

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μάθημα Προπτυχιακών Σπουδών:
Τεχνολογίες Ανάπτυξης Ηλεκτρονικών Παιχνιδιών
Ακαδημαϊκό έτος: 2023 - 2024
Εξάμηνο: 7ο
Εγχειρίδιο Παρουσίασης Απαλλακτικής Εργασίας

Ομάδα Εργασίας:

Θεόδωρος Κοξάνογλου Π20094
Απόστολος Σιαμπάνης Π20173
Δημήτρης Στυλιανού Π20004

Υπεύθυνος Καθηγητής:

Θέμης Παναγιωτόπουλος

Περιεχόμενα

Εκφώνηση.....	3
Εισαγωγή.....	7
Περιγραφή του Προβλήματος.....	7
Αναλυτική Παρουσίαση Φάσεων Ανάπτυξης.....	8
Ανάλυση.....	8
Σχεδίαση.....	9
Υλοποίηση.....	12
Πηγαίος Κώδικας.....	18
Πίνακας C# scripts.....	18
Αναλυτική παρουσίαση των σημαντικότερων scripts.....	26
Animations που χρησιμοποιήθηκαν.....	36
Αναλυτική παρουσίαση για χρήστες.....	37
Menu.....	37
Settings/Ρυθμίσεις.....	38
Help/Βοήθεια.....	39
UI κατά την διάρκεια του παιχνιδιού.....	40
Pause Menu/Μενού Παύσης.....	44
Screenshots από το παιχνίδι.....	45
Φάρμα.....	45
Δάσος.....	47
Πόλη.....	50
Κριτήρια.....	52
Αληθοφάνεια.....	52
Περιεχόμενο.....	52
Πληρότητα.....	52
Σχεδιασμός.....	52
Αισθητική.....	52
Πρωτοτυπία.....	52
Χρηστικότητα.....	52
Κίνηση.....	52
Λειτουργικότητα.....	52
Ανάπτυξη.....	52
Αστοχίες κώδικα, Δυσκολίες, Μελλοντικές επεκτάσεις και Συμπεράσματα.....	53
Πηγές των assets.....	54

Εκφώνηση

Αντικείμενο : (100 μονάδες)

Το μάθημα θα εξεταστεί αποκλειστικά με την πραγματοποίηση απαλλακτικής εργασίας (ατομικά ή σε ομάδες έως 3 ατόμων). Το αντικείμενο της εργασίας είναι η δημιουργία ενός 2D ή 3D παιχνιδιού χρησιμοποιώντας την πλατφόρμα της Unity 3D.

Το παιχνίδι πρέπει οπωσδήποτε να διαθέτει (όταν είναι εφαρμόσιμο αυτό):

- 1) Εισαγωγική Οθόνη (Entry Screen)
- 2) Μενού Εισαγωγικής Οθόνης {Start, Continue, Preferences, Difficulty (Easy, Moderate, Expert), Quit}
- 3) Help screen που θα κάνει pause το παιχνίδι, θα εμφανίζεται με χρήση του πλήκτρου F1 (F1 Keystroke) και θα δείχνει την λειτουργικότητα των πλήκτρων, τους κανόνες του παιχνιδιού, τον στόχο του παίκτη, κ.λπ.
- 4) In-game οθόνη με Μενού {Save, Load, Preferences, Continue, Quit} που θα κάνει pause το παιχνίδι, και θα εμφανίζεται με χρήση του πλήκτρου F2 (F2 Keystroke)
- 5) Έξοδος από το παιχνίδι με την χρήση του πλήκτρου Esc οπότε θα εμφανίζεται η εισαγωγική οθόνη με το μενού της

Το παιχνίδι μπορεί προαιρετικά να διαθέτει ένα εξελιγμένο HUD Interface (ή ιδέες από αυτό)

Παράδειγμα Ιδανικού Basic Game HUD Interface



- 1) Character Status Window : Displays Character name, level, current health (HP), current mana (MP), and stamina(SP).
- 2) Target Page : Displays vitals for targeted enemy or player (name, status, characteristic).
- 3) Buff/DeBuff Check : Shows current buff (or de-buff) effect and the remaining time.
- 4) Hotkey Bars (P) : Emotes, basic behaviors, skills, magic, and consumable items, weapons, gears may be dragged into the hotkey slots and activated using the numerical keys (1,2,3 - 0) on the top of

your keyboard. A second bar to the player's HUD (these slots are activated using the number-pad keys).

- 5) Character Name : The name of the player character. Identity code or rank may appear along the sides of the name while in a party or in player-versus-player (PvP) mode.
 - 6) Minimap (V) : Indicates enemy Locations, NPCs, quests, and quest rewards.
 - 7) Function Buttons from right to left (F10-F9-...-F1) : Options, Quest, Basic Actions, Skills, Inventory, Item Shop, In-Game Menu, Help Screen.
 - 8) Shop : This is where players can buy various items, gears, weapons, pots, consumables, weapons, gears, ...
 - 9) Level Bar : The percentage of experience gained towards the next level.
 - 10) Stats Screen : The current status of your character including class, level, title, stats and other useful information.
- Inventory Screen : Whatever is carried by the character (m x n πινακας)

Το Ιδανικό Basic Game HUD Interface έχει αρκετή δουλειά για να αναπτυχθεί και θα μπορούσε να αποτελέσει από μόνο του απαλλακτική εργασία. Για τους υπόλοιπους είναι ιδέες που μπορείτε να χρησιμοποιήσετε (εφόσον εφαρμόζονται) στο παιχνίδι σας ή στις πτυχιακές σας.

Εναλλακτικό Αντικείμενο Απαλλακτικής (100 μονάδες)

Αν κάποιος είναι μόνος του και όχι σε ομάδα μπορεί εναλλακτικά να χρησιμοποιήσει έτοιμα παιχνίδια του Unity (π.χ. 2D Game Kit, 3D Game Kit) που παρέχονται ως free assets, να πραγματοποιήσει στοχευμένες αλλαγές στά παιχνίδια αυτά, τις οποίες όμως θα εξηγήσει αναλυτικά στο documentation.

Κριτήρια Αξιολόγησης

Ιδανικά, η εργασία που θα αναπτύξετε, θα έπρεπε να πληροί τα παρακάτω κριτήρια:

- **Αληθιοφάνεια.** Ο χώρος σας θα πρέπει να “πείθει” τον χρήστη, να μη δημιουργεί την εντύπωση πως παραβιάζει τους φυσικούς νόμους και, γενικά, να παρέχει τη ζητούμενη εμπειρία χωρίς να βασίζεται σε σχεδιαστικές ή αισθητικές ακρότητες.
- **Περιεχόμενο.** Θα χρησιμοποιήσετε περιεχόμενο που θα θυμίζει τον φυσικό, “πραγματικό” κόσμο για το σύνολο του οποίου θα αναφέρετε πηγές (μέσα στον εικονικό κόσμο, χρησιμοποιώντας κατάλληλα μέσα απεικόνισης πληροφορίας).
- **Πληρότητα.** Ο χώρος θα είναι ολοκληρωμένος σαν συστατικό εφαρμογής εικονικής πραγματικότητας: Θα εμφανίζεται στο χρήστη ως ένας τρισδιάστατος κόσμος, θα έχει λειτουργικότητα σε διάφορα σημεία ανάλογα με το τι αναπαριστά και το τι εξυπηρετεί, θα εξελίσσεται με το πέρασμα του χρόνου χάρη σε κινούμενα στοιχεία και στοιχεία με λειτουργικότητα, θα έχει φωτισμό και άλλα διακοσμητικά στοιχεία, και γενικά θα παρέχει μία ολοκληρωμένη, πολυτροπική, δυναμική εμπειρία αναπαράστασης χωρίς αδικαιολόγητα κενά και σημεία ασυνέχειας ή ασυνέπειας.
- **Σχεδιασμός.** Ο σχεδιασμός του χώρου σας θα χαρακτηρίζεται από ακρίβεια και λεπτομέρεια. Η αυξημένη σχεδιαστική/δομική πολυπλοκότητα δεν αποτελεί στοιχείο το οποίο θα αξιολογηθεί απαραίτητα θετικά.
- **Αισθητική.** Από άποψη αισθητικής, ο χώρος σας θα πρέπει να είναι ευπαρουσίαστος και να μην χαρακτηρίζεται από αισθητικές ακρότητες. Επίσης, θα πρέπει να “προσκαλεί” τον χρήστη και να του κεντρίζει την προσοχή.
- **Πρωτοτυπία.** Η δομή, ο σχεδιασμός, η αισθητική και η λειτουργικότητα του χώρου σας θα αντανακλούν την έκταση στην οποία διερευνήσατε τις σχεδιαστικές δυνατότητες της πλατφόρμας, τις δυνατότητες της οικείας γλώσσας, καθώς και διάφορα σχεδιαστικά πρότυπα και πρακτικές υλοποίησης.
- **Χρηστικότητα.** Ο χώρος θα προορίζεται για χρήση από ανθρώπους και για την κάλυψη υπαρκτών αναγκών τους. (π.χ. δυνατότητα μετακίνησης και επίσκεψης διαφόρων σημείων του τρισδιάστατου κόσμου)

- **Κίνηση** (animation). Η κατασκευή σας θα περιέχει αναπαραστάσεις κινούμενων στοιχείων, ώστε να δημιουργεί στον χρήστη την εντύπωση ότι συμμετέχει σε ένα δυναμικό, εξελισσόμενο με το χρόνο, κόσμο. Τέτοια στοιχεία μπορεί να είναι, για παράδειγμα, “άψυχα” αντικείμενα, αντικείμενα που μετακινούνται σε προκαθορισμένες διαδρομές, ζώα/ρομπότ, κ.ά. Τονίζεται ότι ο όρος κίνηση χρησιμοποιείται εδώ όχι με την έννοια της μετακίνησης, αλλά της οποιασδήποτε μεταβολής της τιμής κάποιου χαρακτηριστικού ενός αντικειμένου. Έτσι, ως κινούμενο συστατικό νοείται και ένα αντικείμενο που αλλάζει χρώμα με την πάροδο του χρόνου (π.χ., ένα φανάρι ρύθμισης κυκλοφορίας, κ.ά.), ένα αντικείμενο που αλλάζει διαστάσεις, κ.ά.
- **Λειτουργικότητα** (functionality). Ανάλογα με τι αναπαριστά, η κατασκευή σας θα περιέχει συστατικά με λειτουργικότητα, δηλαδή, συστατικά πάνω στα οποία θα μπορεί να επιδράσει ο χρήστης ή, γενικότερα, συστατικά τα οποία αντιδρούν με συγκεκριμένο τρόπο σε συγκεκριμένες ενέργειες του χρήστη. Παραδείγματα αντικειμένων με λειτουργικότητα είναι μία πόρτα η οποία ανοίγει και κλείνει όταν ο χρήστης επιδρά σε ένα διακόπτη, ένας μηχανισμός που ξεκινά και σταματά όταν ο χρήστης επιδρά σε ένα μοχλό, ένα ρομπότ ή ένας συναγερμός που ενεργοποιείται όταν ο χρήστης πλησιάσει κάποιο σημείο του εικονικού κόσμου ή βρεθεί μέσα σε κάποια περιοχή του, ένα ραδιόφωνο του οποίου την ένταση ο χρήστης προσαρμόζει με κάποιο χειριστήριο, μία συσκευή τηλεόρασης την οποία ο χρήστης ενεργοποιεί και απενεργοποιεί, κ.ά.
- **Ανάπτυξη.** Η λειτουργικότητα των διαφόρων αντικειμένων θα υλοποιηθεί στην οικεία γλώσσα (C#/Unity 3D). Ο κώδικας που θα συντάξετε θα είναι καλά σχεδιασμένος, θα ενσωματώνει βέλτιστες τεχνικές υλοποίησης, θα είναι ευπαρουσίαστος και, το σημαντικότερο, θα είναι συνολικά, αναλυτικά και κατανοητά σχολιασμένος.

