



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Μορφοποίηση Εικονικού Περιβάλλοντος της Αρχαίας Αθήνας με την Κατασκευή Κτιρίων και Αγαλμάτων Formation of Ancient Athens' Virtual Environment with the Construction of Buildings and Statues
Όνοματεπώνυμο Φοιτητή	Γκίκας Γεώργιος
Πατρώνυμο	Φαίδων
Αριθμός Μητρώου	ΜΠΣΠ/17077
Επιβλέπων	Θεμιστοκλής Παναγιωτόπουλος, καθηγητής

Μεταπτυχιακή Διατριβή

(υπογραφή)

Θεμιστοκλής
Παναγιωτόπουλος
Καθηγητής

Τριμελής Εξεταστική Επιτροπή
(υπογραφή)

Άγγελος Πικράκης
Επίκουρος Καθηγητής

Γκίκας Γεώργιος

(υπογραφή)

Δημήτριος Αποστόλου
Αναπληρωτής Καθηγητής

Πίνακας Περιεχομένων

1. ΠΕΡΙΛΗΨΗ	6
2. ABSTRACT	7
3. ΕΙΣΑΓΩΓΗ	8
3.1. ΓΕΝΙΚΑ-ΟΡΟΛΟΓΙΑ	8
3.2. ΠΑΙΧΝΙΔΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ	8
3.2.1. ΤΑ ΠΡΩΤΑ ΠΑΙΧΝΙΔΙΑ.....	8
3.3. ΑΝΩΤΕΡΑ ΕΠΙΠΕΔΑ ΠΥΡΑΜΙΔΑΣ ΜΑΣΛΟΟΥ	8
3.3.1. ΑΝΑΓΚΕΣ ΕΚΤΙΜΗΣΗΣ Ή ΑΝΑΓΝΩΡΙΣΗΣ.....	8
3.3.2. ΑΝΑΓΚΕΣ ΟΛΟΚΛΗΡΩΣΗΣ Ή ΑΥΤΟΠΡΑΓΜΑΤΩΣΗΣ...9	
3.4. ΕΜΠΕΙΡΙΑ ΚΑΙ ΦΗΜΗ	9
3.4.1. ΟΡΙΣΜΟΣ ΕΜΠΕΙΡΙΑΣ.....	9
3.4.2. ΣΥΓΚΡΙΣΗ ΕΜΠΕΙΡΙΑΣ ΚΑΙ ΠΕΙΡΑΣ.....	9
3.4.3. ΟΡΙΣΜΟΣ ΦΗΜΗΣ.....	9
3.5. ΕΠΙΛΟΓΗ ΜΗΧΑΝΗΣ ΠΑΙΧΝΙΔΙΟΥ	10
3.6. ΓΕΝΙΚΗ ΙΔΕΑ ΚΑΙ ΙΣΤΟΡΙΑ	10
3.6.1. ΧΡΥΣΟΣ ΑΙΩΝΑΣ ΤΟΥ ΠΕΡΙΚΛΗ.....	10
3.6.2. ΕΝΣΩΜΑΤΩΣΗ ΜΕ ΤΗ UNITY.....	10
4. ΥΛΟΠΟΙΗΣΗ	12
4.1. ΕΙΣΑΓΩΓΙΚΗ ΕΙΚΟΝΑ	12
4.2. ΕΠΕΞΗΓΗΣΗ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ	13
4.2.1. ΕΜΠΕΙΡΙΑ ΚΑΙ ΦΗΜΗ.....	13
4.2.2. ΧΡΟΝΟΣ ΚΑΙ ΕΠΟΧΕΣ.....	13
4.2.3. ΕΝΗΜΕΡΩΣΗ ΠΑΡΑΤΗΡΗΤΗ.....	14
4.3. ΔΟΜΗ ΚΩΔΙΚΑ	15
4.3.1. ΕΔΑΦΟΣ ΚΑΙ ΚΤΙΡΙΑ.....	15
4.3.2. ΚΑΜΕΡΑ.....	16
4.3.3. ΧΡΟΝΟΣ ΚΑΙ ΕΠΟΧΕΣ.....	18
4.3.4. ΚΤΙΡΙΑ ΥΠΟ ΚΑΤΑΣΚΕΥΗ.....	20
4.3.5. ΦΕΙΔΙΑΣ.....	23
4.3.6. ΟΜΑΔΑ ΧΤΙΣΤΩΝ.....	24
4.3.7. ΜΗΧΑΝΙΣΜΟΣ ΚΑΤΑΣΤΑΣΕΩΝ.....	25
4.3.8. ΕΜΠΕΙΡΙΑ ΚΑΙ ΦΗΜΗ.....	29

4.3.9. ΕΠΙΛΟΓΗ ΕΠΟΜΕΝΟΥ ΚΤΙΡΙΟΥ.....	31
4.3.10. ΑΓΑΛΜΑΤΑ.....	37
4.3.11. ΗΧΟΙ ΠΡΟΣΟΜΟΙΩΣΗΣ	43
4.3.12. ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	44
5. ΤΕΛΟΣ ΠΡΟΣΟΜΟΙΩΣΗΣ- ΣΥΜΠΕΡΑΣΜΑΤΑ-ΠΕΡΙΛΗΨΗ	47
6. ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	53
7. ASSETS	54
7.1. UNITY ASSETSTORE.....	54
7.2. ΜΙΧΑΜΟ.....	54
7.3. ΥΟΥΤΥΒΕ.....	54
8. Βιβλιογραφία.....	56

Πίνακας Εικόνων

Εικόνα 1 Εισαγωγή προσομοίωσης	12
Εικόνα 2 Εμπειρία και Φήμη	13
Εικόνα 3 Διεπαφή εποχής, μήνα και έτους	13
Εικόνα 4 Πληροφορίες που δίνονται από την προσομοίωση	15
Εικόνα 5 Το terrain χωρίς τα κτίρια και τους χαρακτήρες	15
Εικόνα 6 Αριστερά το Εργαστήριο και δεξιά το σπίτι του Φειδία.....	16
Εικόνα 7 Αριστερά η Ταβέρνα και δεξιά η Εστία	16
Εικόνα 8 Ο Φειδίας	22
Εικόνα 9 Η φάρμα όπως σχεδιάστηκε για τις ανάγκες της προσομοίωσης.....	35
Εικόνα 10 Ο μεγάλος ναός	35
Εικόνα 11 Οι ομοιότητες των αγαλμάτων που χρησιμοποιήθηκαν είναι μεγάλες	36
Εικόνα 12 Όλα τα αγάλματα τοποθετημένα στη σκηνή	37
Εικόνα 13 Αριστερά ο Ερμής , δεξιά ο Άρης	37
Εικόνα 14 Αριστερά το άγαλμα του Δορυφόρου, δεξιά το άγαλμα Κόρη	38
Εικόνα 15 Αριστερά ο άνδρας με το σκύλο και δεξιά η Αφροδίτη με το παιδί της.....	38
Εικόνα 16 Αριστερά η Νύμφη, δεξιά η Πηνελόπη	39
Εικόνα 17 Αρχιτεκτονική αγαλμάτων στη Unity	40
Εικόνα 18 Την ώρα της δουλειάς.....	43
Εικόνα 19 Στον ελεύθερο χρόνο	44
Εικόνα 20 Κατά τη διάρκεια της νύχτας.....	45
Εικόνα 21 Δύο από τους τρεις ναούς που χτίζονται στην προσομοίωση μαζί με τα αγάλματά τους	46
Εικόνα 22 Ο Θόλος και η κεντρική πλατεία.....	46
Εικόνα 23 Το αμφιθέατρο του Περικλή	47
Εικόνα 24 Αμφιθέατρο Περικλή και άγαλμα της Πηνελόπης.....	47
Εικόνα 25 Τα σπίτια της Αθήνας ολοκληρωμένα. Ο Φειδίας με την ομάδα βρίσκονται στο κέντρο της πλατείας, επομένως χτίζουν το Θόλο.	49
Εικόνα 26 Τα σπίτια, ο Θόλος και στο βάθος τα μικρά αγάλματα	49
Εικόνα 27 Ίδια προσομοίωση με την Εικόνα 20. Η ομάδα έχει κατασκευάσει και τη Φάρμα στο βάθος δεξιά.....	50
Εικόνα 28 Στη συνέχεια έχουν χτιστεί δυο ναοί. Η ομάδα βρίσκεται πίσω από τα αγάλματα και κατασκευάζει τον τελευταίο ναό.....	50
Εικόνα 29 Τέλος προσομοίωσης	51
Εικόνα 30 Τέλος προσομοίωσης 2.....	52
Εικόνα 31 Τέλος προσομοίωσης 3.....	52

1. ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική αναφορά έχει ως στόχο την περιγραφή και τη δημιουργία ενός εικονικού περιβάλλοντος στην Αρχαία Αθήνα. Αυτή η σημαντική πόλη του αρχαίου κόσμου που γέννησε τη Δημοκρατία και κυριάρχησε στο εμπόριο, τον πολιτισμό και τις τέχνες, έγραψε παράλληλα ιστορία με τον Χρυσό Αιώνα του Περικλή, ενός σημαντικού πολιτικού, ρήτορα και στρατηγού. Η ανάπτυξη των τεχνών και πολλών οικοδομημάτων γνωστών σε παγκόσμιο επίπεδο, συνέβη τα χρόνια της εξουσίας του.

Για την πειστικότερη αναπαράσταση της μορφοποίησης, δίνεται η ιδιότητα του Αρχιτέκτονα σε έναν εκ των χαρακτήρων της προσομοίωσης. Έτσι, η πόλη σχηματίζεται μπροστά στα μάτια του θεατή. Ο αρχιτέκτονας διαθέτει μια ομάδα χαρακτήρων-ατόμων που τον βοηθούν στην υλοποίηση των κατασκευών, που μέσα από την εμπειρία και την πείρα που αποκτούν σταδιακά στη διάρκεια της προσομοίωσης, καταλήγουν να έχουν μια πολύ καλή φήμη στην πόλη. Η υλοποίηση και κατασκευή οικοδομημάτων όμως, γίνεται παράλληλα με την υπόλοιπη ζωή τους, που αποτελείται από τον ελεύθερο χρόνο τους και την ξεκούραση τους.

Παράλληλα ο χαρακτήρας-αρχιτέκτονας ασχολείται με τη γλυπτική και τον ελεύθερο χρόνο του κοσμεί με αγάλματα την Αθήνα. Τα αγάλματα αυτά κάνουν πιο πειστικό το σκηνικό της αρχαίας Αγοράς της Αθήνας. Επίσης, ανάλογα με τη δραστηριότητα των χαρακτήρων έχουν χρησιμοποιηθεί και σχετικοί ήχοι προκειμένου ο δέκτης της εμπειρίας να βιώσει με κάθε δυνατό τρόπο τη ζωή της τότε Αθήνας.

Ο τρόπος ζωής του αρχιτέκτονα και των υπολοίπων χαρακτήρων διαφέρουν αρκετά. Εκτός από το προγραμματιστικό κομμάτι της εργασίας, σκοπός είναι να αναδειχθεί και ο τρόπος που οι πολίτες της Αρχαίας Αθήνας ζούσαν εκείνα τα χρόνια.

Η διπλωματική γίνεται για το Πρόγραμμα Μεταπτυχιακών Σπουδών του Πανεπιστημίου Πειραιώς.

2. ABSTRACT

This Master's thesis aims to describe and create a virtual environment in Ancient Athens. This important city of the ancient world, which gave birth to democracy and dominated trade, culture and the arts, also made history with the Golden Age of Pericles, an important politician, orator and general. The development of the arts and many world-famous buildings occurred during the years of his rule.

For the most convincing representation of the formatting, one of the characters of the simulation is given the status of Architect. Thus, the city is formed before the eyes of the spectator. The architect has a group of characters-individuals who help him in the implementation of the constructions, who through the experience that they gradually gain during the simulation, end up having a very good reputation in the city. The implementation and construction of buildings, however, takes place in parallel with the rest of their lives, which consists of their free time and their rest.

At the same time, the character-architect deals with sculpture and in his free time adorns Athens with statues. These statues make the setting of the ancient Agora of Athens more convincing. At the same time, depending on the activity of the characters, similar sounds have been used in order for the recipient to experience in every possible way the life of Athens back then.

The lifestyle of the architect and the other characters are quite different. Apart from the programming part of the work, the aim is to highlight the way the citizens of Ancient Athens lived in those years.

The diploma is for the Postgraduate Program of the University of Piraeus.

3. ΕΙΣΑΓΩΓΗ

3.1. ΓΕΝΙΚΑ-ΟΡΟΛΟΓΙΑ

3.2. ΠΑΙΧΝΙΔΙΑ ΠΡΟΣΟΜΙΩΣΗΣ

Τα παιχνίδια προσομοίωσης (simulation video games) αποτελούν μια υποκατηγορία των βιντεοπαιχνιδιών και έχουν σχεδιαστεί προκειμένου να προσομοιώνουν δραστηριότητες του πραγματικού κόσμου. Τέτοιου είδους παιχνίδια αντιγράφουν διάφορες δραστηριότητες από τη ζωή μας για διάφορους σκοπούς όπως εκπαίδευση, ανάλυση ή πρόβλεψη. Σ' αυτά τα παιχνίδια ο παίκτης μπορεί ελεύθερα να χειρίζεται ένα χαρακτήρα ή ένα περιβάλλον, παραδείγματα είναι τα πολεμικά παιχνίδια, τα business και τα παιχνίδια ρόλων. Κατά το παρελθόν παιχνίδια προσομοίωσης έχουν χρησιμοποιηθεί ως μελέτες περιπτώσεων (case studies), ενώ παράλληλα έχουν γίνει συγκρίσεις των παιχνιδιών προσομοίωσης με άλλους τρόπους διδασκαλίας.

3.2.1. ΤΑ ΠΡΩΤΑ ΠΑΙΧΝΙΔΙΑ

Στα μέσα της δεκαετίας του '80 η εταιρεία Codemasters και οι Andrew Nicholas Oliver και Philip Edward Oliver παρουσίασαν έναν αριθμό παιχνιδιών με την ονομασία "Simulator" στον τίτλο, τα BMX Simulator(1986), Grand Prix Simulator(1986), και Pro Boxing Simulator(1988). Από τα πιο γνωστά παιχνίδια προσομοίωσης που όλοι έχουμε παίξει κάποια στιγμή στη ζωή μας είναι το SimCity, ένα παιχνίδι όπου ο παίκτης χτίζει μια πόλη και το οποίο κυκλοφόρησε το 1989, καθώς και η σειρά παιχνιδιών Sims που ανήκουν στην κατηγορία life simulation. Δημιουργός των παιχνιδιών αυτών είναι ο Will Wright.

3.3. ΑΝΩΤΕΡΑ ΕΠΙΠΕΔΑ ΠΥΡΑΜΙΔΑΣ ΜΑΣΛΟΥ

Ο Αμερικανός ψυχολόγος Αβραάμ Μάσλου (Abraham Maslow), ιδρυτής της Ανθρωπιστικής Ψυχολογίας, πραγματοποιώντας κλινικές μελέτες με σκοπό να αποκωδικοποιήσει τους παράγοντες που παρακινούν τις ανθρώπινες συμπεριφορές, κατέληξε σε πέντε κατηγορίες: τις φυσιολογικές, τις ανάγκες ασφάλειας ή σιγουριάς, τις κοινωνικές, τις ανάγκες εκτίμησης ή αναγνώρισης και τις ανάγκες ολοκλήρωσης. Στην παρούσα εργασία θα μιλήσουμε λίγο πιο αναλυτικά για τις δύο τελευταίες κατηγορίες αναγκών.

3.3.1. ΑΝΑΓΚΕΣ ΕΚΤΙΜΗΣΗΣ Ή ΑΝΑΓΝΩΡΙΣΗΣ

Κάθε άνθρωπος πρέπει να έχει μια εκτίμηση για τον εαυτό του, μια ανάγκη για αυτοεκτίμηση ή αναγνώριση. Αυτές οι ανάγκες συνδέονται με την ψυχολογική σύσταση του κάθε ανθρώπου. Η αυτοεκτίμηση όμως, είναι κατασκευασμένη εν μέρει από την εκτίμηση των άλλων, δηλαδή ο άνθρωπος πρέπει να αναγνωρίζεται, να έχει αυτοεκτίμηση, να αισθάνεται έγκυρος στην κοινωνία. Εάν

το άτομο δεν ικανοποιεί συχνά αυτή την ανάγκη εμφανίζει αισθήματα δυστυχίας, χαμηλής αυτοεκτίμησης και αισθάνεται κατώτερος από τους άλλους.

