πρώτα ελέγχουμε αν ένα τουλάχιστον απ'τα τέσσερα διανύσματα Wii0_IR0, Wii0_IR1, Wii1_IR0 και Wii1_IR1 είναι μηδενικό. Αν αυτό ισχύει, τότε το κανάλι ROT_Y γίνεται ίσο με την τιμή του τελευταίου παιδιού του. Εκεί βρίσκεται ένα κανάλι Inertia που εξομαλύνει την τιμή του πρώτου του παιδιού χρησιμοποιώντας την τιμή του δεύτερου παιδιού του, Damping Value (συντελεστής απόσβεσης). Το αποτέλεσμα είναι ότι αν παραδείγματος χάρη η γωνία ROT_Y είναι 0.5 radians και το δεξί άκρο της ράβδου βγει εκτός του πεδίου αντίληψης της δεξιάς κάμερας, η τιμή της ROT_Y μειώνεται σταδιακά μέχρι να γίνει 0. Με αυτό το τρόπο, δε γίνεται άμεση μετάβαση απ'την τιμή 0.5 στην τιμή 0, που θα ήταν ενοχλητική για το χρήστη.

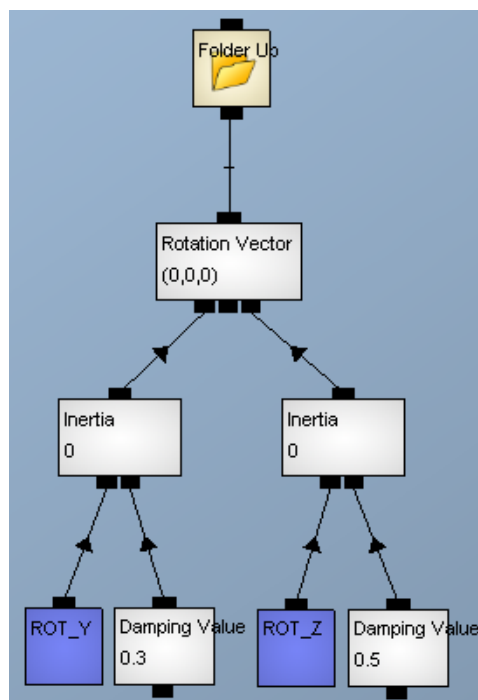
Αν όλα τα διανύσματα είναι μη μηδενικά, τότε το ROT_Y γίνεται ίσο με την τιμή του μεσαίου του παιδιού. Αυτό υπολογίζει την γωνία απ'το τόξο εφαπτομένης (κανάλι ATAN(B/A) της διαφοράς (κανάλι dZ=Zleft-Zright) των αποστάσεων του αριστερού και του δεξιού άκρου διά τη διαφορά (κανάλι dX=Xleft-Xright) των συντεταγμένων X του αριστερού και δεξιού άκρου στο σύστημα συντεταγμένων του πραγματικού κόσμου. Για τις τιμές x_l' και x_r' , χρησιμοποιούμε τις ήδη εξομαλυμένες τιμές για τον υπολογισμό της απόστασης Z των δύο άκρων απ'τον κώδικα της εικόνας 3.44.

Τελικά χρησιμοποιούμε τις τιμές ROT_Y και ROT_Z για να φτιάξουμε το διάνυσμα περιστροφής της κινητής κάμερας που πρόκειται για τον φάκελο Rotation της εικόνας 3.17 και βλέπουμε αναλυτικά στην εικόνα 3.50. Χρησιμοποιούμε πάλι κανάλια Inertia για να γίνεται πιο ομαλή η μεταβολή της τιμής της γωνίας.



Εικόνα 3.49

Κώδικας υπολογισμού γωνίας περιστροφής γύρω από τον άξονα y'y



Εικόνα 3.50

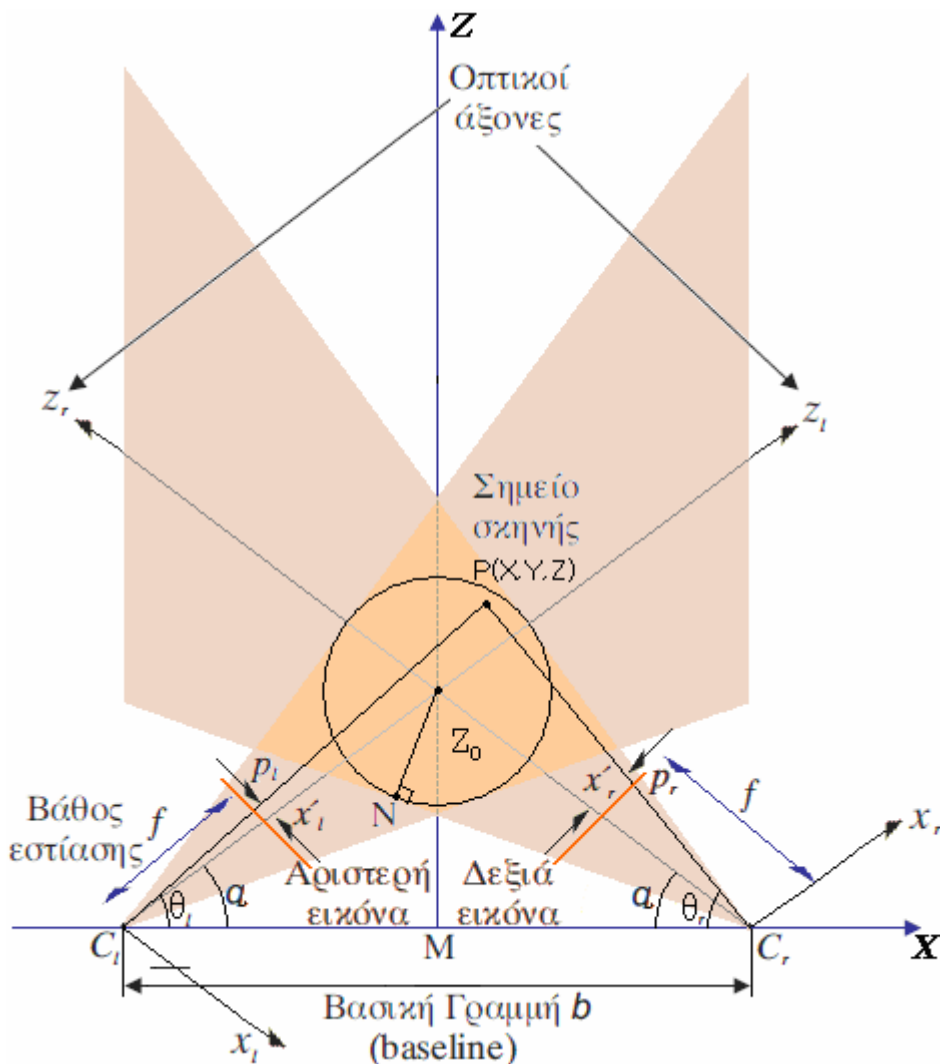
Κώδικας υλοποίησης περιστροφής της κινητής κάμερας

3.4) Οργάνωση και Διαμόρφωση Συστήματος

Λόγω της τοπολογίας του χώρου του εργαστηρίου, δε χρησιμοποιήσαμε τη περίπτωση των παραλλήλων καμερών (εικόνα 1.13), αλλά αυτή των συγκλινουσών [23] καμερών (converging cameras) που φαίνεται στην εικόνα 3.51. Οι δύο κάμερες Wii Remote τοποθετούνται στην αριστερή και δεξιά άκρη της οθόνης και σχηματίζουν γωνία α με το επίπεδό της. Τότε, οι οπτικοί άξονες z_l και z_r των καμερών συγκλίνουν στο σημείο Z_0 , δηλαδή το σημείο του πραγματικού χώρου γύρω του οποίου θα κινείται ο χρήστης, και το θεωρούμε ως δεδομένο που μας βοηθάει να υπολογίσουμε την άγνωστη γωνία α . Ως δεδομένη θεωρείται και η βασική γραμμή b που είναι η απόσταση μεταξύ των δύο καμερών. Έπειτα, γνωρίζοντας τα μεγέθη Z_0 , b και α , μπορούμε να βρούμε και τις συντεταγμένες X , Y και Z ενός σημείου της σκηνής $P(X,Y,Z)$. Υπενθυμίζουμε ότι οι δύο κάμερες βρίσκονται στο ίδιο ύψος, άρα δεν έχουμε κάθετη ανομοιότητα.

