

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΧΕΔΙΟ ΠΑΡΟΥΣΙΑΣΗΣ
ΑΤΟΜΙΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ατομική Διπλωματική Εργασία

**RECONSTRUCTION OF EVERYDAY LIFE IN
19TH CENTURY NICOSIA**

Χριστιάνα Αγάθωνος

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Reconstruction Of Everyday Life In 19th Century Nicosia

Χριστιάνα Αγάθωνος

Επιβλέπων Καθηγητής
Γιώργος Χρυσάνθου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος
Πληροφορικής του Πανεπιστημίου Κύπρου

Ευχαριστίες

Καταρχάς θα ήθελα να ευχαριστήσω τον καθηγητή μου Δρ. Γιώργο Χρυσάνθου για τη βοήθεια που μου έχει προσφέρει, τη συμπαράσταση και κατανόηση την οποία μου έχει επιδείξει. Επίσης θα ήθελα να τον ευχαριστήσω και για την ευκαιρία που μου έχει δώσει να εργαστώ σε ένα τέτοιο έργο, το οποίο μου χάρισε γνώσεις στο πεδίο των γραφικών των υπολογιστών επεκτείνοντάς τες και αποκτώντας εμπειρία στο συγκεκριμένο θέμα.

Είμαι επίσης ευγνώμων στο διδακτορικό ερευνητικό βοηθό Παναγιώτη Χαραλάμπους για τη πολύτιμη βοήθειά και καθοδήγησή του σε θέματα που χρειάζονται βοήθεια, για διάφορες πηγές που μου έχει προτείνει οι οποίες θα με βοηθούσαν μετέπειτα στη μελέτη μου στο θέμα. Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στη συμφοιτήτρια μου Μαρία Χριστοφή για τη πολύτιμη βοήθεια και στήριξη της παρόλη τη δουλειά που είχε η ίδια να επιτελέσει.

Θα ήθελα να αφιερώσω τη διπλωματική μου εργασία στους γονείς και την αδελφή μου για την υπομονή και συμπαράσταση που έχουν δείξει λόγω της συνεχούς απουσίας μου από το σπίτι, καθώς επίσης στους φίλους μου Γιώργο, Κυριάκο και Κωνσταντίνο για τη στήριξη, τη συμπαράσταση, την αγάπη που μου έχουν δείξει, καθώς και για το ότι με είχαν ενθαρρύνει να συνεχίσω τη δουλειά μου επιτυχαίνοντας τους στόχους μου.

Περίληψη

Στόχος αυτού του έργου ήταν να επεκτείνει προηγούμενη δουλειά που είχε γίνει στο πανεπιστήμιό μας δίνοντας ζωή και περισσότερη ρεαλιστικότητα στο έργο. Ο γενικότερος σκοπός αυτού του έργου είναι να παρουσιάσει την καθημερινή ζωή της πόλης της Λευκωσίας κατά τη διάρκεια του 19^{ου} αιώνα με τις διαφορετικές κοινό-οικονομικές ταυτότητες των ανθρώπων της.

Η προηγούμενη δουλειά που είχε γίνει αφορούσε τη γειτονία της Χρυσαινιώτισσας, η οποία είναι κτισμένη με βάση την αρχιτεκτονική της εποχής εκείνης, δίνοντάς της “ζωή” προσθέτοντας ρεαλιστικές φιγούρες πολιτών να κινούνται μέσα στη πόλη. Σε κάθε αυτόνομο χαρακτήρα έχει ανατεθεί μια λίστα από δραστηριότητες τις οποίες πρέπει να υλοποιήσει, για παράδειγμα να κινηθεί προς ένα συγκεκριμένο μέρος, να μιλήσει με άλλους ανθρώπους, να χορέψει, να πάει στη αγορά για ψώνια κλπ.

Η δική μου δουλειά στο project αφορά την προσθήκη ζώων στη πόλη δίνοντάς της περισσότερη ζωή και ρεαλισμό. Άλογα περιπλανιούνται στη αγορά, πουλιά πετούν πάνω από τα σπίτια και την αγορά, κοτούλες αλληλεπιδρούν με τα ζώα γύρω τους. Φωτογραφίες και ζωγραφίες από την εποχή εκείνη μας δίνουν πληροφορίες σχετικά με τα ζώα των ανθρώπων και τη συμβολή τους και εγώ με τη σειρά μου προσπάθησα να το επιτύχω με όσο πιο ρεαλιστικό τρόπο ήταν δυνατό με βάση τα μέσα και τις φιγούρες τις οποίες είχα στη διάθεση μου. Σε κάθε μοντέλο ανατίθενται διάφορες λειτουργίες τις οποίες θα εκτελέσει, όπως να περπατήσει, να πετάξει αν είναι πουλί, να τραφεί από το έδαφος. Για την υλοποίηση όλων αυτών έχω χρησιμοποιήσει το Unity 4 και για την εκτέλεση των εργασιών για κάθε χαρακτήρα το Unity’s Mecanim Animation System.

Περιεχόμενα

Ευχαρισστες.....	4
Περίληψη.....	5
Κεφάλαιο 1 Εισαγωγή.....	8
1.1 Γενικά	8
1.2 Κίνητρο	9
1.3 Στόχος	10
 Κεφάλαιο 2 Προηγούμενη δουλειά στο project.....	 15
2.1 Modeling the Walled City of Nicosia	15
2.2 Reconstruction of Everyday Life in 19th Century Nicosia	16
2.3 Τι είναι το Crowd Simulation	18
2.3.1 Κατηγορίες Κίνησης	20
2.4 Εισαγωγή ζωής στη πόλη	21
2.4.1 Screenshots από την έξοδο του μοντέλου (Πόλη και Αγορά)	23
 Κεφάλαιο 3 Γενική Εικόνα	 25
3.1 Γενικός σκοπός και τα επιμέρους συστατικά του έργου	25
 Κεφάλαιο 4 Ανάλυση Συστατικών του Έργου.....	 27
4.1 Unity	27
4.1.1 Rendering	28
4.1.2 Scripting	29
4.1.3 Physics	29
4.2 Mecanim	30
4.2.1 Retarget Animations	30
4.2.2 Blend Trees and State Machines	31
4.3 Collision Avoidance	32
4.4 Τα 3D Models του Project	32

Κεφάλαιο 5	Υλοποίηση.....	35
5.1	Αρχικές Προσπάθειες	36
5.2	Mecanim (Animation Controller)	39
5.2.1	Mecanims' Tutorial – Robot	39
5.3	Χρήση Mecanim, επικοινωνία και συμπεριφορά των μοντέλων	41
5.4	Collision Avoidance	50
5.5	Tree Creator	52
Κεφάλαιο 6	Συμπεράσματα	54
6.1	Αποτελέσματα	54
6.2	Εντυπώσεις από χρήστες	64
6.3	Μελλοντική δουλειά	64
Βιβλιογραφία		65

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά	8
1.2 Κίνητρο	9
1.3 Στόχος	10

1.1 Γενικά

Παντού γύρο μας μέσα στη καθημερινότητα μας βλέπουμε ανθρώπους να κινούνται, να περιμένουν το λεωφορείο, να οδηγούν τα αυτοκίνητά τους, φίλους να συναντιόνται, να πηγαίνουν για φαγητό, να κάνουν διάφορα αθλήματα και πολλές άλλες δραστηριότητες, οι οποίες περιλαμβάνουν αλληλεπίδραση του ενός με τον άλλο. Ένα μεγάλο κομμάτι της ζωής μας αφορά και την ύπαρξη ζώων στη καθημερινότητά μας. Κατοικίδια, όπως σκύλοι, γάτες, πουλιά, ψάρια ή άλλα ζώα, τα οποία χρησιμοποιούμε για την παροχή τροφής, όπως κότες και κουνέλια υπάρχουν μέχρι και σήμερα στα σπίτια μας. Σε παλαιότερες εποχές, στις οποίες δεν υπήρχαν τα μέσα μαζικής μεταφοράς, οι άνθρωποι διακινούνταν και μετέφεραν τις προμήθειές τους με τη βοήθεια αλόγων, γαϊδουριών και καμήλων.

Αυτή είναι και η πρόκληση την οποία είχα να αντιμετωπίσω για να επιτευχθεί η διατριβή μου μέσω αυτού του έργου. Δηλ., να προσθέσω ζώα που αφορούσαν την εποχή εκείνη όπως για παράδειγμα άλογα, κότες, διάφορα άλλα πουλιά όπως σπουργίτια και κοράκια, τα οποία θα εκτελούν κάποιες λειτουργίες και θα αλληλεπιδρούν με τους ανθρώπους οι οποίοι κυκλοφορούν και ενεργούν φυσιολογικά μέσα στους δρόμους της πόλης.

Για να δώσουμε περισσότερο ρεαλισμό στη πόλη έχω δημιουργήσει δέντρα με τη χρήση γραφικών και του Unity 4.3.2. Πιο συγκεκριμένα θα δούμε οπορωφόρα δέντρα όπως συκιές, δέντρα που χάριζαν σκιά, όπως τα πεύκα. Θα δούμε επίσης φοινικίες και κάποια άλλα φυλλοβόλα δέντρα.

1.2 Κίνητρο

Τα 3D γραφικά υπολογιστών είναι κατά τη γνώμη μου ένας από τους πιο ενδιαφέρον τομείς που έχει να επιδείξει ο κλάδος της Πληροφορικής. Δεν θα μπορούσα να μην πάρω την ευκαιρία και να εργαστώ σε ένα τέτοιο έργο, να συνεργαστώ με την ομάδα του κ. Χρυσάνθου και να πάρω γνώσεις στο τομέα των γραφικών. Αυτό ήταν κάτι το άγνωστο σε εμένα προηγουμένως. Είναι ένας τομέας με πολλές προοπτικές, αφού για τα πάντα πλέον γύρο μας θα υπάρχουν γραφικές αναπαραστάσεις, και φυσικά θα μου έδινε την ευκαιρία να επιλέξω να ακολουθήσω αυτό τον τομέα σε μεταγενέστερες σπουδές μου, αλλά και για επαγγελματικούς σκοπούς.

Κυρίως όμως ο λόγος που έχει τραβήξει την προσοχή μου ο τομέας των γραφικών και με οδήγησε στο να επιλέξω η διπλωματική μου εργασία να αφορά σ' αυτό το θέμα, είναι το “όμορφο και καλλιτεχνικό” κομμάτι των γραφικών, και το πόσο ενδιαφέρον μπορεί να γίνει η μελέτη τους.

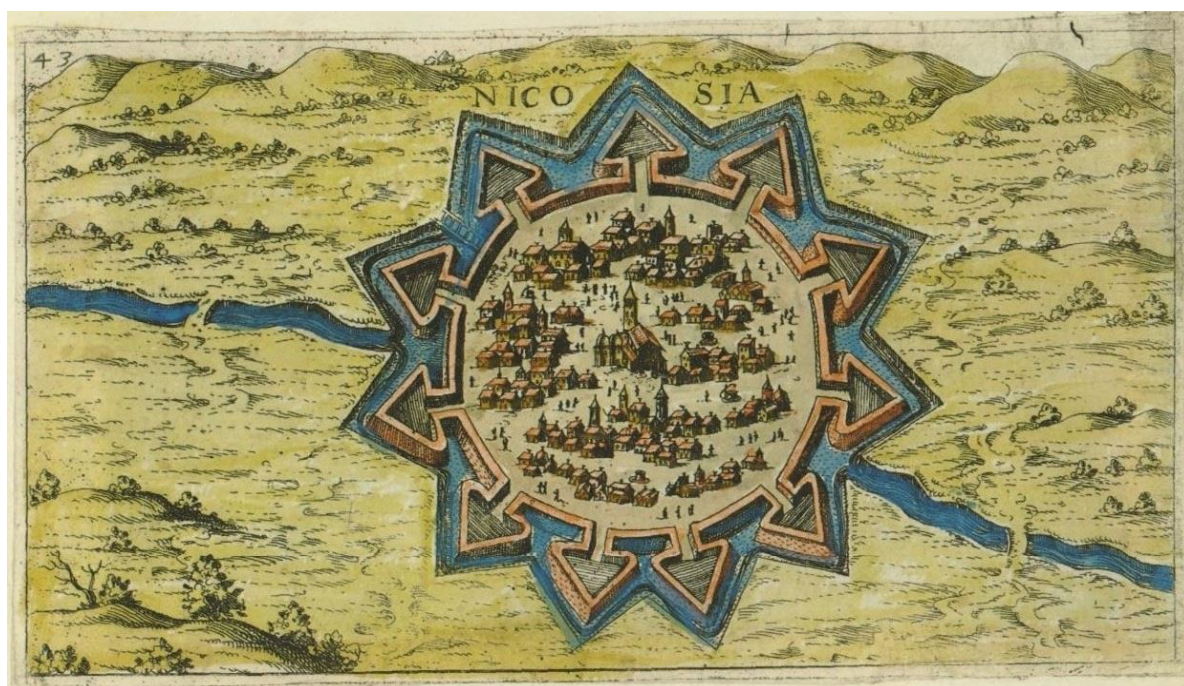
1.3 Στόχος

Στόχος αυτού του έργου είναι να παρουσιάσει και να απεικονίσει με μια μοναδική ματιά την καθημερινή ζωή της πόλης της Λευκωσίας όσον αφορά την κοινωνικό-οικονομική πτυχή του λαού της, την χωρική οργάνωση των κτιρίων εντός των τειχών και την συμμετοχή των ζώων στη ζωή τους κατά τη διάρκεια του 19ου αιώνα.

Το τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου σε συνεργασία με το Λεβέντειο Δημοτικό Μουσείο Λευκωσίας, είχαν σκοπό να δημιουργήσουν ένα εργαλείο που θα μπορούσε να χρησιμοποιηθεί από αρχιτέκτονες, ιστορικούς και αρχαιολόγους για να μπορούν να μελετούν τη χωρική οργάνωση των κτιρίων μέσα στα τείχη της πόλης.

Επίσης θα χρησίμευε και για εκπαιδευτικούς σκοπούς, αφού οι μαθητές θα μπορούσαν μέσω της πλοήγησης τους στην οθόνη να διδαχτούν την ιστορία της πόλης της Λευκωσίας, τις καθημερινές εργασίες και ασχολίες των ανθρώπων της μέσω της αλληλεπίδρασης μεταξύ τους και των ζώων τους.

Εκείνη την εποχή σήμανε το τέλος της Οθωμανικής αυτοκρατορίας και οι Χριστιανοί Ορθόδοξοι συνυπήρχαν με τους Μουσουλμάνους και άλλες μειονότητες. Μέσω της προηγούμενης δουλειάς που έγινε στο έργο θα παρατηρήσουμε τους διαφορετικούς ρόλους και συμπεριφορές των κατοίκων.



Σχήμα 1.1: Λευκωσία μέσα στα τείχη.

Χρησιμοποιώντας τις πληροφορίες που δόθηκαν από το μουσείο και το κτηματολόγιο (παλιές φωτογραφίες και γραπτά έγγραφα) παρατηρούμε πως οι κάτοικοι παρουσιάζονται, από το προηγούμενο έργο, ντυμένοι σύμφωνα με τη θρησκεία, την κοινωνική κατάσταση και το επάγγελμά τους.



Σχήμα 1.2: Η αγορά στη Λευκωσία τον 19^ο αιώνα (<http://reconlife.cs.ucy.ac.cy/>).

Δικός μου στόχος στη συνέχιση του έργου ήταν να προσθέσω ζώα της εποχής εκείνης, όπως για παράδειγμα άλογα, κόττες, διάφορα άλλα πουλιά, όπως σπουργίτια και κοράκια. Όλα θα εκτελούν κάποιες λειτουργίες και θα αλληλεπιδρούν με τους ανθρώπους, οι οποίοι κυκλοφορούν και ενεργούν φυσιολογικά μέσα στους δρόμους της πόλης.

Για τα μοντέλα των ζώων και των πουλιών χρησιμοποίησα τις έτοιμες και όσο το δυνατό ρεαλιστικότερες φιγούρες από το Unity Asset Store.

Έχω δημιουργήσει μια προσομοίωση η οποία αναθέτει σε κάθε αυτόνομο μοντέλο ζώου μια λίστα από εργασίες και κινήσεις. Τα άλογα για παράδειγμα περπατούν μέσα στο χώρο και

αλληλεπιδρούν με τους ανθρώπους, με άλλα ζώα και με αντικείμενα γύρω τους. Εκτελούν διάφορες κινήσεις που χαρακτηρίζουν τα ίδια τα άλογα, όπως το να κοιτάζουν γύρω μετακινώντας ολόκληρο το λαιμό και την κεφαλή τους, να σηκώνονται στα 2 πόδια, να τρώνε από το έδαφος, και πολλά άλλα.

Παράλληλα στο χώρο υπάρχουν και αλληλεπιδρούν με το περιβάλλον κότες και κόκορες. Σαν αρχική και φυσιολογική κίνηση οι κότες κοιτάζουν γύρω το περιβάλλον ή τρώνε, και σε περίπτωση επαφής με άνθρωπο περπατούν μακριά.

Για κάθε ενέργεια των μοντέλων παίζει το ανάλογο animation και απαλά αλλάζει το ένα με το άλλο.

Αρκετή ομορφιά και ρεαλισμό στο περιβάλλον δίνουν τα πουλιά που πετούν μαζί ως ένα σμήνος πάνω από τα τείχη και τους ανθρώπους όπως επίσης και τα κοράκια τα οποία κινούνται κοντά στο πηγάδι.

Για να δώσω περισσότερο ρεαλισμό στην πόλη έχω δημιουργήσει δέντρα με τη χρήση γραφικών και του Unity 4.3.2. Πιο συγκεκριμένα θα δούμε οπορωφόρα δέντρα όπως συκιές, δέντρα που χάριζαν σκιά όπως τα κυπαρίσσια, οι φοινικιές, όπως επίσης και κάποια άλλα φυλλοβόλα.