Εκπόνηση και υλοποίηση

Θα εργαστείτε είτε ατομικά είτε σε ομάδες έως τριών ατόμων. Δεν χρειάζεται να δηλώσετε τις ομάδες στον διδάσκοντα. Μπορείτε να εισάγετε (import) μοντέλα από σχεδιαστικά προγράμματα ή άλλες πηγές. Κάθε μέλος της ομάδας θα συμμετέχει λιγότερο ή περισσότερο σε όλες τις φάσεις της εκπόνησης και σε όλες τις διαδικασίες (ανάλυση, γενικό σχεδιασμό, σχεδιασμό αντικειμένων, προγραμματισμό συμπεριφοράς, δοκιμές, κ.τ.λ.).

Υποβολή και αξιολόγηση

Οι φοιτητές υποβάλουν την εργασία τους σε ειδικό χώρο του Gunet2. Δεδομένου ότι η εργασία είναι συνήθως μεγάλη και δεν την δέχεται το gunet2, μπορείτε εναλλακτικά να την ανεβάσετε σε κάποιου τύπου dropbox και να γράψετε την διεύθυνση της εργασίας στο gunet ώστε ο διδάσκων να την κατεβάσει. Καλό είναι να προσδιορίσετε και ένα e-mail και τηλέφωνο, στην περίπτωση που κάτι πάει στραβά στο download της εργασίας. Η εργασία θα αποτελείται από το

1. project της Unity,
2. τα Build αρχεία,
3. PowerPoint 10-15 σελίδων
4. ολιγόλεπτο video της εφαρμογής,
5. εγχειρίδιο παρουσίασης (documentation) της εργασίας το οποίο θα περιέχει
 - a. Εισαγωγή.
 - b. Περιγραφή του Προβλήματος.
 - c. Αναλυτική παρουσίαση όλων των φάσεων ανάπτυξης
 - i. Ανάλυση
 - ii. Σχεδίαση
 - iii. Υλοποίηση),
 - d. Πηγαίος κώδικας
 - i. Πίνακα με C# scripts και σύντομη περιγραφή (4-5 γραμμών) τι κάνει το κάθε script,
 - ii. Αναλυτική παρουσίαση των 4-5 σημαντικότερων scripts,

- e. Animations που χρησιμοποιήθηκαν
- f. Αναλυτική παρουσίαση για χρήστες (Menu, UI, HUD, keystrokes, Inventory, Character Attributes Interface, λειτουργικότητα, κ.λπ),
- g. screenshots από την εφαρμογή καθώς και
- h. απάντηση για το πως αντιμετωπίζονται τα κριτήρια που θέσαμε παραπάνω.
- i. Αστοχίες κώδικα, Δυσκολίες, Μελλοντικές επεκτάσεις και Συμπεράσματα
- j. Βιβλιογραφία - πηγές των assets,

Εισαγωγή

Η εικονική πραγματικότητα, όντας κάποτε όνειρο επιστημονικής φαντασίας, έχει πλέον λάβει σάρκα και οστά μέσω των σύγχρονων τεχνολογικών εργαλείων. Η ραγδαία εξέλιξη της μηχανικής μάθησης και η διάδοση "έξυπνων" εφαρμογών και πληροφοριών μέσω του διαδικτύου, δίνουν πλέον στον μέσο πολίτη τη δυνατότητα δημιουργίας και ανάπτυξης αληθοφανών εικονικών περιβαλλόντων, ικανά ώστε να ικανοποιήσουν τις ανάγκες των δημιουργών τους. Η παρούσα εργασία εστιάζει στην ανάπτυξη ενός εικονικού περιβάλλοντος που προσομοιάζει την καθημερινότητα ενός μέσου αγρότη. Χρησιμοποιώντας το Unity, ένα από τα κορυφαία εργαλεία στον τομέα της εικονικής πραγματικότητας, δημιουργήσαμε ένα περιβάλλον που αποτυπώνει μια μέρα στη ζωή ενός αγρότη.

Περιγραφή του Προβλήματος

Δημιουργήσαμε ένα 3D παιχνίδι χρησιμοποιώντας την μηχανή 3D του Unity. Το παιχνίδι διαδραματίζεται στο 1995 στο Τέξας. Η ιστορία του παιχνιδιού περιγράφει μία μέρα στη ζωή ενός αγρότη που προσπαθεί να λύσει τα καθημερινά προβλήματα του στη φάρμα που συντηρεί. Το παιχνίδι διαθέτει 3 διαφορετικούς χάρτες, οι οποίοι αντιπροσωπεύουν τα στάδια του ταξιδιού του πρωταγωνιστή για την αποκατάσταση μίας βλάβης στα χωράφια του. Στην πορεία του, ο αγρότης συναντά φίλους και συγγενείς, οι οποίοι, μέσα από συζητήσεις και quests, τον βοηθούν να ξεπεράσει τα εμπόδια της καθημερινότητας και να φέρει εις πέρας τους στόχους του. Το παιχνίδι μας προσφέρει μια ζεστή και νοσταλγική εμπειρία, αγγίζοντας θεματολογίες όπως η φιλία και η σύνδεση με το παρελθόν. Η 3D απεικόνιση, οι ποικίλοι χάρτες, τα quests και οι συζητήσεις συνθέτουν ένα γοητευτικό κόσμο που καλεί τον παίκτη να εξερευνήσει και να συνδεθεί με τον πρωταγωνιστή.

Αναλυτική Παρουσίαση Φάσεων Ανάπτυξης

Ανάλυση

Από την αρχή είχαμε αποφασίσει για την βασική ιστορία του παιχνιδιού. Μετά από καταιγισμό ιδεών καταλήξαμε στο συμπέρασμα ότι θέλουμε μία ιστορία παρένει από την καθημερινότητα για το παιχνίδι. Μετά από αναζήτηση και εύρεση των κατάλληλων assets (τα οποία τα αναφέρουμε στην κατηγορία: “Πηγές των assets”) έπρεπε να τα συνδυάσουμε ώστε να παράξουμε μία πρωτότυπη ιστορία, ιδανική για τα assets που είχαμε διαθέσιμα. Καταλήξαμε στην ιδέα “μια μέρα στη ζωή ενός αγρότη ονόματι Άλεξ”. Ο αγρότης θα ζει σε μια ορεινή περιοχή του Τέξας και θα προσπαθεί να αντεπεξέλθει στις αντιξότητες της καθημερινότητας ώστε να μπορέσει να βιοποριστεί και κατ’ επέκταση να φροντίσει την επιχείρηση του μοναδικού οικογενειακού μέλους που του έχει μείνει, του παππού του.

Έχοντας συμφωνήσει τον βασικό κορμό της ιστορίας, επόμενο βήμα ήταν η δημιουργία του εικονικού περιβάλλοντος που θα αλληλεπιδρά ο Άλεξ και κατ επέκταση ο παίκτης. Το περιβάλλον θα πρέπει να είναι μεγάλο αρκετά ώστε να υπάρχει αρκετό περιεχόμενο για εξερεύνηση από τον παίκτη και θα πρέπει να είναι ποιοτικό ώστε ο παίκτης να απολαμβάνει το παιχνίδι και τα νοήματα που αποτυπώνονται σε αυτό. Χωρίσαμε, λοιπόν, το παιχνίδι σε τρία διαφορετικά περιβάλλοντα:

- 1) **Φάρμα:** Θα είναι το πρώτο και το τελευταίο περιβάλλον όπου θα αλληλεπιδράσει ο παίκτης.
- 2) **Δάσος:** Θα είναι το μεταβατικό περιβάλλον που θα παρεμβάλλεται στο ταξίδι του Άλεξ μεταξύ Φάρμας και Πόλης.
- 3) **Πόλη:** Θα είναι στο περιβάλλον όπου ο Άλεξ θα λύσει το βασικό πρόβλημα της ιστορίας και θα μπορέσει να επιστρέψει πίσω στη φάρμα.

Σχεδίαση

Χρησιμοποιήθηκε η έκδοση Unity: **2022.3.13f1**

Γρώτο βήμα ήταν η βασική σχεδίαση των τριών terrain.

Φάρμα



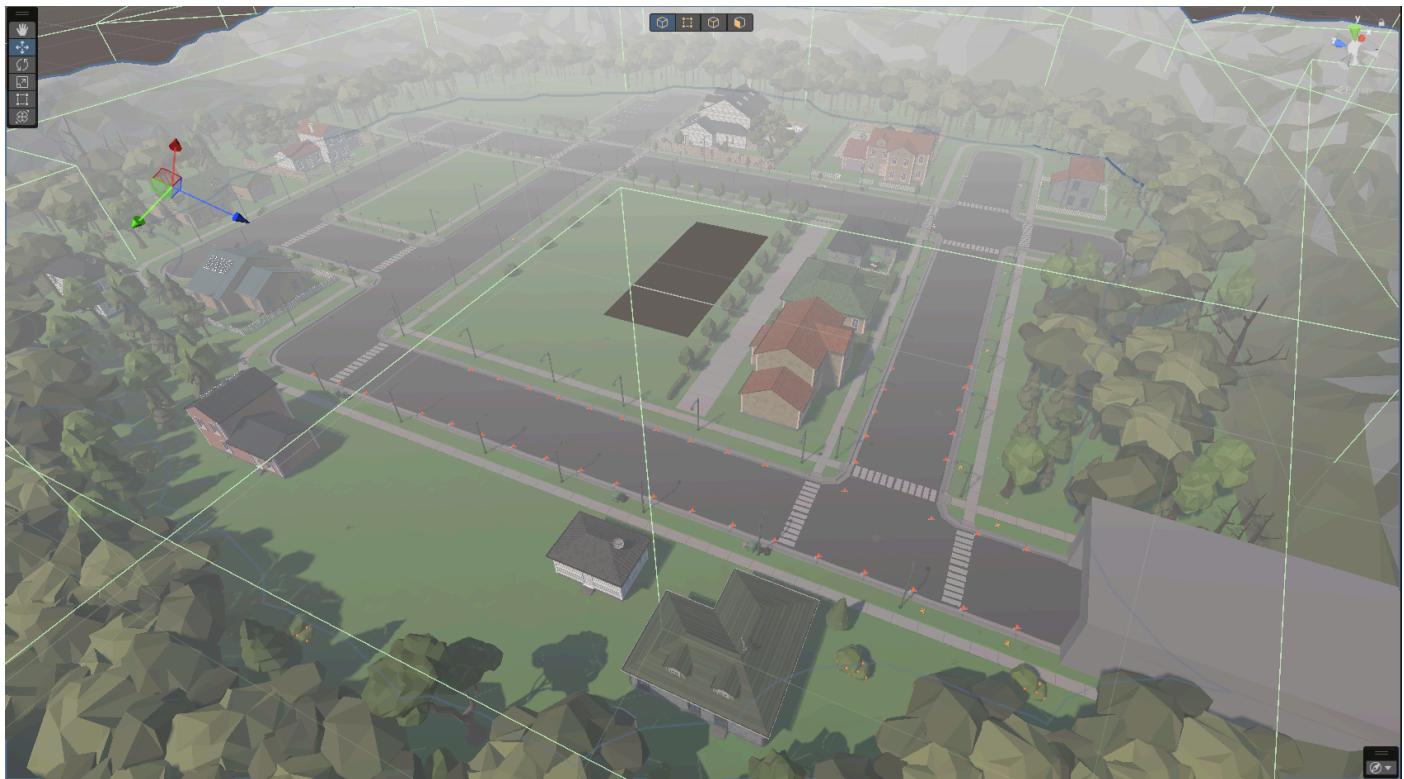
Σχεδίαση μορφολογίας εδάφους και δρόμων που θα συνδέουν τις δομές της φάρμας.

Δάσος



Σχεδίαση του δρόμου, που θα ενώνει την φάρμα με την πόλη, και των ποταμών του δάσους που θα είναι άμεσα συνδεδεμένοι με τον καταρράκτη του δάσους.