3.3.2. ΑΝΑΓΚΕΣ ΟΛΟΚΛΗΡΩΣΗΣ Ή ΑΥΤΟΠΡΑΓΜΑΤΩΣΗΣ

Το επίπεδο αυτό αποτελεί, κατά τον Μάσλοου, την ολοκλήρωση της προσωπικότητας του ανθρώπου, την κατάκτηση του οράματος και του ιδανικού. Ουσιαστικά είναι η πραγματοποίηση των πνευματικών, ψυχικών και σωματικών δυνατοτήτων του ανθρώπου, καθώς και η αίσθηση της πληρότητας που αυτή συνεπάγεται. Π.χ. η εύρεση εργασίας δίνει στον άνθρωπο μια εσωτερική πληρότητα.

3.4. ΕΜΠΕΙΡΙΑ ΚΑΙ ΦΗΜΗ

3.4.1. ΟΡΙΣΜΟΣ ΕΜΠΕΙΡΙΑΣ

Εμπειρία είναι η γνώση και η δεξιότητα που στηρίζεται στην παρατήρηση και στην πρακτική εξάσκηση και αποκτάται με την πάροδο του χρόνου. Π.χ. ένας τεχνίτης μαθαίνει τη δουλειά, παρατηρώντας τον επικεφαλής που είναι εξειδικευμένος πάνω στο εκάστοτε αντικείμενο, ενώ στη συνέχεια δοκιμάζει να φέρει εις πέρας τη δουλειά μόνος του. Προφανώς και απαιτείται ένα χρονικό διάστημα προκειμένου να βελτιωθεί, να μάθει καινούρια πράγματα και πολλές φορές να διδαχθεί από τα λάθη του. Στην επιστήμη η εμπειρία έχει στενή σχέση, το οποίο συμπληρώνει την παρατήρηση. Η εμπειρία έχει και δεύτερη έννοια που είναι το βίωμα μιας κατάστασης, είναι κομμάτι δηλαδή της συνολικής εμπειρίας του ανθρώπου. Στον κλάδο της φιλοσοφίας, εμπειρία είναι η γνώση που αποκτάται από την αισθητηριακή αντίληψη ή με την παρατήρηση. Τέλος στον κλάδο των online βιντεοπαιχνιδιών, ο κάθε παίκτης προσπαθεί να φτάσει στο στόχο του φέρνοντας σε πέρας διάφορες αποστολές μέσα από τις οποίες κερδίζει πόντους εμπειρίας (experience points ή xp).

3.4.2. ΣΥΓΚΡΙΣΗ ΕΜΠΕΙΡΙΑΣ ΚΑΙ ΠΕΙΡΑΣ

Η πείρα δηλώνει τη γνώση που αποκτάται σταδιακά από πολλές εμπειρίες ή το σύνολο των εμπειριών στη ζωή ενός ανθρώπου. Με άλλα λόγια, αποτελεί τη συσσωρευμένη γνώση ενός ανθρώπου, τη σύνθεση και το καταστάλαγμα των εμπειριών του. Π.χ. «είναι νέος ακόμα, δεν έχει την πείρα της ζωής», «οι συνάδελφοι του θεωρούν ότι έχει μακρά πείρα, πράγμα που του επιτρέπει να αναλάβει τη διεύθυνση της επιχείρησης.».

Το ουσιαστικό εμπειρία, ως επιμέρους έννοια, έχει πληθυστικό αριθμό αφού πολλές φορές κάνουμε λόγο για εμπειρίες, σε αντίθεση με το ουσιαστικό πείρα που συναντάται μόνο σε ενικό αριθμό καθώς είναι γενική – περιληπτική έννοια.

3.4.3. ΟΡΙΣΜΟΣ ΦΗΜΗΣ

Το ουσιαστικό φήμη έχει αρκετές ερμηνείες. Η πρώτη αναφέρεται στα λόγια που διαδίδονται από στόμα σε στόμα χωρίς να είναι διασταυρωμένα. Η δεύτερη, που ταιριάζει στη διατριβή, είναι η εντύπωση που επικρατεί στους άλλους ανθρώπους για την ποιότητα και το χαρακτήρα κάποιου, το «καλό» ή «κακό όνομα» που έχει αποκτήσει κάποιος.

3.5. ΕΠΙΛΟΓΗ ΜΗΧΑΝΗΣ ΠΑΙΧΝΙΔΙΟΥ

Η Unity 3D είναι μια κορυφαία μηχανή παιχνιδιών που αναπτύχθηκε από τη Unity Technologies το 2005. Η Unreal είναι επίσης μια κορυφαία μηχανή παιχνιδιών που αναπτύχθηκε από την Epic Games το 1998. Στα πλαίσια αυτής εργασίας προτιμήθηκε η Unity. Αυτό έγινε γιατί αρχικά, υποστηρίζει 3D Gaming Development όπως είναι και η εφαρμογή που έχουμε φτιάξει. Επιπλέον, υπάρχει σε όλες τις γνωστές πλατφόρμες (Android, iOS κ.ά.), ενώ αρχικά κυκλοφόρησε για MacOS. Ένα ακόμα σημαντικό πλεονέκτημα είναι ότι υπάρχει μια τεράστια κοινότητα που ασχολείται με τη δημιουργία παιχνιδιών σε Unity καθώς και ένας μεγάλος αριθμός από tutorials πράγμα που σημαίνει ότι η χρήση και η εξοικείωση μαζί με τη συγκεκριμένη πλατφόρμα είναι εύκολη, ειδικά για αρχάριους χρήστες. Τέλος, η χρήση της γλώσσας C#, που είναι πιο απλή γλώσσα προγραμματισμού από την C++ της Unreal Engine, είναι ένα επίσης σημαντικό πλεονέκτημα και δίνει την ευκαιρία και σε αρχάριους προγραμματιστές να ασχοληθούν με τη δημιουργία 2D και 3D παιχνιδιών. Για την εργασία χρησιμοποιήθηκαν αρκετές διαφορετικές εκδόσεις, αλλά η τελική της μορφή πραγματοποιήθηκε στην 2020.1.5.f.

3.6. ΓΕΝΙΚΗ ΙΔΕΑ ΚΑΙ ΙΣΤΟΡΙΑ

3.6.1. ΧΡΥΣΟΣ ΑΙΩΝΑΣ ΤΟΥ ΠΕΡΙΚΛΗ

Βρισκόμαστε στην Αθήνα κατά τον 5ο αιώνα π.Χ. το λεγόμενο και «Χρυσό Αιώνα του Περικλή». Η πόλη της Θεάς Αθηνάς θα καταφέρει να γνωρίσει, υπό την αρχηγία του περίφημου ρήτορα Περικλή, τη μεγαλύτερη δόξα που θα γνώριζε ποτέ! Έννοιες όπως Βουλή, Άρειος Πάγος, τέχνες, Θέατρο θα επηρεάσουν όχι μόνο την Ελλάδα, αλλά ολόκληρο τον κόσμο. Ο πολιτισμός της Αθήνας όμως θα φτάσει στο Ζενίθ του με τη γέννηση της Δημοκρατίας. Την ίδια περίοδο θα ζήσουν μεγάλοι Έλληνες όπως οι τραγωδοί Σοφοκλής (496-406 π.Χ.) και Ευριπίδης (480-406 π.Χ.), ο Θουκυδίδης (460-395 π.Χ.) ένας από τους κορυφαίους ιστορικούς της Αρχαίας Ελλάδας και ο Αριστοφάνης (450-380 π.Χ.) ο μεγαλύτερος κωμικός ποιητής της Αρχαίας Αθήνας. Τεράστια έργα θα δημιουργηθούν εκείνη την περίοδο που θα αλλάξουν όχι μόνο την ιστορία της Ελλάδας, αλλά και ολόκληρου του κόσμου. Το κορυφαίο μνημείο που κοσμεί την Αθήνα χτίστηκε προς τιμή της Θεάς Αθηνάς και είναι ο Παρθενώνας. Ο Ικτίνος και ο Καλλικράτης αποτέλεσαν τους αρχιτέκτονες και ο γλύπτης Φειδίας έφτιαξε τα γλυπτά που διακοσμούσαν τον εσωτερικό και εξωτερικό χώρο του μνημείου. Αποτελεί μέχρι και σήμερα το σήμα κατατεθέν της Αθήνας.

3.6.2. ΕΝΣΩΜΑΤΩΣΗ ΜΕ ΤΗ UNITY

Σκοπός αυτής της εργασίας είναι να κάνει μια αναπαράσταση για το πώς σχηματίστηκε η Αθήνα των αρχαίων χρόνων, με την παρουσία σημαντικών ανθρώπων που θέλησαν να αυξήσουν το κύρος και να αναδείξουν τη δημοκρατία και τον πολιτισμό της Ελλάδας. Στην προσομοίωση υπάρχουν δυο κατηγορίες- ομάδες:

- Ο Φειδίας, ένας αρχιτέκτονας-γλύπτης
- Μια ομάδα που αποτελείται από πέντε εργάτες-χτίστες.

Ο Φειδίας ως μέντορας και γνώστης της Αρχιτεκτονικής Επιστήμης, αποφασίζει ποιο θα είναι το επόμενο κτίριο που θα χτιστεί. Ενημερώνει τον Έξαρχο, τον αρχηγό της ομάδας και όλοι μαζί

κατευθύνονται προς το μέρος που θα κατασκευαστεί το νέο κτίριο. Αρχίζουν να εργάζονται, έχοντας δίπλα τους το Φειδία που τους δίνει οδηγίες για το πώς να προχωρήσουν στην κατασκευή.

Η ζωή στην Αρχαία Αθήνα, δεν είναι όμως μόνο δουλειά. Για αυτό το λόγο, φεύγοντας από το εργοτάξιο, οι δυο κατηγορίες-ομάδες, ακολουθούν ένα διαφορετικό πρόγραμμα:

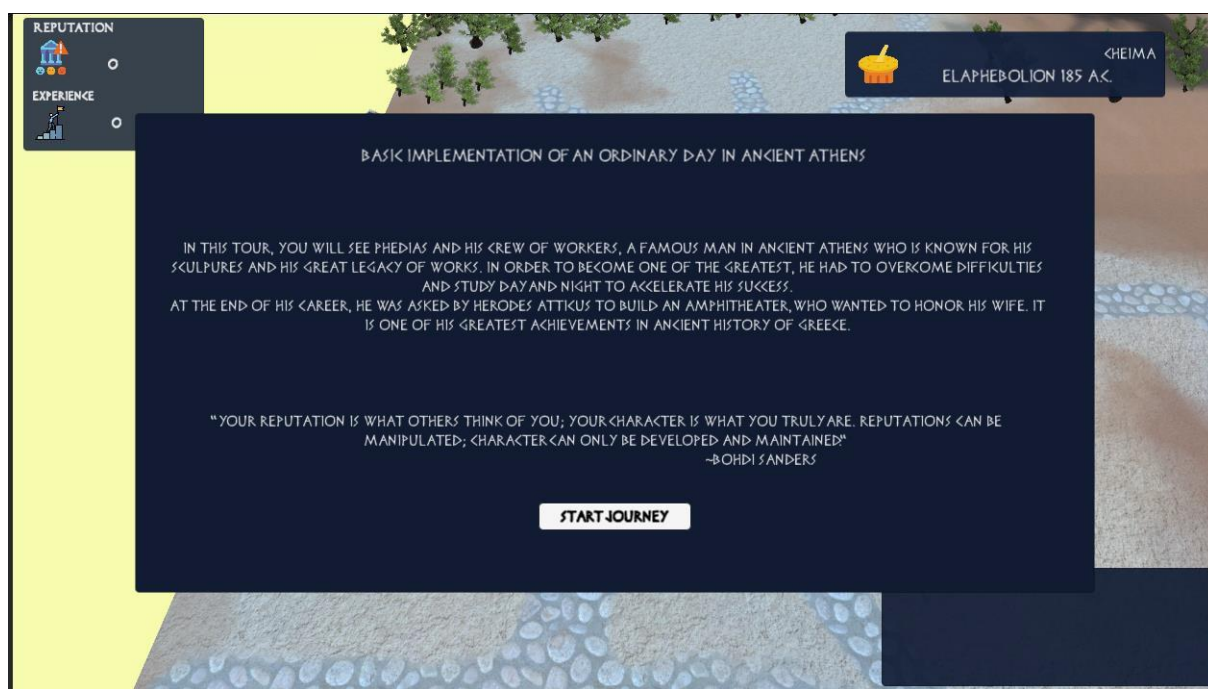
- Ο Φειδίας πηγαίνει σε έναν προσωπικό του χώρο που αποτελεί το Εργαστήρι του, στο οποίο σταδιακά κατασκευάζει αγάλματα. Όταν κατασκευάζει ένα νέο άγαλμα, αυτό εκτίθεται σε ένα συγκεκριμένο σημείο της πόλης.
- Η υπόλοιπη ομάδα, διασκεδάζει σε μια ταβέρνα της Αθήνας.

Το βράδυ όλοι γυρίζουν στα σπίτια τους για να κοιμηθούν. Ο Φειδίας ανήκοντας σε μια πιο υψηλή κοινωνική ομάδα, ζει μόνος του στο σπίτι του. Η υπόλοιπη ομάδα ζει σε ένα κοινό σπίτι που ονομάζεται Εστία.

4. ΥΛΟΠΟΙΗΣΗ

Παρακάτω αναλύεται ο κώδικας της εργασίας. Για τη διευκόλυνση της κατανόησης έχουν χρησιμοποιηθεί και εικόνες από το περιβάλλον.

4.1. ΕΙΣΑΓΩΓΙΚΗ ΕΙΚΟΝΑ



Εικόνα 1 Εισαγωγή προσομοίωσης

Στην εισαγωγή της προσομοίωσης όπως φαίνεται και στην παραπάνω εικόνα, ο παρατηρητής μπορεί να διαβάσει και να κατανοήσει κάποια βασικά πράγματα που θα ακολουθήσουν. Η ιστορία είναι ότι ο Φειδίας αφού απέκτησε κάποια εμπειρία, άρχισε να γίνεται γνωστός στο κοινό για τα κτίρια και τα αγάλματα που κατασκεύαζε. Η φήμη του έγινε αρκετή, ώστε ο Περικλής του ζήτησε να κατασκευάσει ένα μεγάλο αμφιθέατρο. Παράλληλα όμως όλο αυτό τον καιρό, κατασκευάζει αγάλματα συμβάλλοντας ιστορικά στο Χρυσό Αιώνα του Περικλή.

Στην πραγματικότητα, ο Φειδίας δεν ήταν αρχιτέκτονας, αλλά μόνο γλύπτης. Λέγεται ότι το Ωδείο του Περικλή αρχικά κατασκευάστηκε σε ένα πρώιμο στάδιο από το Θεμιστοκλή, μετά τη νίκη στη Ναυμαχία της Σαλαμίνας το 480 π.Χ. Άλλες πηγές αναφέρουν πως το Ωδείο χτίστηκε από τον ίδιο τον Περικλή για την τέλεση μουσικών εκδηλώσεων κατά τα Παναθήναια. Το Ωδείο πυρπολήθηκε το 85 π.Χ. από το Σύλλα, ξαναχτίστηκε ως «Ωδείο της Αρχαίας Αγοράς της Αθήνας περίπου το 15 π.Χ. και η τελική ανακατασκευή του έγινε από τον Ηρώδη τον Αττικό στη μνήμη της γυναίκας του το 2^ο αιώνα μ.Χ.

Στο τέλος παρατίθεται μια φράση ενός διάσημου συγγραφέα πολεμικών τεχνών, του Bohdi Sanders:

«Η φήμη σου είναι αυτό που οι άλλοι σκέφτονται για εσένα· Ο χαρακτήρας σου είναι αυτό που είσαι στην πραγματικότητα. Οι φήμες μπορεί να χειραγωγηθούν, ο χαρακτήρας μπορεί μόνο να αναπτυχθεί και να διατηρηθεί.»

Ο παρατηρητής μπορεί να προχωρήσει στην προσομοίωση χρησιμοποιώντας το κουμπί «START JOURNEY».