Σχήμα 1.3: Ζώα στο περιβάλλον



Κεφάλαιο 2

Προηγούμενη δουλειά στο project

2.1 Modeling the Walled City of Nicosia	15
2.2 Reconstruction of Everyday Life in 19th Century Nicosia	16
2.3 What is Crowd Simulation?	18
2.3.1 Κατηγορίες Κίνησης	20
2.4 Εισαγωγή ζωής στη πόλη	21
2.4.1 Στιγμιότυπα από την έξοδο του μοντέλου (Πόλη και Αγορά)	23

2.1 Modeling the Walled City of Nicosia

Στο δεύτερο κεφάλαιο θα αναλύσω και θα παρουσιάσω τη δουλειά που έχει γίνει σε προηγούμενα projects στο ίδιο θέμα.

Το 2003 έχουν προσπαθήσει να διαμορφώσουν τη συνοικία της Χρυσαινιώτισας στη Λευκωσία, με τη δημιουργία ενός συστήματος μερικώς αυτοματοποιημένης με τη χρήση των δεδομένων GIS της περιοχής ως είσοδο στο σύστημα και δομική ανάλυση των κτιρίων της περιοχής. Για το έργο αυτό, απαιτήθηκε μια επίσημη τυπολογία των κτιρίων, καθώς και η κάτοψη της συγκεκριμένης περιοχής για την οπτικοποίηση σε 2D.

Η ομάδα είχε αναπτύξει μερικά συστατικά όπως παράθυρα, πόρτες, κιόσκια, καμάρες, οροφές και μπαλκόνια και είχαν προσπαθήσει να τα συνδυάσουν κατασκευάζοντας έτσι τα 3D κτίρια.

Στη συνέχεια μετά την ανάγνωση και την ανάλυση των αρχείων εισόδου GIS, άρχισαν να αναγνωρίζονται οι δρόμοι και κάποια χαρακτηριστικά κτιρίων όπως γειτονιές και οι ακμές των δρόμων.

Τα σπίτια είχαν κατηγοριοποιηθεί σε τέσσερις κύριες κατηγορίες, Original Courtyard, Minimal Courtyard, Planned Serial και New Courtyard. Άλλα είδη κτιρίων είχαν δημιουργηθεί αυτόματα με rule-based τεχνικές. Όλη αυτή η καλή δουλειά που είχε γίνει

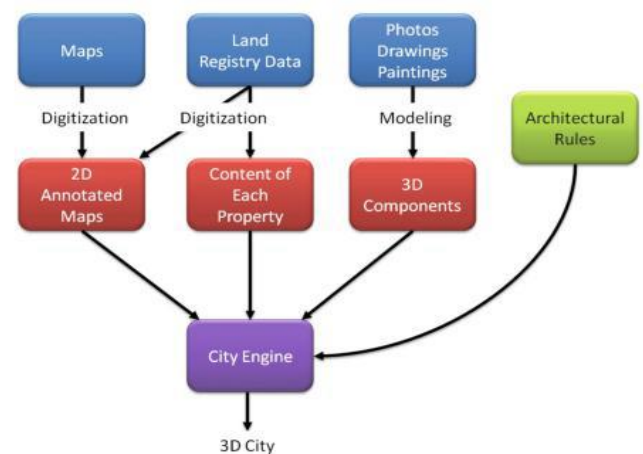
οδήγησε στη παραγωγή πολύ καλών 3D μοντέλων και η έξοδος του προγράμματος είναι πολύ ρεαλιστική.



Σχήμα 2.1: Στιγμιότυπο από το εξωτερικό του μοντέλου (Μαριάννα Δικαιάκου - Modelling the Walled City of Nicosia).

2.2 Reconstruction of Everyday Life in 19th Century Nicosia

Αυτό το έργο περιλαμβάνει την ανάπλαση του αστικού περιβάλλοντος της πόλης βασισμένο σε ιστορικές πληροφορίες και αρχεία του Κτηματολογίου από την Οθωμανική εποχή που λήφθηκαν από τους Βρετανούς μηχανικούς.



Σχήμα 2.2: The reconstruction process

Για την δημιουργία της 3D αναπαράστασης της αρχιτεκτονικής των κτιρίων εντός των τειχών πόλης ελήφθησαν υπόψη τα αρχεία του Κτηματολογίου στη Λευκωσία, η χρήση του κάθε δωματίου, επιπλέον χαρακτηριστικά (δέντρα, γεωτρήσεις), η εθνικότητα και το επάγγελμα του ιδιοκτήτη, έντυποι χάρτες, μερικές φωτογραφίες, σχέδια και πίνακες.

Ένα σύνολο κανόνων και 3D στοιχείων όπως παράθυρα, πόρτες και αψίδες πέρασαν σε ένα διαδικαστικό σύστημα μοντελοποίησης (CityEngine) για να δημιουργήσει με τη σειρά του το μοντέλο της πόλης με 3D μοντέλα για κάθε σπίτι.

Ορίστηκαν κανόνες για διαφορετικούς τύπους κτιρίων (σπίτι με ένα ή δύο ορόφους, κατάστημα): οι κατασκευές δημιουργούνται πρώτα επί της γραμμής του δρόμου, χώροι διαμορφωμένοι για ζωή πήραν ένα ή δύο ορόφους, ενώ τα δωμάτια των καταστημάτων, οι στάβλοι, οι κουζίνες παίρνουν τους πίσω εσωτερικούς χώρους. Επίσης, όλα τα κτίρια, εκτός από τα καταστήματα, οργανώνονται γύρω από μια εσωτερική ανοικτή αυλή, στον πρώτο όροφο υπάρχουν μικρότερα παράθυρα απ' ότι στους επάνω ορόφους και οι πόρτες βρίσκονται η κεντρική στην πρόσοψη του σπιτιού επί του δρόμου.



*Σχήμα 2.3: Στιγμιότυπο από το έργο. Απεικόνιση σπιτιών και της διάταξής τους
(<http://reconlife.cs.ucy.ac.cy/>).*

2.3 What is Crowd Simulation?

Ξεκινώντας, πρέπει να αναλύσουμε κάποιους όρους οι οποίοι θα χρησιμοποιηθούν μέσα από αυτό το έργο. Ο όρος πλήθος αναφέρεται σε ένα σύνολο πραγμάτων, ανθρώπων ή ζώων που συγκεντρώνονται μεταξύ τους ο ένας δίπλα από τον άλλο. Τα μέλη ενός πλήθους μοιράζονται μια εικόνα, μια φήμη και ενεργούν στο χώρο με διάφορες συμπεριφορές και ενέργειες.

Το *Crowd Simulation* είναι η διαδικασία της προσομοίωσης ενός αριθμού ανθρώπων, οι μάζες από πλάσματα ή άλλους χαρακτήρες που αλληλεπιδρούν μαζί στο ίδιο περιβάλλον. Κάθε άτομο (που ονομάζεται actor, ή agent) έχει τους στόχους του, ανταποκρίνεται σε

άλλους και αλληλεπιδρά με το περιβάλλον. Το Crowd Simulation έχει εφαρμογές σε πολλές κατηγορίες όπως για παράδειγμα σε αρχιτεκτονικούς σχεδιασμούς για τη βελτίωση κάποιων περιβαλλόντων κατάρτισης οπτικά και για εικονικές πραγματικότητες με τρόπο τεχνητό - ευφυείς χαρακτήρες σε ταινίες και παιχνίδια (π.χ. σε όλες τις ταινίες The Lord of the Rings όπου υπήρχαν στρατιές χιλιάδων χαρακτήρων να μάχονται ο ένας τον άλλο, για απεικόνιση αεροδρομίου). Το Crowd Simulation χρησιμοποιείται συχνά σε 3D γραφικά υπολογιστών και ασχολείται με πολλά θέματα σχετικά με έρευνα, όπως εύρεση μιας διαδρομής σε πραγματικό χρόνο, το σχεδιασμό κίνησης και αποφυγής συγκρούσεων.



Σχήμα 2.4: Παράδειγμα πεδίου μάχης από ταινίες (εικ. από Google-Crowd Simulation Unity).



Σχήμα 2.5: Παράδειγμα πεδίου μάχης από παιχνίδι (εικ. από Google-Crowd Simulation Unity).

Τα computer animations και η προσομοίωση εφαρμογών μπορούν να χρησιμοποιηθούν για ψυχαγωγικούς και εκπαιδευτικούς σκοπούς, για εξάσκηση και γυμναστική, καθώς και για

ανάλυση του ανθρώπινου παράγοντα για την εκκένωση κτιρίων. Πολλά σενάρια εφαρμογών προσομοιώνουν πλήθη που συγκεντρώνονται ή διαλύονται, που δουλεύουν, παίζουν, έχουν κάποιους στόχους και κάποιες εργασίες που πρέπει να επιτευχθούν. Αυτές οι προσομοιώσεις μπορεί να συμβούν σε μέρη όπως εκδηλώσεις (κοινωνικές, αθλητικές, εκπαιδευτικές), συναυλίες, στους δρόμους της πόλης, σε σχολεία, στο χώρο εργασίας, σε πάρκα, σε ένα σπίτι.



Σχήμα 2.6: Κόσμος σε εσωτερικό και εξωτερικό περιβάλλον (εικ. από Google-Crowd Simulation Unity).

2.3.1 Κατηγορίες κίνησης

Υπάρχουν δύο κατηγορίες για την προσομοίωση της κίνησης και των αλληλεπιδράσεων. Η πρώτη ονομάζεται *Particle motion* και αναφέρεται στην κίνηση σημειακών σωματιδίων, τα οποία παίρνουν κίνηση από την προσομοίωση δυνάμεων, όπως η βαρύτητα, ο άνεμος, σημεία από συγκρούσεις.

Το απλούστερο μηχανικό σύστημα που μπορεί να κινηθεί είναι ένα σωματίδιο. Ένα σωματίδιο έχει μάζα, αλλά όχι όγκο: σημειακή μάζα. Τα σωματίδια είναι τα δομικά στοιχεία ενός στερεού σώματος (Rigid Body). Οι εξισώσεις για ένα σωματίδιο χρησιμοποιούνται για την ανάπτυξη ενός άκαμπτου σώματος.

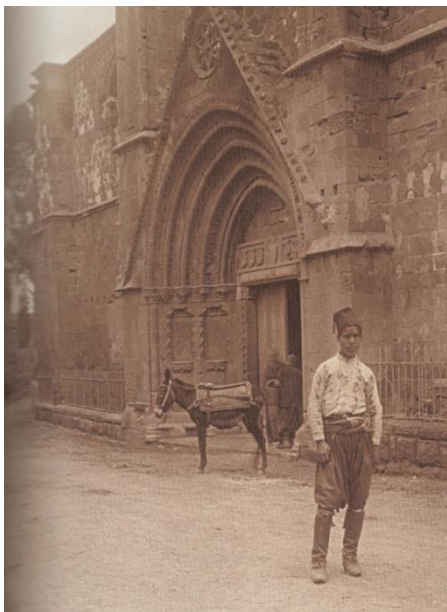
Η δεύτερη ονομάζεται *Crowd Artificial Intelligence*. Η τεχνητή νοημοσύνη είναι η παρουσίαση και αναπαράσταση της ανθρώπινης νοημοσύνης μέσα από μηχανές και λογισμικό. Στη κατηγορία crowd artificial intelligence οι παράγοντες έχουν συγκεκριμένους

στόχους και αλληλεπιδρούν με τους άλλους, όπως κάνουν οι άνθρωποι σε πραγματικά πλήθη. Μερικές φορές οι παράγοντες μπορούν να ανταποκριθούν στις περιβαλλοντικές αλλαγές και το αποτέλεσμα είναι πολύ πιο ρεαλιστικό από τη κίνηση των σωματιδίων (Particle motion), αλλά και πιο δύσκολο να εφαρμοστεί.

Το δικό μου έργο είναι παρόμοιο με το προηγούμενο σε αρκετά σημεία αφού το κάθε ζώο έχει διαφορετική συμπεριφορά, όπως οι άνθρωποι στην αγορά· συμπεριφέρονται ανάλογα με τις αντιδράσεις και συμπεριφορές των ανθρώπων γύρω τους.

2.4 Εισαγωγή ζωής στη πόλη

Στόχος της συμφοιτήτριας μου Καλυψώς Δαυίδ ήταν να δώσει ζωή στο project προσθέτοντας ανθρώπινα μοντέλα που θα έμοιαζαν με τους κατοίκους της πόλης, θα ενεργούσαν και θα συμπεριφέρονταν φυσιολογικά σύμφωνα με τον τρόπο ζωής και συμπεριφοράς των ανθρώπων της πόλης την εποχή εκείνη, βάση των όσων είναι γραμμένα σε ιστορικά βιβλία, φωτογραφίες και σχέδια.



Σχήμα 2.7: Άνθρωποι τον 19^ο αιώνα (Καλυψώ Δαυίδ, *Reconstruction of Everyday Life in 19th Century Nicosia*).

Οι άνθρωποι έχουν κάποιους σκοπούς και στόχους να επιτύχουν και όχι απλά να περιπλανιόνται μέσα στους δρόμους της πόλης. Για αυτό το λόγο εκτελούν ρεαλιστικά animations, κάποιου είδους συμπεριφορά για αποφυγή συγκρούσεων και ακολουθούν ένα αλγόριθμο για να βρύνε το συντομότερο μονοπάτι προς τον προορισμό τους. Για να

επιτευχθεί αυτό παράγεται ένας γράφος από το χάρτη της πόλης έτσι ώστε να βρίσκουν το δρόμο τους μέσα στη πόλη.

Με τη χρήση του Mecanim (το Animation system του Unity), ανατέθηκαν animations στους χαρακτήρες τα οποία αλλάζουν το ένα μετά το άλλο απαλά και δείχνουν απόλυτα φυσικά. Δημιουργείται ένα state machine και η μετάβαση στο επόμενο animation clip συμβαίνει κάτω υπό κάποιες συνθήκες και συγκεκριμένους παραμέτρους. Η high-level συμπεριφορά καθορίζει το πώς θα πράξουν οι ανθρώπινοι παράγοντες.

Οι τίτλοι ιδιοκτησίας που έχουν παραχωρηθεί από το τμήμα Κτηματολογίου στη Λευκωσία, μαζί με την πρώτη έρευνα της γης που διεξήχθη στην πόλη με επικεφαλής τον Λόρδο Kitchener το 1880 (Σχ. 2.8(a)) χρησιμοποιήθηκαν για να δημιουργήσουν ένα ψηφιακό χάρτη (Σχ. 2.8(b)) της πόλης όπως ήταν τον 19^ο αιώνα.



Σχήμα 2.8: (a) Χάρτης Κίτσεενερ από το 1880, αντιπροσωπεύει ένα περίγραμμα των δομημένων τόμων στην πόλη. (b) 2D ψηφιακός χάρτης της Taht el Kale Mahalla, περιέχει την αρχιτεκτονικά ανύψωση του εδάφους των κτιρίων σε σχέση με το Κτηματολογίου και άλλες πληροφορίες.

Αυτές οι πληροφορίες ήταν χρήσιμες όσο αφορά και το πως χρησίμευε το κάθε δωμάτιο στο σπίτι. Για παράδειγμα αν ήταν κουζίνα, καθιστικό, κατάστημα κλπ. Επίσης έχουμε πληροφορίες για το είδος των δέντρων, εξωτερικούς φούρνους και πηγάδια, την εθνικότητα και το επάγγελμα του ιδιοκτήτη κάθε σπιτιού.



Σχήμα 2.9: Φωτογραφία από το εξωτερικό της πόλης.

2.4.1 Στιγμιότυπα από την έξοδο του μοντέλου (Πόλη και Αγορά)

Σχήμα 2.8.1: Το μοντέλο του εσωτερικού χώρου της πόλης, αγορά (Καλυψώ Δαβίδ, *Reconstruction of Everyday Life in 19th Century Nicosia*).



Σχήμα 2.8.2: Το μοντέλο του εσωτερικού χώρου της πόλης, αγορά (Καλυψώ Δαβίδ, *Reconstruction of Everyday Life in 19th Century Nicosia*).



Σχήμα 2.8.3: Ανθρώπινες φιγούρες, κίνηση και συμπεριφορά (Καλυψώ Δαβίδ, *Reconstruction of Everyday Life in 19th Century Nicosia*).



Σχήμα 2.8.4: Η πόλη από ψηλά (Καλυψώ Δαβίδ, *Reconstruction of Everyday Life in 19th Century Nicosia*).



Σχήμα 2.8.5: Η πόλη από ψηλά (Καλυψώ Δαβίδ, *Reconstruction of Everyday Life in 19th Century Nicosia*).



