



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος πτυχιακής εργασίας	Online multiplayer επιτραπέζιο παιχνίδι φτιαγμένο σε unity Online multiplayer board game made with unity
Όνοματεπώνυμο Φοιτητή	ΚΑΠΙΝΙΑΡΗΣ ΧΡΥΣΟΒΑΛΑΝΤΗΣ ΓΕΩΡΓΙΟΣ
Πατρώνυμο	ΘΕΟΔΩΡΟΣ
Αριθμός Μητρώου	Π20229
Επιβλέπων	ΘΕΜΙΣΤΟΚΛΗΣ ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ, ΚΑΘΗΓΗΤΗΣ

Ημερομηνία Παράδοσης: ΙΑΝΟΥΑΡΙΟΣ 2025

Copyright©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου Θεμιστοκλή Παναγιωτόπουλο για την ευκαιρία που μου έδωσε για την ανάπτυξη αυτού του παιχνιδιού.
Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και τους κοντινούς μου φίλους για την βοήθεια και την στήριξη τους κατά την διάρκεια της εργασίας.
Τέλος θα ήθελα να ευχαριστήσω τους φίλους και γνωστούς που απάντησαν στην μικρή ερεύνα που απάντησαν στην ερεύνα και μου έδωσαν έτσι μια ξεκάθαρη πορεία για την δημιουργία της εφαρμογής.

Περίληψη

Η ανάπτυξη ενός multiplayer παιχνιδιού αποτελεί μια από τις πιο απαιτητικές προκλήσεις στον χώρο της προγραμματιστικής δημιουργίας, καθώς απαιτεί ένα στιβαρό και αποτελεσματικό networking σύστημα. Το networking είναι η καρδιά κάθε παιχνιδιού πολλών παικτών, υπεύθυνο για τη συγχρονισμένη ανταλλαγή δεδομένων μεταξύ client και server, εξασφαλίζοντας ότι κάθε παίκτης βιώνει το παιχνίδι με τον ίδιο ρυθμό και τις ίδιες πληροφορίες. Οι απαιτήσεις του networking δεν περιορίζονται απλώς στη μεταφορά δεδομένων· περιλαμβάνουν την ελαχιστοποίηση της καθυστέρησης (latency), την αποφυγή απώλειας δεδομένων (packet loss), και την εξασφάλιση σταθερής εμπειρίας για όλους τους χρήστες, ανεξαρτήτως του επιπέδου συνδεσιμότητάς τους.

Η επιλογή του κατάλληλου networking framework είναι καθοριστική για την επιτυχία ενός τέτοιου εγχειρήματος. Διαφορετικές λύσεις, όπως το Netcode for GameObjects, το FishNet, ή το Photon, προσφέρουν μοναδικές δυνατότητες και πλεονεκτήματα, καθιστώντας την επιλογή τους άμεσα συνδεδεμένη με τις ανάγκες του εκάστοτε παιχνιδιού. Η σωστή διαχείριση του συγχρονισμού δεδομένων, η υποστήριξη διαφορετικών πλατφορμών και η ευκολία ενσωμάτωσης νέων λειτουργιών είναι μερικές μόνο από τις παραμέτρους που πρέπει να ληφθούν υπόψη.

Καθώς η τεχνολογία εξελίσσεται, οι απαιτήσεις για υψηλή απόδοση και κλιμακωσιμότητα αυξάνονται. Το networking δεν είναι απλώς μια τεχνική απαίτηση, αλλά μια στρατηγική επιλογή που επηρεάζει άμεσα την εμπειρία των παικτών, καθιστώντας το θεμέλιο για την ανάπτυξη σύγχρονων, πολυπαικτικών παιχνιδιών που διατηρούν την προσοχή και τη συμμετοχή του κοινού.

Λέξεις Κλειδιά: βιντεοπαιχνίδι, επιτραπέζιο, πολλών παικτών, στρατηγική, unity3D, C#, δικτύωση

Abstract

The development of a multiplayer game is one of the most demanding challenges in the field of programming, as it requires a robust and efficient networking system. Networking is the heart of any multiplayer game, responsible for the synchronized exchange of data between client and server, ensuring that every player experiences the game at the same pace and with the same information. The requirements of networking are not limited to simply transferring data; they include minimizing latency, avoiding packet loss, and ensuring a stable experience for all users, regardless of their connectivity level.

The choice of the appropriate networking framework is crucial for the success of such a project. Different solutions, such as Netcode for GameObjects, FishNet, or Photon, offer unique capabilities and advantages, making their selection directly tied to the specific needs of the game. Proper management of data synchronization, support for different platforms, and ease of integrating new features are just some of the parameters that must be taken into account. As technology evolves, the demand for high performance and scalability increases. Networking is not just a technical requirement but a strategic decision that directly impacts the player experience, forming the foundation for developing modern multiplayer games that retain audience engagement and participation.