4.2. ΕΠΕΞΗΓΗΣΗ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ

4.2.1. ΕΜΠΕΙΡΙΑ ΚΑΙ ΦΗΜΗ

Πάνω αριστερά ο παρατηρητής μπορεί να βρει έναν πίνακα που έχει δυο διαφορετικές τιμές. Η μια αφορά τη φήμη και η άλλη την εμπειρία. Όπως έχει ήδη αναφερθεί



Εικόνα 2 Εμπειρία και Φήμη

4.2.2. ΧΡΟΝΟΣ ΚΑΙ ΕΠΟΧΕΣ

Είναι ένα απλός, αλλά πολύ χαρακτηριστικός τρόπος για να μεταφερθεί ο παρατηρητής στο περιβάλλον της Αρχαίας Αθήνας. Χρησιμοποιήθηκε ως βάση παιχνίδι Civilizations V, στο οποίο κάθε γύρος του παιχνιδιού διαδραματίζεται σε μια διαφορετική χρονιά και μια διαφορετική εποχή. Έτσι και η προσομοίωση της παρούσας διπλωματικής, διαδραματίζεται σε διαφορετικές εποχές, σε ένα συγκεκριμένο εύρος χρόνου.



Εικόνα 3 Διεπαφή εποχής, μήνα και έτους

Οι μήνες και οι εποχές είναι βασισμένες στο Αττικό Ημερολόγιο, το οποίο διέθετε 12 μήνες που ήταν αφιερωμένοι σε κάποιον από τους 12 Θεούς του Ολύμπου. Η νέα χρονιά ξεκινούσε από το

Εκατομβαίων, ένα μήνα του καλοκαιριού που ήταν αφιερωμένος στο θεό Απόλλωνα. Αναλυτικά, προκύπτουν τα εξής δεδομένα που χρησιμοποιήθηκαν για την εργασία:

Μήνας (ημέρες)	Εποχή	Θεός	Πληροφορίες
Εκατομβαίων (30)	Καλοκαίρι (Θέρος)	Απόλλωνας	Πρώτος μήνας του χρόνου ανάλογα με τη Νέα Σελήνη
Μεταγειτινίων (29)	Καλοκαίρι (Θέρος)	Απόλλωνας	Διεξαγωγή Παναθηναίων
Βοηδρομιών (30)	Καλοκαίρι (Θέρος)	Απόλλωνας	Διεξαγωγή Ελευσίνιων Μυστηρίων
Πυανεσιών (29)	Φθινόπωρο (Φθινόπωρον)	Απόλλωνας	Γιορτή Πυανέψια εκ μέρους του Θησέα για το θεό Απόλλωνα
Μαιμακτηριών (30)	Φθινόπωρο (Φθινόπωρον)	Δίας	Γιορτές Μαιμακτηριών για ήπιο χειμώνα και Πομπαία για μελλούμενη καλή σοδειά
Ποσειδεών (29)	Φθινόπωρο (Φθινόπωρον)	Ποσειδώνας	Κάθε 2-3 χρόνια, προστίθετο ένας εμβόλιμος μήνας, ο Ποσειδεών Β', ώστε η πρώτη ημέρα κάθε μήνα να συμπίπτει κατά το δυνατόν με τη Νέα Σελήνη
Γαμηλιών (30)	Χειμώνας (Χειμών)	Δίας και Ήρα	Γιορτή Γαμηλιών προς τιμήν του γάμου της Ήρας με το Δία
Ανθεστηριών (29)	Χειμώνας (Χειμών)	Διόνυσος	Γιορτή Ανθεστηριών που διαρκούσε τρεις μέρες. Διεξαγωγή Μικρών Ελευσίνιων
Ελαφηβολίων (30)	Χειμώνας (Χειμών)	Άρτεμις	Διεξαγωγή Ελαφηβολίων, γιορτή με θυσίες ελαφιών που πραγματοποιούνταν την έκτη μέρα του μήνα προς τιμήν της Άρτεμις
Μουνιχιών (29)	Άνοιξη (Έαρ)	Άρτεμις	Διεξαγωγή γιορτών: Αιάνεια, Βραυρώνια και Ολυμπιεία
Θαργηλιών (30)	Άνοιξη (Έαρ)	Άρτεμις και Απόλλωνας	Διεξαγωγή γιορτών: Θαργήλια, Βενδίδεια, Καλλυντήρια, Πλυντήρια, γιορτές προς τη θεά Αθηνά.
Σκιροφοριών (29)	Άνοιξη (Έαρ)	Αθηνά	Τέλος του έτους, πλήθος γιορτών αφιερωμένες στη θεά Αθηνά

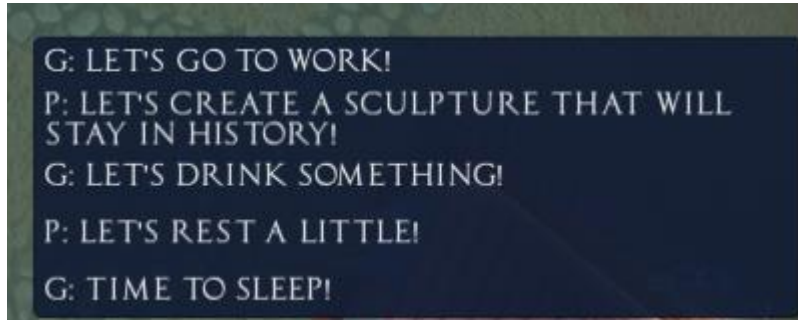
Πίνακας 1 Οι μήνες του Αττικού Ημερολογίου όπως χρησιμοποιήθηκαν στην εργασία

4.2.3. ΕΝΗΜΕΡΩΣΗ ΠΑΡΑΤΗΡΗΤΗ

Πρόκειται για ένα παράθυρο κάτω δεξιά στη διεπαφή χρήστη, όπου ενημερώνει τον παρατηρητή για όλες τις δράσεις που συμβαίνουν μέσα στην προσομοίωση. Τέτοιες δράσεις είναι η ενημέρωση για την αλλαγή της κατάστασης του Φειδία και των χτιστών, δηλαδή η μετάβαση τους για παράδειγμα από τη δουλειά στον ελεύθερο τους χρόνο. Μια άλλη ενημέρωση που προσφέρεται, είναι όταν κατασκευάζεται ένα καινούργιο κτίριο ή ένα νέο άγαλμα. Οι πληροφορίες χωρίζονται σε τρεις κατηγορίες:

- G: είναι οι πληροφορίες που λαμβάνεται από το Group, δηλαδή την ομάδα χτιστών
- P: είναι οι πληροφορίες που λαμβάνονται από το Φειδία

- Ε: είναι οι πληροφορίες που δίνει ο Έξαρχος στο Φειδία

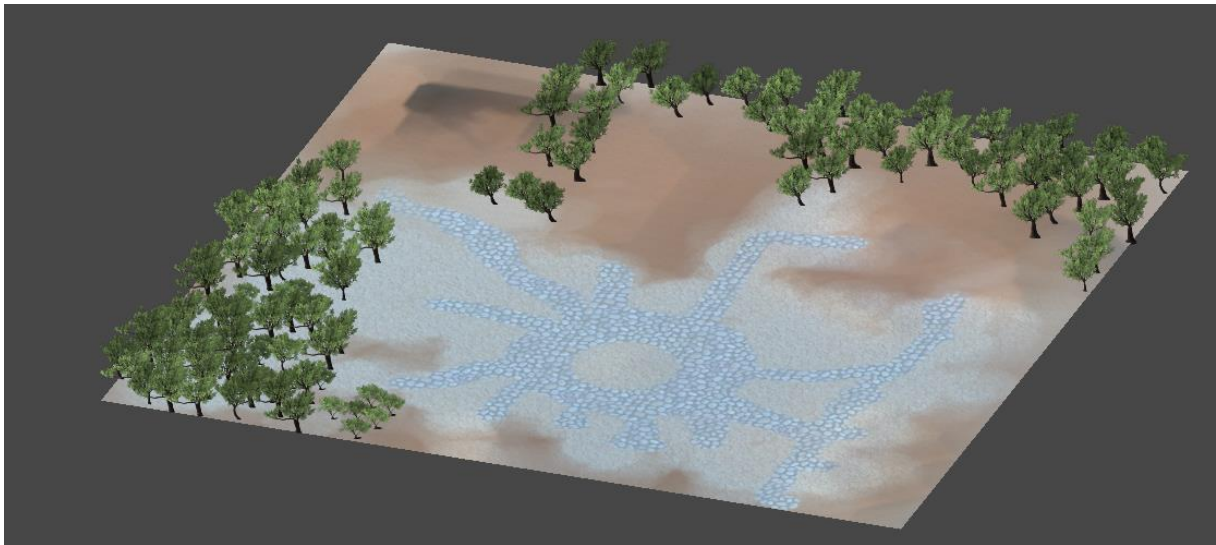


Εικόνα 4 Πληροφορίες που δίνονται από την προσομοίωση

4.3. ΔΟΜΗ ΚΩΔΙΚΑ

4.3.1. ΕΔΑΦΟΣ ΚΑΙ ΚΤΙΡΙΑ

Αρχικά ως έδαφος χρησιμοποιήθηκε ένα απλό 3D Terrain, διαστάσεων 150x150. Εμφανισιακά έπρεπε να επιτευχθεί η αίσθηση ενός δάσους στις δυο του άκρες που θα έμοιαζαν με ένα εξοχικό προάσιο της Αθήνας, ενώ στην υπόλοιπη επιφάνεια του έπρεπε να μοιάζει με το αρχαίο κέντρο της Αθήνας. Στο terrain χρησιμοποιήθηκαν τέσσερα διαφορετικά textures. Δυο που αναπαριστούν το χρώμα της εξοχής, σε διαφορετικές αποχρώσεις για να μοιάζει πιο αληθοφανές, ένα ανοιχτόχρωμο που μοιάζει με σκληρό έδαφος και ένα που μοιάζει με πέτρες, σχηματίζοντας έναν κεντρικό δρόμο στην αρχαία αγορά της Αθήνας.



Εικόνα 5 Το terrain χωρίς τα κτίρια και τους χαρακτήρες

Παράλληλα τοποθετήθηκαν διαφορετικά είδη του δέντρου της ελιάς στις περιοχές που ανήκουν στο δάσος-εξοχή.



Εικόνα 6 Αριστερά το Εργαστήρι και δεξιά το σπίτι του Φειδία



Εικόνα 7 Αριστερά η Ταβέρνα και δεξιά η Εστία

4.3.2. ΚΑΜΕΡΑ

Επειδή το περιβάλλον που σχεδιάστηκε έχει αρκετές λεπτομέρειες, έπρεπε ο παρατηρητής να έχει στη διάθεση του μια κάμερα που να μπορεί να κινείται ελεύθερα μέσα στη σκηνή. Για αυτό το λόγο, σχεδιάστηκε μια, που ο παρατηρητής με τα πλήκτρα A, S, D και W μπορεί να κινείται μέσα στη σκηνή.

Κρατώντας πατημένο το αριστερό κλικ και κινώντας ελαφρώς το ποντίκι, μπορεί να περιστρέψει την κάμερα για να δει περισσότερες λεπτομέρειες. Τέλος, μπορεί με τη χρήση του Mouse Wheel να κάνει zoom in και zoom out. Ο κώδικας για την επίτευξη της κάμερας είναι αναρτημένος παρακάτω:

```
public float panSpeed = 20f;
public float panBorderThickness = 10f;
public Vector2 panLimit = new Vector2(100, 100);
public float scrollSpeed = 2f;
public float minY = 20;
public float maxY = 100f;
private float X;
private float Y;
private float rSpeed = 3.5f;

private void Update()
{
    if (Time.timeScale != 0)
    {
        Vector3 pos = transform.position;

        if (Input.GetKey("w") || Input.mousePosition.y >= Screen.height - panBorderThickness)
        {
            pos.z += panSpeed * Time.deltaTime;
        }

        if (Input.GetKey("s") || Input.mousePosition.y <= panBorderThickness)
        {
            pos.z -= panSpeed * Time.deltaTime;
        }

        if (Input.GetKey("d") || Input.mousePosition.x >= Screen.width - panBorderThickness)
        {
            pos.x += panSpeed * Time.deltaTime;
        }

        if (Input.GetKey("a") || Input.mousePosition.x <= panBorderThickness)
        {
            pos.x -= panSpeed * Time.deltaTime;
        }

        float scroll = Input.GetAxis("Mouse ScrollWheel");
        pos.y -= scroll * scrollSpeed * 100f * Time.deltaTime;

        pos.x = Mathf.Clamp(pos.x, 20, panLimit.x);
        pos.y = Mathf.Clamp(pos.y, minY, maxY);

        if (Input.GetMouseButton(0)) {
            transform.Rotate(new Vector3(Input.GetAxis("Mouse Y") * rSpeed, -
            Input.GetAxis("Mouse X") * rSpeed, 0));
            transform.rotation = Quaternion.Euler(transform.rotation.eulerAngles.x,
            transform.rotation.eulerAngles.y, 0);
        }

        transform.position = pos;
    }
}
```

4.3.3. ΧΡΟΝΟΣ ΚΑΙ ΕΠΟΧΕΣ

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, για τη δημιουργία πιο πραγματικών συνθηκών και αφού πρόκειται για μορφοποίηση και προσομοίωση της Αρχαίας Αθήνας, υλοποιήθηκε η λογική του Αττικού Ημερολογίου. Αρχικά, είναι σημαντικό να αναφερθεί ότι κάθε ημέρα της προσομοίωσης διαρκεί 72 δευτερόλεπτα. Η πρώτη μέρα ξεκινάει στις 7 το πρωί. Όταν η μέρα ολοκληρωθεί, ο μετρητής δευτερολέπτων ξεκινάει πάλι από το μηδέν. Με την InvokeRepeating() καλούμε την CountTime() κάθε ένα δευτερόλεπτο. Έτσι τα δευτερόλεπτα μέσα στο παιχνίδι αυξάνονται κατά ένα ρεαλιστικά.

```
using System.Globalization;
using _Scripts.AncientBuilders.Mechanisms;
using UnityEngine;
using UnityEngine.UI;

public class TimeHandling : MonoBehaviour
{
    public int FullDayNightCycle { get; set; }
    public int Seconds { get; set; }
    private Text time;
    private Text timePassed;
    private string[] theros = {"Hekatombaion", "Metageitnion", "Boedromion"};
    private string[] fthinoporon = {"PaynepSION", "Maimakterion", "Poseideon"};
    private string[] xeima = {"Gamelion", "Anthesterion", "Elaphebolion"};
    private string[] ear = {"Mounichion", "Thargelion", "Skirophorion"};
    private int year;
    private string currentSeason;
    private int seasonInt;
    private string currentMonth;

    private void Start()
    {
        // The below line of code depicts the Day/Night Full Cycle
        FullDayNightCycle = 72;
        // We want the game to start at 7 o'clock in the morning:
        Seconds = FullDayNightCycle * 7 / 24;

        time = GameObject.Find(Constants.TIME_DISPLAY).GetComponent<Text>();
        timePassed = GameObject.Find(Constants.TIME_PASSED_DISPLAY).GetComponent<Text>();
        InitialRandomSeasonAndYear();
        time.text = currentMonth;
        timePassed.text = currentSeason + " " + year + " A.C.";
        InvokeRepeating("CountTime", 1.0f, 1.0f);
    }

    private void CountTime()
    {
        Seconds++;

        // Full Day/Night Cycle
        if (Seconds == FullDayNightCycle)
        {
            Seconds = 0;
            CalculateNextSeason();
        }
    }
}
```

Παράλληλα αρχικοποιούνται τέσσερις πίνακες που αντιπροσωπεύουν τις εποχές και περιλαμβάνουν τρεις μήνες η κάθε μια. Οι ονομασίες προέρχονται από τον Πίνακα 1 στο κεφάλαιο 4.2.2.

```
private void CalculateNextSeason()
{
    year--;
    seasonInt++;
    if (seasonInt == 5)
    {
        seasonInt = 1;
    }
    switch (seasonInt)
    {
        case 1:
            currentMonth = Constants.THEROS;
            currentSeason = theros[Random.Range(1, 3)];
            break;
        case 2:
            currentMonth = Constants.FTHINOPORON;
            currentSeason = fthinoporon[Random.Range(1, 3)];
            break;
        case 3:
            currentMonth = Constants.CHEIMA;
            currentSeason = xeima[Random.Range(1, 3)];
            break;
        case 4:
            currentMonth = Constants.EAR;
            currentSeason = ear[Random.Range(1, 3)];
            break;
    }

    time.text = currentMonth;
    timePassed.text = currentSeason + " " + year + " B.C.";
}
}
```