Κεφάλαιο 3

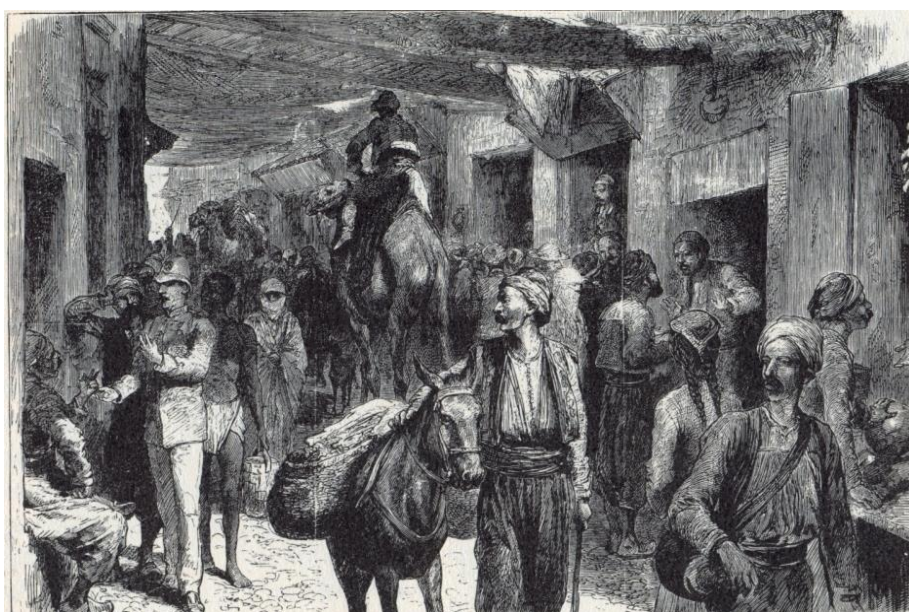
Γενική Εικόνα

3.1 Γενικός σκοπός και τα επιμέρους συστατικά του έργου

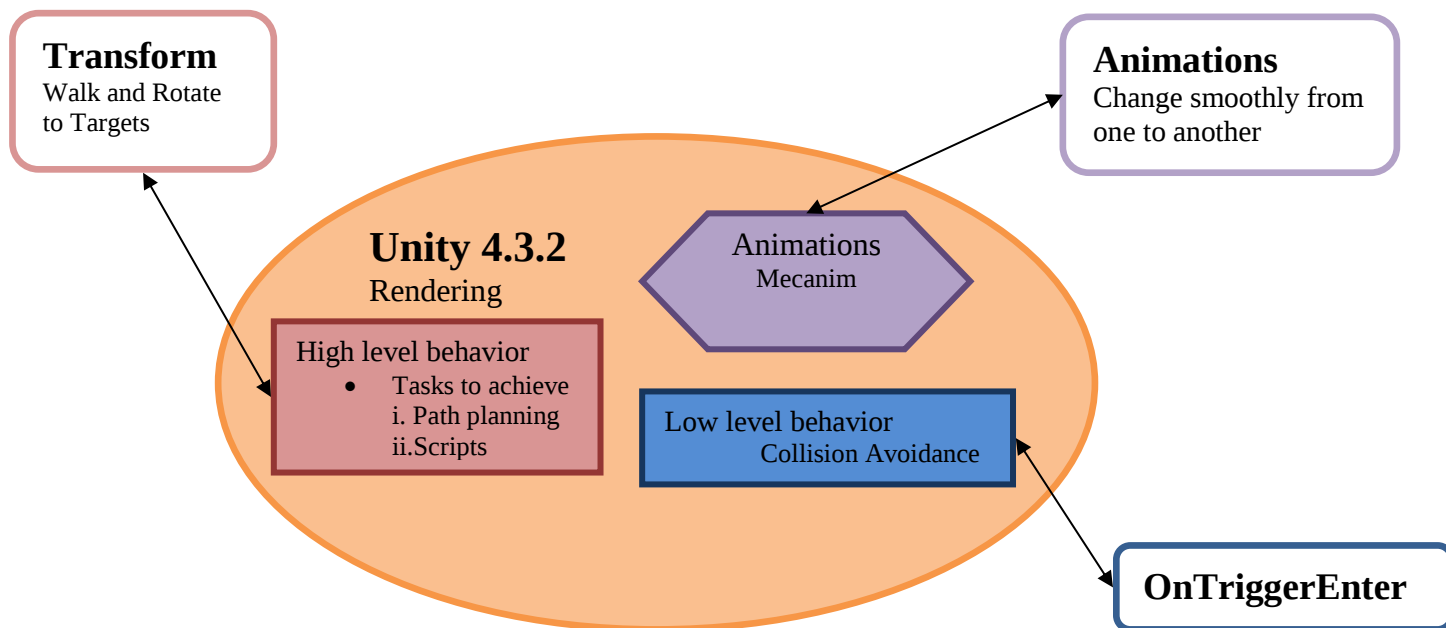
25

3.1 Γενικός σκοπός και τα επιμέρους συστατικά του έργου

Ο γενικός σκοπός του project μας ‘*Reconstruction of Everyday Life in 19th Century Nicosia*’ είναι να δημιουργήσουμε ένα ρεαλιστικό μοντέλο της πόλης μέσα στα τείχη. Η δική μου συμβολή στο έργο έχει σκοπό την προσθήκη χαρακτήρων στο έργο που θα δίνουν περισσότερη ζωή στη πόλη. Οι χαρακτήρες αυτοί αφορούν ζώα, πουλιά και δέντρα. Με την προσθήκη τους δίνουμε ακόμη περισσότερο ρεαλισμό. Οι χαρακτήρες οι οποίοι έχω προσθέσει είναι σύμφωνα με τις πληροφορίες που έχουμε πάρει από ιστορικά βιβλία, φωτογραφίες και σχέδια που αναπαριστούν τη ζωή εκείνη την εποχή.



(a)



Διάγραμμα 3.2: Συστατικά τα οποία χρησιμοποιήθηκαν για την ολοκλήρωση του έργου

Για να επιτευχθεί ένα ρεαλιστικό αποτέλεσμα χρειάζονται μοντέλα που θα απεικονίζουν τα κτίρια μέσα στα τείχη, ανθρώπινα μοντέλα και τα μοντέλα ζώων, πουλιών και δέντρων που θα δώσουν ζωή στη πόλη.

Τα ζώα θα έχουν κάποιους στόχους, θα κινούνται μέσα στη πόλη και θα εκτελούν κάποιες κινήσεις. Για αυτό το λόγο τα ζώα έχουν κάποια ρεαλιστικά animations, μια high level συμπεριφορά με targets στα οποία θα φτάνουν και θα μετακινούνται και μια low level συμπεριφορά για αποφυγή συγκρούσεων.



Paphos is a village situated twenty one kilometers from Paphos. It was the district headquarters, except during Ottoman rule, and in 1875 numbered only 135 inhabitants. The region was known for the many bandits that found refuge there.

(b)

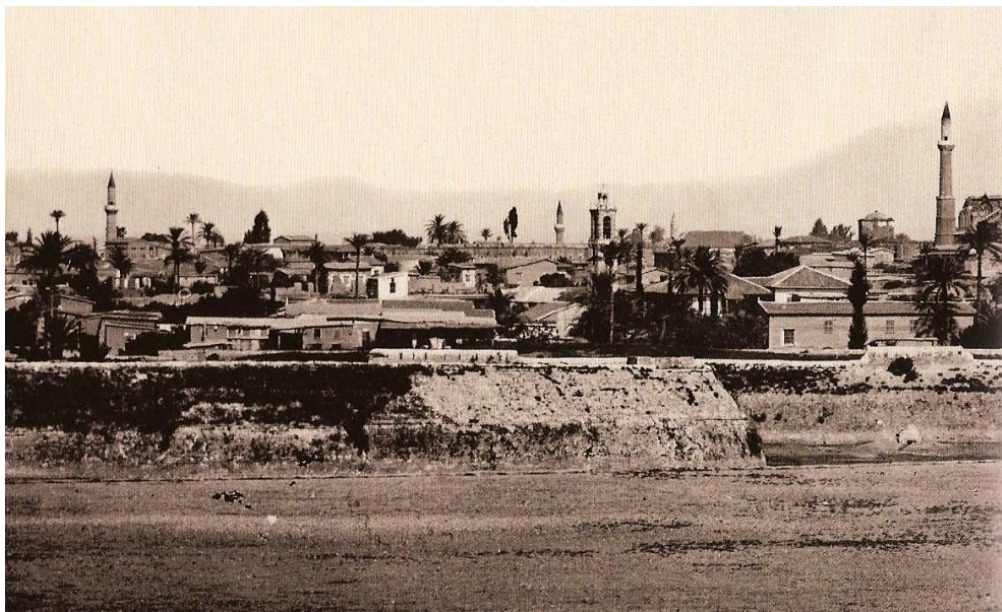
(c)

Σχήμα 3.1: (a) Ζωγραφιά από την Λευκωσία τον 19^ο αιώνα. Μπορούμε να διακρίνουμε ζώα όπως γαϊδούρια, καμήλες. (b) Ανώνυμη φωτογραφία 1905, το χάνι πίσω από τον Καθεδρικό Ναό της Αγίας Σοφίας. (c) Ανώνυμη φωτογραφία 1905, ένας δρόμος στη Πόλις (χωριό έξω από τη Πάφο).

Για την υλοποίηση όλων αυτών έχω χρησιμοποιήσει το Unity, μια ισχυρή μηχανή ενσωματωμένη με ένα πλήρες σύνολο από διαισθητικά εργαλεία και γρήγορες ροές εργασίας για τη δημιουργία διαδραστικού 3D και 2D περιεχομένου.

Για τη διαχείριση των animations χρησιμοποιώ το Mecanim, ένα animation system που προσφέρεται από το Unity. Μέσω του Mecanim τα animations συνδυάζονται και μπορούμε να δούμε μια φυσική συμπεριφορά στα ζώα αφού εναλλάσσονται το ένα μετά το άλλο πολύ απαλά και ρεαλιστικά.

Για τη δημιουργία δέντρων έχω κάνει import στο project μου πακέτο Tree Creator και σταδιακά έχω προσθέσει τον κορμό για αρχή, τα κλαδιά στη συνέχεια και τέλος τα φύλλα.



Σχήμα 3.3: Φωτογραφία της πόλης στην οποία μπορούμε να διακρίνουμε τα διάφορα είδη δέντρων που υπήρχαν (φοινικιές-κυπαρίσσια).

Κεφάλαιο 4

Ανάλυση Συστατικών του Έργου

4.1 Unity	28
4.1.1 Rendering	29
4.1.2 Scripting	30
4.1.3 Physics	30
4.2 Mecanim	31
4.2.1 Retarget Animations	31
4.2.2 Blend Trees and State Machines	32
4.3 Collision Avoidance	33
4.4 Τα 3D Models του project	33

4.1 Unity

Το Unity είναι ένα game development ecosystem.

Με άλλα λόγια είναι μια ισχυρή μηχανή πλήρως ενσωματωμένη με ένα πλήρες σύνολο από διαισθητικά εργαλεία και γρήγορες ροές εργασίας για τη δημιουργία διαδραστικού 3D και 2D περιεχομένου. Είναι μια εύκολη

πολύ-πλατφόρμα δημοσίευσης με τεράστια ποιότητα, έτοιμα assets από το Unity Asset Store και μια κοινότητα ανταλλαγής γνώσεων.

Σπάει τα χρονικά και οικονομικά εμπόδια και επιτρέπει σε όλους τους προγραμματιστές τη δημιουργία μοναδικών όμορφων παιχνιδιών.

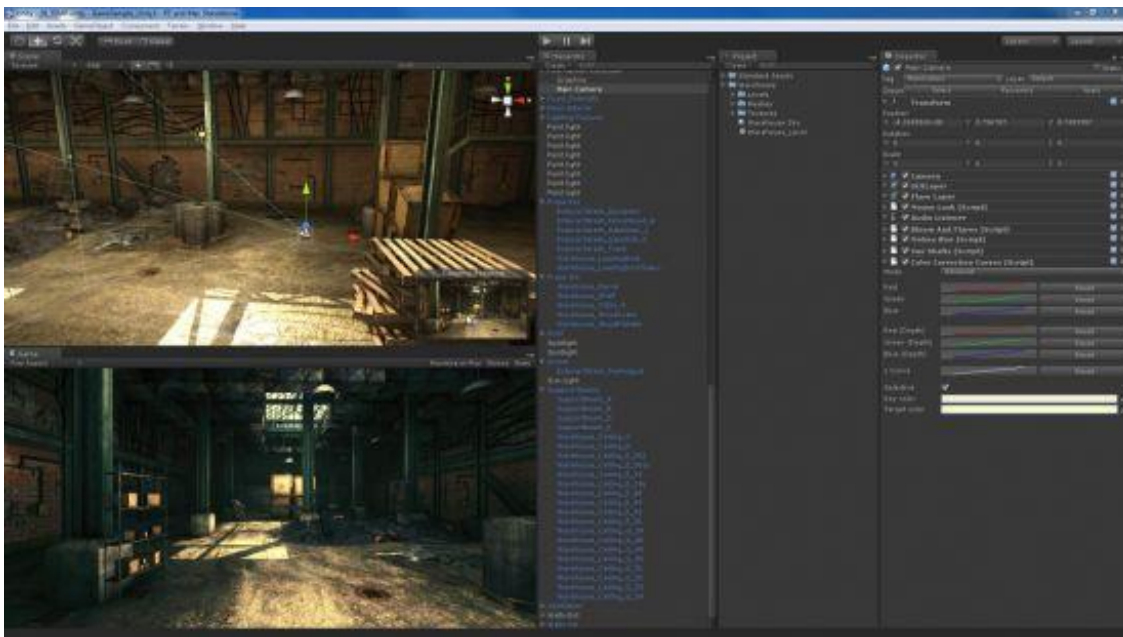
Περιλαμβάνει εκατοντάδες assets πακέτα από textures και materials, particle systems, sound effects μέχρι 3D models, animations, ακόμα και tutorials και έτοιμα projects.



Η τελευταία έκδοση που έχει δημοσιευτεί είναι το *Unity 4.3.2* την οποία και έχω χρησιμοποιήσει για την ανάπτυξη του έργου μου.

Σε αυτή την έκδοση θα υπάρχει το αναμενόμενο 2D εργαλείο, κάποια πραγματικά εντυπωσιακά Mecanim χαρακτηριστικά, ο MonoDevelop 4.0.1, μέρη από τα χαρακτηριστικά των Windows Phone 8 και Windows Store Apps.

Το Unity Pro χρησιμοποιείται μόνο από προχωρημένους χρήστες και δεν διατίθεται δωρεάν. Οι δυνατότητες του είναι αρκετά εξειδικευμένες και προσφέρει αρκετές λειτουργίες.



Σχήμα 4.1: Χρησιμοποιώντας Unity Pro

4.1.1 Rendering

Η μηχανή γραφικών χρησιμοποιεί Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), και ιδιόκτητα API (κονσόλες). Υπάρχει υποστήριξη για reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), δυναμικές σκιές χρησιμοποιώντας shadow maps, render-to-texture και προβολή των αποτελεσμάτων μετά την επεξεργασία σε πλήρη οθόνη.

Το Unity υποστηρίζει art assets και αρχεία με format από 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. Αυτά τα assets μπορούν να προστεθούν στο game project και να διαχειρίζονται μέσω του graphical user interface του Unity.

Η γλώσσα ShaderLab χρησιμοποιείται για προσθήκη σκίασης, υποστηρίζοντας τόσο δηλωτικό προγραμματισμό των fixed-function pipeline όσο και shader προγράμματα γραμμένα σε GLSL ή Cg. Μια σκίαση μπορεί να περιλαμβάνει πολλαπλές παραλλαγές και μια δηλωτική εναλλακτική προδιαγραφή, επιτρέποντας στο Unity να ανιχνεύσει την καλύτερη παραλλαγή για την τρέχουσα κάρτα γραφικών, και αν κανένα από αυτά δεν είναι συμβατό, επιστρέφει πίσω σε μια εναλλακτική σκίαση που μπορεί να θυσιάσει χαρακτηριστικά για την απόδοση.



Σχήμα 4.2: Παράδειγμα αναπαράστασης σκίασης (Google, houses Unity 3d).

4.1.2 Scripting

Το scripting στο game engine είναι χτισμένο στο Mono, η open-source υλοποίηση του .NET Framework. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν UnityScript (μια τυπική γλώσσα με ECMAScript, που αναφέρεται ως JavaScript από το λογισμικό), C#, ή Boo (η οποία είναι εμπνευσμένη από τη σύνταξη της Python).

Στο project μου έχω χρησιμοποιήσει C#, αφού είναι μια πιο κατανοητή γλώσσα κοντά στις γνώσεις μου σε C++.

Η C# είναι μια απλή, σύγχρονη, γενικής χρήσης, και κυρίως object-oriented γλώσσα προγραμματισμού. Είναι αρκετά διαδεδομένη ιδιαίτερα στους Unity developers.

4.1.3 Physics

Το Unity έχει επίσης ενσωματωμένη υποστήριξη για τη PhysX μηχανή φυσικής της Nvidia με πρόσθετη υποστήριξη για real-time cloth simulation για αυθαίρετα και skinned meshes,

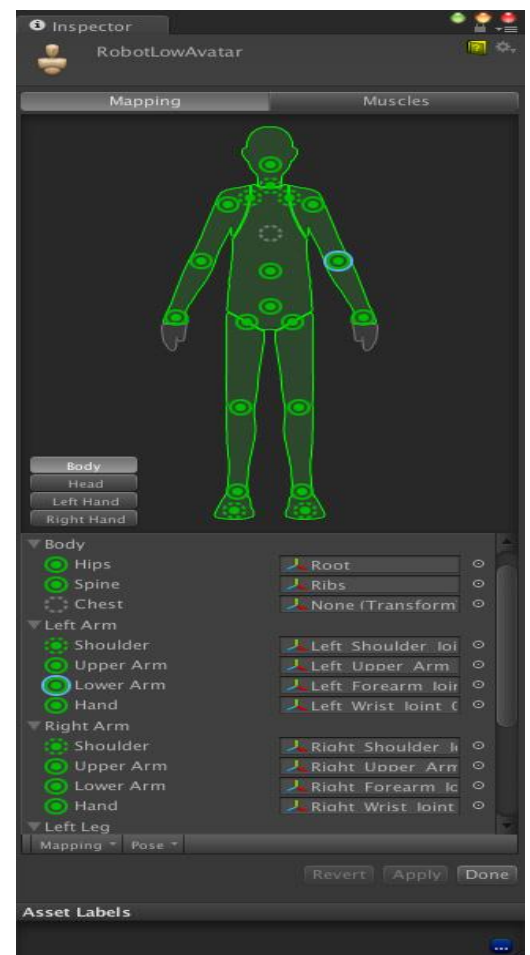
thick ray casts, και collision layers. Στο Unity 4.3 εισήχθηκε ενσωματωμένη υποστήριξη για τη Box2D μηχανή φυσικής για 2D παιχνίδια.