Απ' το τρίγωνο $C_r M Z_0$, της εικόνας 3.51, προκύπτει ότι :

$$\tan \alpha = \frac{(M Z_0)}{(C_r M)} \Rightarrow \tan a = \frac{Z_0}{b/2} \quad (3.3)$$



Εικόνα 3.51

Κάτοψη συστήματος τυχαίου προσανατολισμού (συγκλίνουσες κάμερες)

Προφανώς, μπορούμε να χρησιμοποιήσουμε τη (3.3) και αντίστροφα, δηλαδή γνωρίζοντας τη γωνία a , να βρούμε την απόσταση Z_0 του σημείου σύγκλισης. Για να βρούμε την ακτίνα του κύκλου με κέντρο το Z_0 μέσα στον οποίο ο χρήστης βρίσκεται ταυτόχρονα στα πεδία ανίχνευσης και των δύο καμερών, θεωρούμε το τρίγωνο NC_rZ_0 . Το σημείο N είναι η προβολή του σημείου Z_0 πάνω στην ευθεία που ορίζει το άκρο του πεδίου ανίχνευσης. Απ'το τρίγωνο C_rMZ_0 , προκύπτει ότι :

$$\sin a = \frac{(MZ_0)}{(Z_0C_r)} \Rightarrow$$

$$(Z_0C_r) = \sin a * Z_0$$

Ξέρουμε ότι το πεδίο ανίχνευσης έχει οριζόντια γωνία 33° , άρα η γωνία $\widehat{NC_rZ_0}$ είναι $\frac{33^\circ}{2} = 16.5^\circ$. Τότε, απ'το τρίγωνο C_rNZ_0 , προκύπτει ότι :

$$\sin(16.5^\circ) = \frac{(NZ_0)}{(Z_0C_r)} \Rightarrow$$

$$(NZ_0) = \sin(16.5^\circ) * \sin a * Z_0 \Rightarrow$$

$$(NZ_0) = 0.284 * \sin a * Z_0 \quad (3.4)$$

που είναι η ακτίνα του κύκλου που μέσα του μπορεί να κινείται το κέντρο της sensor bar και να βρίσκεται στα πεδία ανίχνευσης και των δύο καμερών.

Στην εφαρμογή που πραγματοποιήθηκε στο εργαστήριο, τοποθετήσαμε τις δύο κάμερες σε απόσταση 2.4m μεταξύ τους. Θέλαμε ο χρήστης να βρίσκεται και να κινείται σε απόσταση περίπου 75cm απ'την οθόνη. Ας υπολογίσουμε ως παράδειγμα τη γωνία που σχηματίζουν οι κάμερες με το επίπεδο της οθόνης και την ακτίνα του κύκλου στον οποίο θα μπορεί να κινείται ο χρήστης και να βρίσκεται στο πεδίο ανίχνευσης και των δύο καμερών.

$$\text{Απ'την (3.3), προκύπτει ότι : } \tan a = \frac{0.75m}{2.4m/2} \Rightarrow \tan a = 0.625 \Rightarrow a \approx 32^\circ$$

και απ'την (3.4), η ακτίνα του κύκλου είναι :

$$(NZ_0) = 0.284 * \sin 32^\circ * 0.75m \Rightarrow$$

$$(NZ_0) \approx 0.113m = 11.3cm \quad (3.5)$$

Στο σύστημα συγκλινουσών καμερών δεν ισχύουν οι εξισώσεις (1.6) και (1.7) :

$$Z = f \frac{b}{(x'_l - x'_r)} \quad \text{και} \quad X = \frac{b * (x'_l + x'_r)}{2 * (x'_l - x'_r)}$$

με τις οποίες υπολογίζουμε τις τιμές των συντεταγμένων X και Z στο σύστημα συντεταγμένων του πραγματικού κόσμου ενός σημείου της σκηνής σε σύστημα παραλλήλων καμερών. Αντίθετα, στη γενική περίπτωση των συγκλινουσών καμερών, οι εξισώσεις έχουν αυτή τη μορφή [24] :

$$Z = \frac{fb}{(x'_l - x'_r) + \frac{fb}{Z_0}} \quad (3.6)$$

$$X = \frac{b * (x'_l + x'_r)}{2 * \left[(x'_l - x'_r) + \frac{fb}{Z_0} \right]} \quad (3.7)$$

Ακολουθούν οι αποδείξεις τους.

Έστω θ_l και θ_r οι γωνίες που σχηματίζουν τα ευθύγραμμα τμήματα C_lP και C_rP αντίστοιχα με τον άξονα $X'X$. Τότε ισχύει :

$$\tan\theta_l = \frac{Z}{X + \frac{b}{2}} \quad \text{και} \quad \tan\theta_r = \frac{Z}{X - \frac{b}{2}} \quad (3.8)$$

Λύνοντας την πρώτη ως προς X , έχουμε :

$$X = \frac{Z}{\tan\theta_l} - \frac{b}{2}$$

και αντικαθιστώντας στην δεύτερη :

$$\begin{aligned} Z &= \tan\theta_r * \left(\frac{Z}{\tan\theta_l} - \frac{b}{2} - \frac{b}{2} \right) \Rightarrow \\ Z &= Z * \frac{\tan\theta_r}{\tan\theta_l} - b * \tan\theta_r \Rightarrow \\ Z &= \frac{b * \tan\theta_r}{\frac{\tan\theta_r}{\tan\theta_l} - 1} \Rightarrow \\ Z &= \frac{b * \tan\theta_r * \tan\theta_l}{\tan\theta_r - \tan\theta_l} \end{aligned} \quad (3.9)$$

Στην εικόνα 3.51 παρατηρούμε ότι για το σημείο σύγκλισης Z_0 ισχύει :

$$\tan a = \frac{Z_0}{b/2}$$

ενώ για το σημείο P :

$$\begin{aligned} \theta_l &= a - \arctan \frac{x'_l}{f} \\ \theta_r &= \pi - a - \arctan \frac{x'_r}{f} \end{aligned}$$

Χρησιμοποιώντας τις εξής ιδιότητες της εφαπτομένης :

$$\tan(-\phi) = -\tan \phi$$

$$\tan(\pi - \phi) = -\tan \phi$$

$$\tan(\phi + \psi) = \frac{\tan \phi + \tan \psi}{1 - \tan \phi * \tan \psi}$$

έχουμε :

$$\begin{aligned} \tan \theta_l &= \tan \left[\alpha + \left(-\arctan \frac{x_l'}{f} \right) \right] \Rightarrow \\ \tan \theta_l &= \frac{\tan \alpha + \tan \left(-\arctan \frac{x_l'}{f} \right)}{1 - \tan \alpha * \tan \left(-\arctan \frac{x_l'}{f} \right)} \Rightarrow \\ \tan \theta_l &= \frac{\frac{Z_0}{b/2} - \frac{x_l'}{f}}{1 + \frac{Z_0}{b/2} * \frac{x_l'}{f}} \Rightarrow \\ \tan \theta_l &= \frac{fZ_0 - x_l' * b/2}{f * b/2 + Z_0 * x_l'} \end{aligned}$$

και :

$$\begin{aligned} \tan \theta_r &= \tan \left[\pi - \left(\alpha + \arctan \frac{x_r'}{f} \right) \right] \Rightarrow \\ \tan \theta_r &= -\tan \left(\alpha + \arctan \frac{x_r'}{f} \right) \Rightarrow \\ \tan \theta_r &= -\frac{\tan \alpha + \tan \left(\arctan \frac{x_r'}{f} \right)}{1 - \tan \alpha * \tan \left(\arctan \frac{x_r'}{f} \right)} \Rightarrow \\ \tan \theta_r &= -\frac{\frac{Z_0}{b/2} + \frac{x_r'}{f}}{1 - \frac{Z_0}{b/2} * \frac{x_r'}{f}} \Rightarrow \\ \tan \theta_r &= -\frac{fZ_0 + x_r' * b/2}{f * b/2 - Z_0 * x_r'} \end{aligned}$$

Αντικαθιστώντας τις τιμές των $\tan\theta_l$ και $\tan\theta_r$ στην (3.9), προκύπτει :

$$Z = \frac{b \frac{fZ_0 - x_l' b/2}{fb/2 + Z_0 x_l'} * \frac{fZ_0 + x_r' b/2}{Z_0 x_r' - fb/2}}{\frac{fZ_0 + x_l' b/2}{Z_0 x_r' - fb/2} - \frac{fZ_0 - x_l' b/2}{fb/2 + Z_0 x_l'}} \Rightarrow$$

$$Z = b \frac{\frac{(fZ_0 - x_l' b/2) * (fZ_0 + x_r' b/2)}{(fb/2 + Z_0 x_l') * (Z_0 x_r' - fb/2)}}{\frac{(fZ_0 + x_l' b/2) * (fb/2 + Z_0 x_l') - (fZ_0 - x_l' b/2) * (Z_0 x_r' - fb/2)}{(Z_0 x_r' - fb/2) * (fb/2 + Z_0 x_l')}} \Rightarrow$$

$$Z = b \frac{f^2 (Z_0 - x_l' b/2f) * (Z_0 + x_r' b/2f)}{f * (Z_0 + x_l' b/2f) * (fb/2 + Z_0 x_l') - f * (Z_0 - x_l' b/2f) * (Z_0 x_r' - fb/2)}$$

Διαιρώντας αριθμητή και παρανομαστή με το μη μηδενικό Z_0 , έχουμε :

$$Z = fb \frac{(1 - x_l' b/2fZ_0) * (1 + x_r' b/2fZ_0)}{(1 + x_r' b/2fZ_0) * (fb/2Z_0 + x_l') - (1 - x_l' b/2fZ_0) * (x_r' - fb/2Z_0)} \Rightarrow$$

$$Z = \frac{fb}{\frac{x_l' + fb/2Z_0}{1 - \frac{bx_l'}{2fZ_0}} - \frac{x_r' - fb/2Z_0}{1 + \frac{bx_r'}{2fZ_0}}}$$

Τα μετρούμενα μεγέθη x_l' και x_r' είναι πάντα πολύ μικρότερα των f και Z_0 , οπότε μπορούμε να θεωρήσουμε ότι τα δύο κλάσματα $\frac{bx_l'}{2fZ_0}$ και $\frac{bx_r'}{2fZ_0}$ τείνουν προς το μηδέν.