Η παραπάνω συνάρτηση που ονομάζεται CalculateNextSeason() είναι υπεύθυνη για τον υπολογισμό της επόμενης εποχής και μήνα. Αρχικά, το έτος εκκίνησης της προσομοίωσης είναι τυχαίο και βρίσκεται ανάμεσα στο 500 και το 450, όπως επίσης τυχαία είναι και η εποχή. Επομένως έχοντας ως βάση το αρχικό έτος, στο τέλος κάθε ημέρας καλείται η CalculateNextSeason() και κάνει τα ακόλουθα:

- Προσθέτει ένα έτος
- Μετακινεί την προσομοίωση στην επόμενη εποχή του χρόνου
- Επιλέγει τυχαία ένα μήνα αυτής της εποχής
- Ενημερώνει τα πεδία της διεπαφής χρήστη για τα νέα χρονολογικά δεδομένα

4.3.4. ΚΤΙΡΙΑ ΥΠΟ ΚΑΤΑΣΚΕΥΗ

Κατά τη διάρκεια της προσομοίωσης χτίζονται 11 διαφορετικά κτίρια:

- Πέντε σπίτια
- Τρεις ναοί
- Ένας θόλος
- Μια φάρμα
- Ένα αμφιθέατρο, δηλαδή το Ωδείο του Περικλή, που όταν χτιστεί ολοκληρώνεται και το παιχνίδι.

```
using UnityEngine;
namespace _Scripts.AncientBuilders.BuildingLogic
{
    public class BuildingsWeightGenerator : MonoBehaviour
    {
        public string buildingName;
        public int weight;
        public int daysForCompletion;
        public BuildingStatus status;

        public enum BuildingStatus
        {
            UnderConstruction, // Building that is under construction
            Built, // Buildings that have already been built
            Unavailable // Initial status of buildings
        }

        private void Start()
        {
            buildingName = name;
            weight = RandomizeWeight();
            daysForCompletion = GetDaysForCompletionBasedOnTag();
            status = BuildingStatus.Unavailable;
            MakeInternalGameobjectInactive();
        }

        private void MakeInternalGameobjectInactive()
        {
            foreach (Transform child in gameObject.transform)
            {
                child.gameObject.SetActive(false);
            }
        }
    }
}
```

Το κάθε κτίριο διαθέτει ένα σύνολο από χαρακτηριστικά. Αυτά είναι το όνομα του, το βάρος, δηλαδή το κόστος της κατασκευής του, τις μέρες που χρειάζεται να δουλέψουν όλοι για να κατασκευαστεί και την κατάστασή του, η οποία μπορεί να πάρει τρεις τιμές:

- UnderConstruction, δηλαδή το κτίριο αυτή τη στιγμή είναι υπό κατασκευή
- Built, που σημαίνει ότι το κτίριο έχει ήδη χτιστεί
- Unavailable, δηλαδή το κτίριο δεν είναι διαθέσιμο. Αυτή είναι και η αρχική κατάσταση των κτιρίων στην προσομοίωση.

Κατά την εκκίνηση του παιχνιδιού καθένα από τα έντεκα κτίρια που προαναφέρθηκαν παίρνουν μια αρχική τιμή στις παραπάνω μεταβλητές που αναφέρθηκαν. Τα πιο σημαντικά από αυτά είναι το βάρος και οι ημέρες που χρειάζονται οι χτίστες και ο Φειδίας για να τα κατασκευάσουν.

Οι μέρες κατασκευής είναι ανάλογες με τον τύπο του κτιρίου. Το βάρος είναι ένα εύρος ακεραίων αριθμών που διανέμεται με βάση την τυχαιότητα στους τύπους κτιρίων.

```
public int GetDaysForCompletionBasedOnTag()
{
    switch (tag)
    {
        case Constants.B_HOUSE:
            return Constants.D_HOUSE;
        case Constants.B_TEMPLE:
            return Constants.D_TEMPLE;
        case Constants.B_FARM:
            return Constants.D_FARM;
        case Constants.B_AMPHITHEATER:
            return Constants.D_AMPHITHEATER;
        default:
            return Constants.D_THOLOS;
    }
}

private int RandomizeWeight()
{
    switch (tag)
    {
        case Constants.B_HOUSE:
            return Random.Range(30, 60);
        case Constants.B_TEMPLE:
            return Random.Range(75, 95);
        case Constants.B_FARM:
            return Random.Range(50, 70);
        case Constants.B_AMPHITHEATER:
            return Random.Range(80, 100);
        default:
            return Random.Range(60, 85);
    }
}
}
```

Στον παρακάτω πίνακα φαίνονται αναλυτικά οι μέρες και τα βάρη για κάθε τύπο κτιρίου.

ΤΥΠΟΣ ΚΤΙΡΙΟΥ	ΜΕΡΕΣ ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ	ΒΑΡΟΣ
Σπίτι	2	5-15
Ναός	4	16-30
Θόλος	3	31-49
Φάρμα	3	31-49
Αμφιθέατρο	5	50-55

Πίνακας 2 Μέρες για ολοκλήρωση χτισίματος και βάρος ανά τύπο κτιρίου

Το παραπάνω script είναι τοποθετημένο σε κάθε ένα κτίριο. Οπότε κατά την εκκίνηση της προσομοίωσης το κτίριο παίρνει τιμές. Οι τιμές αυτές, πέρα από την κατάσταση του κτιρίου και τις ημέρες που απομένουν για να χτιστεί το κτίριο δεν αλλάζουν ποτέ.

4.3.5. ΦΕΙΔΙΑΣ

Πρόκειται για τον Μέντορα της προσομοίωσης. Είναι ο χαρακτήρας που δίνει εντολές στην ομάδα χτιστών και αποφασίζει ποιο κτίριο θα κατασκευαστεί μετά.



Εικόνα 8 Ο Φειδίας

4.3.6. ΟΜΑΔΑ ΧΤΙΣΤΩΝ

Η ομάδα των χτιστών αποτελείται από πέντε διαφορετικούς χαρακτήρες. Αυτοί είναι ο Έξαρχος που βρίσκεται αριστερά (με το πράσινο βέλος), πίσω του η Άρτεμις και προς τα δεξιά είναι ο Αριστόδημος, ο Φοίβος και η Δανάη.



Πίνακας 3 Η ομάδα των χτιστών

Ο Έξαρχος είναι ο βασικός χαρακτήρας της ομάδας. Όταν ο Φειδίας αποφασίσει ποιο κτίριο πρέπει να κατασκευαστεί, ενημερώνει τον Έξαρχο που ηγείται της υπόλοιπης ομάδας. Μια άλλη σημαντική αρμοδιότητα που έχει είναι να ενημερώσει το Φειδία όταν ολοκληρωθεί το κτίριο με το οποίο ασχολούνται την τρέχουσα περίοδο.

```
public void UpdateRemainingDays()
{
    if (remainingDays - 1 > 0)
    {
        remainingDays--;
    }
    else
    {
        InformMentorAboutEndOfConstruction();
    }
}

private void InformMentorAboutEndOfConstruction()
{
    phedias.GetComponent<Phedias>().ReportFromLeader();
}
```

Εντός κώδικα, ο Έξαρχος αναλαμβάνει να ενημερώσει και τη μεταβλητή του κτιρίου που χτίζεται τη δεδομένη στιγμή. Στο τέλος κάθε ημέρας, πριν μεταβούν στην κατάσταση ελεύθερου χρόνου τη μειώνει κατά ένα. Η λογική είναι ότι στην αρχή της πρώτης ημέρας ο Φειδίας θα αποφασίσει το επόμενο κτίριο που θα χτιστεί. Ενημερώνει τον Έξαρχο που τον ακολουθούν όλα τα υπόλοιπα μέλη της ομάδας. Στο τέλος της ημέρας ο Έξαρχος μειώνει τις μέρες που απομένουν για να χτιστεί το τρέχων κτίριο κατά μία. Αν είναι ήδη η τελευταία μέρα κατασκευής του, τότε το κτίριο εμφανίζεται στη σκηνή. Ο Έξαρχος ενημερώνει το Φειδία για την ολοκλήρωση του κτίσματος και ο δεύτερος ενημερώνει την εμπειρία και τη φήμη που αποκτήθηκε με τη δημιουργία του τελευταίου κτιρίου. Αμέσως μετά αποφασίζει το επόμενο κτίριο που θα χτιστεί με βάση το βάρος και τις ημέρες που χρειάζεται όπως θα γίνει γνωστό και στο κεφάλαιο 4.3.10.

4.3.7. ΜΗΧΑΝΙΣΜΟΣ ΚΑΤΑΣΤΑΣΕΩΝ

Ο μηχανισμός καταστάσεων διανέμεται σε τρεις διαφορετικές καταστάσεις:

- Work, όπου οι χαρακτήρες εργάζονται
- Free Time, όπου οι χαρακτήρες έχουν ελεύθερο χρόνο για διασκέδαση και ψυχαγωγία
- Home, όπου οι χαρακτήρες επιστρέφουν στο σπίτι τους για να κοιμηθούν.

```
public enum States
{
    Work,
    FreeTime,
    Home
}
```

Οι τρεις παραπάνω καταστάσεις εναλλάσσονται μεταξύ τους με τη σειρά που προαναφέρθηκαν. Δηλαδή οι χαρακτήρες μέσα στην ημέρα τους πηγαίνουν στη δουλειά (Work), μετά πηγαίνουν στο αντίστοιχο κτίριο για να περάσουν τον ελεύθερο χρόνο τους (Free Time) και στο τέλος της ημέρας επιστρέφουν στο κτίριο που αντιστοιχεί στο σπίτι τους για να κοιμηθούν.

Η ομάδα των χτιστών ακολουθεί διαφορετικό πρόγραμμα από το Φειδία παρόλα αυτά. Οι δυο διαφορετικές ομάδες χαρακτήρων πηγαίνουν την ίδια ώρα στη δουλειά, δηλαδή στις 8 το πρωί. Ο Φειδίας φεύγει από το χώρο εργασίας στις 17:00 το απόγευμα και σύμφωνα με την επόμενη κατάσταση πηγαίνει στο Εργαστήρι του για να δημιουργήσει ένα άγαλμα. Η ομάδα των χτιστών φεύγει κατά τις 18:00 από το χώρο εργασίας και πηγαίνει στην ταβέρνα για να διασκεδάσει. Οι δυο διαφορετικές ομάδες κατά τις 23:00 το βράδυ επιστρέφουν στα σπίτια και την Εστία αντίστοιχα για να κοιμηθούν.

Οι καταστάσεις αλλάζουν σύμφωνα με την ώρα που προαναφέρθηκε. Για παράδειγμα αν ο Φειδίας βρίσκεται στο χώρο εργασίας και η ώρα του παιχνιδιού γίνει 17:00 το απόγευμα, τότε θα κληθεί η `ChangeCurrentState()` με παράμετρο τη νέα κατάσταση, δηλαδή `FreeTime` και η τρέχουσα κατάσταση που μέχρι στιγμής ήταν η `Work`, θα δώσει τη θέση της στη `FreeTime`.


```

private void Start()
{
    phedias = GameObject.FindWithTag(Constants.MENTOR).GetComponent<Phedias>();
    startGame.transform.parent.gameObject.SetActive(true);
    Time.timeScale = 0;
    currentState = States.Home;
    StartCoroutine(ExecuteAfterTime(1.5f));
}

public void StartGame()
{
    Time.timeScale = 1;
    startGameTitle.GetComponent<Text>().text = Constants.START_GAME_TITLE;
    startGameDesc.GetComponent<Text>().text = Constants.START_GAME_DESC;
    startGame.transform.parent.gameObject.SetActive(false);
}

public void ChangeCurrentState(States newCurrentState)
{
    currentState = newCurrentState;
}

public States GetCurrentState()
{
    return currentState;
}

IEnumerator ExecuteAfterTime(float time)
{
    yield return new WaitForSecondsRealtime(time);
}

GameObject.FindWithTag(Constants.MENTOR).GetComponent<BuildDecisionMaker>().DecisionMaking();

GameObject.FindWithTag(Constants.MENTOR).GetComponent<StatueDecisionMaker>().StatueDecisionMaking();
}

```

Σύμφωνα λοιπόν με την τρέχουσα κατάσταση, ο Φειδίας και ο Έξαρχος διαθέτουν ένα script που ονομάζεται AncientWalk.cs το οποίο έχει μία Update function. Η συνάρτηση αυτή ελέγχει αν ο χαρακτήρας είναι στο κτίριο που ανήκει η τρέχουσα κατάστασή του. Δηλαδή αν ο Collider του Φειδία ή του Έξαρχου βρίσκονται στον Collider του κτιρίου. Στο παραπάνω παράδειγμα, εφόσον ο Φειδίας είναι στην κατάσταση FreeTime πλέον και όχι Work και δεν βρίσκεται στο Εργαστήρι του αλλά στο Εργοτάξιο, η Update() τον μετακινεί με τη συνάρτηση GoTo() στο Εργαστήρι του.

Ο Φειδίας στην GoTo() θα συνεχίσει να κινείται και να εκτελεί το animation «Walk» μέχρι να έρθει σε επαφή με τον Collider του Εργαστηρίου του. Τότε μόνο θα σταματήσει να κινείται. Θα παραμείνει εκεί μέχρι να αλλάξει και πάλι η κατάσταση σε Home, όπου και θα επαναληφθεί η παραπάνω διαδικασία.

```
public void Update()
{
    if (gameObject.tag.Equals(Constants.MENTOR))
    {
        if (Mechanisms.States.Work.Equals(statesMechanism.currentState) &&
            !work.GetComponent<ConstructionSite>().phediasOnConstructionSite)
        {
            GoTo(work, pAnimator, 5.0f);
        }
        else if (Mechanisms.States.Home.Equals(statesMechanism.currentState) &&
            !PhediasHome.atHome)
        {
            GoTo(pHome, pAnimator, 0);
        }
        else if (Mechanisms.States.FreeTime.Equals(statesMechanism.currentState) &&
            !PhediasStudio.atStudio)
        {
            GoTo(pStudio, pAnimator, 0);
        }
    }
    else if (gameObject.tag.Equals(Constants.LEADER))
    {
        if (Mechanisms.States.Work.Equals(statesMechanism.currentState) &&
            !work.GetComponent<ConstructionSite>().exarchosOnConstructionSiteE)
        {
            GoTo(work, bAnimator, 0);
        }
        else if (Mechanisms.States.Home.Equals(statesMechanism.currentState) &&
            !Estia.atEstia)
        {
            GoTo(bHome, bAnimator, 0);
        }
        else if (Mechanisms.States.FreeTime.Equals(statesMechanism.currentState) &&
            !Tavern.atTavern)
        {
            GoTo(bTavern, bAnimator, 0);
        }
    }
}
```

```
public void GoTo(Transform target, Animator animator, float offset)
{
    animator.SetBool(Constants._IS_WALKING, true);
    animator.Play("Walking");

    float step = moveSpeed * Time.deltaTime; // calculate distance to move

    var position = target.position;
    if (target.position.y > 0)
    {
        position.y = 0;
    }
    var targetPosition = new Vector3(position.x+offset, position.y, position.z);
    // targetPosition.y += 0.788f;

    transform.position = Vector3.MoveTowards(transform.position, targetPosition, step);

    if (transform.position != Vector3.zero)
    {
        transform.rotation = Quaternion.LookRotation(targetPosition - transform.position +
        deviation);
    }
}
```

4.3.8. ΕΜΠΕΙΡΙΑ ΚΑΙ ΦΗΜΗ

Όταν ολοκληρώνεται ένα κτίριο ο Φειδίας ενημερώνεται από τον Έξαρχο για την πράξη αυτή. Τότε πραγματοποιείται ένα σύνολο διεργασιών για να θεωρηθεί στον κώδικα το κτίριο ως Χτισμένο και να ενημερωθεί η εμπειρία και η φήμη των χαρακτήρων.

```
public void ReportFromLeader()
{
    var constructedBuilding =
gameObject.GetComponent<BuildDecisionMaker>().currentBuilding;
    foreach (Transform child in constructedBuilding.transform)
    {
        child.gameObject.SetActive(true);
    }
    constructedBuilding.GetComponent<BuildingsWeightGenerator>().status =
BuildingsWeightGenerator_BuildingStatus_Built;
    constructedBuilding.GetComponent<ConstructionSite>().enabled = false;
    UpdateExperienceAndReputation(constructedBuilding);
    chatBox.UpdateChatBox("E: Look Phedias! We just finished a new " +
constructedBuilding.tag);
    gameObject.GetComponent<BuildDecisionMaker>().DecisionMaking();
}

private void UpdateExperienceAndReputation(GameObject building)
{
    var buildingWg = building.GetComponent<BuildingsWeightGenerator>();
    var daysForCompletion = buildingWg.GetDaysForCompletionBasedOnTag();

    // Update Experience
    experience += buildingWg.weight * daysForCompletion;
    experienceText.text = experience.ToString(CultureInfo.CurrentCulture);
    // Calculate Deviation
    var devMultiplier = (100/experience) * Constants.DEVMULTIPLIER;
    var devInt = Random.Range(minDev, maxDev) - devMultiplier;
    var deviation = devInt / 100.0;

    // Update Reputation
    reputation = experience * deviation;
    if (reputation > 100)
        reputation = 100;
    reputationText.text = reputation.ToString(CultureInfo.CurrentCulture);
}
```

Αρχικά όταν ο Έξαρχος ενημερώνει το Φειδία, ο δεύτερος κάνει active το prefab του κτιρίου για να γίνει ορατό στη σκηνή. Επίσης ενημερώνει την κατάσταση του κτιρίου ως Built για να μην μπορεί να επιλεγεί από το μηχανισμό επιλογής επόμενου κτιρίου.