4.2 Mecanim

Στην έκδοση 4 εισήχθηκε ένα νέο σύστημα animation, το *Mecanim*, το οποίο συνδυάζει animations και κάνει την κίνηση ανθρώπινου ή μη ανθρώπινου χαρακτήρα να φαίνεται τελείως ρεαλιστική και φυσική. Το Mecanim είναι μια απλή και ισχυρή τεχνολογία animation, που παρέχει εργαλεία για τη δημιουργία state machines, blend trees, IK rigging και την αυτόματη αναδιάταξη των animations μέσα από τον Unity Editor.

4.2.1 Retarget animations

Το *retargeting rig* μπορεί να δημιουργηθεί πάνω στον χαρακτήρα με μερικά μόνο mouse clicks, χρησιμοποιώντας την αυτόματη χαρτογράφηση των οστών και υπολογιστικών αλγορίθμων. Το rig διασπά την κίνηση του σώματος σε μια σειρά από συσπάσεις των μυών και χρησιμοποιεί ένα μοντέλο μάζας για τον υπολογισμό του κέντρου μάζας του σώματος και το μέσο προσανατολισμό του σώματος για την τροχιά κίνησης.



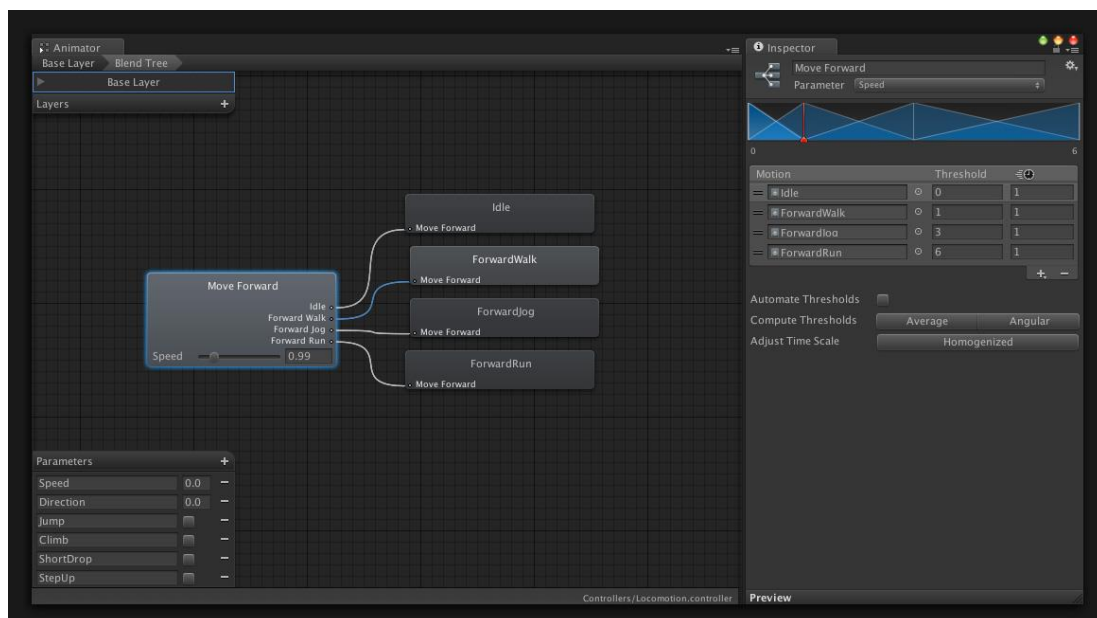
Σχήμα 4.3: Mapping and muscles

4.2.2 Blend Trees και State Machines

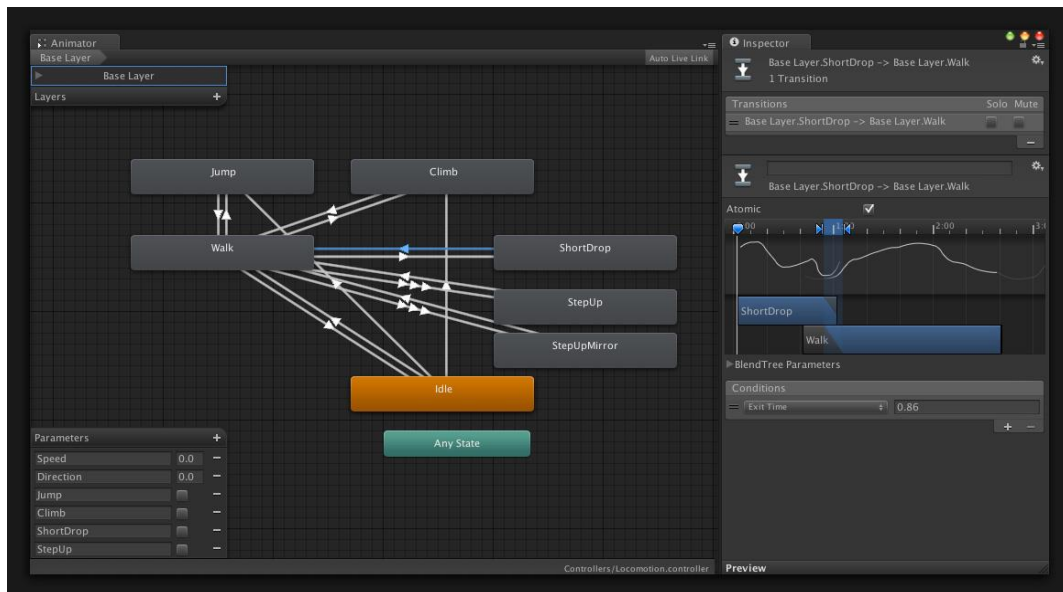
Η κατασκευή των *blend trees* και πιο πολύπλοκων *state machines* αποσκοπούν στο να έχει ο προγραμματιστής τον πλήρη έλεγχο στο πως θα κινείται ο χαρακτήρας του. Ο Unity Editor παρέχει ισχυρά εργαλεία για τη δημιουργία τους. Animation clips μπορούν να χρησιμοποιηθούν ως φύλλα σε ένα πολύ-επίπεδο blend tree ή ως ένα state σε μια ιεραρχία από state machines.

Τα blend trees μας επιτρέπουν να δημιουργήσουμε μια μεγάλη ποικιλία από κινήσεις με πολύ λίγα motion clips. Σε ένα blend tree Editor καθορίζεις τις παραμέτρους που συνδυάζονται μεταξύ τους και μπορείς να δεις τα συνδυασμένα animations σε 3D προβολή.

Ο *Animator Controller* μπορεί να καθορίσει ένα αυθαίρετο αριθμό επιπέδων. Κάθε στρώμα μπορεί να χρησιμοποιήσει το δικό του state machine ή να μοιραστεί ένα state machine με ένα κύριο στρώμα. Μέσω του animator controller μπορούμε να αναθέσουμε σε κάθε απλό state machine ένα animation clip και δημιουργώντας μια μετάβαση σε ένα άλλο επιτυγχάνουμε να δώσουμε μια σειρά από κινήσεις σε ένα χαρακτήρα κάνοντάς τον να ενεργεί φυσιολογικά. Τα animations εναλλάσσονται μεταξύ τους απαλά ανάλογα με τις παραμέτρους που καθορίσαμε.



Σχήμα 4.4: Blend Trees in Mecanim



Σχήμα 4.5: Mecanim's powerful state machine

4.3 Collision Avoidance

Με τον όρο collision avoidance εννοούμε τη διαδικασία της πρόληψης της σύγκρουσης ενός αντικειμένου με ένα άλλο. Στο crowd simulation, το collision avoidance αφορά την αποφυγή σύγκρουσης των ανθρώπινων παραγόντων μεταξύ τους, με άλλους παράγοντες, με άλλα εμπόδια όπως κτίρια ή αντικείμενα στο χώρο.

4.4 Τα 3D Models του project

Όπως είχα προαναφέρει στόχος μου στη συνέχιση του έργου ήταν η τροποποίηση του με σκοπό να γίνει ακόμη πιο ρεαλιστικό. Πέρα από τους ανθρώπινους χαρακτήρες, η προσθήκη ζώων, πουλιών και δέντρων δίνουν ζωή και χαρακτήρα στη πόλη.

Για το σκοπό αυτό χρησιμοποίησα μοντέλα ζώων από το Unity Asset Store τα οποία είχα εισαγάγει στο project μου.

Πρώτη κατηγορία τα πουλερικά. Συγκεκριμένα κόττες, αφού η εκτροφή τους ήταν ένα μέσο τροφής και επιβίωσης. Σε κάθε μοντέλο υπάρχουν ενσωματωμένα animations τα οποία έχω χρησιμοποιήσει. Μερικά από αυτά είναι το “Eat”, “Idle”, “Walk”.



Πολύ ρεαλιστική προσθήκη αποδείχτηκε ένα σμήνος από πουλιά που πετούν πάνω από την πόλη σε διάφορους σχηματισμούς. Υπάρχουν πέντε διαφορετικά είδη πετάγματος του σμήνους, “Free Roam”, “Ultra Synced”, “Wild”, “Wild Few”, “Wild Synced”.

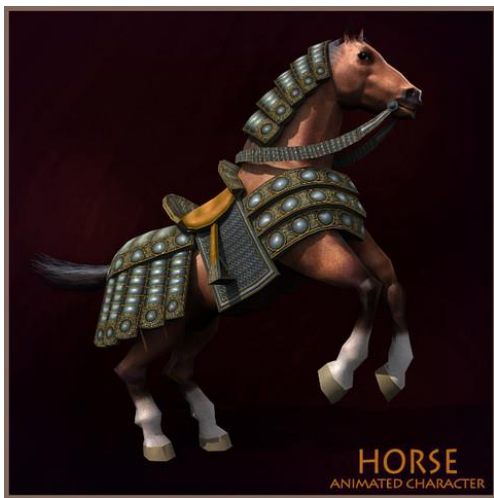
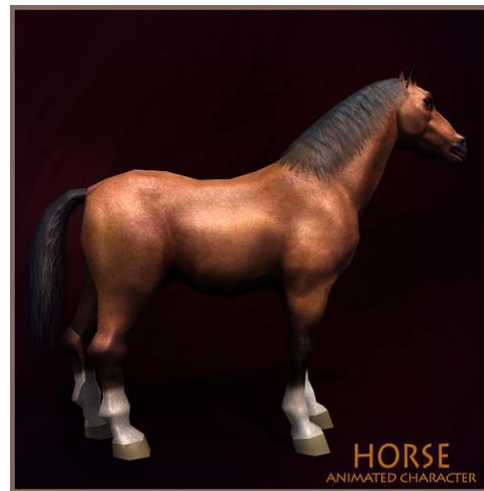
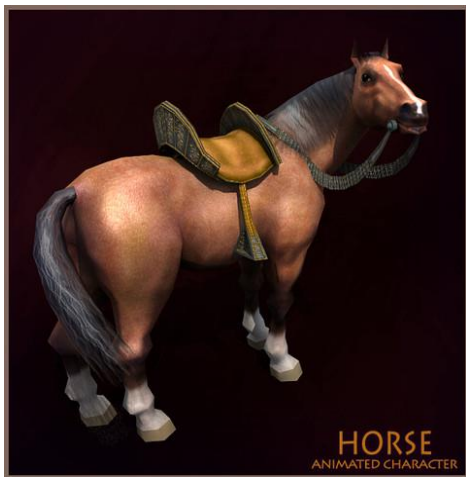


Εκτός από το σμήνος των πουλιών έχω χρησιμοποιήσει κοράκια όπου σύμφωνα με τις πληροφορίες μας ήταν ένα συνηθισμένο είδος πουλιού την εποχή εκείνη.

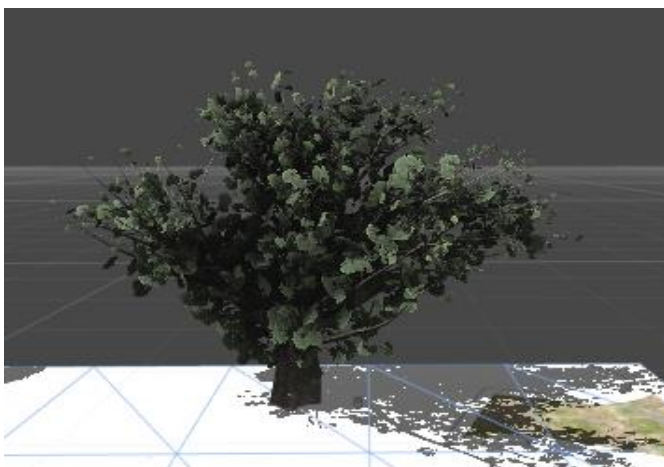


Δεύτερη κατηγορία τα τετράποδα. Ζώα όπως άλογα, γαϊδούρια και καμήλες χρησίμευαν στους ανθρώπους ως μεταφορικό μέσο, τόσο για τους ίδιους, όσο και για τα διάφορα αντικείμενα ή τροφές, και γενικά οτιδήποτε χρειαζόταν να μετακινήσουν και αφορούσε τις γεωργικές τους καλλιέργειες ή προϊόντα για την αγορά.

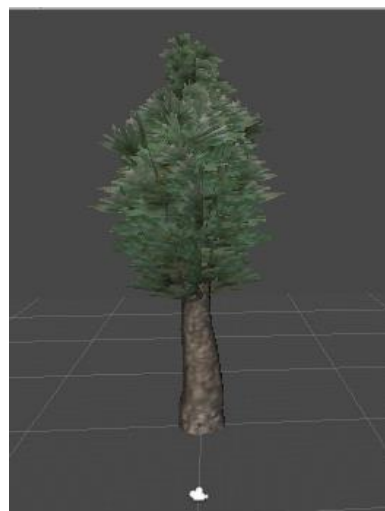
Στο project έχω χρησιμοποιήσει μόνο άλογα αφού δεν ήταν δυνατό να βρούμε μοντέλα καμήλων ή γαϊδουριού. Υπάρχουν ενσωματωμένα animations τα οποία έχω χρησιμοποιήσει, όπως για παράδειγμα “Walk”, “Run”, “Rear up”, “Sniff”, “Looking Around”, “Idle”, “Jump” και πολλά άλλα (σύνολο 19).



Ομορφιά στον χώρο δίνει το φυσικό περιβάλλον που προβάλλεται μέσα από το project. Για το σκοπό αυτό έχω κατασκευάσει δέντρα από το Tree Creator του Unity. Τα δέντρα που αφορούσαν την εποχή εκείνη είναι συκιές, κυπαρίσσια, φοινικιές.



Σχήμα 4.6: Συκιά



Σχήμα 4.7: Κυπαρίσσι

Κεφάλαιο 5

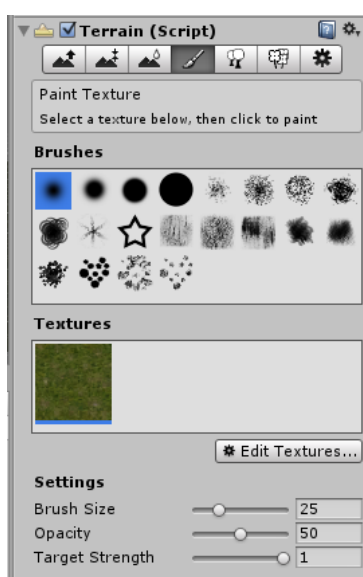
Υλοποίηση

5.1 Αρχικές Προσπάθειες	36
5.2 Mecanim (Animation Controller)	39
5.2.1 Mecanims' Tutorial – Robot	39
5.3 Χρήση Mecanim, επικοινωνία και συμπεριφορά των μοντέλων	41
5.4 Collision Avoidance	50
5.5 Tree Creator	52

5.1 Αρχικές Προσπάθειες

Πρώτη μου επαφή με το Unity απέτελεσαν κάποια έτοιμα projects-tutorials από την ιστοσελίδα του Unity για εξοικείωση με το περιβάλλον, τα εργαλεία και τις δυνατότητες του. Επεξεργάστηκα σκηνές δημιουργώντας έδαφος με textures όπως γρασίδι ή γρασίδι με πέτρες κάνοντας το έδαφος ακόμη πιο ρεαλιστικό.

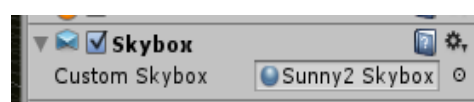
Ακόμη δημιούργησα λόφους, κορυφές, πρόσθεσα δέντρα, καθώς επίσης και skybox (materials με διάφορες σκηνές από ουρανό).