Έτσι, έχουμε την εξής προσεγγιστική εξίσωση υπολογισμού του Z :

$$Z = \frac{fb}{(x_l' + fb/2Z_0) - (x_r' - fb/2Z_0)} \Rightarrow$$

$$Z = \frac{fb}{(x_l' - x_r') + \frac{fb}{Z_0}}$$

που είναι η (3.6).

Ομοίως θα εργαστούμε και για την (3.7).

Λύνοντας την πρώτη εξίσωση των (3.8) ως προς Z, έχουμε :

$$Z = \tan \theta_l * X + \tan \theta_l \frac{b}{2}$$

και αντικαθιστώντας στην δεύτερη :

$$\tan \theta_r * X - \tan \theta_r \frac{b}{2} = \tan \theta_l * X + \tan \theta_l \frac{b}{2} \Rightarrow$$

$$X = \frac{b}{2} * \frac{\tan \theta_l + \tan \theta_r}{\tan \theta_r - \tan \theta_l} \Rightarrow$$

$$X = \frac{b}{2} * \frac{\frac{fZ_0 - x_l' b/2}{fb/2 + Z_0 x_l'} - \frac{fZ_0 + x_r' b/2}{fb/2 - Z_0 x_r'}}{\frac{fZ_0 + x_r' b/2}{fb/2 - Z_0 x_r'} - \frac{fZ_0 - x_l' b/2}{fb/2 + Z_0 x_l'}} \Rightarrow$$

$$X = \frac{b}{2} * \frac{(fZ_0 - x_l' b/2) * (fb/2 - Z_0 x_r') - (fZ_0 + x_r' b/2) * (fb/2 + Z_0 x_l')}{-(fZ_0 + x_r' b/2) * (fb/2 + Z_0 x_l') - (fZ_0 - x_l' b/2) * (fb/2 - Z_0 x_r')} \Rightarrow$$

$$X = \frac{b}{2} * \frac{f^2 Z_0^2 [(1 - x_l' b/2 f Z_0) * (b/2 Z_0 - x_r' / f) - (1 + x_r' b/2 f Z_0) * (b/2 Z_0 + x_l' / f)]}{f^2 Z_0^2 [(-1 + x_r' b/2 f Z_0) * (b/2 Z_0 + x_l' / f) - (1 - x_l' b/2 f Z_0) * (b/2 Z_0 - x_r' / f)]} \Rightarrow$$

Θεωρώντας πάλι ότι τα δύο κλάσματα $\frac{bx_l'}{2fZ_0}$ και $\frac{bx_r'}{2fZ_0}$ τείνουν προς το μηδέν, το X υπολογίζεται προσεγγιστικά ως :

$$X = \frac{b}{2} * \frac{(b/2 Z_0 - x_r' / f - b/2 Z_0 - x_l' / f)}{(-b/2 Z_0 - x_l' / f - b/2 Z_0 + x_r' / f)} \Rightarrow$$

$$X = \frac{b}{2} * \frac{-\frac{x_r'}{f} - \frac{x_l'}{f}}{\left(-\frac{b}{Z_0} - \frac{x_l'}{f} + \frac{x_r'}{f}\right)} \Rightarrow$$

$$X = \frac{b * (x_l' + x_r')}{2 * \left[(x_l' - x_r') + \frac{fb}{Z_0} \right]}$$

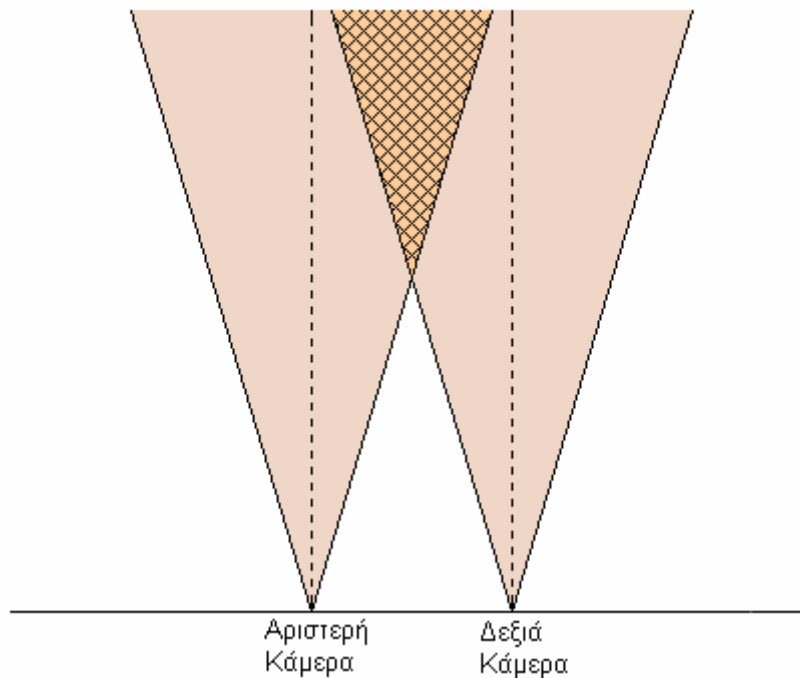
που είναι η (3.7).

Κεφάλαιο 4^ο) ΠΡΟΒΛΗΜΑΤΑ

Το πρώτο πρόβλημα που αντιμετωπίσαμε ήταν η μη ύπαρξη πληροφορίας όσον αφορά την απόσταση της πηγής υπέρυθρου φωτός απ'την κάμερα που περιέχεται μέσα στο Wii Remote. Θεωρώντας το μοντέλο της εικόνας 1.10, βλέπουμε ότι το σημείο P της σκινης έχει στο σύστημα συντεταγμένων του πραγματικού κόσμου διάνυσμα θέσης το (X,Y,Z) . Το σημείο σκινης προβάλλεται στο επίπεδο προβολής της κάμερας και προκύπτει το σημείο p όπου σε αυτό το σύστημα συντεταγμένων έχει συντεταγμένες (x',y') . Αυτές τις δύο τιμές τις πληροφορούμαστε απ'την κάμερα, οπότε μπορούμε να προσομοιώσουμε την κίνηση του ελεγχόμενου αντικειμένου μέσα στο εικονικό περιβάλλον κατά τους άξονες $x'x$ και $y'y$. Μας λείπει όμως η πληροφορία όσον αφορά τον άξονα $Z'Z$ (που ταυτίζεται με τον $z'z$) του συστήματος συντεταγμένων του πραγματικού κόσμου, δηλαδή αν ο χρήστης πλησιάζει ή απομακρύνεται απ'την κάμερα. Η λύση αυτού του προβλήματος προκύπτει απ'την στερεοσκοπία, δηλαδή την τεχνική με την οποία δημιουργείται τρισδιάστατη αναπαράσταση του χώρου με τη βοήθεια δύο εικόνων του χώρου από δύο διαφορετικές θέσεις. Έτσι, χρησιμοποιώντας δύο Wii Remote, άρα και δύο υπέρυθρες κάμερες, παίρνουμε δύο προβολές του ανιχνευόμενου σημείου. Γνωρίζοντας την εστιακή απόσταση f , που είναι η ίδια και για τις δύο κάμερες αφού είναι όμοιες, την απόσταση b που απέχουν μεταξύ τους τα δύο Wii Remote, την γωνία που σχηματίζουν τα Wii Remote με το επίπεδο της οθόνης και την ανομοιότητα d , που ορίζεται ως η διαφορά των συντεταγμένων x'_l και x'_r των προβολών του ανιχνευόμενου σημείου στα αντίστοιχα επίπεδα προβολής των δύο καμερών, απ'την εξίσωση (3.6), μπορούμε να υπολογίσουμε την απόσταση του σημείου απ'τις κάμερες.