Μετά ενημερώνεται η εμπειρία και η φήμη. Αυτό επιτυγχάνεται με βάση τα χαρακτηριστικά του κτιρίου, δηλαδή το βάρος και τις ημέρες που χρειάστηκαν για την υλοποίησή του.

Η εμπειρία είναι το γινόμενο του βάρους επί τις ημέρες που χρειάστηκαν για την κατασκευή του κτιρίου. Η εμπειρία αυτή προστίθεται στη συνολική εμπειρία, η οποία ενημερώνει και το κείμενο που βλέπει ο παρατηρητής στη διεπαφή χρήστη.

Η φήμη είναι όμως κάτι τελείως διαφορετικό από την εμπειρία όπως έχει προαναφερθεί. Οπότε ο υπολογισμός της γίνεται με άλλο τρόπο. Αρχικά έχει ληφθεί υπόψιν ότι όταν ο Φειδίας ξεκινάει να χτίσει το πρώτο κτίριο με την ομάδα του, το αποτέλεσμα δεν θα είναι το ιδανικό. Δηλαδή αυτό που έχει σκεφτεί, είναι διαφορετικό από αυτό που υλοποιείται στην πραγματικότητα. Αυτή η απόκλιση όμως, όσο αποκτάει εμπειρία μειώνεται και μετά από ένα σημείο μηδενίζεται. Για να βγει λογική από τον υπολογισμό της απόκλισης ορίστηκε η μεταβλητή `devMultiplier` που αποτελείται από το αντίστροφο του ποσοστού της τρέχουσας εμπειρίας επί μια μεταβλητή που για να είναι λογικά τα αποτελέσματα ορίστηκε ως ο αριθμός 2. Μετέπειτα υπολογίζεται η μεταβλητή `devInt` που είναι ένας τυχαίος αριθμός από το 50 ως το 75 από τον οποίο αφαιρείται το `devMultiplier`. Ορίζουμε το τελικό `deviation`, δηλαδή την απόκλιση ως το πηλίκο του `devInt` με το 100.

Τέλος υπολογίζουμε το `reputation` ως το γινόμενο της τρέχουσας συνολικής εμπειρίας επί της απόκλισης. Δηλαδή πρόκειται για το αποτέλεσμα του Φειδία και της ομάδας, αλλά με βάση την εικόνα που δίνει σε έναν τρίτο, για παράδειγμα σε κάποιον άλλο πολίτη της Αρχαίας Αθήνας.

```
private void UpdateExperienceAndReputation(GameObject building)
{
    var buildingWg = building.GetComponent<BuildingsWeightGenerator>();
    var daysForCompletion = buildingWg.GetDaysForCompletionBasedOnTag();

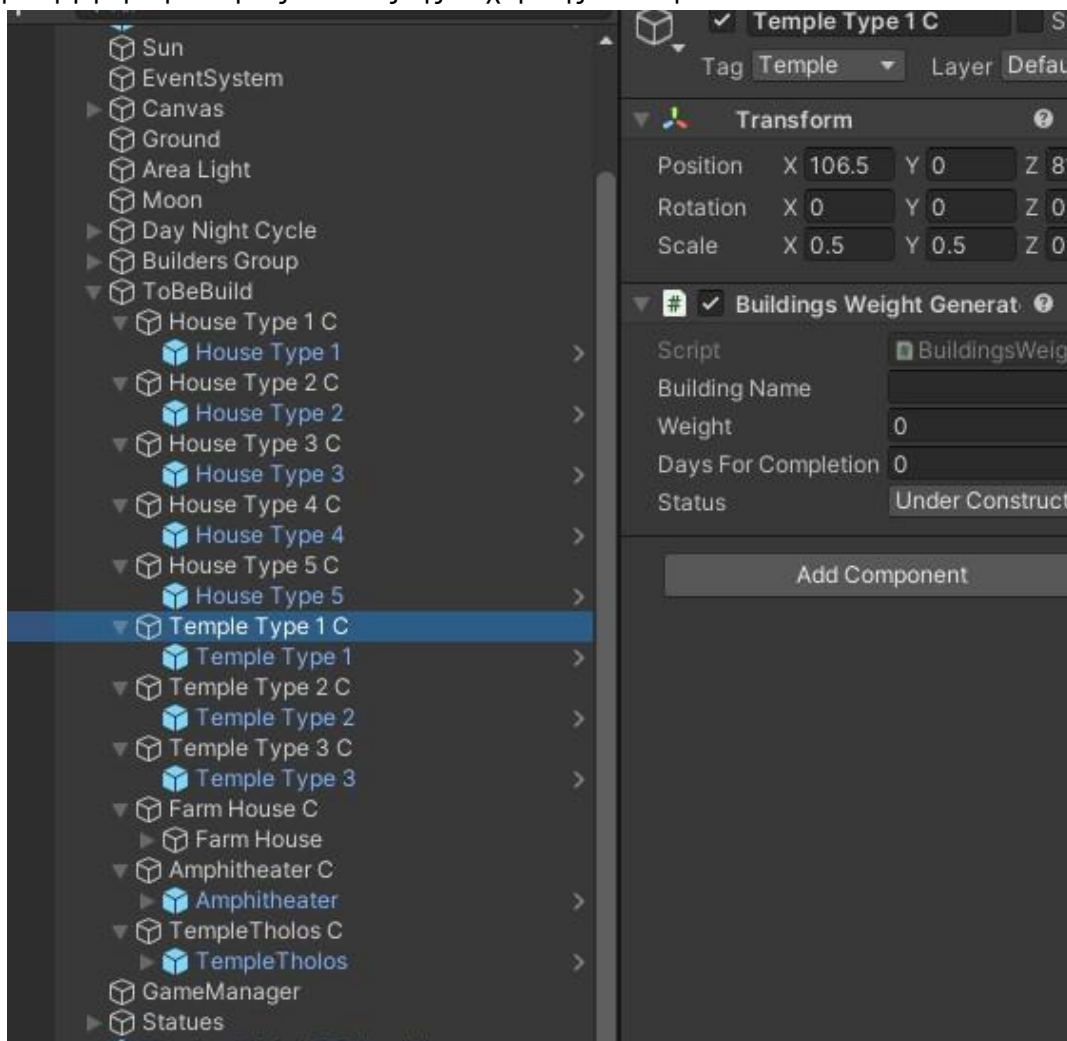
    // Update Experience
    experience += buildingWg.weight * daysForCompletion;
    experienceText.text = experience.ToString(CultureInfo.CurrentCulture);
    Debug.Log("Building weight: " + buildingWg.weight);
    Debug.Log("daysForCompletion: " + daysForCompletion);
    // Calculate Deviation
    var devMultiplier = (100/experience) * Constants.DEVMULTIPLIER;
    var devInt = Random.Range(minDev, maxDev) - devMultiplier;
    var deviation = devInt / 100.0;

    // Update Reputation
    reputation = experience * deviation;
    if (reputation > 100)
        reputation = 100;
    reputationText.text = reputation.ToString(CultureInfo.CurrentCulture);
}
```

Αξίζει να παρατηρηθεί ότι όσο περισσότερη εμπειρία αποκτούν ως ομάδα με την κατασκευή κτιρίων, τόσο πιο μεγάλος αριθμός είναι το `devMultiplier`. Επομένως το `devInt` κάθε φορά θα γίνεται μικρότερο, μέχρι να εξαλειφθεί τελείως. Η φήμη μετά από ένα σημείο μέσα στην προσομοίωση λόγω της μεγάλης εμπειρίας της ομάδας φτάνει το 100, που σημαίνει ότι ο Φειδίας με την ομάδα του είναι πλέον γνωστός σε όλη την Αθήνα. Παρόλα αυτά, η εμπειρία του συνεχίζει να αυξάνεται κανονικά, κάτι που είναι απόλυτα λογικό και θα συνέβαινε και στην πραγματικότητα.

4.3.9. ΕΠΙΛΟΓΗ ΕΠΟΜΕΝΟΥ ΚΤΙΡΙΟΥ

Η επιλογή του επόμενου κτιρίου είναι η μεγαλύτερη διαδικασία της προσομοίωσης. Όπως έχει ήδη αναλυθεί, το κάθε κτίριο κατά την εκκίνηση της προσομοίωσης παίρνει δυο τιμές: το βάρος (weight) και τις ημέρες που χρειάζονται για την ολοκλήρωση της κατασκευής του κτιρίου (daysForCompletion). Αυτές οι τιμές λαμβάνονται από το BuildingsWeightGenerator script. Αρχιτεκτονικά, όλα τα κτίρια που πρόκειται να χτιστούν τοποθετούνται κάτω από ένα κενό GameObject με όνομα ToBeBuild και ετικέτα Buildings. Κάθε κτίριο είναι ένα κενό GameObject κάτω από το ToBeBuild με ετικέτα τον τύπο του κτιρίου. Κάτω από κάθε ένα κτίριο υπάρχει και το prefab. Το ενδιάμεσο κενό GameObject δημιουργήθηκε για λόγους ευκολίας της διαχείρισης των κτιρίων.



Οπότε, υπάρχουν οι παρακάτω πληροφορίες:

Όνομα	Ετικέτα
House Type 1 C	House
House Type 2 C	House
House Type 3 C	House
House Type 4 C	House
House Type 5 C	House
Temple Type 1 C	Temple
Temple Type 2 C	Temple
Temple Type 3 C	Temple
Farm House C	Farm
Amphitheater C	Amphitheater
Tholos C	Tholos

Πίνακας 4 Τα κτίρια και οι προγραμματιστικές ονομασίες τους

Με βάση αυτά τα δεδομένα, μετά την έλευση 2 δευτερολέπτων γιατί δημιουργούνται συγκρούσεις λόγω χρόνου (Race Conditions), εκτελείται μια συνάρτηση που ονομάζεται NextBuildingDecider().

Η συνάρτηση αυτή έχει ως στόχο να βρει το επόμενο κτίριο που αξίζει περισσότερο να κατασκευαστεί. Αυτό σημαίνει ότι ψάχνει το κτίριο με το περισσότερο βάρος και τις λιγότερες μέρες για να αυξήσει πιο γρήγορα το experience, μοιραία να αποκτήσει ένα αρχικό reputation το οποίο δεν θα είναι το καλύτερο και σταδιακά με τη συνεχόμενη αύξηση της εμπειρίας να βελτιώσει όλα τα στατιστικά.

Η `NextBuildingDecider()` αποθηκεύει όλα τα παιδιά του `ToBeBuild GameObject` που έχουν ως status το `Unavailable`, σε ένα `Dictionary`. Στη λίστα δηλαδή δε συμπεριλαμβάνονται και τα κτίρια που πιθανόν να έχουν ήδη χτιστεί. Αρχικά κάνει έναν έλεγχο και αφαιρεί τα κτίρια που έχουν τις περισσότερες ημέρες μέχρι να ολοκληρωθούν. Αυτό είναι λογικό, γιατί η ομάδα δεν έχει αρκετή εμπειρία για να κατασκευάσει εξ αρχής έναν ναό για παράδειγμα. Οπότε πρέπει να αρχίσουν από απλά κτίσματα, όπως είναι τα σπίτια. Μετά την πρώτη αφαίρεση, ακολουθεί μια δεύτερη με βάση το βάρος. Δηλαδή από τα κτίρια με τις ίδιες ημέρες κατασκευής παραμένει μόνο αυτή που έχει το μεγαλύτερο βάρος. Αυτό είναι και το ιδανικό κτίριο που επιλέγει ο Φειδίας για να χτίσει.

```
private GameObject NextBuildingDecider(Dictionary<int, BuildingsWeightGenerator>
buildingsWeightGenerator)
{
    // Set initial value of buildingZero
    var buildingZeroKey = 0;
    var buildingZero = buildingsWeightGenerator[0];

    // Sort by Days of Completion
    foreach (var building in buildingsWeightGenerator.ToList())
    {
        // Find the minimum DaysForCompletion
        if (building.Value.daysForCompletion >
buildingZero.daysForCompletion)
        {
            buildingsWeightGenerator.Remove(building.Key);
            // Debug.Log("buildingZero DOC " + buildingZero.buildingName);
        }
    }

    // Sort by Building Weight
    foreach (var building in buildingsWeightGenerator.ToList())
    {
        // Find the maximum Building Weight for the least Days For Completion
        if (building.Value.weight > buildingZero.weight)
        {
            buildingsWeightGenerator.Remove(buildingZeroKey);
            buildingZero = building.Value;
            buildingZeroKey = building.Key;
            // Debug.Log("buildingZero BW " + buildingZero.buildingName);
        }
    }

    // Find the relevant GameObject from toBeBuilt
    var children = toBeBuild.GetComponentInChildren<Transform>(true);
    foreach (var t in children)
    {
        if (t.name.Equals(buildingZero.buildingName))
            return t.gameObject;
    }

    return null;
}
}
```


Δεν αρκεί όμως μόνο να βρεθεί το κτίριο, αλλά πρέπει να παραμετροποιηθεί έτσι ώστε να γνωρίζουν όλοι πού πρέπει να κατευθυνθούν για να χτίσουν. Οπότε το κτίριο παίρνει την καινούργια κατάσταση UnderConstruction. Κάθε φορά είναι εφικτό μόνο ένα κτίριο να έχει τη συγκεκριμένη κατάσταση. Επιπλέον προστίθεται ένας Collider στο κτίριο και ένα script που ονομάζεται OnConstructionSite, περισσότερο για να ενημερώνονται τα animations των χαρακτήρων.

```
private void AddCustomBoxCollider()
{
    currentBuilding.AddComponent<BoxCollider>();
    foreach (Transform child in currentBuilding.transform)
    {
        if (child.name.Equals(Constants.HOUSE_TYPE + 3) ||
            child.name.Equals(Constants.HOUSE_TYPE + 4) ||
            child.name.Equals(Constants.HOUSE_TYPE + 5) ||
            child.name.Equals(Constants.TEMPLE_TYPE + 2) ||
            child.name.Equals(Constants.TEMPLE_TYPE + 3))
        {
            currentBuilding.GetComponent<BoxCollider>().size = new Vector3(1, 1, 1);
        }
        else
        {
            currentBuilding.GetComponent<BoxCollider>().size = new Vector3(15, 15,
15);
        }
    }

    currentBuilding.GetComponent<BoxCollider>().isTrigger = true;
}

private Dictionary<int, BuildingsWeightGenerator> GetAllWeightGenerators()
{
    var buildings = new Dictionary<int, BuildingsWeightGenerator>();
    var id = 0;
    foreach (Transform building in toBeBuild.transform)
    {
        //Add only buildings that are not already built.
        if (building.GetComponent<BuildingsWeightGenerator>().status !=
            BuildingsWeightGenerator.BuildingStatus.Built)
        {
            buildings.Add(id, building.GetComponent<BuildingsWeightGenerator>());
            id++;
        }
    }

    return buildings;
}
```

Παράλληλα ενημερώνεται το AncientWalk script του Φειδία και του Έξαρχου για το νέο μέρος όπου πρέπει να τους οδηγήει η συνάρτηση GoTo().