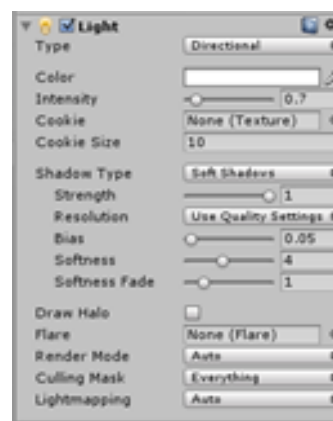
(a)



(b)



(c)



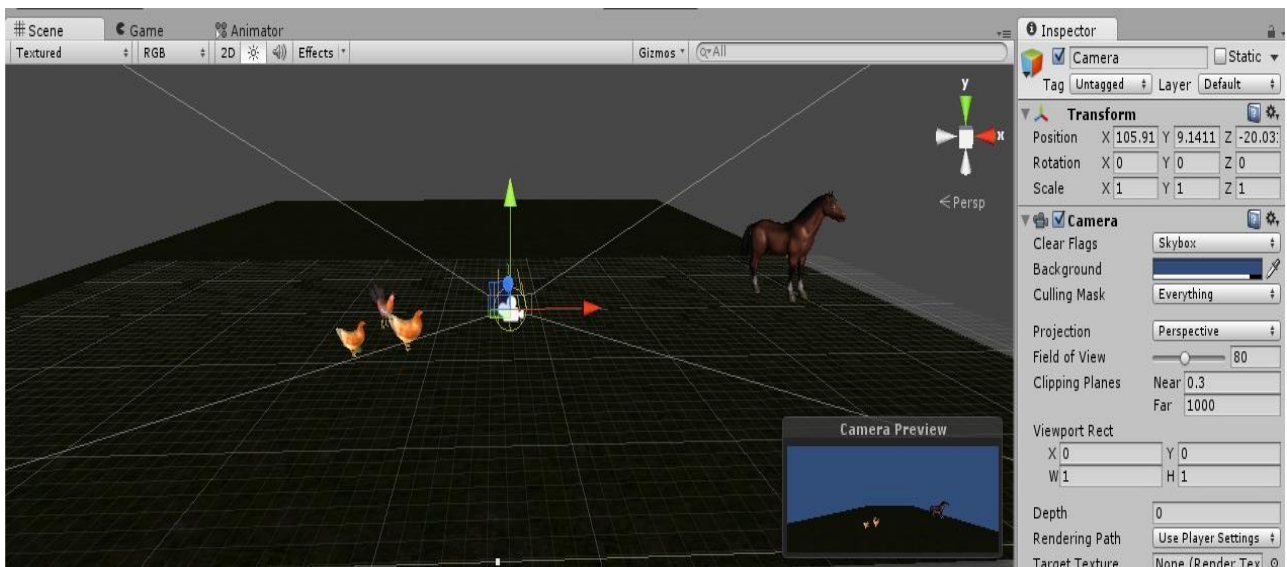
(d)

Σχήμα 5.1: (a) προσθήκη texture στο έδαφος, (b) προσθήκη δέντρων, (c) Skybox, (d) Light

Απαραίτητη προϋπόθεση είναι η camera στο project έτσι ώστε κατά την εκτέλεση ο χρήστης να μπορεί να βλέπει τη σκηνή ή να αλληλεπιδρά με το παιχνίδι του. Η θέση της cameras καθορίζεται από το χρήστη μέσω του camera preview. Εκτός από camera μπορείς να χειρίζεσαι τη σκηνή σου με ένα character controller, δηλ. να παρακολουθείς το παιχνίδι μέσα από τα μάτια ενός χαρακτήρα μετακινώντας το mouse ανάλογα μέσα στη σκηνή.

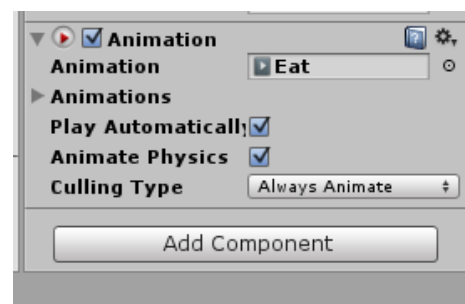
Μέσω της cameras προσθέτεις φως στη σκηνή. Ένα directional light το οποίο θα φωτίζει προς το μέρος το οποίο κατευθύνεται η camera. Εκτός από το φως, περισσότερη ρεαλιστικότητα στη σκηνή δίνει ο άνεμος προσθέτοντας wind zone και ένα bend factor στα δέντρα που καθορίζει τη δύναμη με την οποία θα επηρεάζονται από τον άνεμο.

Επόμενη προσπάθεια ήταν η φόρτωση των μοντέλων (κότες και κόκορες, αργότερα άλογα) από το Unity Asset Store στο project μου και ο χειρισμός τους.



Σχήμα 5.2: Σκηνή με camera, directional light και μοντέλα ζώων.

Είχα αναθέσει σε κάθε μοντέλο από ένα animation και μπόρεσα να δω τη συμπεριφορά τους. Για παράδειγμα όπως βλέπουμε στην εικόνα 5.2, το animation “Eat” σε ένα από τα μοντέλα.



Σχήμα 5.3: Ανάθεση animation σε ένα μοντέλο

Πρώτη προσέγγιση με τον κώδικα αποτελούσε η ανάγκη ανάθεσης στα μοντέλα κάποιων λειτουργιών, όπως η μετακίνηση τους από ένα σημείο σε άλλο και η περιστροφή του σώματος τους. Ο κώδικάς μου σε C# και οι εντολές που χρησιμοποίησα είναι το *transform.position* και *transform.rotation*.

Με το *transform.position* παίρνουμε τη θέση την οποία βρισκόμαστε και μπορούμε να του αναθέσουμε μια νέα θέση. Έτσι μέσω του *Vector3.Lerp* ή του *Vector3.MoveFoward* μετακινούμε το gameObject μας από τη θέση στη οποία βρίσκεται στη νέα θέση που επιθυμούμε.

Το *Vector3.Lerp* παίρνει το διάνυσμα μεταξύ δύο σημείων και ουσιαστικά μετακινεί σταδιακά το object μέσα σε αυτό το διάνυσμα, από το ένα σημείο στο άλλο. Με την ίδια λογική λειτουργεί το *MoveFoward*. Μέσω του *Vector3* παίρνουμε τις 3 διαστάσεις x, y, z.

Με το *transform.rotation* περιστρέφουμε το object και αυτό επιτυγχάνεται μέσω του Quaternion.

Συγκεκριμένα στο παράδειγμα (εικόνα 5.3) δηλώνουμε από το Unity το object στο οποίο θέλουμε να φθάσει το μοντέλο μας, δηλ. το target, και ένα step το οποίο υποδηλώνει την ταχύτητα περιστροφής και μετακίνησης του μοντέλου που υπολογίζεται βάση του *Time.deltaTime* (ο χρόνος σε δευτερόλεπτα που χρειάζεται για να ολοκληρωθεί το προηγούμενο frame).

```
using UnityEngine;
using System.Collections;

public class chicken: MonoBehaviour {
    float step = 0.1f * Time.deltaTime;
    Vector3 myPosition;
    Vector3 newDir;
    public Vector3 target;

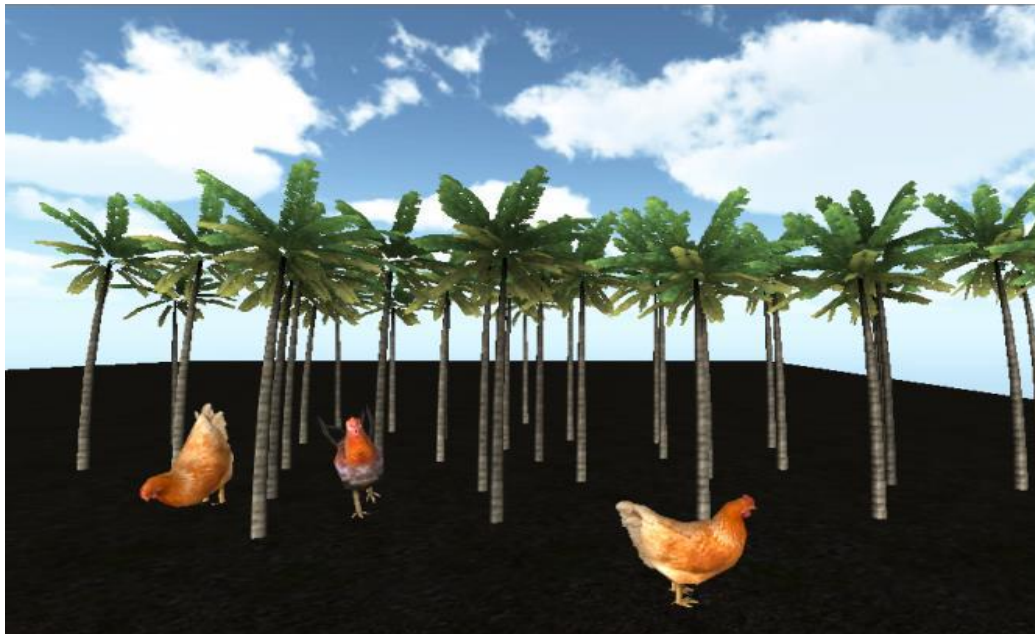
    void Start() { }

    void Update(){

        newDir= Vector3.RotateTowards(transform.forward, target, step, 0.0F);
        transform.rotation = Quaternion.LookRotation(newDir);
        transform.position = Vector3.Lerp(myPosition, target, step);

    }
}
```

Σχήμα 5.4: Παράδειγμα κώδικα move και rotate.



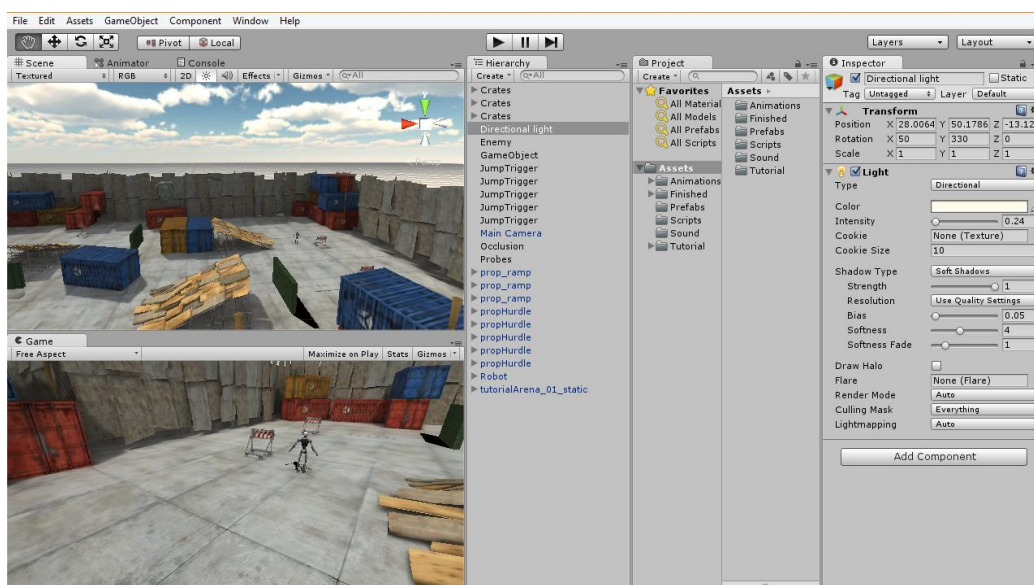
Σχήμα 5.5: Έξοδος προγράμματος.

5.2 Mecanim (Animation Controller)

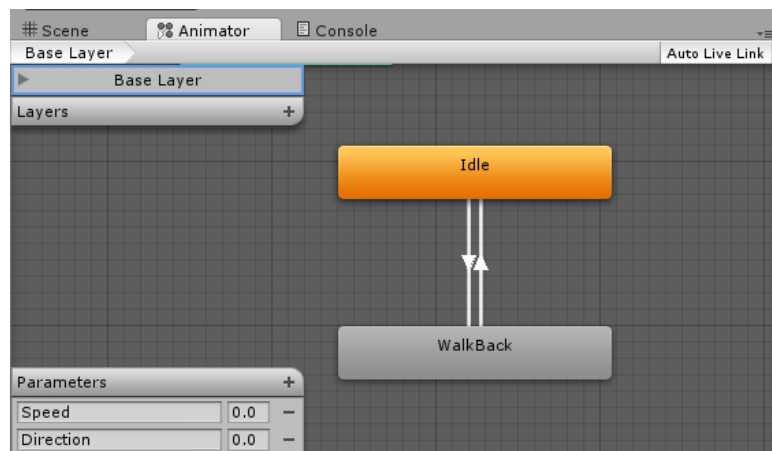
Όπως έχω αναφέρει και εξηγήσει σε προηγούμενα κεφάλαια, για τη διαχείριση των animations σε ένα gameObject και την απαλή εναλλαγή τους με σκοπό την προσθήκη λειτουργιών και συμπεριφοράς σε αυτό, το Unity προσφέρει το εργαλείο Mecanim.

5.2.1 Mecanims' Tutorial - Robot

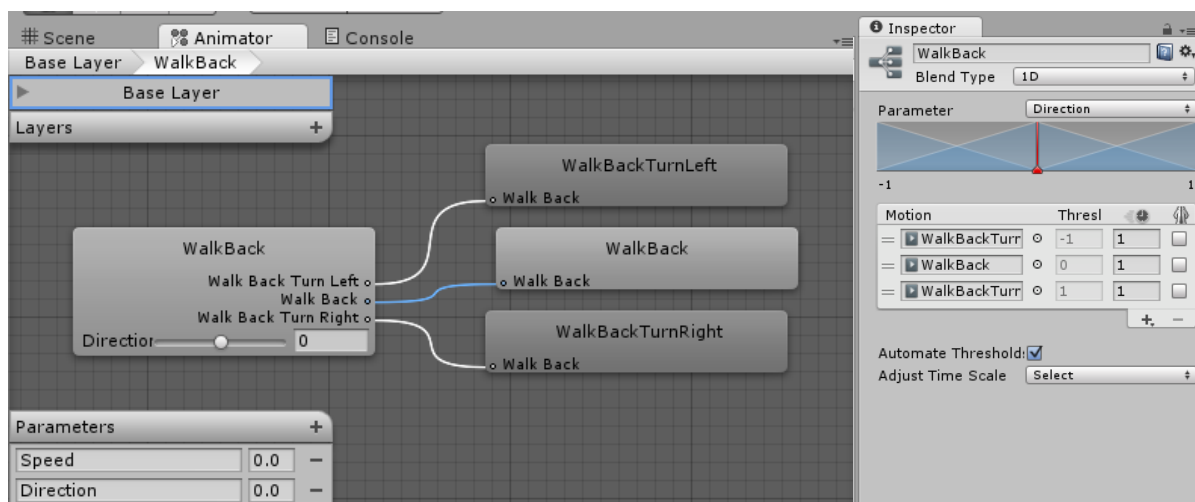
Το Unity προσφέρει ένα εκπαιδευτικό tutorial με σκοπό την εξοικείωση του προγραμματιστή με το Mecanim και τη διαχείριση των animations σε ένα gameObject.



Στα περιεχόμενα του tutorial περιλαμβάνεται η δημιουργία blend trees, απλών state machines, transactions και παραμέτρων βάση των οποίων θα εναλλάσσονται τα animations.



Καθορίζουμε το αρχικό state machine (με κίτρινο χρώμα), το οποίο θα είναι το αρχικό animation του object μας και η μετάβαση στο επόμενο (WalkBack) θα γίνει μέσω της παραμέτρου Speed, δηλ. όταν το Speed υπερβεί ένα αριθμό ή γίνει λιγότερο από ένα αριθμό.



Blend Tree: Ένα blend tree είναι στην ουσία ένα πολύπλοκο state machine όπου συνδυάζει πολλά animation σε ένα state machine τα οποία εκτελούνται στη σειρά σύμφωνα με την παράμετρο που καθορίζουμε. Στην εικόνα πιο πάνω το WalkBackTurnLeft ακολουθείται από το WalkBack και αυτό από το WalkBackTurnRight. Η παράμετρος εδώ είναι η κατεύθυνση (direction), οπότεν το robot μας θα κινηθεί πρώτα πίσω αριστερά και αν η παράμετρος

direction αλλάξει σε ευθεία το robot θα κινηθεί πίσω ευθεία και όταν αλλάξει ξανά προς τα δεξιά θα κινείται προς τα πίσω δεξιά.

Σε αυτό το tutorial βλέπουμε ένα διαφορετικό τρόπο αλληλεπίδρασης του robot με τον χρήστη. Ο χρήστης κινεί το robot από τα βέλη του πληκτρολογίου και μπορεί να τον βλέπει μέσα από τα “μάτια” ενός character controller. Πατώντας τα βελάκια δεξιά ή αριστερά ανάλογα αλλάζει η παράμετρος direction και με αυτό τον τρόπο βλέπουμε το robot να κινείται σύμφωνα με το blend tree που έχουμε δηλώσει και φαίνεται πιο πάνω.

5.3 Χρήση Mecanim, επικοινωνία και συμπεριφορά των μοντέλων

Πολύ σημαντικό εργαλείο για την υλοποίηση του project αποτελούσε το Mecanim. Δημιουργώντας animation controller για κάθε χαρακτήρα μπόρεσα να καθορίσω τη συμπεριφορά του και τις κινήσεις του ανάλογα με τους υπόλοιπους χαρακτήρες γύρω του.

➤ Κότες-Κόκορες

Τα μοντέλα αυτά είχαν έτοιμα τρία animations. Τα animations αυτά είναι idle, walk, eat.

Στο project υπάρχουν 2 ομάδες από κότες-κόκορες.

Η μία ομάδα βρίσκεται στο εσωτερικό μιας αυλής σε μια γειτονία μέσα στη πόλη. Στο κάθε μοντέλο έχω αναθέσει διαφορετικές συμπεριφορές. Για παράδειγμα κάποιες θα βρίσκονται στο χώρο τρώγοντας από το έδαφος, ενώ κάποιες άλλες θα είναι ακίνητες κοιτάζοντας γύρω. Όλες μαζί συντονίζονται με κάποιο χρόνο μεταξύ τους έτσι ώστε να ξεκινά ο ένας με idle, ο άλλος μετά από λίγο με eat, και αυτός ο οποίος ήταν idle να αρχίσει το eat ή να κινηθεί στο χώρο (walk). Την ίδια λογική θα ακολουθήσει και αυτός που είχε ξεκινήσει με eat. Μπορεί στη συνέχεια να είναι idle, να ξαναμπει σε eat ή να κινηθεί μαζί με τον άλλο στο χώρο.