Στη παράγραφο 3.3.3 εξετάσαμε πως γίνεται ο υπολογισμός των γωνιών που σχηματίζει η ράβδος (άρα και το ελεγχόμενο αντικείμενο του εικονικού κόσμου) όταν περιστρέφεται γύρω απ'τον άξονα $Z'Z$ και $Y'Y$. Σύμφωνα με την εικόνα 1.10 πάλι, αν p_l και p_r είναι οι προβολές του αριστερού και του δεξιού άκρου της ράβδου στο επίπεδο προβολής της μίας κάμερας, η πρώτη γωνία υπολογίζεται απ'το τόξο εφαπτομένης της διαφοράς των y' των δύο προβολών προς τη διαφορά των x' τους. Ομοίως, αν έχουμε ήδη βρει την απόσταση Z του αριστερού και του δεξιού άκρου, μπορούμε να υπολογίσουμε τη δεύτερη γωνία απ'το τόξο εφαπτομένης της διαφοράς των Z των δύο άκρων προς τη διαφορά των X των προβολών τους, αφού εφαρμόσουμε την εξίσωση (3.7) για να βρούμε τα X στο σύστημα συντεταγμένων του πραγματικού κόσμου. Όμως, περιστρέφοντας τη ράβδο γύρω απ'τον άξονα $X'X$, διαπιστώνουμε ότι και οι συντεταγμένες x',y' των προβολών των δύο άκρων της ράβδου στο επίπεδο προβολής δε μεταβάλλονται, αλλά και οι αποστάσεις τους απ'τις δύο κάμερες παραμένουν σταθερές, με αποτέλεσμα να μην έχουμε δεδομένα για να υπολογίσουμε τη γωνία περιστροφής γύρω απ'τον $X'X$. Μια λύση αυτού του προβλήματος θα ήταν αν αντί για τη ράβδο, χρησιμοποιούσαμε μια διάταξη σε σχήμα ρόμβου με τα LED στα τέσσερα άκρα. Σε αυτή τη περίπτωση θα βρίσκαμε τη γωνία απ'τις διαφορές των Z και των Y (δηλαδή στο σύστημα συντεταγμένων του πραγματικού κόσμου) του πάνω και κάτω άκρου του ρόμβου. Η κάμερα του Wii Remote είναι ικανή να ανιχνεύει ταυτόχρονα μέχρι τέσσερις στόχους, αλλά το πρόβλημα βρίσκεται στο κανάλι Get IR camera data from Wii, εικόνα 3.40, το οποίο μπορεί να μας δίνει δεδομένα για τις προβολές μόνο δύο φωτεινών πηγών. Όταν δοκιμάσαμε να χρησιμοποιήσουμε τη διάταξη ρόμβου με τα LED στα τέσσερα άκρα, διαπιστώσαμε ότι το κανάλι Get IR camera data from Wii μας παρείχε δεδομένα μόνο για τα δύο πρώτα LED που ανίχνευε η κάμερα, ενώ για τα υπόλοιπα δύο LED δεν είχαμε καθόλου πληροφορίες. Έτσι, με τη δυνατότητα ανίχνευσης δύο μόνο σημείων δε μπορούσαμε να υπολογίσουμε τη γωνία περιστροφής γύρω απ'τον άξονα $X'X$.

Ένα ακόμα πρόβλημα είναι ότι όπως αναφέρθηκε στη παράγραφο 1.3, η μέγιστη οριζόντια γωνία του πεδίου θέασης της κάμερα του Wii Remote είναι περίπου 33° . Στην εικόνα 4.1 βλέπουμε ένα σκαρίφημα των πεδίων αντίληψης των δύο καμερών, όταν αυτές είναι παράλληλες, όπου διακρίνεται και ο χώρος τομής τους. Πρόβλημα δημιουργείται όταν ο χρήστης βγει εκτός του χώρου τομής, καθώς τότε ένα τουλάχιστον απ'τα διανύσματα Wii0_IR0, Wii0_IR1, Wii1_IR0 και Wii1_IR1, δηλαδή οι συντεταγμένες των προβολών των δύο άκρων της ράβδου στα συστήματα προβολής των δύο καμερών, θα είναι μηδενικό με αποτέλεσμα να μας λείπουν δεδομένα για τον υπολογισμό των διανυσμάτων θέσης και περιστροφής της ράβδου. Έτσι, για την καλύτερη απόδοση της εφαρμογής, προσπαθούμε να μεγιστοποιήσουμε το χώρο τομής. Αυτό μπορεί να επιτευχθεί είτε αυξάνοντας την απόσταση της αρχικής θέσης του χρήστη απ' τις κάμερες όπου τότε το θεωρητικό μέγιστο εξαρτάται απ' την ισχύ των υπέρυθρων LED, είτε μειώνοντας την απόσταση μεταξύ των δύο καμερών, γεγονός όμως που εξαρτάται απ' την τοπολογία του χώρου.

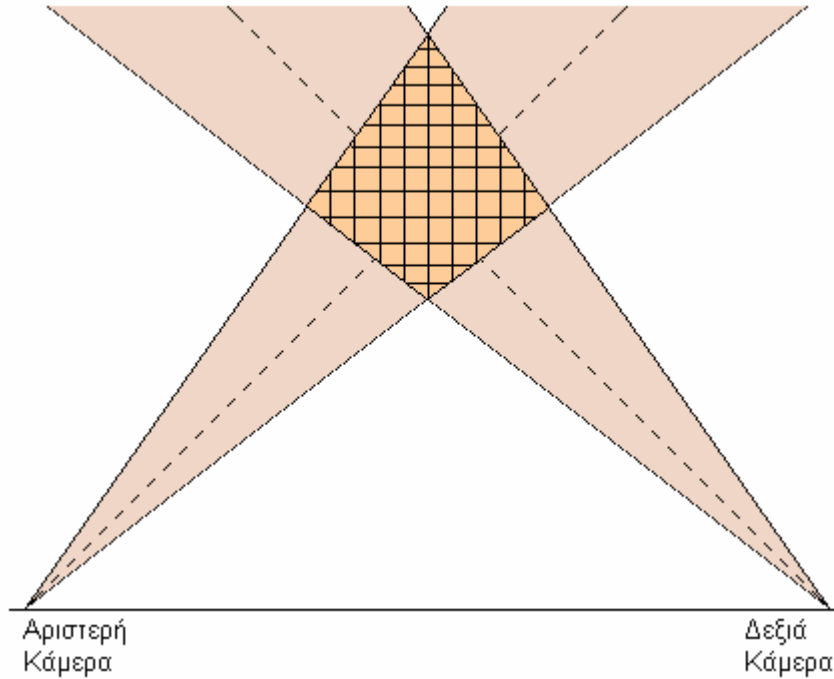


Εικόνα 4.1

Σκαρίφημα των πεδίων αντίληψης παράλληλων καμερών

Αν οι κάμερες είναι συγκλίνουσες, όπως αυτές της εικόνας 4.2, τότε για την εύρεση του χώρου τομής, θα χρησιμοποιήσουμε την εξίσωση (3.3) : $\tan a = \frac{Z_0}{b/2}$,

είτε υπολογίζοντας το σημείο Z_0 γύρω απ' το οποίο θα κινείται ο χρήστης με δεδομένη τη γωνία a που σχηματίζουν οι κάμερες με το επίπεδο της οθόνης, είτε υπολογίζοντας την γωνία a με δεδομένο το σημείο Z_0 . Η βασική γραμμή b θεωρείται πάντα γνωστή. Η μεγιστοποίηση του χώρου τομής επιτυγχάνεται πάλι αυξάνοντας την απόσταση του χρήστη απ' τις κάμερες (δηλαδή το Z_0 άρα και τη γωνία a) ή μειώνοντας την απόσταση μεταξύ των δύο καμερών.



Εικόνα 4.2

Σκαρίφημα των πεδίων αντίληψης συγκλινουσών καμερών

Ακόμα, υπήρχε το πρόβλημα της μη ομαλής κίνησης του αντικειμένου υποδοχέα που ελέγχει ο χρήστης. Όπως φαίνεται στην εικόνα 3.12 και αναλύσαμε στην παράγραφο 3.1.1, ο χρήστης δεν ελέγχει απευθείας το αντικείμενο υποδοχέα, κανάλι Basket στην εικόνα 3.12, αλλά την κινητή κάμερα. Ο υποδοχέας απλά χρησιμοποιεί την μήτρα κίνησης της κινητής κάμερας, με αποτέλεσμα το διάλυμα θέσης και περιστροφής να είναι ίδια με αυτά της κάμερας, δίνοντας έτσι την ψευδαίσθηση στο χρήστη ότι μετακινεί τον υποδοχέα, ενώ ουσιαστικά κινεί την άορατη κάμερα. Κάναμε αυτό το τέχνασμα, γιατί όταν δοκιμάσαμε να μετακινούμε απευθείας τον υποδοχέα, παρατηρήσαμε ότι η κίνηση του δεν ήταν ομαλή καθόλη τη χρονική διάρκεια της εφαρμογής. Αυτό οφείλεται στη διαρκή ενεργοποίηση και απενεργοποίηση του συστήματος φυσικής του εικονικού περιβάλλοντος. Το σύστημα φυσικής ενεργοποιείται στην αρχή της χρονικής περιόδου που αντιστοιχεί σε κάθε σφαίρα, ώστε να μπορούν να εφαρμοστούν δυνάμεις πάνω της και η σφαίρα να εκτοξευθεί προς το χρήστη. Στο πέρας της χρονικής περιόδου ή όταν η σφαίρα ακουμπάει το έδαφος ή όταν αποθέτουμε τη σφαίρα στο σωστό πλαίσιο, το σύστημα απενεργοποιόταν, ώστε οι σφαίρες να ξαναγυρίσουν στην αρχική τους θέση και να είναι έτοιμες για την επόμενη επανάληψη. Όμως, όταν το σύστημα φυσικής ήταν απενεργοποιημένο η κίνηση του υποδοχέα ήταν ομαλή σε μεγάλο βαθμό, ενώ όταν ήταν ενεργοποιημένο, η κίνησή του γινόταν μη ομαλή. Αυτό οφείλεται στην αύξηση του πλήθους των αριθμητικών υπολογισμών που κάνει το Quest3D για να εμφανίζει τα αντικείμενα στη σωστή θέση μέσα στο εικονικό περιβάλλον, όταν ενεργοποιείται το σύστημα φυσικής, καθώς προστίθενται πολλά νέα δεδομένα όπως ασκούμενες δυνάμεις στα φυσικά σώματα, η βαρύτητα, οι συγκρούσεις με άλλα σώματα, κτλ. Έτσι, χρησιμοποιώντας το τέχνασμα η κίνηση του υποδοχέα να μην γίνεται άμεσα απ' το χρήστη, αλλά εμμέσως απ' τη μήτρα κίνησης της κινητής κάμερας, ενός μη φυσικού σώματος που δεν ενδιαφέρει το σύστημα φυσικής, διαπιστώσαμε ότι η μετακίνηση του υποδοχέα διατηρείται ομαλή για όλη τη διάρκεια της εφαρμογής.