Key Words: video game, board game, multiplayer, strategy, Unity3D, C#, networking

Πίνακας Περιεχομένων

Περιεχόμενα

Copyright©.....	ii
Ευχαριστίες.....	3
Περίληψη.....	iv
Abstract.....	5
Πίνακας Περιεχομένων	6
Πίνακας Εικόνων.....	8
Εισαγωγή	1
1. Unity Services.....	2
1.1 Λειτουργικότητα.....	2
1.2 Unity lobby	2
1.2.1 Γενική Περιγραφή.....	2
1.2.2 Scripts	2
1.3 Unity Relay.....	6
2. Το παιχνίδι	7
2.1 Σκοπός του παιχνιδιού.....	7
2.2 Τρόποι αύξησης networth	7
2.2.3 Αναβάθμιση περιοχών	10
3. Λειτουργίες παίκτη	15
3.1 Σειρά του παίκτη	15
3.2 Γενικές ενέργειες	16
3.3 Πλήκτρα παίκτη.....	19
4. Χάρτης	20
4.1 Διεκδικήσασες περιοχές.....	20
4.1.1 Χρωματιστές περιοχές	20
4.2 Ειδικές περιοχές.....	21
5. Συγχρονισμός δεδομένων.....	23
5.1 Δικτύωση/ Netcode for Game Objects.....	23
5.1.3 Network Variables.....	25
5.1.4 Network Variables vs ClientRpc.....	26
5.2 Συγχρονισμός Αμφιέσεων	26
5.3 Συγχρονισμός Animation.....	27
5.4 Συγχρονισμός περιοχών	28
6. Netcode vs άλλες επιλογές.....	29
6.1 Netcode vs Mirror.....	29
6.2 Netcode vs Fishnet	30
6.3 Netcode vs Photon.....	32
7. Εξέλιξη και μελλοντικές αλλαγές.....	34

7.1 Δημιουργία Καινούργιων Χαρτών και Περιβαλλόντων.....	34
7.2 Αναβάθμιση της Αλληλεπίδρασης Μεταξύ των Παικτών	34
7.3 Βελτίωση του User Interface (UI).....	34
7.4 Προσαρμοστικότητα και Δυναμική Αντίδραση στις Στρατηγικές του Παίκτη.....	34
7.5 Διαχείριση Εξωτερικών Πηγών και Διαδικασιών.....	34
7.6 Αναβάθμιση Στρατηγικών Στοιχείων και Αντικειμένων	35
8. Προβλήματα που πρόέκυψαν.....	36
8.1 Μη συγχρονισμός interface.....	36
8.2 Αγορά Περιοχών	36
8.3 Ανταλλαγή	36
8.4 Μη έναρξη του παιχνιδιού.....	37
9. Αλλαγές από την original monopoly	38
9.1 Μεταφορά σε υπολογιστή	38
Συμπεράσματα.....	40
Βιβλιογραφία.....	41

Πίνακας Εικόνων

Εικόνα 1: Επιλογή Αμφίεσης	3
Εικόνα 2: Lobby interface	4
Εικόνα 3: Lobby List interface	5
Εικόνα 4: Create Lobby interface	5
Εικόνα 5: interface Αγοράς περιοχής	9
Εικόνα 6: Interface πλειστηριασμού	10
Εικόνα 7: Επίπεδο Αναβάθμισης 1	11
Εικόνα 8: Επίπεδο Αναβάθμισης 2	12
Εικόνα 9: Επίπεδο Αναβάθμισης 3	12
Εικόνα 10: Επίπεδο Αναβάθμισης 4	13
Εικόνα 11: Επίπεδο Αναβάθμισης 5	14
Εικόνα 12: Μενού ανταλλαγής	17
Εικόνα 13: Προβολή χρωματιστής περιοχής χωρίς Ιδιοκτήτη	18
Εικόνα 14: Προβολή περιοχής υπηρεσίας	19

Εισαγωγή

Το παιχνίδι είναι ένα σύγχρονο επιτραπέζιο δημιουργημένο με το unity3D. Εμπνευσμένο από τη Monopoly, που συνδυάζει στρατηγική και ανταγωνισμό, προσφέροντας τη δυνατότητα παγκόσμιας σύνδεσης παικτών. Οι συμμετέχοντες μπορούν να αγοράζουν, να επενδύουν και να διαπραγματεύονται μεταξύ τους.

Βασικός σκοπός της επιλογής παιχνιδιού ήταν η μάθηση και η εξοικείωση με τα διάφορα εργαλεία που υπάρχουν για την δικτύωση στα βιντεοπαιχνίδια πολλών παικτών, μέσω μιας εφαρμογής που δίνει λιγότερο σημασία σε θέματα όπως το interface και τα visuals και επικεντρώνεται πλήρως στην λειτουργικότητα αυτών των εργαλείων.