Η άλλη ομάδα βρίσκεται στην αγορά δίπλα από τον πάγκο κάποιου πωλητή. Η ομάδα αυτή κάνει κάτι πιο εξειδικευμένο καθώς βρίσκεται σε εξωτερικό χώρο με ανθρώπους να κινούνται και άλλα ζώα να περιφέρονται. Σκεπτόμενοι μια φυσική και ρεαλιστική συμπεριφορά των κοτών, σε κάθε επαφή τους με άλλα ζώα (άλογα στη συγκεκριμένη περίπτωση), οι κότες θα ενοχληθούν από την παρουσία τους και θα απομακρυνθούν μακριά σαν ομάδα. Για την επίτευξη του σκοπού αυτού υπάρχει στο χώρο ένα άδριο `gameObject` με ένα `collider` κάποιας ακτίνας έτσι ώστε σε περίπτωση `collision` με το άλογο παίρνουμε τη θέση του αλόγου εκείνη τη στιγμή, τις θέσεις της κάθε κότας και υπολογίζουμε το διάνυσμα της μεταξύ τους απόστασης. Στη συνέχεια παίρνουμε αυτό το διάνυσμα και το προσθέτουμε στην αντίθετη κατεύθυνση από τη θέση που βρίσκεται εκείνη τη χρονική στιγμή η κότα. Με αυτό τον τρόπο θα γυρίσει και θα κατευθυνθεί αντίθετα μακριά από το άλογο.

Έχω δημιουργήσει ένα tag (“Chicken”) έτσι ώστε να μαρκάρω όλες τις κότες μέσα στην εμβέλεια του `emptyGameObject` και να τις τοποθετήσω μέσα σε ένα πίνακα. Με αυτό τον τρόπο αποθηκεύουμε τη θέση του κάθε μοντέλου και την αλλάζουμε με βάση το διάνυσμα.

Όταν το άλογο έρθει σε επαφή με την εμβέλεια του `emptyGameObject` παίρνουμε τη θέση του και την περνούμε μέσα στο script του `emptyGameObject` με σκοπό τον υπολογισμό του διανύσματος και της νέας θέσης της κάθε κότας.



```

void Update () {
    float step = 0.1f * Time.deltaTime;
    if (fracComplete < 1) {
        GameObject []array = GameObject.FindGameObjectsWithTag("Chicken");
        for(int i=0;i<array.Length;i++){
            GameObject obj=array[i];
            fracComplete = (Time.time - StartTime) / journeyTime;

            myPosition[i].y=0;
            targetDir[i].y=0;
            newDir[i].y=0;

            if (obj.gameObject.GetComponent<Animator>().GetCurrentAnimatorStateInfo(0).IsName ("walk")){
                newDir[i] = Vector3.RotateTowards(transform.forward, targetDir[i], step, 0.0F);
                obj.transform.rotation = Quaternion.LookRotation(newDir[i]);
                obj.transform.position = Quaternion.Slerp (myPosition[i], targetDir[i], fracComplete);
            }
        }
    }
}

```

Σχήμα 5.3.1: Χρήση μοντέλων μέσα από τον πίνακα, όταν φτάσουν σε κατάσταση “walk” μέσω του animator controller τους, θα γυρίσουν το σώμα τους και θα περπατήσουν μακριά με βάση το targetDir. Στη μεταβλητή fracComplete αναθέτουμε τη πραγματική χρονική στιγμή που βρισκόμαστε (real time).

```

void OnTriggerEnter(Collider col) {

    if (col.gameObject.name.StartsWith("Horse")) {

        col.gameObject.GetComponent<Animator>().SetBool("isWalking", false);

        horsepos = col.gameObject.GetComponent<horse_Emily>().currentPosition;
        GameObject []array = GameObject.FindGameObjectsWithTag("Chicken");
        myPosition=new Vector3[array.Length];
        targetDir=new Vector3[array.Length];
        newDir=new Vector3[array.Length];

        for(int i=0;i<array.Length;i++){

            GameObject obj=array[i];
            myPosition[i] = obj.transform.position;
            StartTime=Time.time;
            targetDir[i] =myPosition[i] -(2*(myPosition[i] -horsepos))/array.Length;
            fracComplete=0;

        }
    }
}

```

Σχήμα 5.3.2: Όταν γίνει το collision με το αλόγο θέτουμε το state του animator του false (για να συνεχίσει να εκτελεί τα επόμενα states), παίρνουμε τη θέση του αλόγου από το αλόγο, τοποθετούμε τα gameObjects με tag “Chicken” και αρχικοποιούμε τους πίνακες με τις θέσεις τους. Τοποθετούμε τη

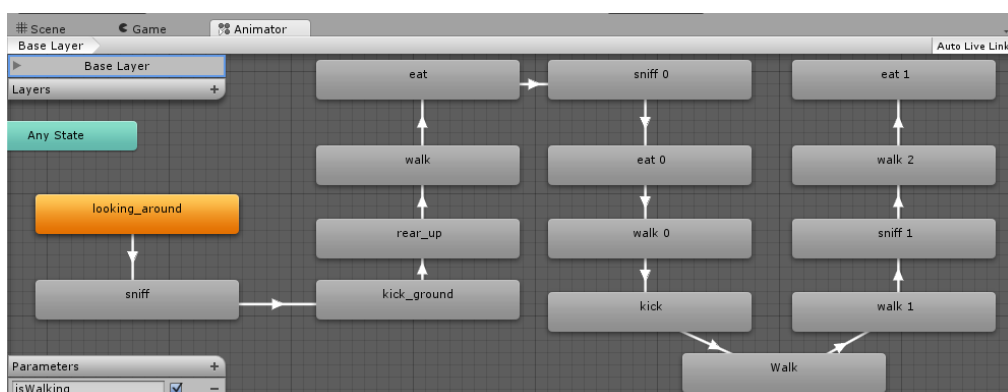
θέση της κάθε κότας στον πίνακα *myPosition*, το *real-time* χρόνο και στο *targetDir* πίνακα θα υπολογίσουμε και θα αναθέσουμε τη νέα θέση τους σύμφωνα με τη δική τους τώρα και αυτή του αλόγου την ώρα του *collision*.

$$\text{newPosition} = \text{previousPosition} - (\text{previousPosition} - \text{otherObjectPosition})$$

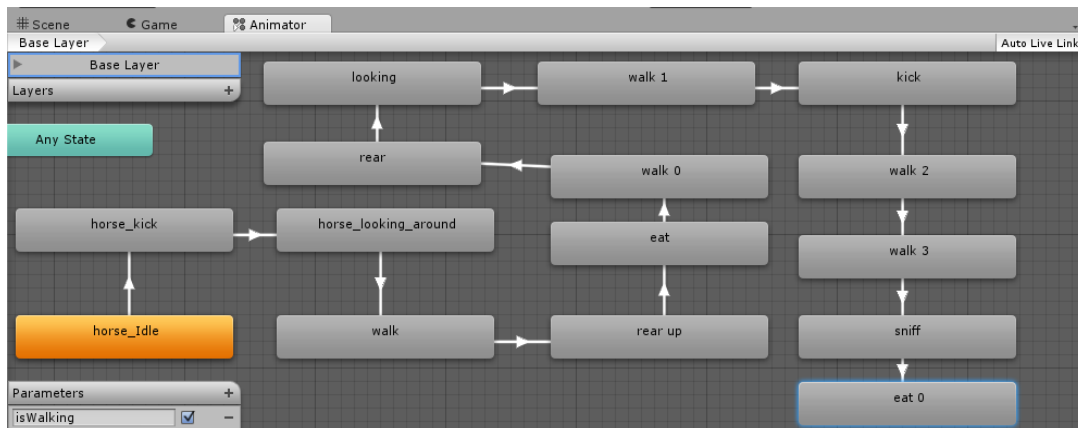
$$(0,-3) = (3,0) - [(3,0) - (6,-3)]$$

➤ Άλογα

Το project περιλαμβάνει 4 άλογα. Το πρώτο άλογο (Emily) βρίσκεται μέσα στην αγορά κοιτάζοντας γύρω και εκτελώντας μια σειρά από animations. Στη συνέχεια κατευθύνεται προς τη δεύτερη ομάδα από κότες που βρίσκονται στην αγορά και ακολούθως συνεχίζει να κινείται προς το δεύτερο άλογο (Jenn). Τα δύο άλογα αλληλεπιδρούν μεταξύ τους και το καθένα θα συνεχίσει την πορεία του προς διαφορετικές κατευθύνσεις ή θα ακολουθήσουν μια κοινή πορεία. Τα 2 άλογα εκτελούν animations όπως walk, eat, rear up, kick ground, sniff, looking around, kick. Κατά την πρώτη επαφή τους θα εκτελέσουν και τα 2 το animation rear up, ακολούθως το άλογο Jenn θα κατευθυνθεί προς την αγορά όπου και θα εκτελέσει το eat animation και στη συνέχεια θα κατευθυνθεί προς τα δέντρα στην άκρη της αγοράς μαζί με το άλογο Emily. Όταν φθάσουν και τα 2 στο προορισμό τους θα εκτελούν παρόμοιες κινήσεις όπως θα ήταν μια φυσιολογική συμπεριφορά 2 αλόγων που συναντώνται (στη περίπτωση αυτή το animation kick ground). Τα δύο άλογα στη συνέχεια θα κατευθυνθούν πίσω προς την αγορά σε κάποιο σημείο εκτελώντας και πάλι παρόμοια animations όπως eat, looking around.



Σχήμα 5.3.3: Animation Controller 2^ο αλόγου Jenn.



Σχήμα 5.3.4: Animation Controller 1^{ov} αλόγου Emily.

Στο 3^ο άλογο (Chris) ήθελα να δώσω διαφορετική λειτουργία κάνοντάς το πιο διασκεδαστικό. Το άλογο αλληλεπιδρά με τον ιδιοκτήτη του, ο οποίος βρίσκεται και κινείται δίπλα του. Τα 2 μοντέλα κινούνται μαζί από το χρήστη μέσω του πληκτρολογίου συμμετέχοντας έτσι ενεργά μέσα στο project. Πατώντας το δεξί τόξο τα δύο μοντέλα κινούνται δεξιόστροφα, ενώ πατώντας το αριστερό αριστερόστροφα. Πατώντας το πάνω τόξο κινούνται ευθεία μπροστά ενώ με το κάτω τόξο σταματούν εκτελώντας το καθένα το idle animation του.

Λόγω της ταχύτητας που κινείται το κάθε μοντέλο (άνθρωπος - άλογο), υπάρχουν σημεία όπου ο ένας θα απομακρυνθεί σε μικρή απόσταση από τον άλλο αλλά χρησιμοποιώντας σωστά τα τόξα θα επανέλθουν. Σε αυτήν την υλοποίηση για τα δύο μοντέλα υπάρχει πιθανότητα να μην αναγνωρίζουν το σώμα ο ένας του άλλου παρά τις προσπάθειες μου.

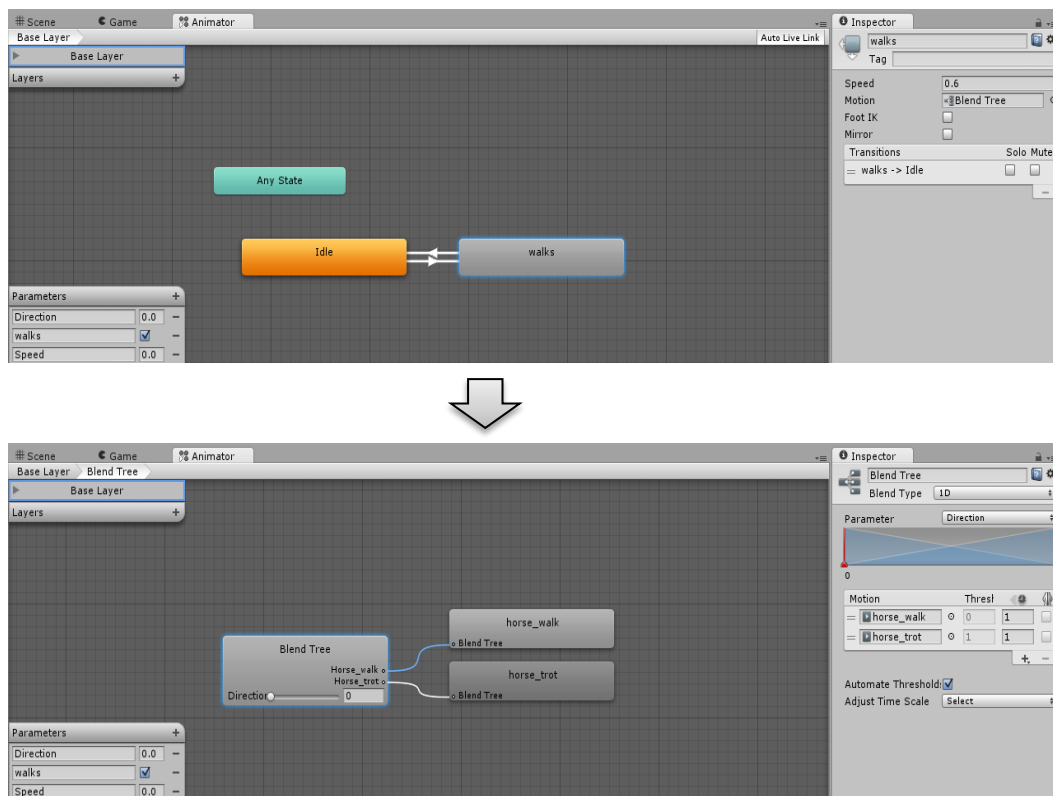
Το άλογο κινείται με ένα Blend Tree και περιλαμβάνει τρεις παραμέτρους, Speed, walks, Direction. Με τις παραμέτρους Speed και walks μεταφερόμαστε από το ένα state (idle) στο άλλο (walks – Blend Tree). Η παράμετρος walks είναι Boolean και η παράμετρος Speed float. Όταν γίνει true η walks και η ταχύτητα του μεγαλύτερη από 0 το άλογο θα περπατήσει πατώντας το πάνω τόξο ενώ όταν πατήσουμε το κάτω τόξο η walks θα γίνει false και το Speed (ταχύτητα) μικρότερη από 0 οπότε και το άλογο θα σταματήσει να κινείται μπαίνοντας στο idle state του. Το blend tree αποτελείται από δύο motion fields, το animation walk και trot που υπάρχουν στο άλογο. Τα δύο αυτά animations αλλάζουν μεταξύ τους βάση της κατεύθυνσης που ακολουθεί το άλογο. Δηλ. όταν η κατεύθυνση αλλάξει και κινείται δεξιόστροφα πατώντας το δεξί τόξο τότε το άλογο θα εκτελεί το trot animation (άλλος τρόπος περπατήματος πιο εντυπωσιακός), ενώ με το αριστερό τόξο θα εκτελεί το walk.

```

if (currentBaseState.nameHash == locoState){
    if(Input.GetKey(KeyCode.UpArrow)){
        anim.SetBool("walks", true);
    }
    if(Input.GetKey(KeyCode.RightArrow)){
        transform.Rotate (new Vector3 (0, 30 * Time.deltaTime, 0));
    }
    if(Input.GetKey(KeyCode.LeftArrow)){
        transform.Rotate (new Vector3 (0, -30 * Time.deltaTime, 0));
    }
    if(Input.GetKey(KeyCode.DownArrow)){
        anim.SetBool("walks", false);
    }
}
else if (currentBaseState.nameHash == idleState){
    if(Input.GetKey(KeyCode.UpArrow)){
        anim.SetBool("walks", true);
    }
}
}

```

Σχήμα 5.3.5: Αν βρίσκεσαι σε κίνηση και από το input πάρουμε το πάνω τόξο αναθέτουμε την boolean μεταβλητή walk true από τον animator έτσι ώστε να εκτελέσει το walk animation και να κινηθεί ευθεία μπροστά (Speed > 0). Παίρνοντας από το πληκτρολόγιο το δεξί ή αριστερό τόξο κάνει rotate το σώμα του ανάλογα στην κατεύθυνση που πρέπει και εκτελώντας το ανάλογο animation με βάση το Direction. Με το κάτω τόξο θέτουμε τη μεταβλητή walk false αναθέτοντάς του από τον animator το idle animation σταματώντας την κίνησή του (Speed < 0).



Σχήμα 5.3.6: Blend tree, animation states, parameters.



Σχήμα 5.3.7: Κίνηση μοντέλων από το πληκτρολόγιο

Το 4^ο άλογο βρίσκεται μέσα στη πόλη, στην ίδια γειτονιά με την πρώτη ομάδα κοτών. Με αυτό τον τρόπο δείχνουμε ότι οι άνθρωποι είχαν στις αυλές τους τα ζώα τους προστατευμένα. Το άλογο εδώ βρίσκεται στην αυλή αλληλεπιδρώντας με το περιβάλλον κοιτάζοντας γύρω, τρώγοντας από το έδαφος και εκτελώντας διάφορα άλλα animations όπως kick ground, rear up, bite επαναλαμβανόμενα.

➤ Bird Flock

Πιο πολλή ζωή στη πόλη δίνουν τα πουλιά πετώντας ρεαλιστικά πάνω από τα σπίτια και την αγορά. Στο Unity Asset Store έχω βρει μικρά πουλιά που πετούν μαζί σαν μια ομάδα με διάφορους τρόπους. Αυτοί οι τρόποι αφορούν τον τρόπο κίνησης τους δηλ. πόσο γρήγορα, πόσο έντονα, πόσο άγρια θα πετούν, την κατεύθυνση και πορεία που ακολουθούν.