Όσον αφορά το τρέμουλο (jittering) στην κίνηση της κάμερας, είδαμε πως αντιμετωπίστηκε για την κίνηση στους άξονες $x'x$, $y'y$, $z'z$ στην παραγράφο 3.3.2 και για την περιστροφή γύρω απ'τους ίδιους άξονες στη παράγραφο 3.3.3. Για την κίνηση στους άξονες $x'x$ και $y'y$, χρησιμοποιήσαμε δύο κανάλια Envelope (εικόνα 3.47), καθώς ξέρουμε ότι, με το καινούργιο σύστημα συντεταγμένων (εικόνα 3.43) που δημιουργήσαμε για τη κάθε κάμερα, οι μετρούμενες τιμές x και y της προβολής του ανιχνευόμενου σημείου του χώρου πάνω στο επίπεδο προβολής της κάμερας παίρνουν ελάχιστη τιμή -0.5 και μέγιστη 0.5 με ακρίβεια έκτου δεκαδικού ψηφίου. Οπότε, για να μην υπάρχει μετακίνηση όταν έχουμε μεταβολή τόσο μικρών τιμών, χρησιμοποιήσαμε τα κανάλια Envelope για να φτιάξουμε ένα γράφημα με πεπερασμένες σε πλήθος τιμές εξόδου τις οποίες λαμβάνουν ως τιμές εισόδου τα κανάλια που είναι υπεύθυνα για τη κίνηση σε αυτούς τους άξονες. Το αποτέλεσμα είναι ότι η κάμερα κινείται στις μεταβολές αυτών των τιμών και όχι στις μεταβολές έκτου δεκαδικού ψηφίου των μετρούμενων x και y . Δεν μπορούμε να χρησιμοποιήσουμε την ίδια τεχνική για την κίνηση στον άξονα $z'z$, καθώς δε μπορούμε να προκατασκευάσουμε ένα κανάλι Envelope γιατί δε γνωρίζουμε την ελάχιστη και μέγιστη τιμή της απόστασης του χρήστη απ'τις κάμερες. Σε αυτή τη περίπτωση, το κανάλι που υπολογίζει αυτήν την απόσταση χρησιμοποιεί ένα άλλο κανάλι με ανατροφοδότηση (εικόνα 3.44) που ελέγχει αν η απόλυτη τιμή της διαφοράς της τιμής της μετρούμενης συντεταγμένης x στο συγκεκριμένο frame πλην την τιμή της τιμής x στο αμέσως προηγούμενο frame είναι μικρότερη από μια τιμή κατωφλίου που έχουμε θέσει. Αν είναι μικρότερη, τότε το ανατροφοδοτούμενο κανάλι παραμένει ίσο με την τιμή x του προηγούμενου frame, αλλιώς αν η απόλυτη τιμή της διαφοράς είναι μεγαλύτερη απ'την τιμή κατωφλίου, τότε το κανάλι γίνεται ίσο με την τιμή x του τωρινού frame. Έτσι, μπορούμε να ελέγχουμε την ευαισθησία στην κίνηση στον άξονα $z'z$, μεταβάλλοντας απλά την τιμή κατωφλίου.

Οι γωνίες περιστροφής γύρω απ'τους άξονες προκύπτουν απ'τα τόξα εφαπτομένων (εικόνας 3.48 και 3.49) των διαφορών των τιμών x και z που είναι ήδη ομαλοποιημένες, άρα και οι τιμές των γωνιών περιστροφής είναι επίσης ομαλοποιημένες. Επιπροσθέτως, για να φαίνεται ακόμα πιο φυσική η περιστροφή του σώματος, χρησιμοποιούμε ένα κανάλι Inertia (εικόνα 3.50), που του έχουμε ορίσει τον συντελεστή απόσβεσής του, για να εξομαλύνουμε ακόμα περισσότερο την τιμή της γωνίας.

Ένα πρόβλημα που δεν έχει ολοκληρωτική λύση είναι η αρίθμηση των φωτεινών πηγών όταν ανιχνεύονται απ'τις κάμερες. Όπως είπαμε και στην παράγραφο 3.3.2, η κάθε κάμερα αριθμεί με τον αριθμό 0 τη φωτεινή πηγή που ανιχνεύει πρώτη και με τον αριθμό 1 τη δεύτερη. Αλλά, αν τύχει και ο χρήστης κινηθεί εκτός του πεδίου ανίχνευσης της κάμερας και έπειτα ξαναεισέλθει, τότε υπάρχει η περίπτωση η φωτεινή πηγή που πριν είχε τον αριθμό 0 να ανιχνευθεί τώρα δεύτερη οπότε να αποκτήσει τον αριθμό 1. Έτσι όμως, μπερδεύονται οι μετρούμενες τιμές x_l και x_r των εξισώσεων (3.6) και (3.7) με αποτέλεσμα να μην έχουμε σωστούς υπολογισμούς για την τιμή Z και τις γωνίες. Μια μερική λύση του προβλήματος είναι να βγάλουμε απ'το πεδίο ανίχνευσης των δύο καμερών και τα δύο άκρα με LED της ράβδου. Ύστερα, κατά σύμβαση, να εισάγουμε το αριστερό άκρο και μετέπειτα το δεξιό, οπότε να διατηρείται πάντα η ίδια αρίθμηση για τα δύο άκρα.

Κεφάλαιο 5^ο) ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Το θέμα της εργασίας ήταν η μελέτη και η υλοποίηση λογισμικού ενός συστήματος ανίχνευσης στόχου (tracking). Η ανίχνευση πραγματοποιείται με την εκπομπή υπέρυθρου φωτός από μία ράβδο με LED που κρατά ο χρήστης και τη λήψη αυτού από κάμερες με υπέρυθρο φίλτρο. Έπειτα, οι συντεταγμένες των προβολών των φωτεινών πηγών στο επίπεδο προβολής των καμερών μεταφέρονται σε ένα υπολογιστικό σύστημα. Σε αυτό γίνονται οι κατάλληλοι υπολογισμοί, βασισμένοι στις εξισώσεις της στερεοσκοπίας, για την αποτίμηση της απόστασης του χρήστη απ' τις κάμερες. Επίσης, χρησιμοποιήσαμε το λογισμικό πρόγραμμα Quest3D σε έναν ηλεκτρονικό υπολογιστή για να δημιουργήσουμε το περιβάλλον της εικονικής πραγματικότητας που βλέπει ο χρήστης στην οθόνη του. Το εικονικό αυτό περιβάλλον αξιοποιεί τα δεδομένα απ' τις κάμερες, δηλαδή τα διανύσματα θέσης και κατεύθυνσης του χρήστη, ώστε οι πληροφορίες που μεταδίδει στο χρήστη, όπως η εικόνα και ο ήχος, να ανταποκρίνονται στην κίνησή του στον χώρο και σε πραγματικό χρόνο.

Το συγκεκριμένο σύστημα ανίχνευσης αποτελεί μια μεγάλης ακρίβειας και μικρού κόστους λύση. Τόσο οι υπέρυθροι φωτοδίοδοι όσο και τα Wii Remote που χρησιμοποιούμε ως κάμερες με υπέρυθρο φίλτρο, είναι εύκολα προσβάσιμα στο εμπόριο και σε χαμηλή τιμή, διευκολύνοντας έτσι τη δημιουργία ενός τέτοιου συστήματος. Το σύστημα ανίχνευσης μπορεί να συνεργασθεί με διαφορετικού είδους εικονικά περιβάλλοντα, αναλόγως ποιος είναι ο σκοπός του συστήματος εικονικής πραγματικότητας. Παραδείγματος χάρη, η ράβδος με τα LED θα μπορούσε να λειτουργήσει ως το πηδάλιο ενός εκπαιδευτικού αεροσκάφους [25] του εικονικού περιβάλλοντος, όπου ο χρήστης μετακινώντας τη ράβδο κατά τους άξονες X'X και Y'Y ή μεταβάλλοντας τη γωνία που σχηματίζει η ράβδος με τους άξονες αλλάζει την κατεύθυνση του αεροσκάφους. Ακόμα, ο χρήστης πλησιάζοντας τις κάμερες θα μπορούσε να αυξάνει την ταχύτητα του αεροσκάφους και αντίστοιχα αν απομακρύνεται απ' αυτές να μειώνεται η ταχύτητά του. Ένα άλλο παράδειγμα χρήσης του εικονικού περιβάλλοντος είναι η κινητική αποκατάσταση ενός ασθενούς. Αυτός θα εισάγεται σε ένα τέτοιο εικονικό κόσμο όπου θα απαιτείται να κάνει κάποιες συγκεκριμένες σωματικές κινήσεις (που θα ανιχνεύονται απ' το σύστημα ανίχνευσης) για να επιτύχει τους στόχους της εικονικής πραγματικότητας. Προφανώς οι κινήσεις του χρήστη στον πραγματικό κόσμο θα έχουν προβλεφθεί να είναι τέτοιες ώστε να επιταχύνουν την αποκατάστασή του.