Πόλη



Σχεδίαση δρόμων και πιθανές τοποθεσίες σπιτιών για την πόλη.

Επόμενο βήμα ήταν η προσθήκη επιπλέον χαρακτηριστικών για να κάνει τα terrain πιο ρεαλιστικά. Προσθέσαμε από τα έτοιμα assets δέντρα, βουνά, ποτάμια, σπίτια, αυτοκίνητα.

Φάρμα



Συνέχεια υλοποίησης της φάρμας με δέντρα, χωράφια με σιτηρά, θερμοκήπια και αγροτικά οχήματα.

Δάσος



Προσθήκη σπιτιού και αυτοκινήτων που θα διέρχονται από τον βασικό δρόμο του δάσους.

Τελειώσαμε τη σχεδίαση των τριών κόσμων με προσθήκη λεπτομερειών που θα κάνει το κόσμο του παιχνιδιού πιο φυσικό.

Φάρμα



Δάσος



Πόλη



Υλοποίηση

Τελικό βήμα για την υλοποίηση των κόσμων ήταν η προσθήκη “ζωντανών” χαρακτηριστικών με τα οποία ο πρωταγωνιστής θα μπορεί να αλληλεπιδρά μαζί τους. Προσθέσαμε φωτισμό, ήχους, NPCs και κίνηση στους χώρους αυτούς:

Φάρμα



Προσθέσαμε ήχο και κίνηση για το αυτόματα πότισμα.



Προσθέσαμε NPCs με κίνηση από το mixamo με τα οποία ο παίκτης θα μπορεί να αλληλεπιδρά.

Δάσος



Προσθέσαμε ήχους της φύσης, εφέ στο νερό και στον αέρα.



Προσθέσαμε κίνηση στα αυτοκίνητα και στις ρόδες τους. Τα αυτοκίνητα έχουν προκαθορισμένη και επαναλαμβανόμενη διαδρομή. Επιπλέον, στα αυτοκίνητα τοποθετήσαμε generic NPCs που εκτελούν μία κίνηση οδήγησης την οποία κάναμε λήψη από το mixamo.

Πόλη



Προσθέσαμε κίνηση στα αυτοκίνητα και στα NPCs. Κάθε στοιχείο που έχει κίνηση εκτελεί μία προκαθορισμένη διαδρομή με χρήση του NavMeshPath.



Ορισμένα NPCs έχουν μοναδικές κινήσεις όταν πλησιάζει ή τους κοιτάει ο παίκτης - με τη χρήση του RayCast. Για παράδειγμα, στη φωτογραφία εμφανίζεται ο πωλητής, δίπλα από το NPC ντυμένο Hot Dog που χαιρετάει όταν τον πλησιάζει ο παίκτης. Η κίνηση αυτή και η χορευτική κίνηση του συναδέλφου του κατέβηκαν από το mixamo.

Τα τελικά terrain:

Φάρμα



Δάσος



Πόλη



Πηγαίος Κώδικας

Πίνακας C# scripts

Path: “Assets/Scripts/”

Όνομασία Αρχείου	Περιγραφή Κώδικα
WheelsRotation.cs	<ul style="list-style-type: none">Κινεί τις ρόδες των αυτοκινήτων
Player.cs	<ul style="list-style-type: none">Ορίζει το inventory του player
MainMenuManager.cs	<ul style="list-style-type: none">Δημιουργεί την αρχική οθόνη του παιχνιδιούΟρίζει όλα τα απαραίτητα settings για τη ρύθμιση του παιχνιδιού από τον παίκτη.
MainLevelLoader.cs	<ul style="list-style-type: none">Διεκπεραιώνει την σωστή επιλογή σκηνής ανάλογα με την πρόοδο του παίκτη στην τελευταία αποθήκευση.

Path: “Assets/Scripts/Common/”

Όνομασία Αρχείου	Περιγραφή Κώδικα
LevelLoader.cs	<ul style="list-style-type: none">Φορτώνει το κατάλληλο επίπεδο που ορίζεται από το παιχνίδι
PauseManager.cs	<ul style="list-style-type: none">Παγώνει τα στοιχεία της σκηνήςΕμφανίζει το pause menuΠαίζει τη μουσική του menu

Path: “Assets/Scripts/Common/QuestSystem/”

Όνομασία Αρχείου	Περιγραφή Κώδικα
CarryQuest.cs	<ul style="list-style-type: none">Ορίζει τα αντικείμενα τα οποία μπορεί να μεταφέρει ο χρήστης ανά questΟρίζει την γενική λογική πίσω από τα quest ανάλογα με την φάση στην οποία βρίσκονται
CollectQuest.cs	<ul style="list-style-type: none">Χρησιμοποιείται στα quests που ο χρήστης πρέπει να μαζέψει αντικείμενα και όχι να μεταφέρει.Έχει ελέγχους ανάλογα με το στάδιο του quest
InteractQuest.cs	<ul style="list-style-type: none">Δημιουργεί ένα νέο quest
Quest.cs	<ul style="list-style-type: none">Ορίζει τις ιδιότητες ενός quest
QuestManager.cs	<ul style="list-style-type: none">Θέτει τα UI στοιχεία του παιχνιδιού ανάλογα με το event που έχει δημιουργηθεί

Path: "Assets/Scripts/Common/InteractionSystem/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
NPC.cs	<ul style="list-style-type: none"> Αναπαριστά ένα μη εκτελέσιμο χαρακτήρα σε ένα παιχνίδι που μπορεί να αλληλεπιδρά με τον παίκτη, εκτελώντας διάλογο, προσφέροντας αποστολές και αλλάζοντας κατάσταση ανάλογα με την επικοινωνία.
Interactor.cs	<ul style="list-style-type: none"> Υλοποιεί λειτουργίες αλληλεπίδρασης και διαχείρισης inventory για τον πρωταγωνιστή του παιχνιδιού, Έχει λειτουργίες όπως ο εντοπισμός αντικειμένων με χρήση raycasting, η αλληλεπίδραση με αντικείμενα, καθώς και η δυνατότητα μεταφοράς αντικειμένων από ένα μέρος σε ένα άλλο.
InteractHint.cs	<ul style="list-style-type: none"> Αναλαμβάνει την εμφάνιση ενός μηνύματος υπόδειξης για αλληλεπίδραση με ένα αντικείμενο, χρησιμοποιώντας το Text Mesh Pro
Interactable.cs	<ul style="list-style-type: none"> Παρέχει λειτουργίες όπως επιλογή, έναρξη, και διακοπή αλληλεπίδρασης Ενσωματώνει στοιχεία για τη διαχείριση αποστολών (Quests)
IFixable.cs	<ul style="list-style-type: none"> Διασφαλίζει τη δυνατότητα επισκευής ενός αντικειμένου.
ICollector.cs	<ul style="list-style-type: none"> Μέθοδος Collect για τη συλλογή συγκεκριμένου τύπου αντικειμένου.
Collectable.cs	<ul style="list-style-type: none"> Παρέχει λειτουργίες για τη διαχείριση της αλληλεπίδρασης με τον παίκτη, συμπεριλαμβανομένης της συλλογής αντικειμένων και της απόκρυψης του αντικειμένου μετά τη συλλογή.
ICarrier.cs	<ul style="list-style-type: none"> Περιλαμβάνει μεθόδους PickUp, Drop, και GetCarryingObject

Path: "Assets/Scripts/Common/DialogueSystem/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
ISpeak.cs	<ul style="list-style-type: none"> Ορίζει μια μέθοδο OnDialogueEnd για την ολοκλήρωση διαλόγου.
DialogueTrigger.cs	<ul style="list-style-type: none"> Ελέγχει τους διαλόγους του παιχνιδιού, ξεκινώντας τους
DialogueSentence.cs	<ul style="list-style-type: none"> Αντιπροσωπεύει μία πρόταση διαλόγου με πεδία για όνομα χαρακτήρα και κείμενο μηνύματος.
DialogueManager.cs	<ul style="list-style-type: none"> Εμφάνιση κειμένου και ονομάτων ομιλητών με χρήση TextMeshPro
Dialogue.cs	<ul style="list-style-type: none"> Δημιουργεί ένα αντικείμενο διαλόγου που περιέχει μια λίστα προτάσεων διαλόγου.

Path: "Assets/Scripts/Farm/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
WindmillSound.cs	<ul style="list-style-type: none"> Χρησιμοποιείται για να ξεκινήσει ο ήχος του ανεμόμυλου σε κάθε επανάληψη της κίνησής του
RandomAnimatorOffset.cs	<ul style="list-style-type: none"> Δημιουργεί ένα τυχαίο χρονισμό αναπαραγωγής για ένα συγκεκριμένο animation
NPCPatrollingController.cs	<ul style="list-style-type: none"> Ο ελεγκτής NPC Patrolling Controller ελέγχει τα μονοπάτια των NPC. Χρησιμοποιείται NavMeshAgent για την πλοϊγηση του εκάστοτε χαρακτήρα στα σημεία που έχουν οριστεί από εμάς.
FruitTreeCollector.cs	<ul style="list-style-type: none"> Χρησιμοποιεί το στοιχείο NavMeshAgent για να χαραχτεί το μονοπάτι προς τα δέντρα και τον Animator για την αναπαραγωγή της αντίστοιχης κίνησης για τη συλλογή μήλων.
Grandpa.cs	<ul style="list-style-type: none"> Διαχειρίζεται την κατάσταση του παππού στο αγρόκτημα.
FarmLevelLoader.cs	<ul style="list-style-type: none"> Αναλαμβάνει τη φόρτωση του επιπέδου του φάρμας και την αρχικοποίησή του, λαμβάνοντας υπόψη την πρόοδο του παίκτη και τις αποθηκευμένες πληροφορίες.
Alice.cs	<ul style="list-style-type: none"> Διαχειρίζεται την κατάσταση της Alice στο αγρόκτημα.

Path: "Assets/Scripts/Farm/InteractionSystem/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
SprinklerInteractable.cs	<ul style="list-style-type: none"> Αλληλεπίδραση με αυτόματο πότισμα στη φάρμα
Car.cs	<ul style="list-style-type: none"> Επιτρέπει την αλληλεπίδραση με το αυτοκινητο του Άλεξ στο παιχνίδι. Κατά την αλληλεπίδραση, απενεργοποιεί το gameCanvas και ενεργοποιεί ένα exitCutScene.
TreeCollectable.cs	<ul style="list-style-type: none"> Δημιουργεί ένα συλλέκτη για συλλογή μήλων από τα δέντρα. Καθορίζει την ποσότητα φρούτων που είναι διαθέσιμα για συλλογή και μειώνει την ποσότητα με κάθε αλληλεπίδραση.

Path: "Assets/Scripts/Farm/Quests/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
BeforeQuests.cs	<ul style="list-style-type: none"> Αρχικοποιεί τον τίτλο και τη περιγραφή μιας αποστολής ανάλογα με το στάδιο που βρίσκεται η ιστορία.
InteractWithCarDoorQuest.cs	<ul style="list-style-type: none"> Ορίζει το quest για την διαχείριση του αυτοκινήτου (έξοδος από τη φάρμα).
InteractWithSprinkler.cs	<ul style="list-style-type: none"> Ορίζει το quest για την διαχείριση με το σύστημα αυτόματου ποτίσματος.

Path: "Assets/Scripts/Farm/Quests/{Apple/Carrot} Collect Quest/",
 "Assets/Scripts/Farm/Quests/Water Crops Quest/" και
 "Assets/Scripts/Farm/Fix watering system Quest"

Όνομασία Αρχείου	Περιγραφή Κώδικα
Localization.cs	<ul style="list-style-type: none"> Ορίζει τα μηνύματα για τα quests σε όλες τις υποστηριζόμενες γλώσσες του παιχνιδιού.