Η τοποθέτηση τους μέσα στο project ήταν αρκετά απλή αφού απλά κάνεις drag and drop τη ανάλογη σκηνή στο σημείο που επιθυμείς. Στη συνέχεια αναθέτεις το script το οποίο δημιουργεί παιδιά (τα πουλιά) και προσαρμόζεις κάποιες παραμέτρους που υπάρχουν, όπως

για παράδειγμα πόσο diving να κάνουν, πόσο ψηλά να ανεβαίνουν, πόση απόσταση να έχει το ένα από το άλλο.

(α)



(β)



Σχήμα 5.3.8: (α)-(β) Bird Flock στη σκηνή

➤ Raven

Εκτός από το flock στη σκηνή υπάρχουν και κοράκια. Τα κοράκια ξεκινούν από μια idle κατάσταση και στη συνέχεια απογειώνονται από το έδαφος και πετούν μακριά μέχρι το πηγάδι. Τα animations που έχω χρησιμοποιήσει είναι το take off, landing, idle, flight, gliding flight. Για την απογείωση έχω χρησιμοποιήσει τη παράμετρο του Vector 3.up η οποία προσθέτει ύψος στο μοντέλο με βάση τον άξονα των y.



Σχήμα 5.3.9: Τα 3 κοράκια ξεκινούν από τις θέσεις τους πετώντας προς το πηγάδι.

```

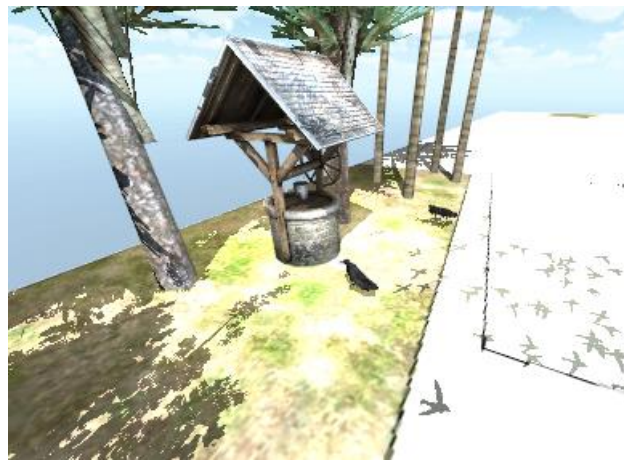
if (rav.GetCurrentAnimatorStateInfo (0).IsName ("take off")) {

    Quaternion targetRotation = Quaternion.LookRotation (target1.position -
transform.position);
    float str = Mathf.Min (strength * Time.deltaTime, 1);
    transform.rotation = Quaternion.Lerp (transform.rotation, targetRotation, str);
    transform.position += Vector3.up * Time.deltaTime;

}

```

Σχήμα 5.3.10: Αν βρίσκεσαι στο take off animation γύρισε το σώμα σου προς την κατεύθυνση που σου ανατίθεται, ανέβα πιο ψηλά από το έδαφος και πέταξε μακριά (flight, gliding flight).



Σχήμα 5.3.11: Όταν έχουν φθάσει στο πηγάδι.

Όταν το πρώτο κοράκι νιώσει την παρουσία του αλόγου Emily που θα έρθει κοντά του προς το πηγάδι θα κινηθεί μακριά περπατώντας προς τα άλλα 2 κοράκια. Εκεί από μόνο του θα αρχίσει να τρέφεται από το έδαφος (eat animation).



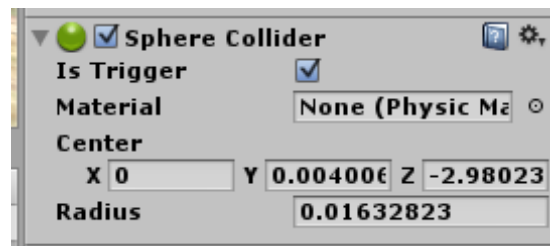
Σχήμα 5.3.12: Μόλις γίνει το collision με το άλογο το κοράκι περπατά προς τα άλλα δύο κοράκια όπου εκεί θα αρχίσει να τρέφεται από το έδαφος.

5.4 Collision Avoidance

Πολύ σημαντικός παράγοντας για την επιτυχία μιας γραφικής αναπαράστασης της καθημερινότητας των ανθρώπων είναι η παρουσίαση της συμπεριφοράς τους απόλυτα φυσιολογικά. Δηλ. η επικοινωνία μεταξύ τους, η αναγνώριση μιας φιγούρας για αποφυγή συγκρούσεων, αλλαγή πορείας, χαιρετισμός και διάφορες άλλες συμπεριφορές που μπορεί να έχουν. Το ίδιο ισχύει για τα ζώα. Πως θα αντιδράσουν δύο άλογα όταν βρεθούν το ένα απέναντι από το άλλο; Πως θα αντιδράσουν και θα κινηθούν οι κότες όταν τις πλησιάσει οποιαδήποτε φιγούρα; Πως θα επιτευχθεί η αναγνώριση ενός μοντέλου από ένα άλλο όταν πλησιάζει;

Αυτό επιτυγχάνεται με *collision avoidance* και *triggers*.

Σε κάθε μοντέλο ανατίθεται ένα collider (σε σχήμα sphere ή capsule) το οποίο μεγαλώνει ανάλογα με το μέγεθος του μοντέλου δημιουργώντας μια διάμετρο γύρω του. Σε αυτό το component προσθέτεις την παράμετρο isTrigger.



Όταν ένα μοντέλο έρθει σε σύγκρουση με την περίμετρο του collider του άλλου μοντέλου τότε μπορούμε να χειριστούμε τη σύγκρουση μέσω κάποιων συναρτήσεων όπως την OnTriggerEnter, η οποία καλείται σε κάθε αναγνώριση σύγκρουσης.

```
void OnTriggerEnter(Collider col){  
  
    if(col.gameObject.name.StartsWith("P_CY")){  
        animator.SetBool("idle", false);  
    }  
  
}
```

Σχήμα 5.4.1: Σε αυτό το παράδειγμα κώδικα βλέπουμε τη χρήση της OnTriggerEnter. Αν το όνομα του αντικείμενου με το οποίο έγινε η σύγκρουση αρχίζει από "P_CY" (στη περίπτωση αυτή εννοούμε τους ανθρώπους), τότε θέσε τη μεταβλητή idle σε false και μεταφέρσου στο επόμενο animation το οποίο καθορίστηκε από τον animator.

Δυστυχώς δεν έχω επιτύχει την πλήρη αποφυγή σύγκρουσης των ανθρώπων με τα ζώα και για αυτό το λόγο μέσα στην πόλη θα κυκλοφορούν μερικοί και όχι όλοι οι άνθρωποι εκτελώντας τις διάφορες εργασίες που τους έχουν ανατεθεί από το προηγούμενο project. Οι προσπάθειες και οι προσεγγίσεις που είχα κάνει ήταν πολλές αλλά παρόλα αυτά και λόγω χρόνου δεν το έχω επιτύχει. Στο project μου θα παρουσιάσω την συμπεριφορά των ζώων αλληλεπιδρώντας μεταξύ τους.

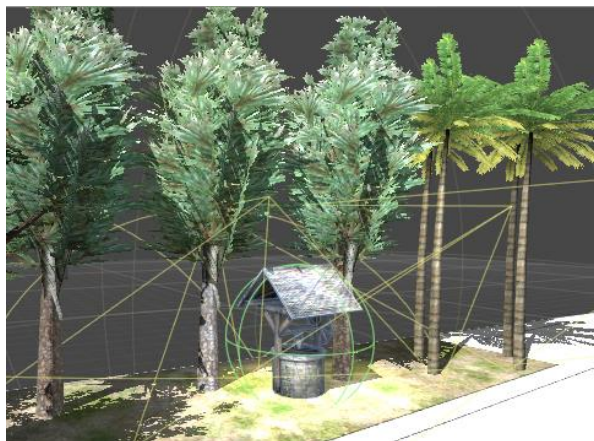
Όπως έχω αναφέρει και πιο πάνω σε κάθε επαφή του collider του emptyGameObject με το άλλο οι κόττες θα κινηθούν μακριά αποφεύγοντας το άλλο, ενώ αυτό ακολούθως θα κινηθεί μακριά. Το ίδιο θα συμβεί σε κάθε επαφή των δύο άλλων αλόγων οπότε το κάθε ένα θα ακολουθήσει την πορεία του σύμφωνα με το τι έχει καθοριστεί στον animator του κάθε ενός.

Όπως έχω εξηγήσει σε πιο πάνω υποκεφάλαιο τα μοντέλα κατευθύνονται προς τα σημεία τα οποία καθορίζω με το transform.position. Έτσι στο κάθε άλλο σημείο, το οποίο μπορεί να είναι ένα βαρέλι από την αγορά, ένα δοχείο, ένα δέντρο, ανατίθεται ένας collider έτσι ώστε να καθορίζονται οι επακόλουθες ενέργειες των ζώων. Για παράδειγμα το αλόγο Jenn όταν φθάσει στο βαρέλι στην άκρη της αγοράς θα αλλάξει συμπεριφορά εκτελώντας άλλα animations όπως sniff, eat, θα αλλάξει κατεύθυνση και θα κινηθεί προς τα δέντρα.



Σχήμα 5.4.2: Sphere Collider στο βαρέλι στην αγορά.

Το ίδιο συμβαίνει στα κοράκια. Όταν φθάσουν στο πηγάδι θα γίνει collision με το Sphere Collider του πηγαδιού όπου και θα αναθέσει τη μεταβλητή isFlying σε false. Έτσι τα κοράκια θα εκτελέσουν το landing animation τους και θα προσγειωθούν στο έδαφος. Όπως έχω εξηγήσει και πιο πάνω το κοράκι από τη μεριά που πλησιάζει το αλόγο θα ανιληφθεί το collision μαζί του και θα κατευθυνθεί προς τα άλλα δύο κοράκια.



Σχήμα 5.4.3: Το collider στο πηγάδι.

5.5 Tree Creator

Το Unity προσφέρει ένα εργαλείο δημιουργίας δέντρων, το *Tree Creator*.

Σε κάθε βήμα προσθέτεις ένα-ένα κομμάτι φτάνοντας σταδιακά στο τελικό αποτέλεσμα. Για αρχή δημιουργείται ο κορμός στον οποίο καθορίζεις το μέγεθος του σε ύψος και πάχος. Πολύ σημαντικό ανάλογα με το είδος του δέντρου είναι το σχήμα της βάσης του κορμού, πόσο κυκλικός θα είναι ή πόσο καμπυλωτός αλλά και ο κορμός ολόκληρος πόσο λυγιστός, πόσο ευθύς και πόσο καμπυλωτός. Επόμενο βήμα πριν την προσθήκη κλαδιών είναι η δήλωση του texture, δηλ. το χρώμα και σχέδιο του κορμού.



Σχήμα 5.5.1: Texture για πεύκο.

Προσθέτοντας τα κλαδιά στο επόμενο βήμα καθορίζεις και πάλι το texture, πόσο μεγάλα ή πόσο κοντά θα είναι, πόσο ίσια ή πόσο λυγιστά, όπως επίσης τη συχνότητά τους (πόσα κλαδιά) και τη διάταξη τους πάνω στο κορμό. Η διάταξη τους μπορεί να είναι random, whorled, alternate, opposite. Μέσω της διάταξης καθορίζεις μέσα από γραφικές παραστάσεις το πόσο ψηλά πάνω στον κορμό θα διαταχθούν, την γωνιά κλίσης τους, δηλ. πόσο ψηλά ή πόσο χαμηλά θα κλίνουνται.

Τελευταίο βήμα είναι η προσθήκη των φύλλων. Και πάλι καθορίζεις τη συχνότητα τους και τον τρόπο διάταξης τους πάνω στα κλαδιά, καθώς επίσης το μέγεθος και την κλίση τους.

Κατά την παραγωγή της συκιάς έχω χρησιμοποιήσει ένα έτοιμο texture του Unity για τον κορμό και τα κλαδιά. Για τα φύλλα έχω διαμορφώσει μια εικόνα που αναπαριστούσε ένα φύλλο συκιάς και κόβοντας το περίγραμμά της φορτώνεται στο Unity.

Μέσα από το Unity η φωτογραφία γίνεται transparent, και με τη δημιουργία ενός material δημιουργείται η αναπαράσταση του φύλλου.



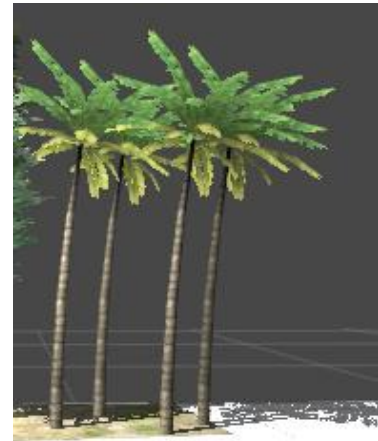
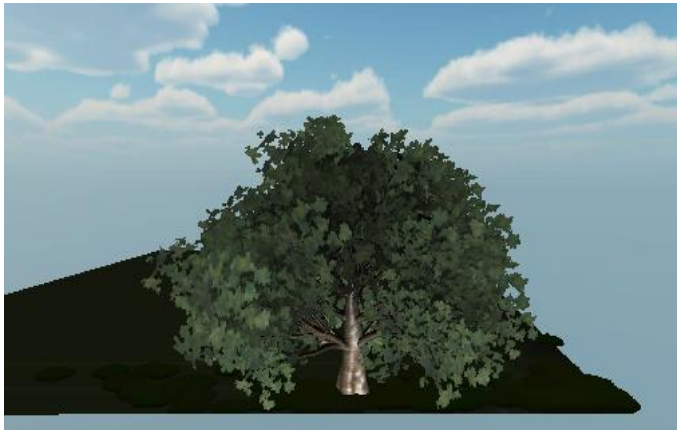
Σχήμα 5.5.2: Τελικό σχήμα συκιάς.

Κατά την παραγωγή του κυπαρισσιού δεν ήταν απαραίτητη η προσθήκη φύλλων αφού το material των φύλλων από μόνο του αποτελείτο από κλαδί και φύλλα κυπαρισσιού. Υψώνοντας τα κλαδιά όσο πιο ψηλά ήταν δυνατόν να ανεβούν και με μια whorled διάταξη προσπάθησα να πετύχω το καλύτερο δυνατό αποτέλεσμα για απεικόνιση ενός κυπαρισσιού.



Σχήμα 5.5.3: Τελικό σχήμα κυπαρισσιού.

Μέσα στα πλαίσια του project για περισσότερη ρεαλιστικότητα έχω προσθέσει και τις έτοιμες φοινικιές που προσφέρει το Unity. Επίσης έχω δημιουργήσει ένα είδος δέντρου για ομορφιά δίνοντας περισσότερο πράσινο στο περιβάλλον.



Τέλος, σε κάθε κομμάτι δέντρου υπάρχει η επιλογή προσθήκης άνεμου καθορίζοντας το πόσο δυνατά θα κινούνται τα κλαδιά, ο κορμός ή τα φύλλα, πόση κλίση θα έχουν και πόσο έντονο θα είναι το λύγισμα και κούνημα τους.

Κεφάλαιο 6

Συμπεράσματα

6.1 Αποτελέσματα	54
6.2 Εντυπώσεις από χρήστες	64
6.3 Μελλοντική δουλειά	64

6.1 Αποτελέσματα



Σχήμα 6.1: Η πόλη από ψηλά.



Σχήμα 6.2: Κοτούλες και άλογο μέσα στην αυλή (Φάση I").



Σχήμα 6.3: Κοτούλες και άλογο μέσα στην αυλή (Φάση 2^η).



Σχήμα 6.4: Κοτούλες και άλογο μέσα στην αυλή (Φάση 3^η).



Σχήμα 6.5: Κοτούλες και άλογο μέσα στην αγορά (Φάση 1η).



Σχήμα 6.6: Κοτούλες και άλογο μέσα στην αγορά (Φάση 2η).



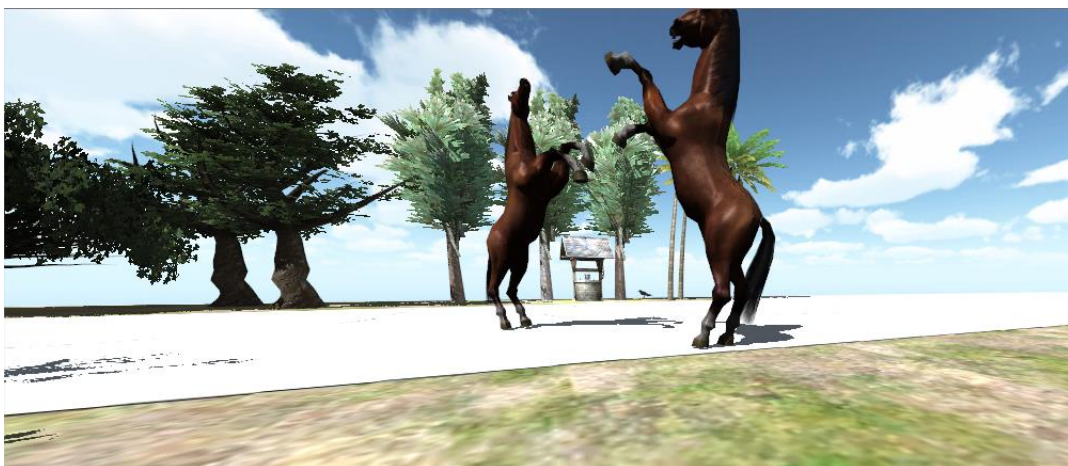
Σχήμα 6.7: Κοτούλες και άλογο μέσα στην αγορά (Φάση 3η).



Σχήμα 6.8: Κοτούλες και άλογο μέσα στην αγορά (Φάση 4η).