Τέλος, ο Φειδίας ενημερώνει τον Έξαρχο για τα καινούργια δεδομένα, δηλαδή για τις ημερες που θα χρειαστούν για την κατασκευή του νέου κτιρίου και το όνομα του κτιρίου για να μπορεί να πηγαίνει ο Έξαρχος μέσω της AncientWalk.GoTo().

```
public class BuildDecisionMaker : MonoBehaviour
{
    private GameObject toBeBuild;
    private Dictionary<int, BuildingsWeightGenerator> buildingsWeightGenerators;
    public GameObject currentBuilding;

    public void DecisionMaking()
    {
        toBeBuild = GameObject.FindGameObjectWithTag
            (Constants.B_BUILDINGS);
        buildingsWeightGenerators = GetAllWeightGenerators();

        // Decision algorithm on how the selection of next building will be done
        currentBuilding = NextBuildingDecider(buildingsWeightGenerators);
        AddCustomBoxCollider();

        currentBuilding.AddComponent<ConstructionSite>();
        currentBuilding.GetComponent<BuildingsWeightGenerator>().status =
            BuildingsWeightGenerator.BuildingStatus.UnderConstruction;

        // Update workplace of Phedias (to walk to)
        gameObject.GetComponent<AncientWalk>().BuildingInitialization();

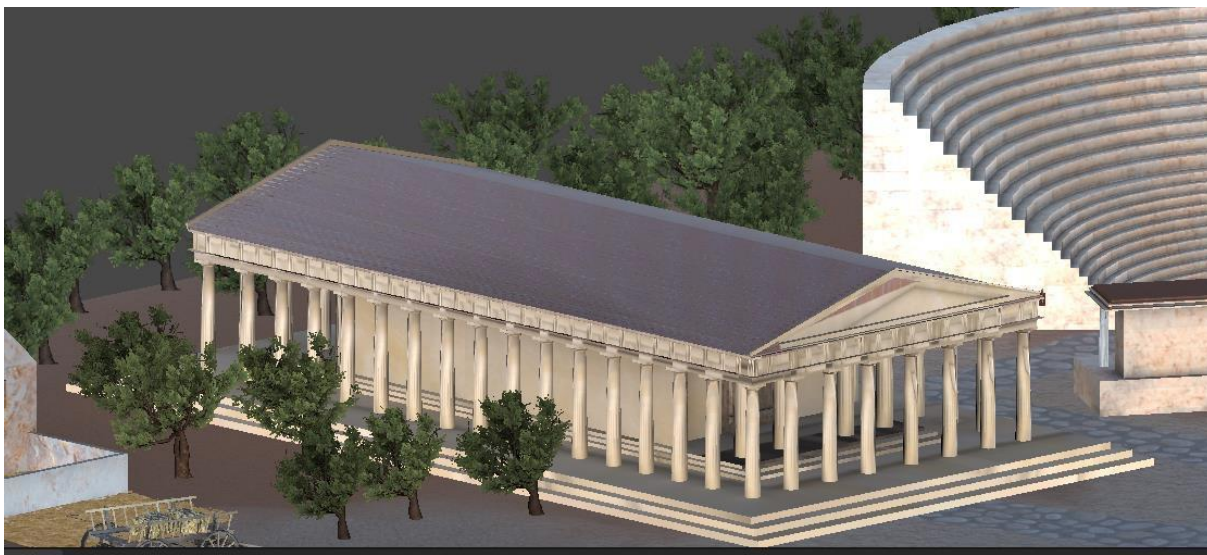
        // Inform Exarchos about Phedias's decision
        GameObject.FindWithTag(Constants.LEADER).GetComponent<Exarchos>().building =
        currentBuilding;

        GameObject.FindWithTag(Constants.LEADER).GetComponent<Exarchos>().remainingDays =
        currentBuilding.GetComponent<BuildingsWeightGenerator>().daysForCompletion;

        GameObject.FindWithTag(Constants.LEADER).GetComponent<AncientWalk>().BuildingInitiali
        zation();
    }
}
```



Εικόνα 9 Η φάρμα όπως σχεδιάστηκε για τις ανάγκες της προσομοίωσης



Εικόνα 10 Ο μεγάλος ναός

4.3.10. ΑΓΑΛΜΑΤΑ

Τα αγάλματα στην Αρχαία Ελλάδα έπαιζαν ιδιαίτερα σημαντικό ρόλο. Πολλοί γλύπτες κατάφεραν να αναδείξουν το ταλέντο τους και να αναπαραστήσουν με πολύ ρεαλιστικό τρόπο το ανθρώπινο σώμα και τα συναισθήματα. Η ύπαρξη τους στην εργασία είναι καθοριστική και συμβάλλει στην επίτευξη του στόχου για ρεαλιστικότερη απεικόνιση της Αρχαίας Αθήνας. Τα prefabs που βρέθηκαν είναι επίσης πολύ καλοσχεδιασμένα. Τα μέλη των αγαλμάτων είναι ολόκληρα και φαίνεται να έχουν μόλις χτιστεί. Δυστυχώς δεν υπήρχε η δυνατότητα να βρεθούν έγχρωμα αντίγραφα των αγαλμάτων, όπως ήταν και στην πραγματικότητα.



Εικόνα 11 Οι ομοιότητες των αγαλμάτων που χρησιμοποιήθηκαν είναι μεγάλες



Εικόνα 12 Όλα τα αγάλματα τοποθετημένα στη σκηνή



Εικόνα 13 Αριστερά ο Ερμής , δεξιά ο Άρης



Εικόνα 14 Αριστερά το άγαλμα του Δορυφόρου, δεξιά το άγαλμα Κόρη



Εικόνα 15 Αριστερά ο άνδρας με το σκύλο και δεξιά η Αφροδίτη με το παιδί της



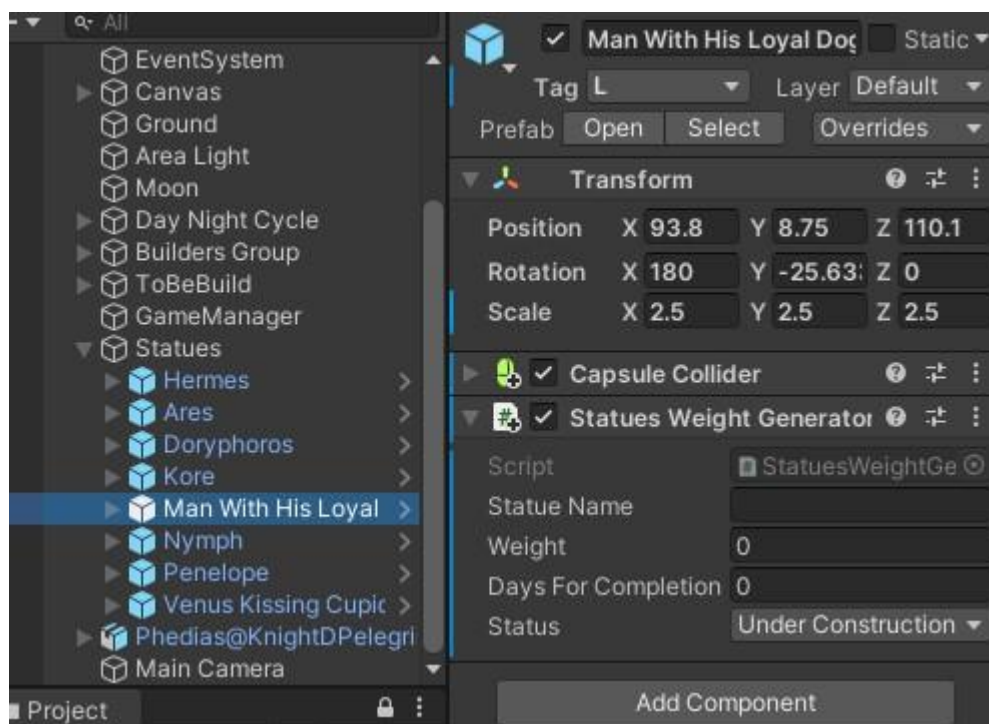
Εικόνα 16 Αριστερά η Νύμφη, δεξιά η Πηνελόπη

Η λογική δημιουργίας των αγαλμάτων είναι σχεδόν η ίδια με των κτιρίων. Κάθε άγαλμα έχει ένα βάρος και ανήκει σε μια κατηγορία από τις ακόλουθες: Μικρό, Μεσαίο, Μεγάλο. Έτσι διαμορφώνεται ο παρακάτω πίνακας:

Όνομα Αγάλματος	Μέγεθος	Βάρος	Μέρες για να ολοκληρωθεί
Ερμής	Μικρό	10-35	3
Άρης	Μικρό	10-35	3
Δορυφόρος	Μικρό	10-35	3
Κόρη	Μικρό	10-35	3
Αφροδίτη	Μεσαίο	45-75	6
Νύμφη	Μεσαίο	45-75	6
Πηνελόπη	Μεγάλο	80-95	9
Άνδρας με σκύλο	Μεγάλο	80-95	9

Πίνακας 5 Προγραμματιστικές πληροφορίες για τα αγάλματα

Αντίστοιχα λοιπόν με το BuildingWeightGenerator.cs, που υπάρχει σε κάθε κτίριο και παράγει το βάρος του κτιρίου, έτσι δημιουργήθηκε και η StatueWeightGenerator.cs που με βάση τον παραπάνω πίνακα, αρχικοποιεί τα αγάλματα.



Εικόνα 17 Αρχιτεκτονική αγαλμάτων στη Unity

Ο Φειδίας επιλέγει ποιο άγαλμα θα χτίσει στη συνέχεια με την `NextStatueDecider()`, που είναι παρόμοια με τη `NextBuildingDecider()`. Η συνάρτηση βάζει σε ένα Dictionary τα «παιδιά» του κενού `GameObject Statues`. Πρώτα απορρίπτει αυτά που χρειάζονται πολλές μέρες για να χτιστούν και μετά από τα εναπομένοντα διαλέγει αυτό με το μεγαλύτερο βάρος. Αμέσως μετά ενημερώνεται μια μεταβλητή (`remainingDays`) του `Phedias.cs` για να γνωρίζει πόσες ημέρες χρειάζεται ακόμα για να ολοκληρωθεί το άγαλμα. Στα κίρια αυτή η διαδικασία της ενημέρωσης γίνεται από τον Έξαρχο (`Exarchos.cs`).

Όταν ο Φειδίας ολοκληρώσει ένα νέο άγαλμα, δηλαδή οι `remainingDays` μηδενιστούν, καλείται η `PlaceStatueAndDecideNext()`, όπου:

- ενημερώνεται η κατάσταση του αγάλματος σε `Built`
- γίνεται ορατό στη θέση που έχει προσχεδιαστεί να εμφανιστεί
- ενημερώνεται η φήμη και η απόκλιση σε αυτά που σχεδιάζει και υλοποιεί ο Φειδίας
- ενημερώνεται η διεπαφή χρήστη ότι ολοκληρώθηκε το συγκεκριμένο άγαλμα με βάση το όνομά του
- αποφασίζει ποιο θα είναι το επόμενο άγαλμα που θα χτίσει.


```

public void UpdateStatueRemainingDays()
{
    if (statueRemainingDays - 1 > 0) {
        statueRemainingDays--;
    }
    else
    {
        PlaceStatueAndDecideNext();
    }
}

private void UpdateDeviation(GameObject statue)
{
    var statueWg = statue.GetComponent<StatuesWeightGenerator>();
    var daysForCompletion = statueWg.GetDaysForCompletionBasedOnTag();

    // Update Experience
    var personalExperience = statueWg.weight * daysForCompletion;

    // Calculate Deviation
    var devMultiplier = (100/personalExperience) * Constants.DEVMULTIPLIER;
    var devInt = Random.Range(minDev, maxDev) - devMultiplier;
    var deviation = devInt / 100.0;

    // Update Reputation
    reputation = personalExperience * deviation;
}

```

Στο σημείο αυτό πρέπει να επισημανθεί πως η εμπειρία αφορά το σύνολο της ομάδας και όχι το Φειδία αποκλειστικά. Εξάλλου στην κατασκευή κτιρίων ο Φειδίας επιβλέπει, αλλά η ομάδα χτιστών είναι αυτή που κάνει το κίσμα πράξη. Οπότε η εμπειρία είναι θέμα του συνόλου και όχι του αρχιτέκτονα. Επομένως το μόνο που μπορεί να βελτιωθεί με τη δημιουργία αγαλμάτων, είναι η φήμη και η πιο γρήγορη αποκατάστασή της. Για αυτό το λόγο ακολουθείται η διαδικασία που ακολουθήθηκε στην κατασκευή κτιρίων, για την ενημέρωση της απόκλισης και της φήμης, και στα αγάλματα.

```

public void PlaceStatueAndDecideNext()
{
    var sculptedStatue =
gameObject.GetComponent<StatueDecisionMaker>().currentStatue;
    sculptedStatue.SetActive(true);
    sculptedStatue.GetComponent<StatuesWeightGenerator>().status =
StatuesWeightGenerator.StatuesStatus.Built;
    UpdateDeviation(sculptedStatue);
    chatBox.UpdateChatBox("P: I finished my personal work of art. I named it " +
sculptedStatue.name);
    gameObject.GetComponent<StatueDecisionMaker>().StatueDecisionMaking();
}

```

4.3.11. ΗΧΟΙ ΠΡΟΣΟΜΟΙΩΣΗΣ

Στην προσομοίωση χρησιμοποιήθηκαν ήχοι για να βοηθήσουν τον παρατηρητή να βιώσει παραπάνω την Αρχαία Αθήνα.

Αρχικά χρησιμοποιήθηκε ένας ήχος παρασκηνίου για να φαίνεται πως υπάρχουν κάποιες δράσεις που συμβαίνουν στη διάρκεια της προσομοίωσης.

Όταν είναι βράδυ και οι χαρακτήρες πηγαίνουν για να κοιμηθούν, ακούγεται ένας άλλος ήχος πιο ήπιος για να δίνει την αίσθηση της ηρεμίας και της γαλήνης. Ο ήχος παρασκηνίου διακόπτεται για όσο διαρκεί ο παραπάνω ήχος.

Την ώρα που πηγαίνουν στο εκάστοτε κτίριο που βρίσκεται υπό κατασκευή, ακούγονται ήχοι από σφυριά και φτυάρια.

Όταν ο Φειδίας βρίσκεται στο Εργαστήριο και ο παρατηρητής με την κάμερα πλησιάσει το συγκεκριμένο κτίριο, μπορεί να ακούσει το Φειδία να σμιλεύει ένα άγαλμα. Αυτό έχει επιτευχθεί με τη χρήση 3D sound. Αντίστοιχα έξω από την Ταβέρνα όπου πηγαίνει τον ελεύθερο χρόνο η ομάδα, ακούγονται συνομιλίες, ποτήρια και γέλια.

Οι ήχοι που έχουν χρησιμοποιηθεί για κάθε περίπτωση δεν είναι μόνο ένας. Έχουν ανακτηθεί διαφορετικά τμήματα του ίδιου .wav αρχείου, κόπηκαν σε μικρότερα τμήματα και χρησιμοποιούνται με τη χρήση της συνάρτησης Random.

Όλοι οι παραπάνω ήχοι έχουν ανακτηθεί από την πλατφόρμα του YouTube και αναγράφονται στη Βιβλιογραφία στο τέλος της παρούσας διπλωματικής.

4.3.12. ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

Παρακάτω θα δοθεί παράδειγμα μιας τυπικής ημέρας των εργατών και του Φειδία.



Εικόνα 18 Την ώρα της δουλειάς

Στην αρχή της πρώτης ημέρας, ο Φειδίας αποφασίζει πού θα χτίσουν το επόμενο κτίσμα. Το πρωί πηγαίνουν και ξεκινούν τις εργασίες της κατασκευής.



Εικόνα 19 Στον ελεύθερο χρόνο

Το απόγευμα που ολοκληρώνουν τη δουλειά τους, κατευθύνονται προς τα ήδη χτισμένα σπίτια που αντιπροσωπεύουν την ταβέρνα και το εργαστήριο του Φειδία. Η ταβέρνα είναι στην παραπάνω εικόνα το σπίτι στα αριστερά και το εργαστήριο του Φειδία το μεσαίο σπίτι στα δεξιά. Τα μηνύματα έχουν ανανεωθεί στη διεπαφή χρήστη κάτω δεξιά.