Path: "Assets/Scripts/Forest/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
MonologueTrigger.cs	<ul style="list-style-type: none"> Εκκινεί ένα μονόλογο όταν ο παίκτης αλληλεπιδρά με ένα NPC.
ForestManNavMesh.cs	<ul style="list-style-type: none"> Ρυθμίζει την κίνηση ενός NPC στο δάσος χρησιμοποιώντας το NavMesh
CarMovement.cs	<ul style="list-style-type: none"> Ελέγχει την κίνηση ενός αυτοκινήτου χρησιμοποιώντας το NavMesh
BookCollectable.cs	<ul style="list-style-type: none"> Ορίζει ένα αντικείμενο ως "επιλέξιμο" στο δάσος Καθορίζει τον τύπο του αντικειμένου ως βιβλίο, χρησιμοποιώντας το Inventory System για τη διαχείριση του inventory του παίκτη.
ForestLevelLoader.cs	<ul style="list-style-type: none"> Αναλαμβάνει τη φόρτωση του επιπέδου του δάσους και την αρχικοποίησή του, λαμβάνοντας υπόψη την πρόοδο του παίκτη και τις αποθηκευμένες πληροφορίες.
Samir	<ul style="list-style-type: none"> Δεν επιτρέπει στο συγκεκριμένο NPC να γυρίσει να κοιτάξει τον παίκτη.

Path: "Assets/Scripts/Forest/Quests/Car Repair Quest",
 "Assets/Scripts/Forest/Quests/Find Book Quest" και
 "Assets/Scripts/Forest/Quests/Find Help Quest"

Όνομασία Αρχείου	Περιγραφή Κώδικα
Localization.cs	<ul style="list-style-type: none"> Ορίζει τα μηνύματα για τα quests σε όλες τις υποστηριζόμενες γλώσσες του παιχνιδιού.

Path: "Assets/Scripts/Inventory/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
UI_Inventory.cs	<ul style="list-style-type: none"> Διαχειρίζεται την εμφάνιση του αποθέματος του παίκτη στη UI χρήστη. Δημιουργεί δυναμικά κουτιά αντικειμένων στην οθόνη, απεικονίζοντας τα αντικείμενα που έχει ο παίκτης στο inventory του. Κάθε κουτί αντικειμένου περιλαμβάνει ένα εικονίδιο και, αν υπάρχει περισσότερα από ένα αντικείμενα του ίδιου τύπου, εμφανίζει τον αριθμό τους.
ItemAssets.cs	<ul style="list-style-type: none"> Αποθηκεύει τα assets των αντικειμένων που χρησιμοποιούνται στο σύστημα του Inventory.
Items.cs	<ul style="list-style-type: none"> Ορίζει τον τύπο του αντικειμένου και την ποσότητα του.
Inventory.cs	<ul style="list-style-type: none"> Αντιπροσωπεύει τον κατάλογο του Inventory, που περιέχει τα αντικείμενα που έχει συλλέξει ο παίκτης. Ο κατάλογος υλοποιείται με μια λίστα αντικειμένων.

Path: "Assets/Scripts/Localization/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
QuestsHint.cs	<ul style="list-style-type: none"> Ορίζει την βοήθεια που δίνουν τα NPCs στον παίκτη όταν δεν ξέρει τι να κάνει.
LocaleSelector.cs	<ul style="list-style-type: none"> Κατά την εκκίνηση του παιχνιδιού, φορτώνει την γλώσσα που έχει αποθηκευτεί προηγουμένως και αλλάζει τη γλώσσα σύμφωνα με αυτήν. Κάθε φορά που ο χρήστης επιλέγει μια νέα γλώσσα από το dropdown, εκτελείται η μέθοδος ChangeLocale, η οποία ενεργοποιεί τη μέθοδο SetLocale με τη χρήση ενός coroutine για να αλλάξει την τοπική ρύθμιση.
DefaultNpcMonologue.cs	<ul style="list-style-type: none"> Αντικαθιστά τον προηγούμενο μονόλογο ενός χαρακτήρα NPC με τον νέο μονόλογο, σύμφωνα με την γλώσσα που έχει επιλέξει ο χρήστης κατά την εκκίνηση.

Όνομασία Αρχείου	Περιγραφή Κώδικα
AlexHoldingBox.cs	<ul style="list-style-type: none"> • Ελέγχει την κίνηση του παίκτη που κρατά ένα κουτί • Αν ο παίκτης κρατά το κουτί, ελέγχει αν ο παίκτης προσπαθεί να κινηθεί και ανάλογα ρυθμίζει τον animator για περπάτημα και την ταχύτητα sprint. • Αν ο παίκτης δεν κινείται, απλώς ρυθμίζει τον animator σε idle.
Bob.cs	<ul style="list-style-type: none"> • Αλλάζει την κατάσταση του NPC "Bob", εξαρτώμενη από το εάν ο χαρακτήρας είναι σε κατάσταση ομιλίας ή όχι.
Box.cs	<ul style="list-style-type: none"> • Όταν ο παίκτης αλληλεπιδρά με ένα κουτί, αν ο παίκτης δεν κουβαλά κάποιο άλλο αντικείμενο, τότε το κουτί παίρνεται από τον παίκτη και ενεργοποιείται η αντίστοιχη αντίδραση από τον Quest Handler, αν υπάρχει κάποια τρέχουσα αποστολή σχετική με τη μεταφορά αντικειμένων.
MarketGirlWave.cs	<ul style="list-style-type: none"> • Ελέγχει την κίνηση ενός χαρακτήρα NPC, συγκεκριμένα μιας κοπέλας, που κάνει κίνηση χειροκροτήματος όταν ο παίκτης πλησιάζει
MoveTo.cs	<ul style="list-style-type: none"> • Χρησιμοποιείται για να κινήσει έναν χαρακτήρα προς ένα στόχο χρησιμοποιώντας το NavMesh.
RayCastState.cs	<ul style="list-style-type: none"> • Ελέγχει την ενεργοποίηση ή απενεργοποίηση ενός RayCast component ανάλογα με την εμφάνιση του παίκτη στον χώρο
RayCastFromCamera.cs	<ul style="list-style-type: none"> • Χρησιμοποιείται για να απεικονίσει ένα RayCast από την κύρια κάμερα του scene.
RayCast.cs	<ul style="list-style-type: none"> • Ελέγχει ένα RayCast από τον χαρακτήρα NPC προς τον χαρακτήρα παίκτη, έτσι ώστε να αναγνωρίσει αν ο παίκτης βρίσκεται μπροστά του.
PolicePatrol.cs	<ul style="list-style-type: none"> • Ελέγχει την περιπολία της πόλης από τους αστυνομικούς, χρησιμοποιώντας NavMeshAgent
PlacementHint.cs	<ul style="list-style-type: none"> • Ορίζει την τοποθεσία του hint για τα κουτιά που μεταφέρει ο παίκτης.
Patrol.cs	<ul style="list-style-type: none"> • Ελέγχει την διαδρομή NPCs/Αυτοκινήτων στην πόλη
TownLevelLoader.cs	<ul style="list-style-type: none"> • Αναλαμβάνει τη φόρτωση του επιπέδου της πόλης και την αρχικοποίησή του, λαμβάνοντας υπόψη την πρόοδο του παίκτη και τις αποθηκευμένες πληροφορίες.

Path: "Assets/Scripts/Town/MaleJacket/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
TransitionState.cs/MaleJacketPatrol.cs	<ul style="list-style-type: none"> • Ειδική κίνηση για ένα συγκεκριμένο τύπο NPC.

Path: "Assets/Scripts/Town/Quests/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
BeforeQuests.cs	<ul style="list-style-type: none"> • Αρχικοποιεί τον τίτλο και τη περιγραφή μιας αποστολής ανάλογα με το στάδιο που βρίσκεται η ιστορία.
GetInTheCarQuest.cs	<ul style="list-style-type: none"> • Ορίζει το quest για την διαχείριση του αυτοκινήτου (έξοδος από τη πόλη).

Path: "Assets/Scripts/Town/Quests/Find toolbox Quest" και
"Assets/Scripts/Town/Quests/Truck Unloading Quest"

Όνομασία Αρχείου	Περιγραφή Κώδικα
Localization.cs	<ul style="list-style-type: none"> • Ορίζει τα μηνύματα για τα quests σε όλες τις υποστηριζόμενες γλώσσες του παιχνιδιού.

Path: "Assets/Scripts/Save/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
SaveSystem.cs	<ul style="list-style-type: none"> • Χειρίζεται την αποθήκευση και τη φόρτωση δεδομένων προόδου για το παιχνίδι.

Path: "Assets/Scripts/Save/Data/"

Όνομασία Αρχείου	Περιγραφή Κώδικα
TownData.cs	<ul style="list-style-type: none"> • Τύποι δεδομένων αποθήκευσης των δεδομένων, των ρυθμίσεων και της προόδου στο παιχνίδι.
ForestData.cs	
ProgressData.cs	
SettingsData.cs	
TownData.cs	

Path: “Assets/Scripts/Timelines/”

Όνομασία Αρχείου	Περιγραφή Κώδικα
SceneLoader.cs	<ul style="list-style-type: none">• Φορτώνει την επόμενη σκηνή.

Αναλυτική παρουσίαση των σημαντικότερων scripts

QuestManager.cs

```
1  using System.Collections.Generic;
2  using Common.InteractionSystem;
3  using Inventory;
4  using Save;
5  using SaveData;
6  using TMPro;
7  using UnityEngine;
8
9  namespace Common.QuestSystem
10 {
11     public class QuestManager : MonoBehaviour
12     {
13         public static QuestManager Instance { get; private set; }
14
15         public GameObject questPanel;
16         public TextMeshProUGUI questTitleText;
17         public TextMeshProUGUI questDescriptionText;
18
19         public GameObject player;
20
21         public List<Quest> quests;
22         private int _currentQuestIdx;
23
24         public Quest currentQuest;
25
26         public void Awake()
27         {
28             if (Instance == null)
29             {
30                 Instance = this;
31                 _currentQuestIdx = -1;
32
33                 NextQuest();
34             }
35             else
36             {
37                 Destroy(gameObject);
38             }
39         }
40
41         private void UpdateQuestUI()
42         {
43             if (currentQuest == null)
44             {
45                 questPanel.SetActive(false);
46             }
47             else
48             {
49                 questPanel.SetActive(currentQuest.state == Quest.State.InProgress);
50                 questTitleText.text = quests[_currentQuestIdx].title;
51                 questDescriptionText.text = quests[_currentQuestIdx].description;
52             }
53         }
54
55         public void StartQuest(NPC requester)
56         {
57             if (currentQuest.responsibleNPC != null && currentQuest.responsibleNPC != requester) return;
58             if (requester != null) requester.ShowQuestHint(false);
59             UpdateQuestUI();
60         }
61
62         public void CompleteQuest()
63         {
64             if (currentQuest.responsibleNPC != null) currentQuest.responsibleNPC.availableQuest = null;
65
66             // Auto save quest
67             AutoSave();
68
69             NextQuest();
70         }
71     }
72 }
```

```

72     private void AutoSave()
73     {
74         var scene = currentQuest.scene;
75         switch (scene)
76         {
77             case Quest.Scene.Farm:
78                 List<Collectable> carrots = null;
79
80                 if (currentQuest is CollectQuest { itemType: Item.ItemType.Carrot } quest)
81                 {
82                     carrots = quest.collectables;
83                 }
84
85                 if (currentQuest.questIndex == 5)
86                 {
87                     SaveSystem.SaveMainProgress(new MainProgressData((int)LevelLoader.Scene.Forest));
88                 }
89
90                 SaveSystem.SaveFarmProgress(new FarmData(currentQuest.questIndex, player.transform.position, carrots));
91
92                 break;
93             case Quest.Scene.Forest:
94                 if (currentQuest.questIndex == 2)
95                 {
96                     SaveSystem.SaveMainProgress(new MainProgressData((int)LevelLoader.Scene.Town));
97                 }
98                 SaveSystem.SaveForestProgress(new ForestData(currentQuest.questIndex, player.transform.position));
99                 break;
100            case Quest.Scene.Town:
101                if (currentQuest.questIndex == 3)
102                {
103                    SaveSystem.SaveMainProgress(new MainProgressData((int)LevelLoader.Scene.Farm));
104                }
105                SaveSystem.SaveTownProgress(new TownData(currentQuest.questIndex, player.transform.position));
106                break;
107            }
108        }
109    }
110
111    private void NextQuest()
112    {
113        if (++_currentQuestIdx >= quests.Count)
114        {
115            currentQuest = null;
116            Debug.Log("No more available quests");
117            UpdateQuestUI();
118            return;
119        }
120
121        currentQuest = quests[_currentQuestIdx];
122
123        if (currentQuest.responsibleNPC == null)
124        {
125            currentQuest.StartQuest(null);
126        }
127        else
128        {
129            currentQuest.responsibleNPC.availableQuest = currentQuest;
130            currentQuest.responsibleNPC.ShowQuestHint(true);
131        }
132
133        UpdateQuestUI();
134    }
135
136    public void SetCurrentQuestIndex(int questIndex)
137    {
138        _currentQuestIdx = questIndex;
139        NextQuest();
140    }
141}
142
143

```

- Διαχειρίζεται τα quests και την πρόοδο του παίκτη σε σχέση με τα quests.
- Παρέχει μεθόδους για την ενεργοποίηση, ενημέρωση και ολοκλήρωση της κατάστασης των quests.
- Κρατάει το τρέχον quest και το επίπεδο προόδου του παίκτη.