Όμως, όπως είδαμε και στην παράγραφο 3.4 όπου μελετήσαμε την εφαρμογή στο χώρο του εργαστηρίου, όταν η απόσταση μεταξύ των δύο συγκλινουσών καμερών είναι μεγάλη και εμείς θέλουμε να κινούμαστε σε μια περιοχή κοντά στην οθόνη, τότε η ακτίνα του κύκλου, στον οποίο βρισκόμαστε ταυτόχρονα στο πεδίο ανίχνευσης και των δύο καμερών, είναι της τάξης των λίγων δεκάδων εκατοστών (εξίσωση (3.5)). Αυτό σημαίνει ότι αν βρεθούμε εκτός αυτού του κύκλου, κάποιο άκρο της sensor bar δε θα ανιχνεύεται τουλάχιστον από μία κάμερα, με αποτέλεσμα να μην υπάρχουν αρκετά διαθέσιμα δεδομένα στο Quest3D για να αναπαραστήσει σωστά την κίνηση και περιστροφή του ελεγχόμενου αντικειμένου στον εικονικό κόσμο και μειώνεται έτσι ο χώρος σωστής και αξιόπιστης λειτουργίας της εφαρμογής. Άρα, ο χρήστης αναγκάζεται να περιορίζει την μετακίνησή του σε μια μικρή περιοχή στον πραγματικό κόσμο.

Παρατηρώντας τις εξισώσεις (3.3) και (3.4), διαπιστώνουμε ότι η ακτίνα του κύκλου είναι ανάλογη προς την απόσταση της αρχικής θέσης του χρήστη (Z_0) και αντιστρόφως

ανάλογη της απόστασης των δύο καμερών (βασική γραμμή b). Επομένως, αυξάνοντας την απόσταση Z_0 και μειώνοντας την απόσταση b , καταφέρνουμε να μεγιστοποιήσουμε την περιοχή στην οποία μπορεί να κινείται ο χρήστης και να ανιχνεύεται απ' το σύστημα των δύο καμερών. Βέβαια, τα όρια αυτών των τιμών, όπως και της γωνίας α που σχηματίζουν οι κάμερες με το επίπεδο της οθόνης, εξαρτώνται απ' την τοπολογία του χώρου.

Οι μελλοντικές εργασίες θα πρέπει να επικεντρωθούν κυρίως γύρω απ' τα εξής δύο σημεία :

- Μεγιστοποίηση του χώρου δράσης χρήστη. Όπως είδαμε στο προηγούμενο κεφάλαιο, οι κάμερες των Wii Remote έχουν μικρό πεδίο ανίχνευσης λόγω της μικρής οριζόντιας γωνίας, περίπου 33° και της μικρής ισχύος των υπέρυθρων LED. Το πεδίο ανίχνευσης θα μπορούσε να αυξηθεί χρησιμοποιώντας καλύτερες υπέρυθρες κάμερες ή πιο ισχυρά LED που όμως θα προκαλούσαν και την παράλληλη αύξηση του κόστους δημιουργίας του συστήματος ανίχνευσης. Επίσης, το ιδανικό είναι ο χρήστης να βρίσκεται ταυτόχρονα στο πεδίο ανίχνευσης και των δύο καμερών, ώστε να υπάρχουν δεδομένα και για τα δύο άκρα της ράβδου και στις δύο κάμερες. Άρα, θα πρέπει να γίνει μια μελέτη του πραγματικού χώρου, ώστε να επιλεγεί το σύστημα παραλλήλων ή το σύστημα συγκλινουσών καμερών και έπειτα να μεγιστοποιηθεί ο χώρος τομής των δύο πεδίων ανίχνευσης αναλόγως τις θέσεις των καμερών και την απόσταση μεταξύ τους αλλά και της αρχικής θέσης του χρήστη.
- Καινούργιο κανάλι Get IR camera data from Wii. Η κάμερα του Wii Remote δύναται να ανιχνεύει ταυτόχρονα μέχρι τέσσερις στόχους και να καταγράφει τις συντεταγμένες των προβολών τους στο επίπεδο προβολής της. Όμως το κανάλι Get IR camera data from Wii της τωρινής έκδοσης (4.0) του λογισμικού Quest3D λαμβάνει και αξιοποιεί τα δεδομένα μόνο για τους δύο πρώτους στόχους που ανιχνεύει η κάμερα. Έτσι, όπως αναλύσαμε στο προηγούμενο κεφάλαιο, δεν είναι δυνατή η μέτρηση της γωνίας περιστροφής γύρω απ' τον άξονα $X'X$. Αν σε κάποια μελλοντική έκδοση του Quest3D, το κανάλι Get IR camera data from Wii μπορεί να καταμετρά τις συντεταγμένες των προβολών τεσσάρων φωτεινών πηγών, τότε με μια διαφορετική διάταξη των φωτοδιόδων θα μπορούμε να υπολογίσουμε και αυτή τη γωνία. Σε αυτή τη περίπτωση, τοποθετώντας τα τέσσερα υπέρυθρα LED στις άκρες ενός ρόμβου, μετρώντας τη διαφορά των αποστάσεων του πάνω και κάτω άκρου του ρόμβου απ' τις κάμερες και τη διαφορά των συντεταγμένων τους Y στο σύστημα συντεταγμένων του πραγματικού κόσμου, μπορούμε να βρούμε τη γωνία γύρω απ' τον άξονα $X'X$ απ' το τόξο εφαπτομένης του ηλίκου αυτών των διαφορών. Η συντεταγμένη Y ενός σημείου υπολογίζεται με την ίδια μέθοδο που χρησιμοποιήσαμε στη παράγραφο 3.4 όπου αποδείχθηκαν οι εξισώσεις (3.6) και (3.7)

και τελικά είναι :
$$Y = \frac{b * y'}{(x_l' - x_r') + \frac{fb}{Z_0}}$$
, όπου b είναι η βασική γραμμή δηλαδή η

απόσταση των δύο καμερών, f είναι η εστιακή απόσταση των καμερών, Z_0 είναι το σημείο σύγκλισης των οπτικών αξόνων των δύο καμερών, y' είναι η συντεταγμένη y της προβολής του σημείου στα επίπεδο προβολής των καμερών και οι x_l' και x_r' είναι οι συντεταγμένες x των σημείων προβολής του ανιχνευόμενου σημείου στα επίπεδα προβολής της αριστερής και δεξιάς κάμερας αντίστοιχα. Ακόμα, το ιδανικό θα ήταν η αρίθμηση των ανιχνευμένων σημείων να διατηρούταν σταθερή.

Βιβλιογραφία

- [1] Αμαλία Γιαννακά (2006). “*Εικονική Πραγματικότητα*”, Μεταπτυχιακή Εργασία, Ιόνιο Πανεπιστήμιο.
- [2] Δάφνη Οικονόμου (2006). “*Εικονικοί Χώροι*”. Σημειώσεις Μαθήματος για Εικονικό Μουσείο, Πανεπιστήμιο Αιγαίου.
- [3] Ronald Azuma, Yohan Baillet, Reinhold Behringer, Steven Feiner, Simon Julier and Blair MacIntyre (2001). “*Recent Advances in Augmented Reality*”, IEEE Computer Graphics and Applications, Volume 21, Nov/Dec 2001, Issue 6, Pages : 34-47.
- [4] Διμηνιαίο Περιοδικό “SPACEVISION News”, Τεύχος 5, Απρίλιος 2009
- [5] Δήμητρα Μπαντέλα (2007). “*Εικονική Πραγματικότητα και οι Εφαρμογές στην Εκπαίδευση*”, Πτυχιακή Εργασία, Τ.Ε.Ι. Λάρισας.
- [6] Παρασκευή Μπρατσικά και Ερμονέλα Ράμα (2007). “*Εικονική Αίθουσα Κινηματογράφου*”, Πτυχιακή Εργασία, Τ.Ε.Ι. Μεσολογγίου.
- [7] Ιωάννης Γ. Γκέκας (2005). “*Τεχνικές Εκτίμησης Θέσης και Κατεύθυνσης Κινούμενου Στόχου σε Συστήματα Εικονικής Πραγματικότητας*”, Διπλωματική Εργασία, Ε.Μ.Π.
- [8] Χρόνης Κυνηγός (2008). “*Black-and-white-box perspectives to distributed control and constructionism in learning with robotics*”, Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice(Italy) 2008 November, Pages : 1-9
- [9] Γεώργιος Π. Γκλέκας (2005). “*Σχεδιασμός και Κατασκευή Συστήματος Ανίχνευσης Κινητού Στόχου με Υπερήχους για Συστήματα Εικονικής Πραγματικότητας*”, Διπλωματική Εργασία, Ε.Μ.Π.
- [10] Yang-Wai Chow (2009). “*Low-Cost Multiple Degrees-of-Freedom Optical Tracking for 3D Interaction in Head-Mounted Display Virtual Reality*”, International Journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009.
- [11] Ιστοσελίδα WiiBrew : <http://wiibrew.org/wiki/Wiimote>
- [12] Ιστοσελίδα giimote : <http://code.google.com/p/giimote/wiki/Camera>
- [13] Πασχάλης Γονίδης (2007). “*Σχεδιασμός και Υλοποίηση Ηλεκτρονικών Συστημάτων Στερεοσκοπικής Όρασης Μηχανής*”, Μεταπτυχιακή Διατριβή, Δημοκρίτειο Πανεπιστήμιο Θράκης.
- [14] Gary Bradski and Adrian Kaehler (2008). “*Learning OpenCV*”, Chapters 10 & 11, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA.