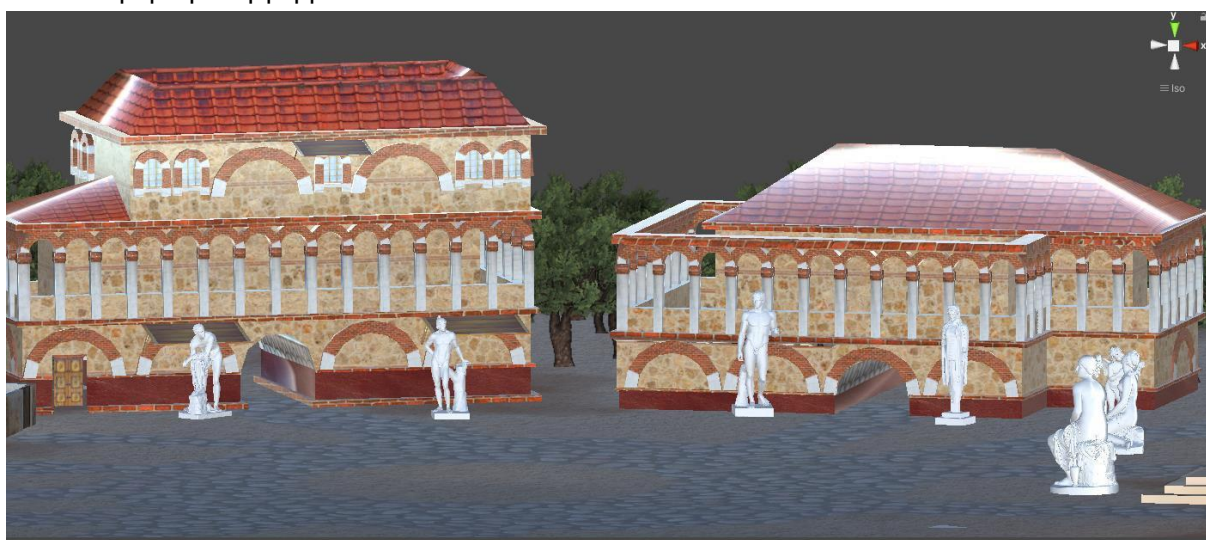


Εικόνα 20 Κατά τη διάρκεια της νύχτας

Μετά τον ελεύθερο χρόνο τους η ομάδα κατευθύνεται στην Εστία που είναι το δεύτερο κτίσμα από τα αριστερά. Στην εικόνα φαίνονται οι εργάτες να κατευθύνονται προς τα εκεί. Επίσης ο Φειδίας έχει φύγει από το εργαστήριο του και βρίσκεται στην αυλή του σπιτιού του στα δεξιά, έτοιμος να μπει για να κοιμηθεί.

5. ΤΕΛΟΣ ΠΡΟΣΟΜΟΙΩΣΗΣ- ΣΥΜΠΕΡΑΣΜΑΤΑ-ΠΕΡΙΛΗΨΗ

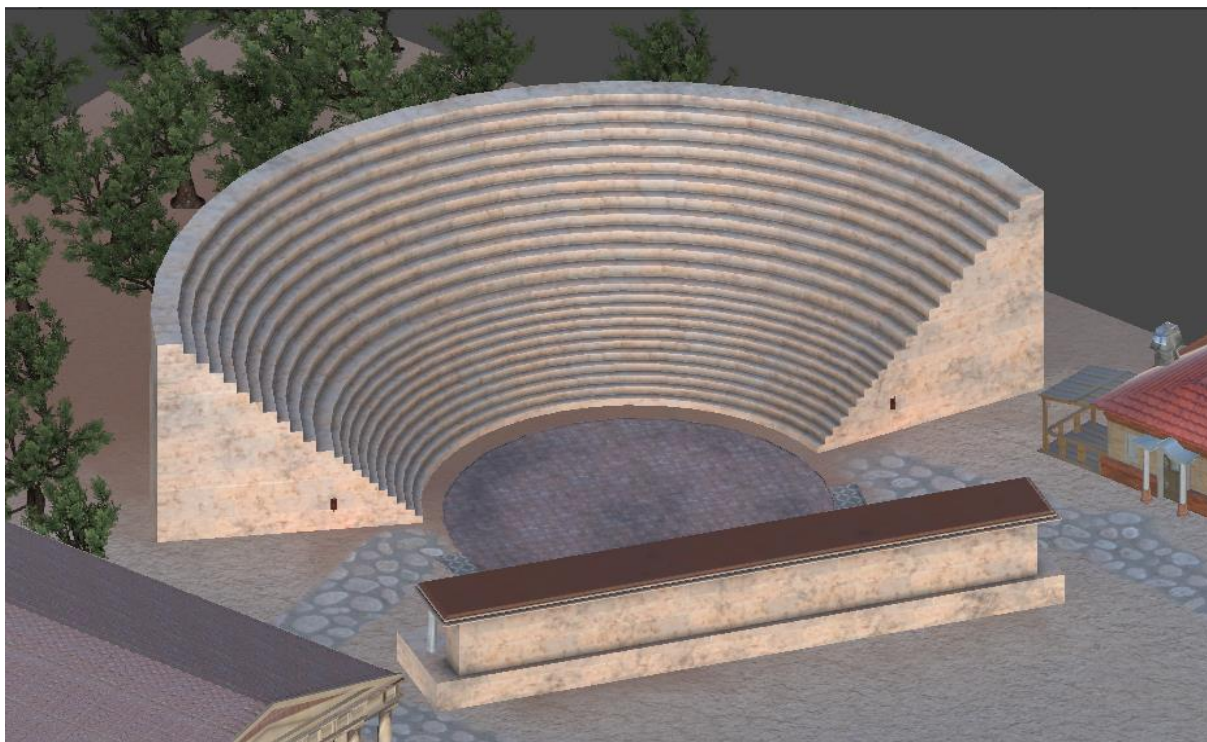
Παρακάτω υπάρχουν εικόνες από τη σκηνή ολοκληρωμένη, που περιλαμβάνει τα σπίτια, τους ναούς, το θόλο και το αμφιθέατρο. Η φάρμα έχει τοποθετηθεί στη σκηνή με τέτοιο τρόπο ώστε να παραπέμπει σε ένα οίκημα εκτός της κεντρικής πόλης. Ο Θόλος βρίσκεται στη μέση της πλατείας στην Αρχαία Αγορά της Αθήνας. Τα αγάλματα είναι τοποθετημένα μπροστά στους κεντρικούς ναούς, το αμφιθέατρο και στο δρόμο για τη φάρμα.



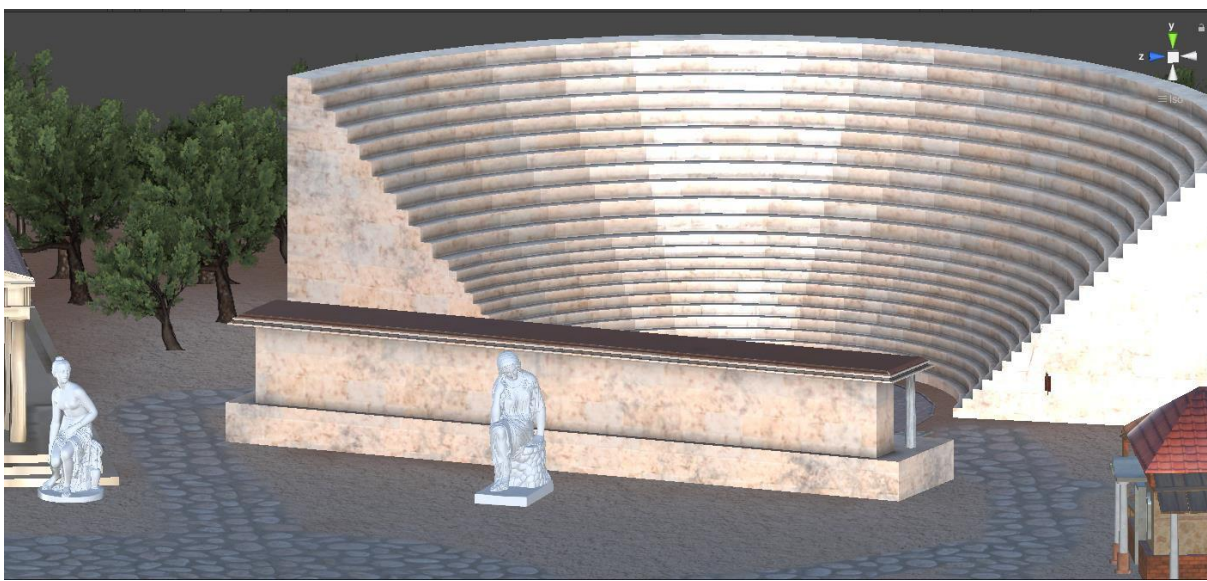
Εικόνα 21 Δύο από τους τρεις ναούς που χτίζονται στην προσομοίωση μαζί με τα αγάλματά τους



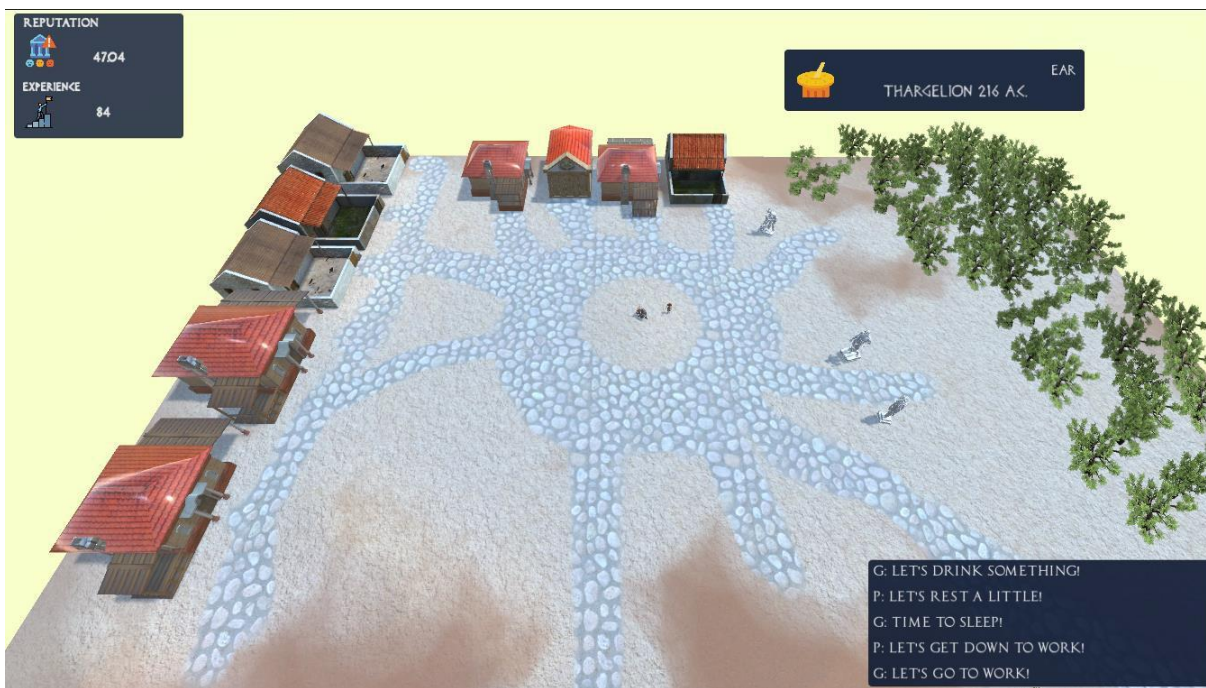
Εικόνα 22 Ο Θόλος και η κεντρική πλατεία



Εικόνα 23 Το αμφιθέατρο του Περικλή



Εικόνα 24 Αμφιθέατρο Περικλή και άγαλμα της Πηνελόπης



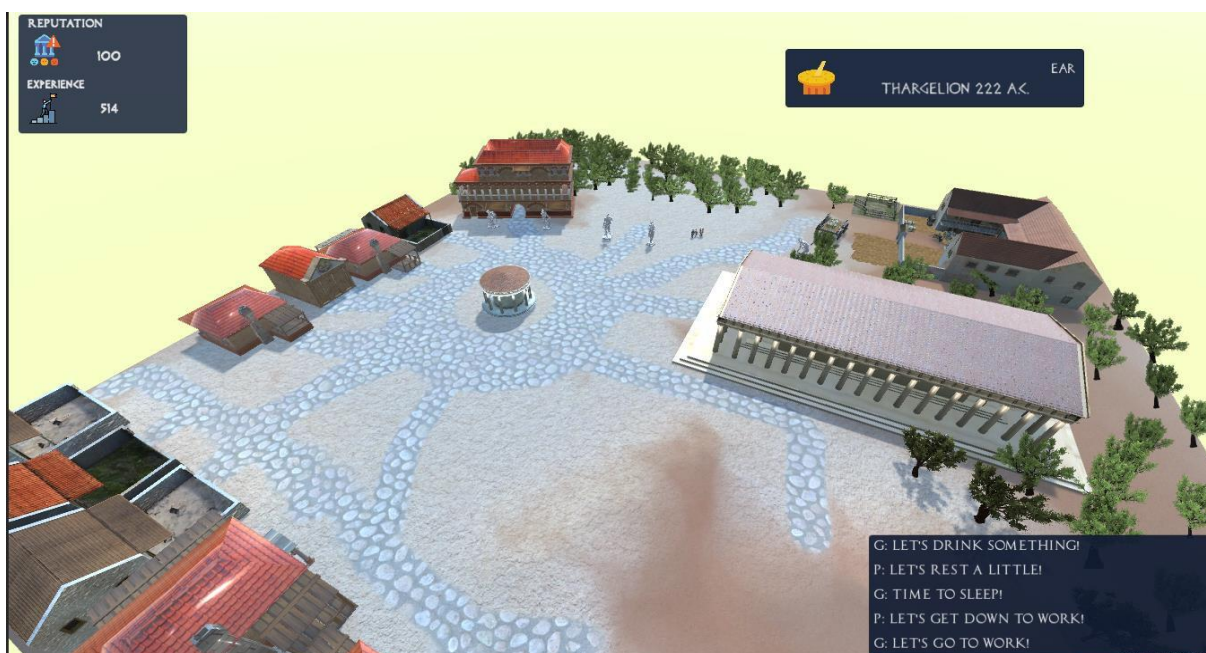
Εικόνα 25 Τα σπίτια της Αθήνας ολοκληρωμένα. Ο Φειδίας με την ομάδα βρίσκονται στο κέντρο της πλατείας, επομένως χτίζουν το Θόλο.