LevelLoader.cs

```
1  using System;
2  using System.Collections;
3  using Common.InteractionSystem;
4  using Save;
5  using UnityEngine;
6  using UnityEngine.SceneManagement;
7
8  namespace Common
9  {
10     public abstract class LevelLoader : MonoBehaviour
11     {
12         public enum Scene
13         {
14             MainMenu,
15             Farm,
16             Forest,
17             Town
18         }
19
20         public Interactor interactor;
21         public GameObject player;
22
23         // Transition animation
24         public Animator transition;
25         public float transitionTime = 1f;
26
27         public void LoadScene(int sceneIndex)
28         {
29             StartCoroutine(LoadScreen(sceneIndex));
30         }
31
32         protected IEnumerator LoadScreen(int sceneIndex)
33         {
34             transition.SetTrigger("Start");
35             yield return new WaitForSeconds(transitionTime);
36             SceneManager.LoadScene(sceneIndex);
37         }
38
39         public abstract void InitScene();
40     }
41 }
42 }
```

- Αναλαμβάνει τη φόρτωση ενός επιπέδου στο παιχνίδι.
- Είναι υπεύθυνο για την αρχικοποίηση του επιπέδου, τη φόρτωση της αποθηκευμένης προόδου του παίκτη και τη διαχείριση των quests του επιπέδου.

RayCast.cs

```
1  using UnityEngine;
2
3  namespace Town
4  {
5      public class RayCast : MonoBehaviour
6      {
7          [SerializeField] private Transform alexTransform;
8          [SerializeField] private Animator npcAnimator;
9          private static readonly int WAVE = Animator.StringToHash("wave");
10         private Vector3 _mNpcEyesPosition, _mAlexEyesPosition;
11         private bool _mAlreadyWaved;
12
13     private void Start()
14     {
15         npcAnimator = GetComponent<Animator>();
16         alexTransform = GameObject.FindGameObjectWithTag("Player").transform;
17         _mNpcEyesPosition = transform.position + new Vector3(0, 1.7f, 0);
18     }
19
20     private void FixedUpdate()
21     {
22         _mAlexEyesPosition = alexTransform.position + new Vector3(0, 1.65f, 0);
23
24         Vector3 direction = _mAlexEyesPosition - _mNpcEyesPosition;
25
26         Debug.DrawRay(_mNpcEyesPosition, direction, Color.yellow);
27
28         if (!Physics.Raycast(_mNpcEyesPosition, direction, out var hit, 10f)) return;
29         if (hit.collider.transform == alexTransform)
30         {
31             if (_mAlreadyWaved) return;
32             npcAnimator.SetTrigger(WAVE);
33             _mAlreadyWaved = true;
34         }
35         else _mAlreadyWaved = false;
36     }
37 }
38 }
```

- Υλοποιεί μια ακτίνα (raycast) που χρησιμοποιείται για την ανίχνευση του παίκτη στο παιχνίδι.
- Εκτελεί μια ακτίνα από ένα σημείο και ανιχνεύει τον παίκτη που είναι μπροστά της.

PlacementHint.cs

```
1  using System.Collections;
2  using Common.InteractionSystem;
3  using Common.QuestSystem;
4  using UnityEngine;
5
6  namespace Town
7  {
8      public class PlacementHint : Interactable
9      {
10         public float flashDuration = 1f;
11         private Renderer _myRenderer;
12         public GameObject objectToActivate;
13
14         public InteractableObject interactableTargetObject;
15
16         private void Start()
17         {
18             _myRenderer = GetComponent<Renderer>();
19             // Start the flashing coroutine when the script is enabled
20             StartCoroutine(FlashHint());
21         }
22
23         private IEnumerator FlashHint()
24         {
25             float elapsedTime = 0f;
26             while (true)
27             {
28                 float lerpValue = Mathf.PingPong(elapsedTime / flashDuration, 1f);
29                 _myRenderer.material.color = new Color(_myRenderer.material.color.r, _myRenderer.material.color.g,
30                     _myRenderer.material.color.b, lerpValue);
31
32                 elapsedTime += Time.deltaTime;
33
34                 yield return null;
35             }
36         }
37
38         // You can stop the flashing by calling this method
39         public void StopFlashing()
40         {
41             StopAllCoroutines();
42             // Ensure the renderer is fully opaque when stopped
43             _myRenderer.material.color = new Color(_myRenderer.material.color.r, _myRenderer.material.color.g,
44                 _myRenderer.material.color.b, 1f);
45         }
46
47         public override void OnInteract(Interactor interactor)
48         {
49             base.OnInteract(interactor);
50
51             if (interactor.GetCarryingObject() == interactableTargetObject)
52             {
53                 // Instant interaction
54                 interactor.Drop();
55                 objectToActivate.SetActive(true);
56                 gameObject.SetActive(false);
57
58                 if (QuestManager.Instance.currentQuest is CarryQuest quest)
59                 {
60                     quest.OnObjectPlaced(interactableTargetObject);
61                 }
62             }
63
64             interactor.EndInteraction(this);
65         }
66     }
67 }
```

- Υλοποιεί ένα σημείο ενεργοποίησης (hint) για την τοποθέτηση αντικειμένων (κουτιών) στο παιχνίδι.
- Χρησιμοποιείται για να εμφανίζει ένα flash effect σε ένα αντικείμενο που μπορεί να τοποθετηθεί σε μια συγκεκριμένη θέση.
- Παρέχει μεθόδους για την αλληλεπίδραση με το αντικείμενο και τη διαχείριση του quest που σχετίζεται με αυτό.

SaveSystem.cs

```
1  using UnityEngine;
2  using System.IO;
3  using System.Runtime.Serialization.Formatters.Binary;
4  using Save.Data;
5
6  namespace Save
7  {
8      public static class SaveSystem
9      {
10         private const string SETTINGS_PATH = "/settings.save";
11         private const string LOCALE_PATH = "/locale.save";
12
13         private const string MAIN_PROGRESS_PATH = "/main_progress.save";
14         private const string FARM_PROGRESS_PATH = "/farm_progress.save";
15         private const string FOREST_PROGRESS_PATH = "/forest_progress.save";
16         private const string TOWN_PROGRESS_PATH = "/town_progress.save";
17
18         public static void SaveSettings(SettingsData settingsData)
19         {
20             var formatter = new BinaryFormatter();
21             var path = Application.persistentDataPath + SETTINGS_PATH;
22
23             try
24             {
25                 using var stream = new FileStream(path, FileMode.Create);
26                 formatter.Serialize(stream, settingsData);
27                 Debug.Log("Settings have been saved at " + path);
28             }
29             catch (System.Exception e)
30             {
31                 Debug.LogError("Error saving settings: " + e.Message);
32             }
33         }
34
35         public static SettingsData LoadSettings()
36         {
37             var path = Application.persistentDataPath + SETTINGS_PATH;
38             try
39             {
40                 if (File.Exists(path))
41                 {
42                     using var stream = new FileStream(path, FileMode.Open);
43                     var formatter = new BinaryFormatter();
44                     return formatter.Deserialize(stream) as SettingsData;
45                 }
46             }
47             catch (System.Exception e)
48             {
49                 Debug.LogError("Error loading settings: " + e.Message);
50             }
51
52             Debug.Log("Settings save file not found in " + path);
53             return null;
54         }
55
56         public static void SaveLocale(int localeId)
57         {
58             var formatter = new BinaryFormatter();
59             var path = Application.persistentDataPath + LOCALE_PATH;
60
61             try
62             {
63                 using var stream = new FileStream(path, FileMode.Create);
64                 formatter.Serialize(stream, localeId);
65                 Debug.Log("Locale has been saved at " + path);
66             }
67             catch (System.Exception e)
68             {
69                 Debug.LogError("Error saving locale: " + e.Message);
70             }
71         }
72     }
```

```

73     public static int LoadLocale()
74     {
75         var path = Application.persistentDataPath + LOCALE_PATH;
76         try
77         {
78             if (File.Exists(path))
79             {
80                 using var stream = new FileStream(path, FileMode.Open);
81                 var formatter = new BinaryFormatter();
82                 return (int) formatter.Deserialize(stream);
83             }
84         }
85         catch (System.Exception e)
86         {
87             Debug.LogError("Error loading locale: " + e.Message);
88         }
89         Debug.Log("Locale save file not found in " + path);
90         return -1;
91     }
92
93     public static void SaveTownProgress(TownData townData)
94     {
95         var formatter = new BinaryFormatter();
96         var path = Application.persistentDataPath + TOWN_PROGRESS_PATH;
97
98         try
99         {
100            using var stream = new FileStream(path, FileMode.Create);
101            formatter.Serialize(stream, townData);
102            Debug.Log("Town progress has been saved at " + path);
103        }
104        catch (System.Exception e)
105        {
106            Debug.LogError("Error saving town progress: " + e.Message);
107        }
108    }
109
110    public static TownData LoadTownProgress()
111    {
112        var path = Application.persistentDataPath + TOWN_PROGRESS_PATH;
113        try
114        {
115            if (File.Exists(path))
116            {
117                using var stream = new FileStream(path, FileMode.Open);
118                var formatter = new BinaryFormatter();
119                return formatter.Deserialize(stream) as TownData;
120            }
121        }
122        catch (System.Exception e)
123        {
124            Debug.LogError("Error loading town progress: " + e.Message);
125        }
126
127        Debug.Log("Town progress save file not found in " + path);
128        return null;
129    }
130
131    public static void SaveFarmProgress(FarmData farmData)
132    {
133        var formatter = new BinaryFormatter();
134        var path = Application.persistentDataPath + FARM_PROGRESS_PATH;
135
136        try
137        {
138            using var stream = new FileStream(path, FileMode.Create);
139            formatter.Serialize(stream, farmData);
140            Debug.Log("Farm progress has been saved at " + path);
141        }
142        catch (System.Exception e)
143        {