- [15] Γεώργιος Δ. Ευαγγελίδης (2008). “*Ανάπτυξη Αποδοτικών Παραμετρικών Τεχνικών Αντιστοίχισης Εικόνων με εφαρμογή στην Υπολογιστική Όραση*”, Διδακτορική Διατριβή, Πανεπιστήμιο Πατρών.
- [16] Κωσταντίνος Τζαφέστας (2008). “*Εικονική Πραγματικότητα – Απτικά Συστήματα και Εφαρμογές*”, Σημειώσεις Μεταπτυχιακού Μαθήματος, Ε.Μ.Π.
- [17] Stefan Florczyk (2005). “*Robot Vision : Video-based Indoor Exploration with Autonomous and Mobile Robots*”, Chapter 6, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany.
- [18] Arun K. Dalmia and Mohan Trivedi (1996). “*Depth Extraction Using a Single Moving Camera : an Integration of Depth from Motion and Depth from Stereo*”, Springer-Verlag, Machine Vision and Applications (1996), Vol.9, Pages : 43-55
- [19] Ευριπίδης Πετράκης (2003). “*Image Formation*”, Σημειώσεις Μαθήματος, Πολυτεχνείο Κρήτης.
- [20] Ιστοσελίδα του Quest3D : <http://quest3d.com/>
- [21] Ιστοσελίδα του 3D Studio Max :
<http://usa.autodesk.com/adsk/servlet/pc/index?id=13567410&siteID=123112>
- [22] Ιστοσελίδα της OPTEK : <http://www.optekinc.com/>
- [23] Arnar Birgisson and Bjarki E. Kristjánsson (2008). “*Immersion through Head-tracking*”, Reykjavík University, Iceland.
- [24] David Young (1994). “*Stereoscopic Vision and Perspective Projection*”, Σημειώσεις Μαθήματος, University of Sussex.
- [25] Frederick P. Brooks, Jr. (1999). “*What’s Real About Virtual Reality?*”, IEEE Computer Graphics and Applications, Volume 19, November 1999, Issue 6, Pages : 16 – 27.

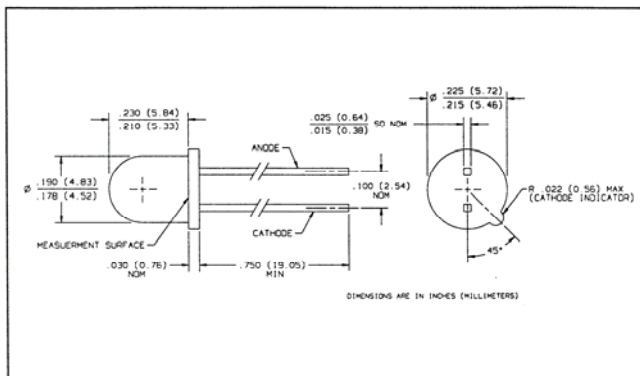
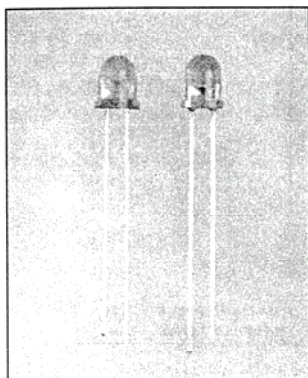
Παράρτημα

Έντυπο τεχνικών δεδομένων (data sheet) της διόδου εκπομπής φωτός OP293



Product Bulletin OP293
June 1996

GaAlAs Plastic Infrared Emitting Diodes Types OP293 and OP298 Series



Features

- Wide irradiance pattern (OP293 series)
- Narrow irradiance pattern (OP298 series)
- Mechanically and spectrally matched to the OP593 and OP598 series phototransistors
- Variety of power ranges
- Significantly higher power output than GaAs at equivalent drive currents
- Wavelength matched to silicon's peak response
- Low cost replacement for TO-46 hermetic package

Description

The OP293 and OP298 series devices are 890nm high intensity gallium aluminum arsenide infrared emitting diodes molded in IR transmissive packages. The broad irradiance pattern of the OP293 series provides relatively even illumination over a large area. The OP298 series is focused with an emission angle of 25°.

Absolute Maximum Ratings (T_A = 25° C unless otherwise noted)

Reverse Voltage	2.0 V
Continuous Forward Current, Free Air	100 mA ⁽²⁾
Continuous Forward Current, Board Mounted	133 mA ⁽³⁾
Continuous Forward Current, Full Heat Sink	200 mA ⁽⁴⁾
Peak Forward Current (25 μs pulse width)	2.0 A
Maximum Duty Cycle (250 μs pulse width, @ 2 A)	5.0%
Storage and Operating Temperature Range	-40° C to +100° C
Lead Soldering Temperature [1/16 inch (1.6 mm) from case for 5 sec with soldering iron]	260° C ⁽¹⁾
Power Dissipation, Free Air	142 mW ⁽²⁾
Power Dissipation, Board Mounted	200 mW ⁽³⁾
Power Dissipation, Full Heat Sink	400 mW ⁽⁴⁾

Notes:

- (1) RMA flux is recommended. Duration can be extended to 10 sec max. when flow soldering. Max. 20 grams force may be applied to the leads when soldering.
- (2) Measured in Free-Air. Derate power dissipation linearly 1.43 mW/° C above 25° C.
- (3) Mounted on 1/16" (1.6 mm) thick PC board with each lead soldered through 80 mil square lands 0.250" (6.35 mm) below flange of device. Derate power dissipation linearly 2.00 mW/° C above 25° C. (Normal Use)
- (4) Immersed in silicone fluid to simulate infinite heat sink. Derate power dissipation linearly 2.50 mW/° C above 25° C.
- (5) Measurement is taken at the end of a single 100 μs pulse. Heating due to increased pulse rate or pulse width will cause a decrease in reading.
- (6) E_{e(APT)} is a measurement of the average apertured radiant energy incident upon a sensing area 0.250" (6.35 mm) in diameter perpendicular to and centered on the mechanical axis of the lens and 0.420" (10.7 mm) from the measurement surface. E_{e(APT)} is not necessarily uniform within the measured area.
- (7) Typical Total Power Out (P_o) @ I_F = 100 mA pulsed on OP293C = 13 mW; OP293B = 18 mW; OP293A = 22 mW.
- (8) E_{e(APT)} is a measurement of the average apertured radiant energy incident upon a sensing area 0.250" (6.5 mm) in diameter perpendicular to and centered on the mechanical axis of the lens and 1.429" (36.30 mm) from the measurement surface. E_{e(APT)} is not necessarily uniform within the measured area.
- (9) For press fit, drill 0.184 ± 0.001" diameter hole.
- (10) This dimension is held to within ± 0.005" on the flange edge and may vary ± 0.020" in the area of the leads.
- (11) Cathode lead is 0.070" nom shorter than anode lead.