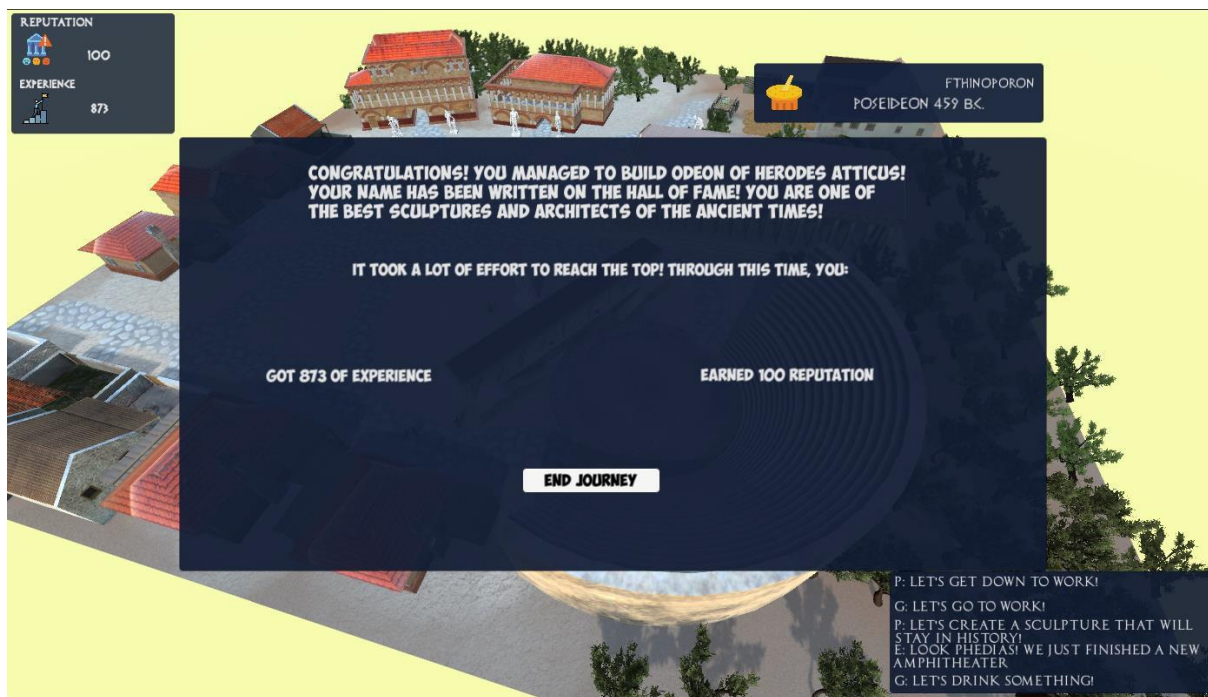
Εικόνα 26 Τα σπίτια, ο Θόλος και στο βάθος τα μικρά αγάλματα



Εικόνα 27 Ίδια προσομοίωση με την Εικόνα 20. Η ομάδα έχει κατασκευάσει και τη Φάρμα στο βάθος δεξιά



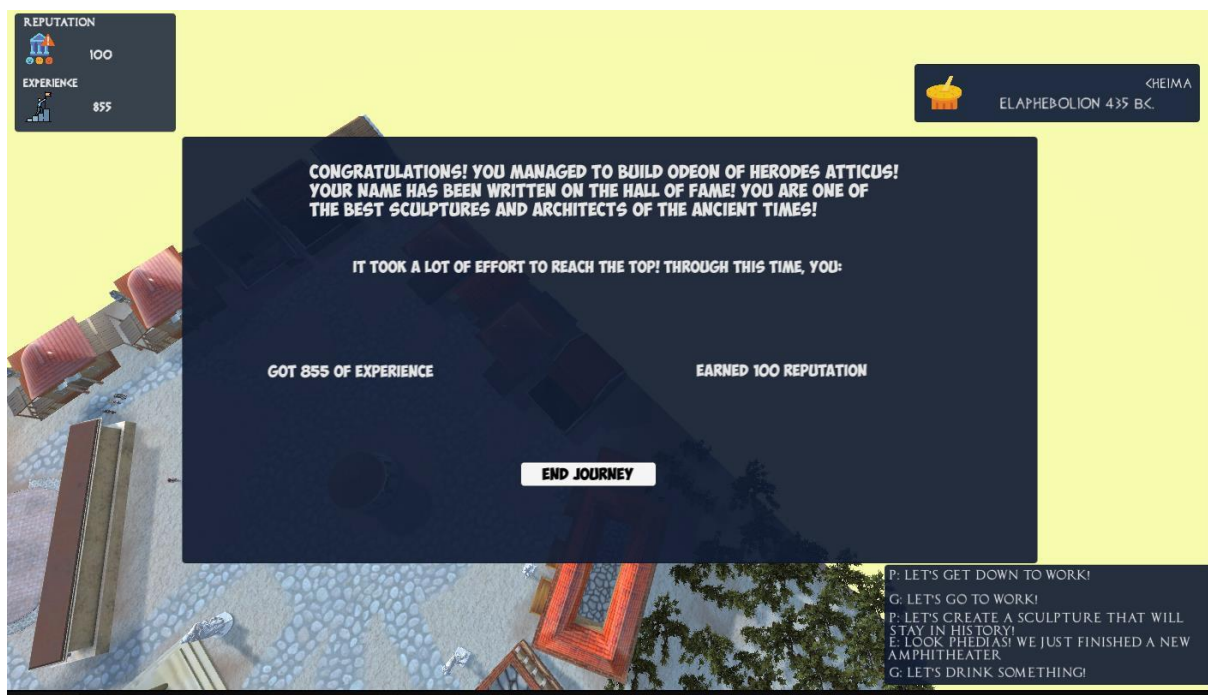
Εικόνα 28 Στη συνέχεια έχουν χτιστεί δυο ναοί. Η ομάδα βρίσκεται πίσω από τα αγάλματα και κατασκευάζει τον τελευταίο ναό



Εικόνα 29 Τέλος προσομοίωσης

Έχοντας πλέον χτίσει και το αμφιθέατρο, η προσομοίωση τελειώνει. Το ενημερωτικό μήνυμα επικροτεί την προσπάθεια του Φειδία και της ομάδας του. Παράλληλα παρατίθενται και στατιστικά σχετικά με την εμπειρία και τη φήμη του. Στο πρώτο παράδειγμα, είναι 873 πόντοι εμπειρίας και 100% φήμη. Στις ακόλουθες φωτογραφίες παρατίθενται και άλλα παραδείγματα που αποδεικνύουν ότι η φήμη φτάνει πάντα το 100% και η εμπειρία φτάνει κάπου στα 790-880.

Αυτό οδηγεί στο συμπέρασμα ότι παρόλο που η φήμη φτάνει σε μια κορυφή, στο σημείο της προσομοίωσης όπου έχουν χτιστεί τα περισσότερα σπίτια ή μόλις χτιστεί ο Θόλος ή η Φάρμα, η εμπειρία συνεχίζει να αυξάνεται. Οι εργάτες συνεχίζουν να μαθαίνουν και να εξελίσσονται καθ' όλη τη διάρκεια της προσομοίωσης, παρόλο που η γνώμη των κατοίκων της Αθήνας είναι ήδη η καλύτερη για αυτούς.



Εικόνα 30 Τέλος προσομοίωσης 2



Εικόνα 31 Τέλος προσομοίωσης 3

6. ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ

Οι επεκτάσεις που μπορούν να εφαρμοστούν στη σκηνή είναι αρκετές. Στην εργασία υπάρχουν και χαρακτήρες που δε χρησιμοποιήθηκαν αλλά απεικονίζουν πολίτες της Ρώμης που μοιάζουν περισσότερο με Αρχαίους Έλληνες από αυτούς που τελικά χρησιμοποιήθηκαν. Η χρήση των τελευταίων θα μπορούσε να επιφέρει πιο ρεαλιστικά αποτελέσματα. Παράλληλα θα μπορούσαν να εφαρμοστούν οι εποχές και στην πραγματικότητα με την ένταξη έστω κάποιου έτοιμου μηχανισμού από το Asset Store της Unity. Θα μπορούσε επίσης να εφαρμοστεί μια λογική που θα απέτρεπε τους χαρακτήρες να περνούν μέσα από τα κτίρια όταν περπατούν. Η υλοποίηση αυτή θα μπορούσε να γίνει με τη χρήση πρακτόρων. Εναλλακτικά, αντί για ελεύθερη προσομοίωση θα μπορούσε να αποτελέσει τη βάση για ένα παιχνίδι. Ο Φειδίας θα μπορούσε να αποτελέσει τον χαρακτήρα που θα χειριζόταν ο χρήστης, ελέγχοντας έτσι το επόμενο κτίριο ή άγαλμα που θα χτιστεί.

Το υπόβαθρο της Αρχαίας Ελλάδας, σε συνδυασμό με τη Μυθολογία, τους θρύλους και τα ιστορικά δεδομένα που υπάρχουν, δίνουν ένα μεγάλο εύρος δυνατοτήτων εξέλιξης. Για παράδειγμα θα μπορούσαν να υπάρχουν οι 12 θεοί του Ολύμπου που παίρνοντας αποφάσεις να επηρεάζουν τη ζωή των χαρακτήρων. Επίσης θα ήταν ενδιαφέρουσα η παρουσία ηρώων όπως ο Ηρακλής που τυχαία θα συναναστρεφόταν -ή και όχι, με το Φειδία και να του δώσει κάποια πληροφορία χτίζοντας έτσι κάποιο μυθικό κτίριο που θα του έδινε επιπλέον εμπειρία και αναγνωρισιμότητα.

Φυσικά, εκτός από το Φειδία θα μπορούσε ο παίκτης να παίρνει τις αποφάσεις του ίδιου του Περικλή, έχοντας έτσι τη συνολική διοίκηση της Αθήνας. Με αυτό τον τρόπο θα έλεγχε όλη την πόλη σε πολιτιστικό, στρατιωτικό, πολιτικό και οικονομικό επίπεδο. Καθημερινές δυσκολίες, αποφάσεις και επιλογές θα μπορούσαν να οδηγήσουν το Χρυσό Αιώνα του Περικλή σε μια μεγάλη αποτυχία ή στην επιτυχία που πραγματικά αποτέλεσε.

Από ιστορικά γεγονότα και δεδομένα είναι πολύ γνωστή η διαμάχη ανάμεσα στη Σπάρτη και την Αθήνα. Η μεταξύ τους έριδα θα μπορούσε να μεταφραστεί και αυτή μέσα από το παιχνίδι. Για παράδειγμα ο ίδιος μηχανισμός για την Αθήνα ή τα ίδια Assets θα μπορούσαν να χρησιμοποιηθούν μεταγενέστερα χτίζοντας μια Σπάρτη. Ο παίκτης θα μπορούσε να επιλέξει με ποια πόλη θα ήθελε να είναι. Έτσι το παιχνίδι θα έμοιαζε περισσότερο με το παιχνίδι που ενέπνευσε σε ένα βαθμό τη διπλωματική, το Civilizations V. Στο τέλος κάθε ημέρας ο παίκτης που θα διαχειριζόταν τη μια πόλη θα μάθαινε τι κατάφερε να κάνει ή άλλη μέσα στην ίδια μέρα και θα σχεδίαζε καλύτερα τις κινήσεις του για την επόμενη.

Μια εναλλακτική θα ήταν η στρατηγική που εφάρμοσε ένα άλλο πολύ γνωστό παιχνίδι, το Assassins Creed: Odyssey. Στο παιχνίδι αυτό βλέπουμε τη ζωή ενός χαρακτήρα στη διάρκεια του Πελοποννησιακού Πολέμου μεταξύ Αθήνας και Σπάρτης. Δηλαδή συμβαίνουν διάφορα γεγονότα στα οποία μπορεί να παρέμβει, αλλά στην ουσία εξελίσσονται και χωρίς τη βοήθειά του. Οπότε ένας χαρακτήρας της διπλωματικής θα μπορούσε να εκτελεί διάφορες εργασίες είτε ως απλός πολίτης (Simulation Game) είτε ως μισθοφόρος που εκτελεί αποστολές (Role Play Game) σε μια εκ των δυο πόλεων που προαναφέρθηκαν και αν εκτελεί με επιτυχία τις παραπάνω εργασίες ή αποστολές, η πόλη του να κερδίζει προβάδισμα απέναντι στην άλλη πόλη. Στη συγκεκριμένη εκδοχή θα μπορούσαν να υπάρχουν και επίπεδα δυσκολίας, για παράδειγμα η άλλη πόλη να μπορεί να νικηθεί εύκολα ή δύσκολα.

7. ASSETS

7.1. UNITY ASSETSTORE

- i. Olive Tree by **FOUR BLACK COLORS**

<https://assetstore.unity.com/packages/3d/vegetation/trees/olive-tree-1760>

- ii. Rome house Package by **HAVOLK**

<https://assetstore.unity.com/packages/3d/environments/historic/rome-house-package-116205>

- iii. Ancient Greece Volume I by **Methexis Studios**

<https://assetstore.unity.com/packages/3d/environments/historic/ancient-greece-volume-i-51069>

7.2. MIXAMO

- i. Phedias: Knight D Pelegri

<https://www.mixamo.com/#/?page=1&query=knight+d+pelegri&type=Character>

- ii. Exarchos: Abe

<https://www.mixamo.com/#/?page=1&query=Abe&type=Character>

- iii. Aristodemos: Peasant Man

<https://www.mixamo.com/#/?page=1&query=Peasant+Man&type=Character>

- iv. Phoebos: Morak

<https://www.mixamo.com/#/?page=1&query=Morak&type=Character>

- v. Artemis: Erika Archer (without bow)

<https://www.mixamo.com/#/?page=1&query=Erika+Archer&type=Character>

- vi. Danae: Kachujin G Rosales

<https://www.mixamo.com/#/?page=1&query=Kachujin+rosales&type=Character>

7.3. YOUTUBE

1. ASMR Ancient Athens by **1 Hour Ambience Therapy**

https://www.youtube.com/watch?v=vOUF9Yqim_M

2. D&D Ambience | Construction Site | Medieval, Workers, Citylife by **Michael Ghelfi - RPG Ambiences & Music**

<https://www.youtube.com/watch?v=Ly9Ci1ITAnY>

3. hammering Sound Effects All Sounds 2 by **All Sounds**

<https://www.youtube.com/watch?v=n7smCNFcb1k>

4. Ancient Greek Music Vol.1 Spirit Of Aristotle by **FM Records**

<https://www.youtube.com/watch?v=UoQUy9po5bk>

5. Night Forest (Cricket/Owl Hoot/Wolves Howl) Sound for Relaxation, Sleep, Study, Yoga **by Pixelated Chill**

<https://www.youtube.com/watch?v=XX6Xt54Ed4c>

8. Βιβλιογραφία

1. Gamedesigning, Unity vs Unreal. <https://www.gamedesigning.org/engines/unity-vs-unreal/>. [Ηλεκτρονικό]
2. Working with character animations in unity. <https://www.rokoko.com/en/explore/blog/rokoko-guides-working-with-character-animations-in-unity>. [Ηλεκτρονικό]
3. How to make RTS Camera Movement in Unity. <https://www.youtube.com/watch?v=cfjLQrMGEb4>. [Ηλεκτρονικό]
4. Ωδείον του Περικλέους. https://el.wikipedia.org/wiki/%CE%A9%CE%B4%CE%B5%CE%AF%CE%BF%CE%BD_%CF%84%CE%BF%CF%85_%CE%A0%CE%B5%CF%81%CE%B9%CE%BA%CE%BB%CE%AD%CE%BF%CF%85%CF%82. [Ηλεκτρονικό]
5. Αρχαίων Τόπος - Ο Χρυσός Αιώνας του Περικλή. <https://theancientwebgreece.wordpress.com/2015/04/26/%CE%BF%CF%87%CF%81%CF%85%CF%83%CF%8C%CF%82-%CE%B1%CE%B9%CF%8E%CE%BD%CE%B1%CF%82-%CF%84%CE%BF%CF%85-%CF%80%CE%B5%CF%81%CE%B9%CE%BA%CE%BB%CE%AD%CE%BF%CF%85%CF%82-5%CE%BF%CF%82-%CE%B1%CE%B9%CF%8E-2/>. [Ηλεκτρονικό]
6. Οι μήνες των Αρχαίων Ελλήνων. <http://www.kairatos.com.gr/mineselinon.htm>. [Ηλεκτρονικό]
7. Wikipedia-Εμπειρία. <https://el.wikipedia.org/wiki/%CE%95%CE%BC%CF%80%CE%B5%CE%B9%CF%81%CE%AF%CE%B1>. [Ηλεκτρονικό]
8. Εμπειρία και Πείρα. <https://www.in.gr/2017/12/25/culture/glossakailogotexnia/empeiria-kai-peira/>. [Ηλεκτρονικό]
9. Μηχανή του Χρόνου. <https://www.mixanitouxronou.gr/pia-anagki-ine-i-pio-simantiki-gia-kathe-anthropo-ti-lei-i-perifimi-piramida-tou-psichologou-maslow-pou-tin-axiopiisan-i-manatzer-gia-na-dinoun-bonous-ke-se-idos/>. [Ηλεκτρονικό]
10. Η ανθρώπινη ανάγκη πυραμίδας του Maslow (με εικόνες). <https://el.thpanorama.com/articles/desarrollo-personal/piramide-de-maslow-las-necesidades-humanas-con-imagenes.html>. [Ηλεκτρονικό]
11. Simulation Video Game. https://en.wikipedia.org/wiki/Simulation_video_game. [Ηλεκτρονικό]
12. Χαρατσή, Αλεξάνδρα. *Βασική Υλοποίηση Πυραμίδας των Αναγκών του Maslow σε Εικονικό Περιβάλλον*. σ.λ. : Πανεπιστήμιο Πειραιώς, 2020.