```

```

145     }
146     }
147   }
148 
149   public static FarmData LoadFarmProgress()
150   {
151     var path = Application.persistentDataPath + FARM_PROGRESS_PATH;
152     try
153     {
154       if (File.Exists(path))
155       {
156         using var stream = new FileStream(path, FileMode.Open);
157         var formatter = new BinaryFormatter();
158         return formatter.Deserialize(stream) as FarmData;
159       }
160     }
161     catch (System.Exception e)
162     {
163       Debug.LogError("Error loading farm progress: " + e.Message);
164     }
165 
166     Debug.Log("Farm progress save file not found in " + path);
167     return null;
168   }
169 
170   public static void SaveForestProgress(ForestData forestData)
171   {
172     var formatter = new BinaryFormatter();
173     var path = Application.persistentDataPath + FOREST_PROGRESS_PATH;
174 
175     try
176     {
177       using var stream = new FileStream(path, FileMode.Create);
178       formatter.Serialize(stream, forestData);
179       Debug.Log("Forest progress has been saved at " + path);
180     }
181     catch (System.Exception e)
182     {
183       Debug.LogError("Error saving forest progress: " + e.Message);
184     }
185   }
186 
187   public static ForestData LoadForestProgress()
188   {
189     var path = Application.persistentDataPath + FOREST_PROGRESS_PATH;
190     try
191     {
192       if (File.Exists(path))
193       {
194         using var stream = new FileStream(path, FileMode.Open);
195         var formatter = new BinaryFormatter();
196         return formatter.Deserialize(stream) as ForestData;
197       }
198     }
199     catch (System.Exception e)
200     {
201       Debug.LogError("Error loading forest progress: " + e.Message);
202     }
203 
204     Debug.Log("Forest progress save file not found in " + path);
205     return null;
206   }
207 
208   public static void SaveMainProgress(MainProgressData mainProgressData)
209   {
210     var formatter = new BinaryFormatter();
211     var path = Application.persistentDataPath + MAIN_PROGRESS_PATH;
212 
213     try
214     {
215       using var stream = new FileStream(path, FileMode.Create);
216       formatter.Serialize(stream, mainProgressData);

```

```

217     Debug.Log("Main progress has been saved at " + path);
218 }
219 catch (System.Exception e)
220 {
221     Debug.LogError("Error saving main progress: " + e.Message);
222 }
223 }

224 public static MainProgressData LoadMainProgress()
225 {
226     var path = Application.persistentDataPath + MAIN_PROGRESS_PATH;
227     try
228     {
229         if (File.Exists(path))
230         {
231             using var stream = new FileStream(path, FileMode.Open);
232             var formatter = new BinaryFormatter();
233             return formatter.Deserialize(stream) as MainProgressData;
234         }
235     }
236     catch (System.Exception e)
237     {
238         Debug.LogError("Error loading main progress: " + e.Message);
239     }
240 }

241 Debug.Log("Main progress save file not found in " + path);
242 return null;
243 }

244 public static bool DeleteProgressFiles()
245 {
246     var success = true;

247     // Paths to the progress files
248     var progressPaths = new[] { MAIN_PROGRESS_PATH, FARM_PROGRESS_PATH, FOREST_PROGRESS_PATH, TOWN_PROGRESS_PATH };

249     foreach (var progressPath in progressPaths)
250     {
251         try
252         {
253             var fullPath = Application.persistentDataPath + progressPath;

254             if (File.Exists(fullPath))
255             {
256                 File.Delete(fullPath);
257                 Debug.Log("Deleted progress file: " + fullPath);
258             }
259             else
260             {
261                 Debug.Log("Progress file not found: " + fullPath);
262             }
263         }
264         catch (System.Exception e)
265         {
266             Debug.LogError("Error deleting progress file: " + e.Message);
267             success = false; // Set success to false if any deletion fails
268         }
269     }
270 }

271     return success;
272 }

273 }

274 }

275 }

276 }

277 }

278 }

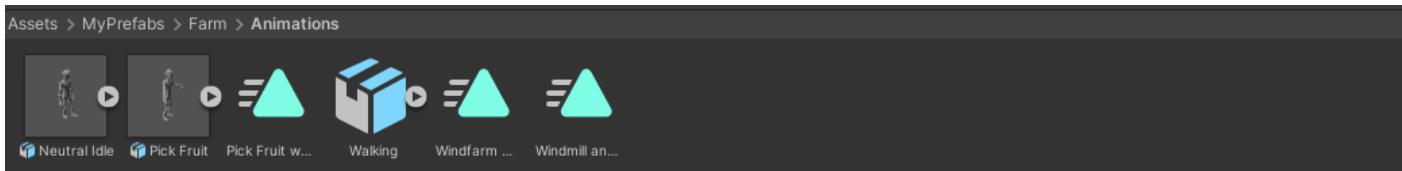
279 }

280 }

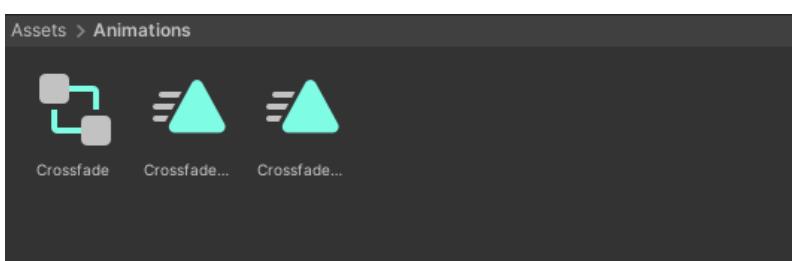
```

- Αναλαμβάνει την αποθήκευση και τη φόρτωση δεδομένων, των ρυθμίσεων και της προόδου του παιχνιδιού από το δίσκο.
- Παρέχει μεθόδους για την αποθήκευση και τη φόρτωση των ρυθμίσεων, της προόδου του παίκτη και άλλων δεδομένων του παιχνιδιού.

Animations που χρησιμοποιήθηκαν



Όλα τα παραπάνω animations προήλθαν από το site: <https://www.mixamo.com/>



Όλα τα παραπάνω animations προήλθαν από εμάς.

Αναλυτική παρουσίαση για χρήστες

Όταν ανοίξει ο παικτης για πρώτη φορά το παιχνίδι θα αντικρίσει το βασικό **Menu/Μενού** της εφαρμογής:

Menu



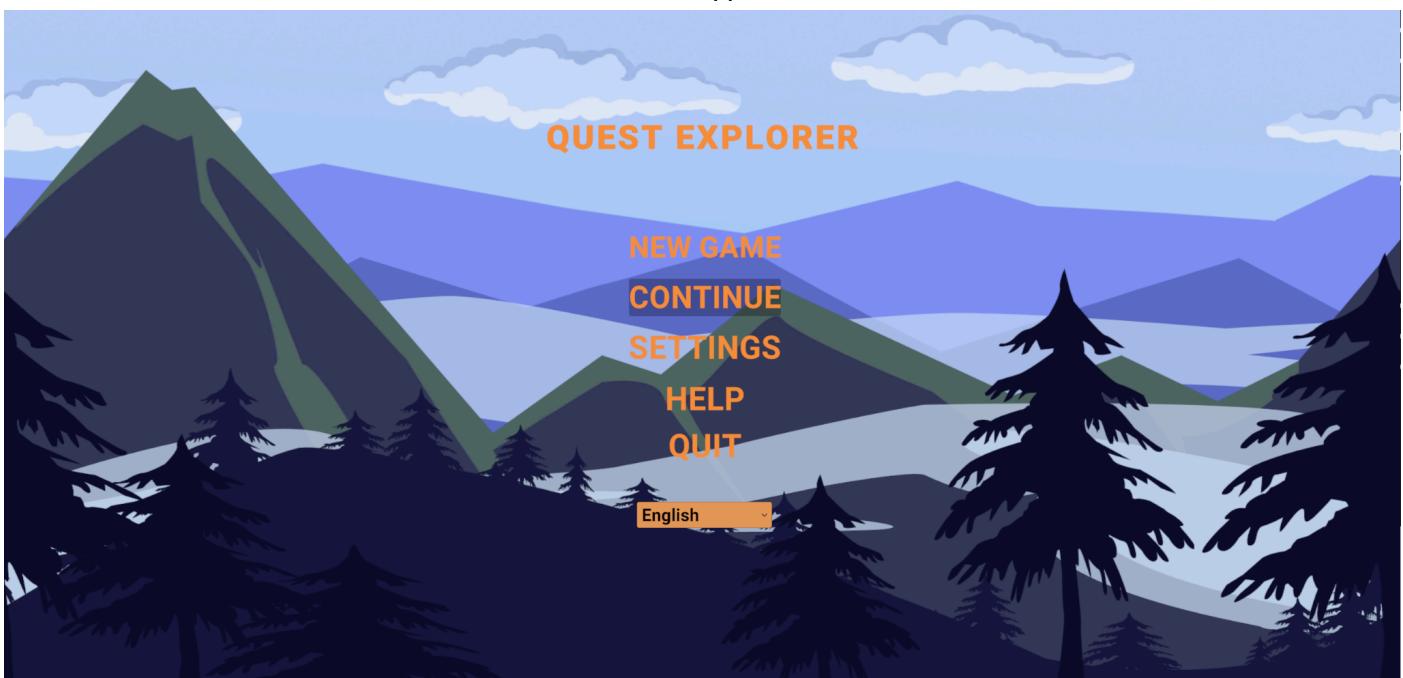
Όλες οι επιλογές είναι διαθέσιμες εκτός της επιλογής **Continue/Συνέχεια**. Η επιλογή αυτή ενεργοποιείται όταν έχει ξεκινήσει ήδη μία φορά το παιχνίδι ο παίκτης.

Για να ξεκινήσει το παιχνίδι επιλέγει το **New Game/Νέο Παιχνίδι**.

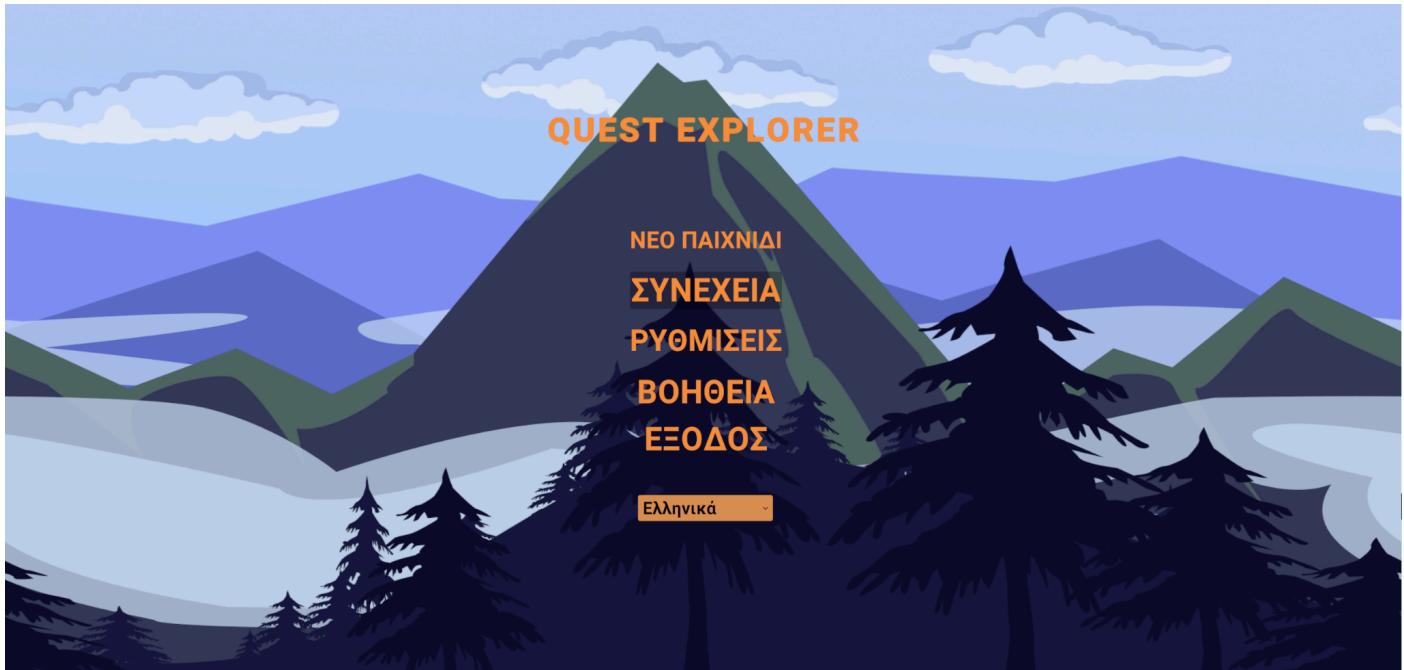
Το παιχνίδι υποστηρίζει πλήρως 2 γλώσσες: **Αγγλικά** και **Ελληνικά**.