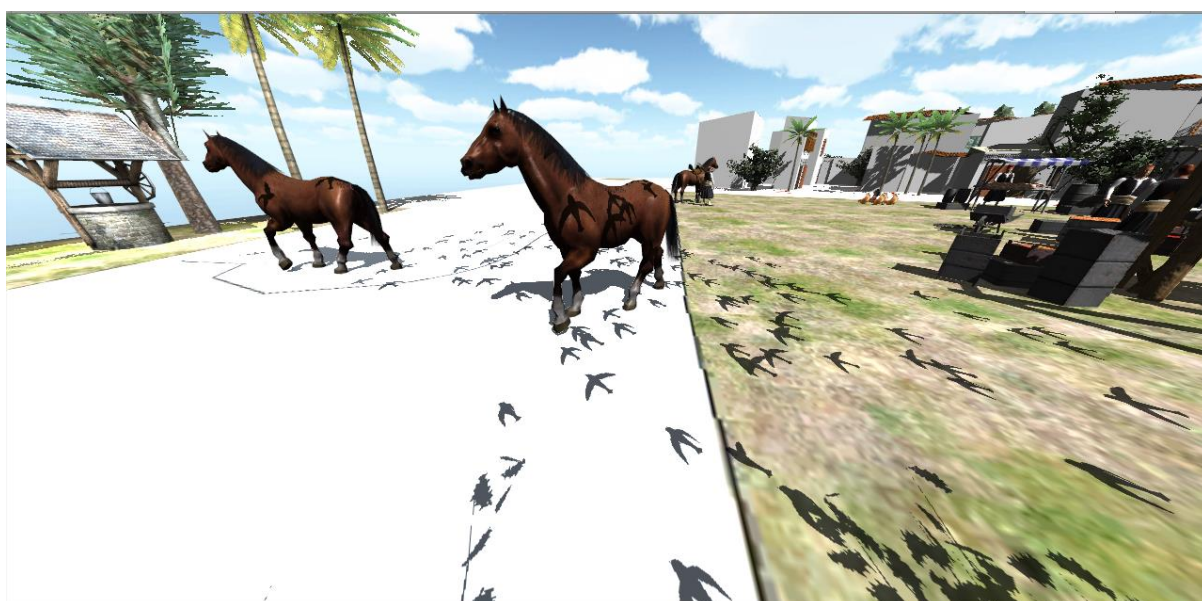
Σχήμα 6.9: Άλογο Emily(δεξιά) και άλογο Jenn(αριστερά), Φάση 1η.



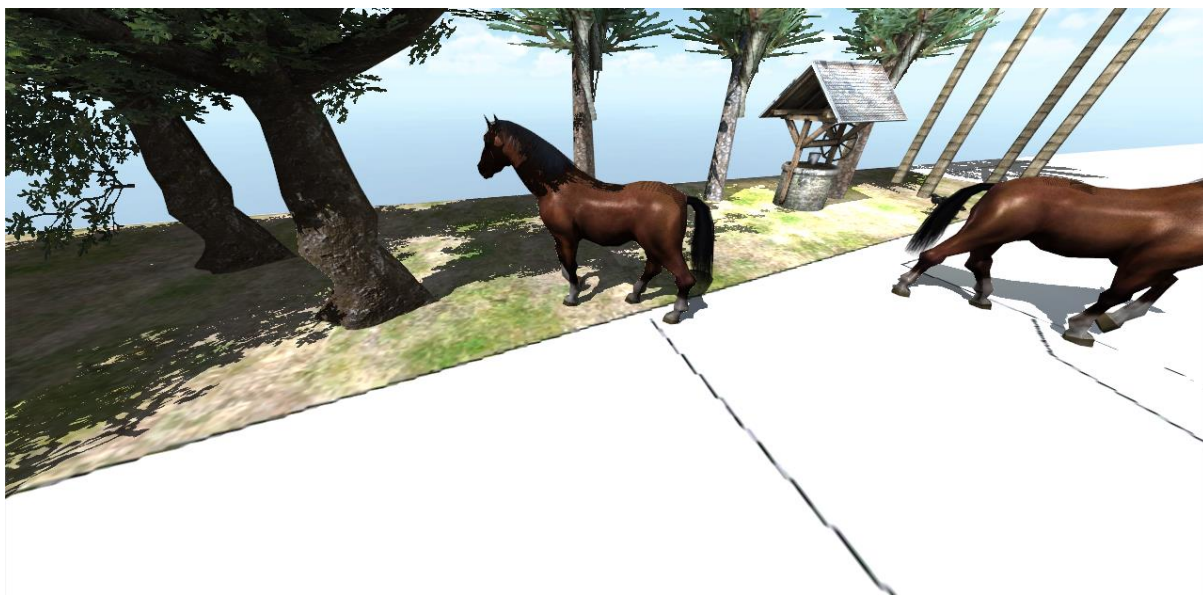
Σχήμα 6.10: Άλογο Emily(δεξιά) και άλογο Jenn(αριστερά), Φάση 2η.



Σχήμα 6.11: Άλογο Emily(δεξιά) και άλογο Jenn(αριστερά), Φάση 3η.



Σχήμα 6.12: Άλογο Emily(αριστερά) και άλογο Jenn(δεξιά), Φάση 4η.



Σχήμα 6.13: Άλογο Emily(δεξιά) και άλογο Jenn(αριστερά), Φάση 5η.



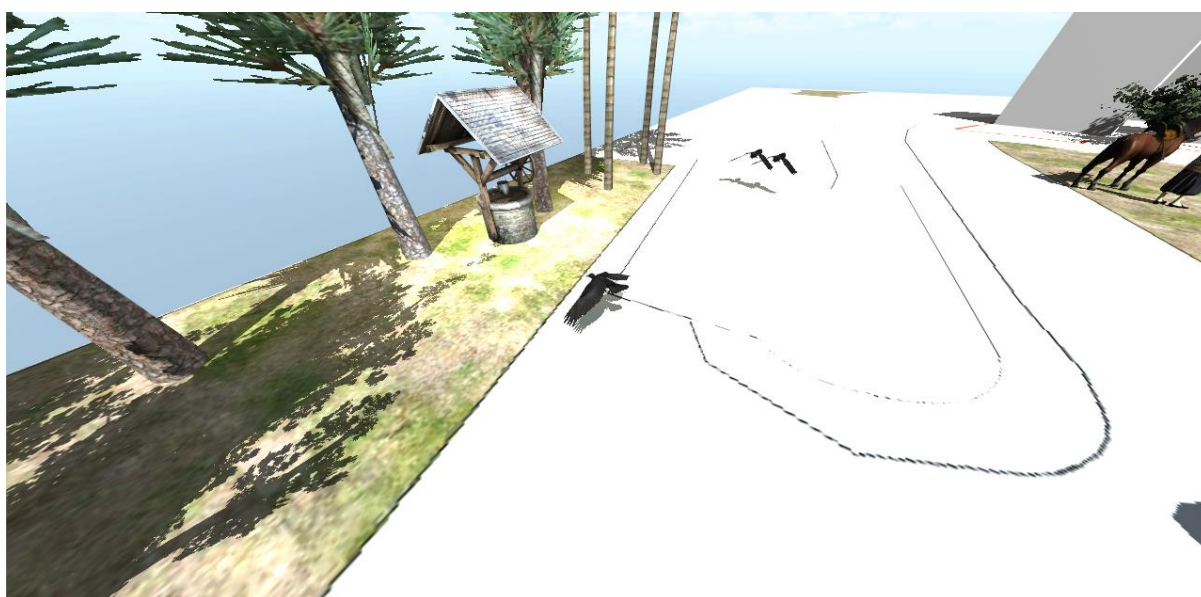
Σχήμα 6.14: Άλογο Emily(δεξιά) και άλογο Jenn(αριστερά), Φάση 6η.



Σχήμα 6.15: Κοράκι, Φάση 1^η



Σχήμα 6.16: Κοράκι, Φάση 2^η



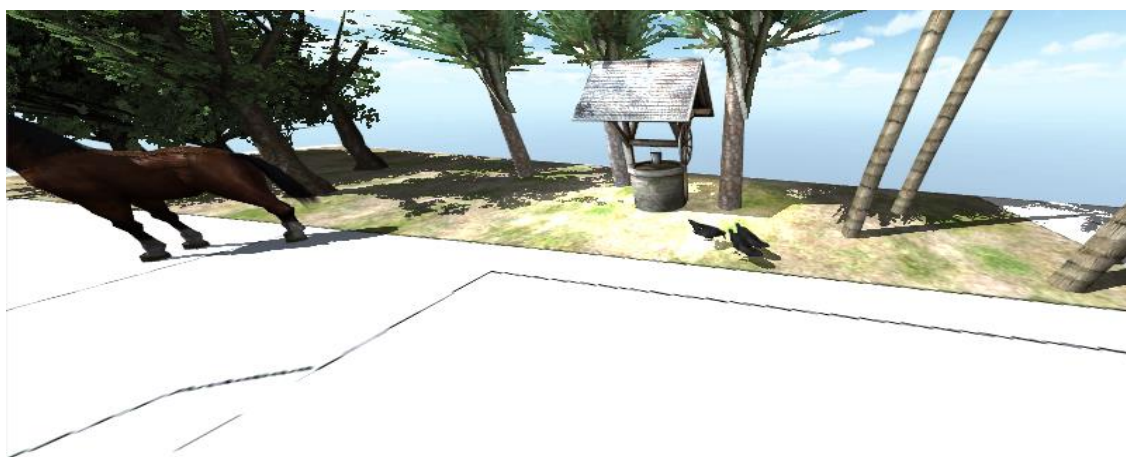
Σχήμα 6.17: Κοράκια, Φάση 3^η



Σχήμα 6.18: Κοράκια, Φάση 4"



Σχήμα 6.19: Κοράκι - Αλογο, Φάση 1"



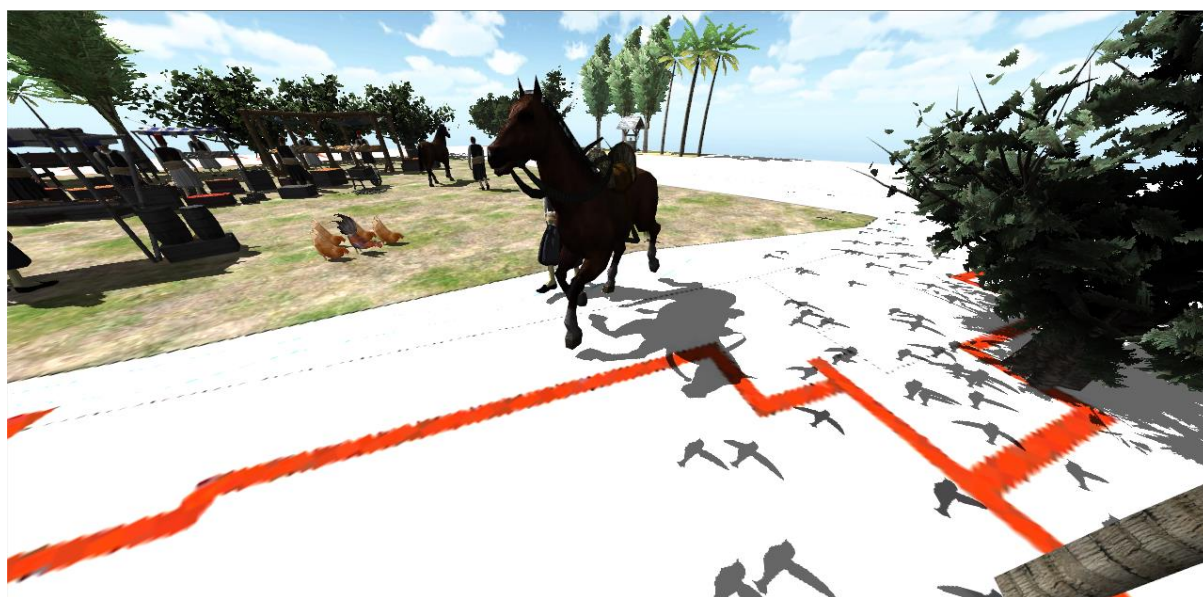
Σχήμα 6.20: Κοράκι - Αλογο, Φάση 5"



Σχήμα 6.21: Άνθρωπος – Άλογο κινούμενοι από το χρήστη (walk).



Σχήμα 6.22: Άνθρωπος – Άλογο κινούμενοι από το χρήστη ανάμεσα στα κτίρια της πόλης.



Σχήμα 6.23: Άνθρωπος – Άλογο κινούμενοι από το χρήστη (trot).

6.2 Εντυπώσεις από χρήστες

Μέσα στα πλαίσια της αξιολόγησης μου αποτελούσε η εμπλοκή διάφορων χρηστών αποσπώντας έτσι εντυπώσεις και απόψεις για τη ρεαλιστικότητα του project.

Οι χρήστες ήταν ένας φοιτητής και μία μαθήτρια λυκείου.

Πιο πολλή ενθουσιασμό έδειξε η μαθήτρια αφού πρώτη φορά ερχόταν σε επαφή με ένα τέτοιο περιβάλλον. Οι κινήσεις και οι συμπεριφορές των ζώων της φάνηκαν πολύ ρεαλιστικές και όμορφες. Όταν της δόθηκε η ευκαιρία να κινήσει από το πληκτρολόγιο το άλογο με τον άνθρωπο της έδωσε την εντύπωση ότι έπαιζε ένα παιχνίδι. Οι παρατηρήσεις της με ώθησαν να σκεφτώ περισσότερες ενέργειες που θα μπορούσε να έχει το άλογο όπως για παράδειγμα να κινηθεί πιο γρήγορα εκτελώντας άλλο animation.

Ενθουσιασμό και ικανοποίηση από αυτό που έβλεπε έδειξε και ο φοιτητής. Τρέχοντας το παιχνίδι από το πληκτρολόγιο με τη σειρά του παρατήρησε ότι τα μοντέλα μπορούσαν να περάσουν μέσα από τα κτίρια. Αυτό είναι ένας παράγοντας ο οποίος πρότεινε η συμφοιτήτρια μου Καλυψώ για μελλοντική βελτίωση και κάτι το οποίο θα προτείνω και εγώ.

Οι κινήσεις και οι συμπεριφορές των ζώων του φάνηκαν πολύ ρεαλιστικές κυρίως η κίνηση των κοτών μακριά όταν πλησίασε το άλογο και οι αντιδράσεις των δύο αλόγων όταν πλησίασαν ο ένας τον άλλο.

6.3 Μελλοντική δουλεία

Με τη σειρά μου εγώ λοιπόν παρατήρησα κάποιες συμπεριφορές που δεν μπόρεσα εγώ να επιτύχω και κάποιες λειτουργίες που θα μπορούσαν να προστεθούν από κάποιο άλλο φοιτητή οι οποίες δεν ήταν μέσα στις δικές μου αρμοδιότητες.

Ξεκινώντας, ενθαρρύνω την κατασκευή περισσότερων μοντέλων ζώων από κάποιο αρμόδιο όπως σκύλους, γαϊδούρια και καμήλες έτσι ώστε ένας άλλος φοιτητής να μπορέσει να δώσει περισσότερη ζωή στη πόλη και περισσότερες λειτουργίες στα ζώα. Θα μπορούσε να προστεθεί ένα κάρτο το οποίο θα κινεί ένα γαϊδούρι ή ένα άλογο μεταφέροντας τρόφιμα στην αγορά. Επίσης, αν είναι εφικτό, καλό θα ήταν η αποφυγή σύγκρουσης των μοντέλων με τα κτίρια.

Ταυτόχρονα η προσθήκη κι άλλων συμπεριφορών στα ήδη υπάρχοντα ζώα θα έδινε περισσότερο ρεαλισμό στη πόλη.

Βιβλιογραφία

- [1] C. Panayiotis, D. Kalypso, “Reconstruction of Everyday Life in 19th Century Nicosia”, 2014.
- [2] Youtube, <https://www.youtube.com/watch?v=Xx21y9eJq1U>, “Unity 4.0 - Mecanim Animation Tutorial”, 2014.
- [3] C.Yiorgos,
<http://docs.unity3d.com/Documentation/Manual/MecanimAnimationSystem.html>, “Mecanim Animation System”, 2014.
- [4] <https://www.assetstore.unity3d.com/en/>, “Unity Asset Store”, 2013.
- [5] <http://unity3d.com/unity/animation>, “Simple and Powerful animation Technology”, 2014
- [6] <http://docs.unity3d.com/Documentation/ScriptReference/Transform.html>, “Script Reference - Transform”, 2013
- [7] <http://docs.unity3d.com/Documentation/ScriptReference/Transform-rotation.html>, “Script Reference – Transform.rotation”, 2013
- [8] <http://docs.unity3d.com/Documentation/ScriptReference/AnimationInfo.html>, “Script Reference – AnimationInfo”, 2014
- [9] <http://docs.unity3d.com/Documentation/ScriptReference/AnimationState.html>, “Script Reference – AnimationState”, 2014
- [10] <http://docs.unity3d.com/Documentation/ScriptReference/Collider.html>, “Script Reference – Collider”, 2014

- [11] <http://docs.unity3d.com/Documentation/ScriptReference/Terrain.html>, “Script Reference - Terrain”, 2014
- [12] <http://docs.unity3d.com/Documentation/ScriptReference/Camera.html>, “Script Reference - Camera”, 2014
- [13] <http://docs.unity3d.com/Documentation/ScriptReference/Vector3.html>, “Script Reference – Vector 3”, 2014
- [14] Wikipedia, <http://en.wikipedia.org/wiki/>, 2014
- [15] <http://answers.unity3d.com/index.html>, “Unity Answers”, 2013-2014
- [16] M. Dikaiakou, A. Efthymiou, and Y. Chrysanthou, “Modelling the Walled City of Nicosia,” 2003.
- [17] Anonymous photographs, Nicosia 19th Century.