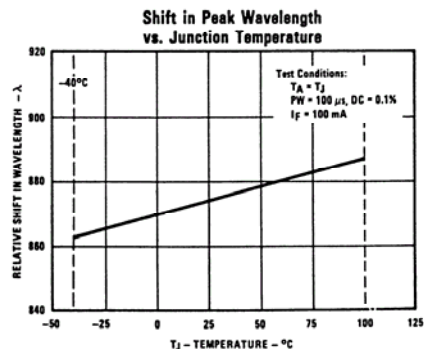
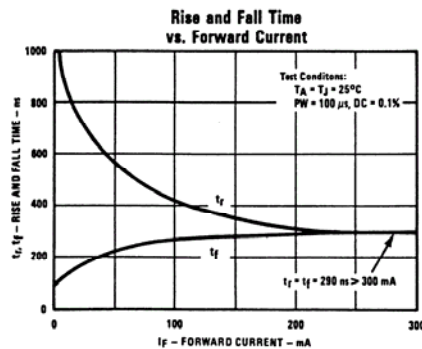
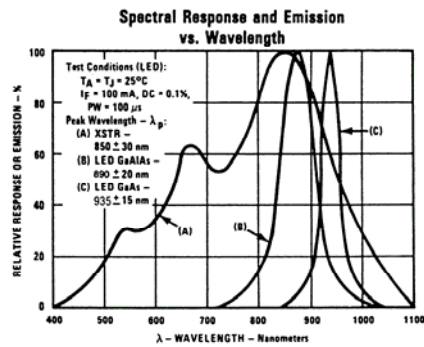
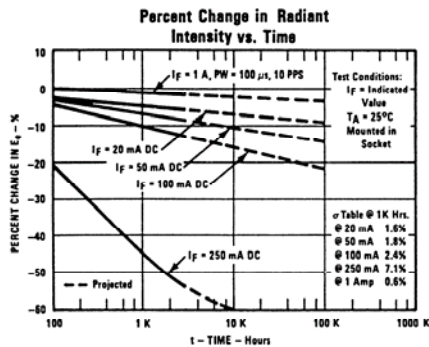
Types OP293 and OP298 Series

Electrical Characteristics ($T_A = 25^\circ\text{C}$ unless otherwise noted)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITIONS
$E_{e(\text{APT})}$	Apertured Radiant Incidence *OP293 is measured with a 30° cone angle at 0.420" (10.67 mm)	OP293C	10	26	mW/cm^2	$I_F = 100\text{ mA}^{(5)(6)(7)}$
		OP293B	13			$I_F = 100\text{ mA}^{(5)(6)(7)}$
		OP293A	16			$I_F = 100\text{ mA}^{(5)(6)(7)}$
	*OP298 is measured with a 10° cone angle at 1.429" (36.30 mm)	OP298C	1.8	4.8	mW/cm^2	$I_F = 100\text{ mA}^{(5)(6)(7)}$
		OP298B	2.4			$I_F = 100\text{ mA}^{(5)(6)(7)}$
		OP298A	3.0			$I_F = 100\text{ mA}^{(5)(6)(7)}$
V_F	Forward Voltage			2.0	V	$I_F = 1.50\text{ A}^{(5)}$
I_R	Reverse Current			100	μA	$V_R = 2\text{ V}$
λ_p	Wavelength at Peak Emission		890		nm	$I_F = 10\text{ mA}$
B	Spectral Bandwidth Between Half Power Points		80		nm	$I_F = 10\text{ mA}$
$\Delta\lambda_p/\Delta T$	Spectral Shift with Temperature		+0.18		$\text{nm}/^\circ\text{C}$	$I_F = \text{Constant}$
θ_{HP}	Emission Angle at Half Power Points	OP293	60		Deg.	$I_F = 20\text{ mA}$
		OP298	25		Deg.	$I_F = 20\text{ mA}$
t_r	Output Rise Time		500		ns	$I_{F(\text{PK})} = 100\text{ mA}$, $\text{PW} = 10\text{ }\mu\text{s}$, D.C. = 10%
t_f	Output Fall Time		250		ns	

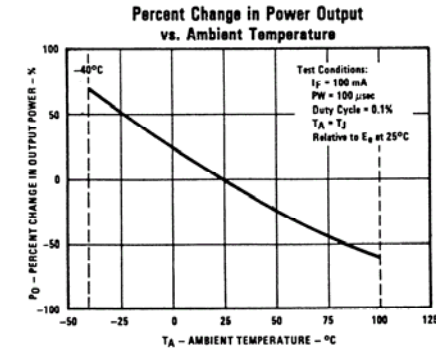
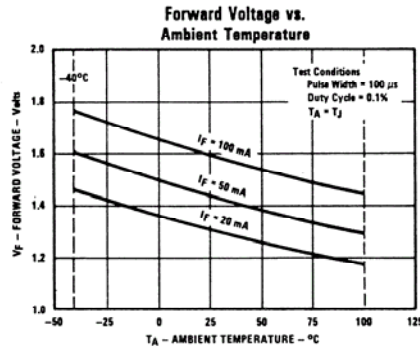
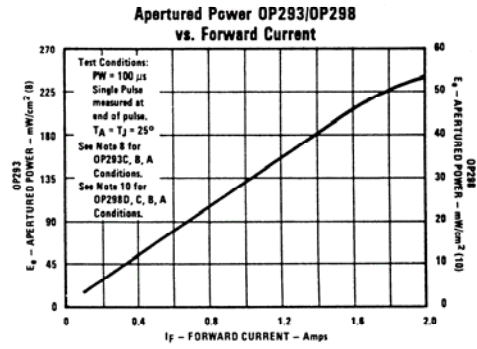
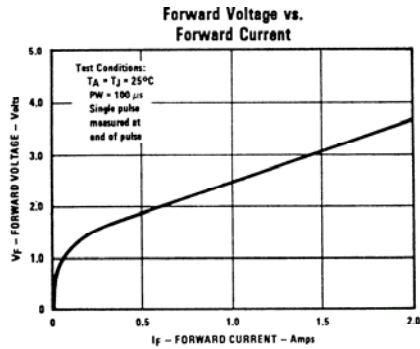
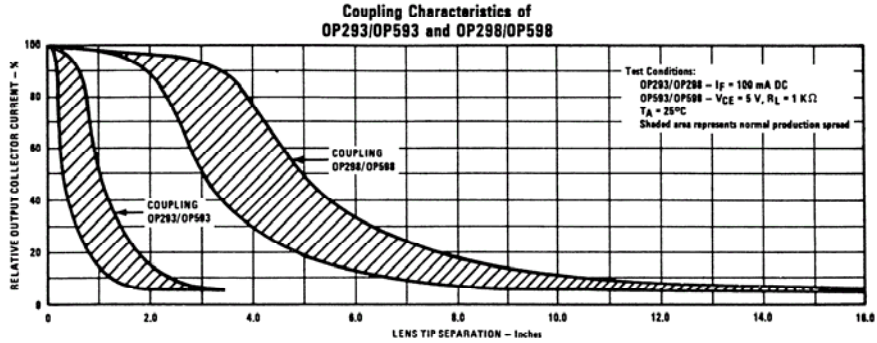
INFRARED EMITTING DIODES

Typical Performance Curves



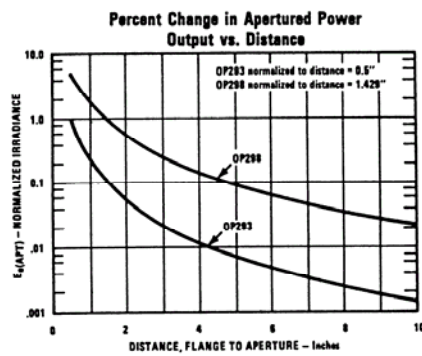
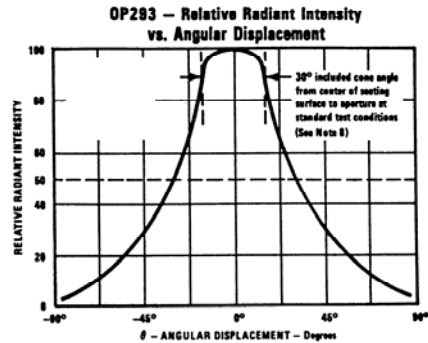
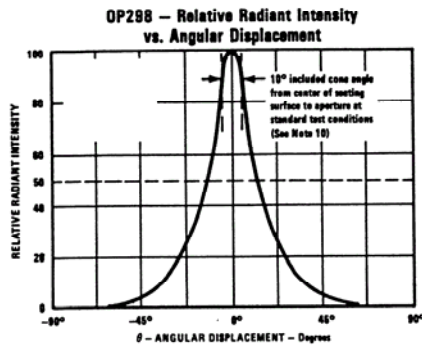
Optek reserves the right to make changes at any time in order to improve design and to supply the best product possible.
 Optek Technology, Inc. 1215 W. Crosby Road Carrollton, Texas 75006 (972)323-2200 Fax (972)323-2396
 2-53

Typical Performance Curves



Types OP293 and OP298 Series

Typical Performance Curves



Thermal Parameters

Type Units	RTHJA (°C/W)			C _{TH} (10 ⁻⁶ W/°C)	T _{TH} (10 ⁻² s)	K
	Free Air ⁽¹⁾	Normal ⁽²⁾	Infinite Heat Sink ⁽³⁾			
AN	700	500	250	4.0	1.5	0.008

Notes:

- (1) Heat transfer minimized by holding unit in still air with minimum heat transferred through leads by conduction.
- (2) Unit mounted in double sided printed circuit board 0.250 inches (6.35 mm) below plastic. The land areas are 0.080 inches square. This simulates normal use.
- (3) Unit immersed in circulating silicone fluid holding T_{case} @ 25°C. This simulates an infinite heat sink.

Refer to Application Bulletin 200 for use of these constants.

INFRARED
EMITTING
DIODES

Optek reserves the right to make changes at any time in order to improve design and to supply the best product possible.
Optek Technology, Inc. 1215 W. Crosby Road Carrollton, Texas 75006 (972)323-2200 Fax (972)323-2396