Μπορεί να αλλάξει την γλώσσα του παιχνιδιού από την αρχική οθόνη στο **Drop Down μενού**:

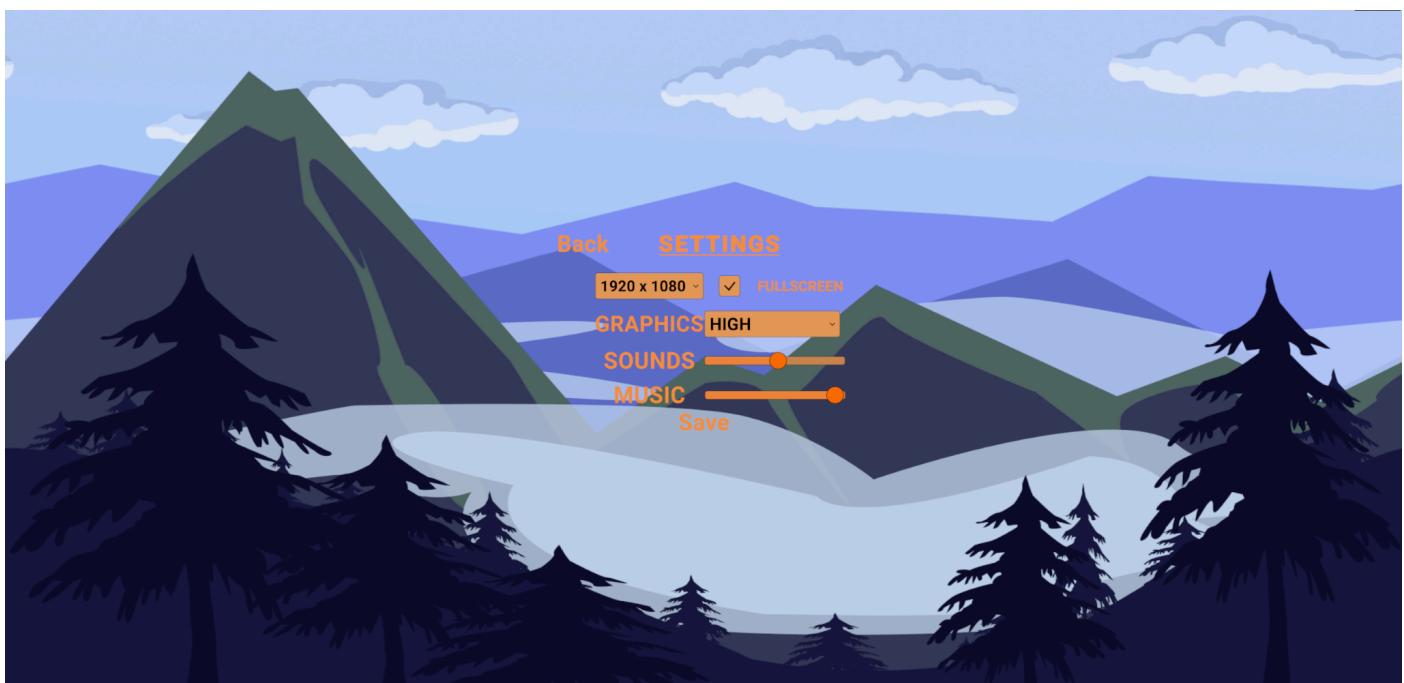
UI στα Αγγλικά:



UI στα Ελληνικά:



Settings/Ρυθμίσεις



Ο παίκτης μπορεί να αλλάξει τις ρυθμίσεις του παιχνιδιού, όπως:

- Resolution/Ανάλυση Οθόνης,
- Graphics/Ποιότητα Γραφικών,
- Sounds/Ήχοι
- και τη Music/Μουσική

Όταν κάνει τις αλλαγές που επιθυμεί, για να τις αποθηκεύσει, πατάει την επιλογή Save/Αποθήκευση.

Help/Βοήθεια



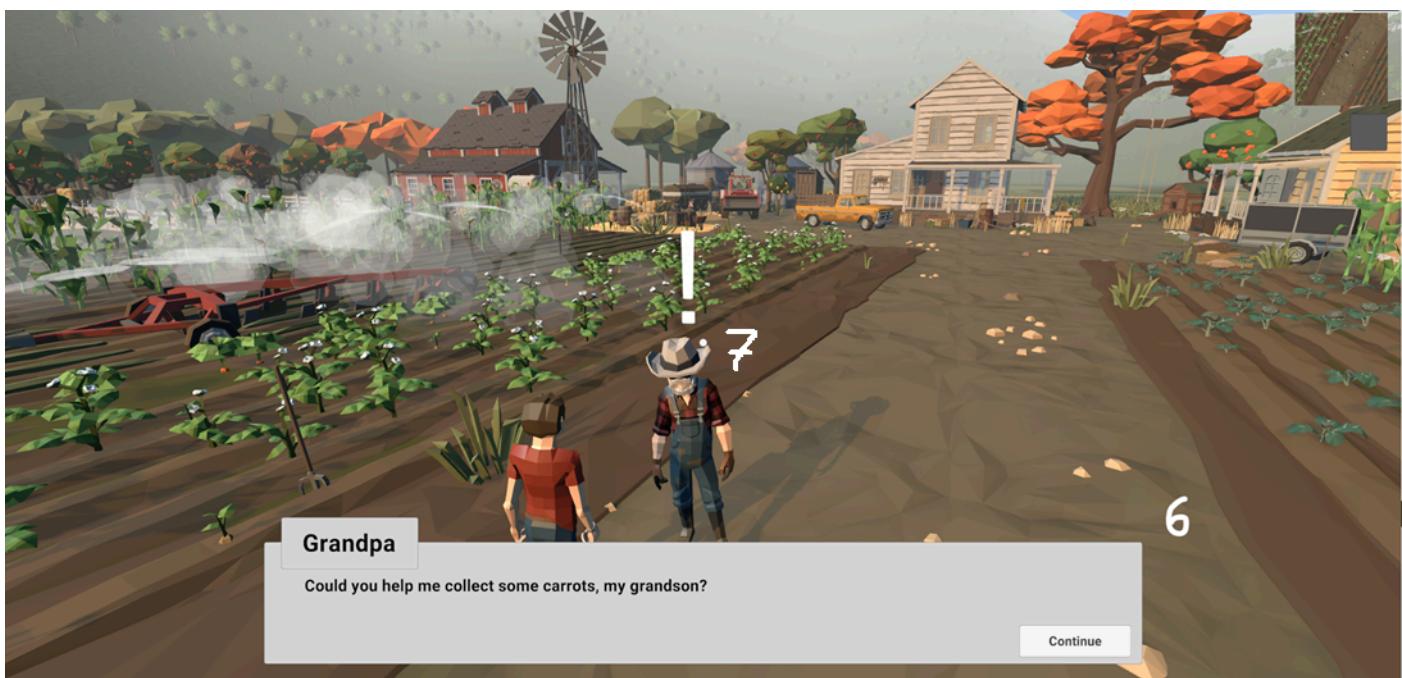
Στην οθόνη αυτή ο χρήστης μπορεί να δει όλα τα κουμπιά με τα οποία μπορεί να χειριστεί τον παίκτη στο παιχνίδι. Τα κουμπιά είναι:

- **W:** Move forward/Μετακίνηση μπροστά
- **A:** Turn Left/Στροφή αριστερά
- **S:** Move backward/Μετακίνηση πίσω
- **D:** Turn right/Στροφή δεξιά
- **SPACE:** Jump/Άλμα
- **Esc:** Pause/Παύση
- **SHIFT:** Run/Τρέξιμο
- **E:** Interaction/Αλληλεπίδραση
- **Q:** End interaction/Τέλος αλληλεπίδρασης

UI κατά την διάρκεια του παιχνιδιού



- 1) Εμφάνιση περιγραφής για την εκπλήρωση του quest
- 2) Εμφάνιση hint για την αλληλεπίδραση του παίκτη με το περιβάλλον.
- 3) Μίνι χάρτης που δηλώνει την τοποθεσία του παίκτη στην περιοχή.
- 4) Inventory χαρακτήρα. Εμφανίζει τα αντικείμενα που κρατάει εκείνη την στιγμή ο παίκτης.
- 5) Η αλληλεπίδραση με τα NPC ή τα αντικείμενα του κόσμου γίνεται όταν ο παίκτης τα προσεγγίζει με την άσπρη τελεία που φαίνεται στο κέντρο της οθόνης.



- 6) Εμφάνιση ομιλητή και κειμένου διαλόγου. Για την εμφάνιση του επόμενου διαλόγου επιλέγει με τον κέρσορα το κουμπί Continue/Συνέχεια.
- 7) Μπορεί ο χρήστης να αλληλεπιδρά με NPC που έχουν διαθέσιμα Quests όταν αυτά έχουν (!) θαυμαστικό πάνω από το κεφάλι τους.

Η συλλογή αντικειμένων έχει αντίκτυπο στον κόσμο του παιχνιδιού.

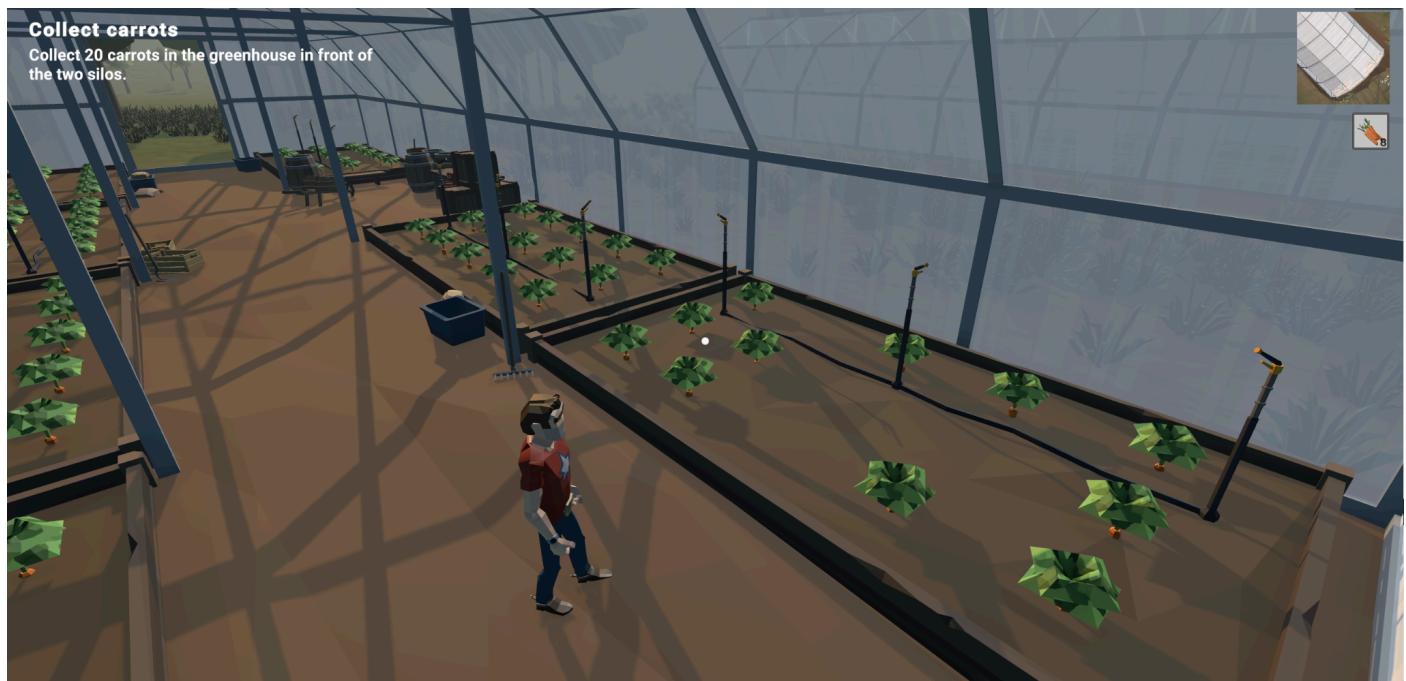
Παραδείγματα:

- 1) Συλλογή καρότων:

Πριν την συλλογή καρότων



Μετά την συλλογή καρότων

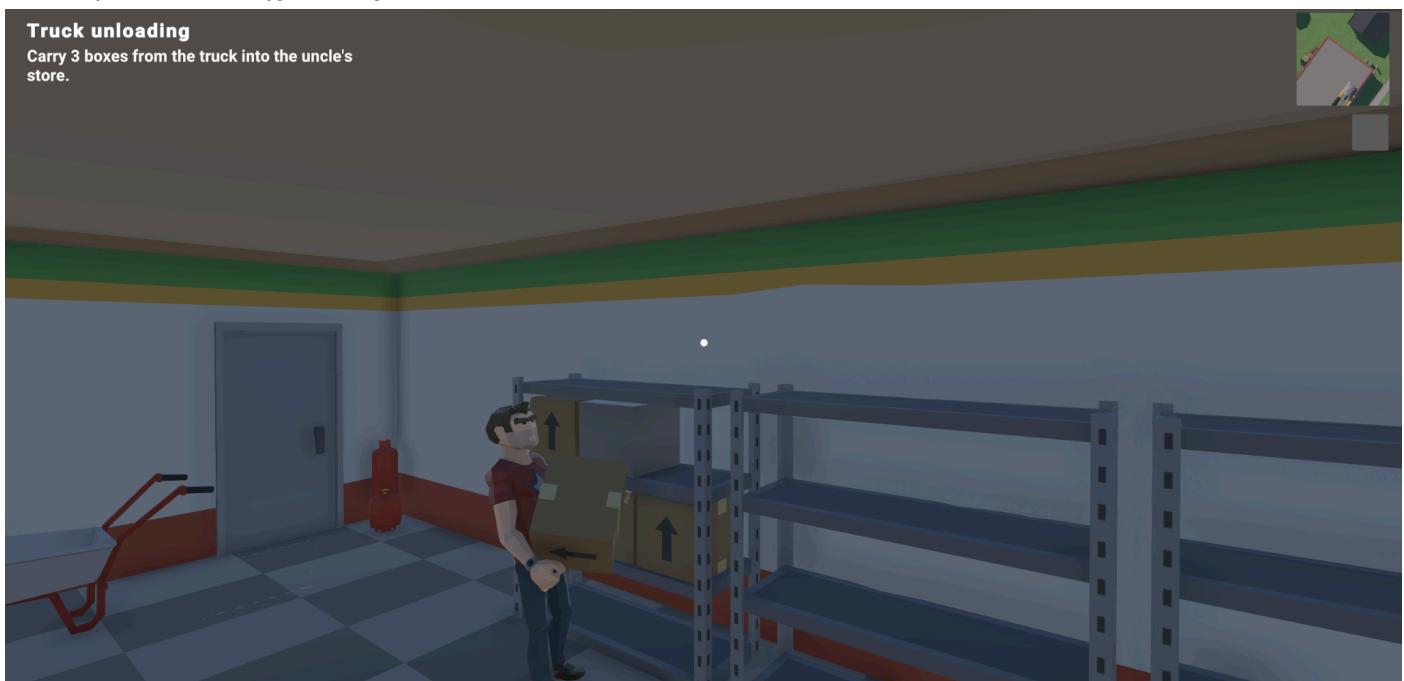


2) Μεταφορά κούτας:

Πριν την μετακίνηση κούτας:



Κατά τη μεταφορά της κούτας:



Μετά την μετακίνηση κούτας:

Στην τελική τοποθεσία



Στην αρχική τοποθεσία



Pause Menu/Μενού Παύσης



Εμφανίζονται οι λειτουργίες του αρχικού μενού - εκτός της αλλαγής γλώσσας. Για να αλλάξει ο παίκτης την γλώσσα θα μπορεί να την αλλάξει μόνο από το αρχικό μενού.

Screenshots από το παιχνίδι

Φάρμα





Δάσος







Πόλη





Κριτήρια

Αληθοφάνεια

Ο κόσμος είναι ρεαλιστικός όσον αφορά τα γραφικά και την αλληλεπίδραση. Η κίνηση των χαρακτήρων και των αντικειμένων είναι φυσική και η προσθήκη δυναμικών ήχων και εφέ φωτισμού συμβάλλουν στην αίσθηση ρεαλισμού.

Περιεχόμενο

Το περιεχόμενο του κόσμου είναι πλούσιο και ποικίλο. Υπάρχουν διάφορα σημεία ενδιαφέροντος και δραστηριότητες για τον παίκτη. Η ιστορία που διαδραματίζεται στο παιχνίδι είναι σύνθετη και ενδιαφέρουσα. Άλλαζουν διαρκώς τα σκηνικά του παιχνιδιού χωρίς να χάνεται το ενδιαφέρον του παίκτη.

Πληρότητα

Ο κόσμος είναι πλήρης και ολοκληρωμένος. Ο παίκτης μπορεί να εξερευνήσει ελεύθερα και να αλληλεπιδράσει με όλα τα στοιχεία. Η εργασία δεν παρουσιάζει αδικαιολόγητα κενά ή ασυνέπειες. Έχουν υλοποιηθεί τα πάντα.

Σχεδιασμός

Ο σχεδιασμός του **Quest Explorer** είναι λεπτομερής και αισθητικά όμορφος. Τα γραφικά είναι low poly. Δίνεται με αυτή την υλοποίηση η δυνατότητα να παίζουν όσοι θέλουν το παιχνίδι αυτό χωρίς να έχει το παιχνίδι υψηλές απαιτήσεις για να παίξει στα συστήματά τους. Τέλος, η οργάνωση και η δομή του περιβάλλοντος είναι εύχρηστες για τον παίκτη.

Αισθητική

Ο κόσμος του **Quest Explorer** είναι ευπαρουσίαστος και οπτικά ελκυστικός. Τα χρώματα, τα υλικά και οι υφές συνδυάζονται αρμονικά. Η ατμόσφαιρα του κόσμου ταιριάζει με τον χώρο και τον χρόνο της ιστορίας που αφηγείται η εργασία.

Πρωτοτυπία

Η εργασία παρουσιάζει πρωτότυπες ιδέες και υλοποιήσεις. Ο σχεδιασμός του κόσμου, η ιστορία και οι μηχανισμοί παιχνιδιού δεν είναι αντιγραφές υπαρχόντων παιχνιδιών.

Χρηστικότητα

Ο κόσμος είναι εύχρηστος και ο παίκτης μπορεί να τον εξερευνήσει εύκολα. Τα μενού και οι οδηγίες είναι σαφείς και κατανοητές. Η εργασία δεν παρουσιάζει σφάλματα ή προβλήματα που δυσχεραίνουν την εμπειρία του παίκτη. Το παιχνίδι καθοδηγεί τον παίκτη σε κάθε στάδιο της ιστορίας.

Κίνηση

Η εργασία περιλαμβάνει κινούμενα στοιχεία, όπως:

- Κίνηση χαρακτήρων (NPCs, παίκτης)
- Κίνηση αντικειμένων (αυτοκίνητα, ανεμόμυλος)
- Εφέ φωτισμού και ατμοσφαιρικά φαινόμενα

Η κίνηση είναι ομαλή και ρεαλιστική.

Λειτουργικότητα

Η εργασία περιλαμβάνει λειτουργικά στοιχεία, όπως είναι η αλληλεπίδραση με αντικείμενα (συλλογή, χρήση), η ενεργοποίηση μηχανισμών (πότισμα, αυτοκίνητο, κουτιά, μήλα, καρότα), τα *quests* με στόχους και ανταμοιβές ως προς τον παίκτη για την συνέχεια της πλοκής της ιστορίας. Η τελική έκδοση του παιχνιδιού είναι σταθερή και χωρίς σφάλματα.

Ανάπτυξη

Χρησιμοποιήθηκε η 3D μηχανή του Unity σε συνδυασμό της γλώσσας προγραμματισμού C#. Έχουν προστεθεί οδηγίες για την αλληλεπίδραση του παίκτη με τον κόσμο του παιχνιδιού.

Αστοχίες κώδικα, Δυσκολίες, Μελλοντικές επεκτάσεις και Συμπεράσματα

Ο κώδικας δεν περιέχει κάποιου είδους λογική αστοχία. Το παιχνίδι είναι πολύ εύκολο να επεκταθεί με την προσθήκη νέων quests, νέων διαλόγων και νέων προσώπων στην ήδη υπάρχουσα δομή. Η 3D μηχανή του Unity σε συνδυασμό της γλώσσας προγραμματισμού C# έχουν ένα μεγάλο πλήθος δυνατοτήτων και assets ώστε να δίνεται στον developer η δυνατότητα να ξεδιπλώνει την φαντασία του και να μπορεί να δημιουργήσει οτιδήποτε θελήσει.

Πηγές των assets

Nature Sounds Pack - Free

Δημιουργός: Goumain Antoine

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/audio/sound-fx/nature-sounds-pack-free-202076>

Χρήση: Χρησιμοποιήθηκαν ήχοι για τη φάρμα και το δάσος

POLYGON City - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/environments/urban/polygon-city-low-poly-3d-art-by-synty-95214>

Χρήση: Χρησιμοποιήθηκαν NPCs και prefabs για την δημιουργία της πόλης

POLYGON City Characters - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/characters/humanoids/humans/polygon-city-characters-low-poly-3d-art-by-synty-106757>

Χρήση: Χρησιμοποιήθηκαν NPCs

POLYGON Farm - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/environments/industrial/polygon-farm-low-poly-3d-art-by-synty-146192>

Χρήση: Χρησιμοποιήθηκαν NPCs, σπίτια, χωράφια και αγροτικές μηχανές για την δημιουργία της φάρμας

POLYGON Gang Warfare - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/environments/industrial/polygon-gang-warfare-low-poly-3d-art-by-synty-150589>

Χρήση: Χρησιμοποιήθηκαν props για προσθήκη λεπτομέρειας στο design του κόσμου

POLYGON Heist - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/environments/urban/polygon-heist-low-poly-3d-art-by-synty-97949>

Χρήση: Χρησιμοποιήθηκαν οχήματα για τη δημιουργία της πόλης

POLYGON Icons Pack - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/gui/polygon-icons-pack-low-poly-3d-art-by-synty-202117>

Χρήση: Χρησιμοποιήθηκαν props για προσθήκη λεπτομέρειας στο design του κόσμου

POLYGON Nature - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/vegetation/trees/polygon-nature-low-poly-3d-art-by-synty-120152>

Χρήση: Χρησιμοποιήθηκαν asset για τη δημιουργία της φύσης στο δάσος και στη φάρμα

POLYGON Office - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/environments/urban/polygon-city-low-poly-3d-art-by-synty-95214>

Χρήση: Χρησιμοποιήθηκαν props για προσθήκη λεπτομέρειας στο design του κόσμου

POLYGON Town Pack - Low Poly 3D Art by Synty

Δημιουργός: Synty Studios

Σύνδεσμος στο asset:

<https://assetstore.unity.com/packages/3d/environments/urban/polygon-town-pack-low-poly-3d-art-by-synty-121115>

Χρήση: Χρησιμοποιήθηκαν asset για τη δημιουργία της πόλης