1. Unity Services

1.1 Λειτουργικότητα

Το **Unity Lobby** και το **Unity Relay** είναι ισχυρές βιβλιοθήκες της Unity που διευκολύνουν τη δημιουργία σύνδεσης μεταξύ παικτών σε online περιβάλλοντα, αναλαμβάνοντας την πιο περίπλοκη υλοποίηση των σχετικών λειτουργιών. Για τη σωστή χρήση τους, είναι απαραίτητο ο χρήστης να περάσει από μια διαδικασία πιστοποίησης μέσω κώδικα, κατά την οποία του αποδίδεται ένας μοναδικός ατομικός κωδικός (token). Αυτός ο κωδικός χρησιμεύει για την ασφαλή αναγνώριση και στις δύο υπηρεσίες, επιτρέποντας τη διαχείριση των συνδέσεων και των συνεδριών παιχνιδιού. Η πιστοποίηση πραγματοποιείται κατά την έναρξη της εφαρμογής, όταν ο χρήστης επιλέγει την επιλογή "Play". Με αυτόν τον τρόπο, διασφαλίζεται ότι οι χρήστες εισέρχονται με ασφάλεια στο online περιβάλλον και είναι έτοιμοι να συμμετάσχουν στο παιχνίδι, χρησιμοποιώντας τις δυνατότητες που παρέχουν οι δύο αυτές υπηρεσίες.

Το **Netcode for GameObjects (NGO)** είναι μια βιβλιοθήκη δικτύωσης της Unity που παρέχει ένα ισχυρό και ευέλικτο εργαλείο για την ανάπτυξη online multiplayer παιχνιδιών με βάση τα game objects. Σχεδιασμένο για απλότητα και προσαρμοστικότητα, το NGO διευκολύνει την υλοποίηση μηχανισμών δικτύωσης όπως συγχρονισμό κατάστασης, αποστολή μηνυμάτων μεταξύ πελατών (clients) και διακομιστών (server), καθώς και τη διαχείριση αντικειμένων στο δίκτυο. Με ενσωματωμένες δυνατότητες όπως ο συγχρονισμός μεταβλητών (variables), η διαχείριση spawn/despawn αντικειμένων και η υποστήριξη εξουσιοδότησης (ownership), το Netcode for GameObjects προσφέρει όλα τα απαραίτητα εργαλεία για τη δημιουργία αποδοτικών και σταθερών multiplayer εμπειριών. Υποστηρίζει επίσης πλήρη προσαρμογή και επέκταση, επιτρέποντας στους προγραμματιστές να προσαρμόζουν τη δικτύωση στις ανάγκες του παιχνιδιού τους, είτε πρόκειται για peer-to-peer είτε για server-hosted αρχιτεκτονική.

1.2 Unity lobby

1.2.1 Γενική Περιγραφή

Το **Unity Lobby** είναι μια υπηρεσία της Unity που επιτρέπει τη δημιουργία και διαχείριση εικονικών χώρων συνάντησης (lobbies) για online multiplayer παιχνίδια. Τα lobbies λειτουργούν ως χώροι συγκέντρωσης των παικτών, όπου μπορούν να συνδεθούν, να ανταλλάξουν πληροφορίες, να ρυθμίσουν παραμέτρους του παιχνιδιού και να προετοιμαστούν πριν την έναρξη. Με το Unity Lobby, οι προγραμματιστές έχουν τη δυνατότητα να δημιουργούν, να ενημερώνουν και να διαγράφουν lobbies μέσω ενός εύχρηστου API, παρέχοντας έναν τρόπο συγχρονισμού δεδομένων, όπως προτιμήσεις παικτών ή ρυθμίσεις παιχνιδιού, μεταξύ όλων των συμμετεχόντων.

Η υπηρεσία υποστηρίζει διαφορετικά επίπεδα πρόσβασης, επιτρέποντας τη δημιουργία δημόσιων, ιδιωτικών ή προσβάσιμων μέσω πρόσκλησης lobbies, ενώ παράλληλα ενσωματώνεται με συστήματα matchmaking, διευκολύνοντας τη σύνδεση παικτών με παρόμοιες δεξιότητες ή προτιμήσεις. Οι παίκτες μπορούν να συμμετέχουν, να αποχωρούν ή να προσκαλούν άλλους, ενισχύοντας την κοινωνική διάσταση του παιχνιδιού.

1.2.2 Scripts

1.2.2.1 Customize_character

Το παιχνίδι προσφέρει στους παίκτες τη δυνατότητα να προσαρμόσουν την εμφάνιση του χαρακτήρα τους, επιλέγοντας από έναν μεγάλο αριθμό διαθέσιμων αμφιέσεων. Αυτές οι αμφιέσεις καλύπτουν ευρύ φάσμα ιστορικών περιόδων, περιλαμβάνοντας τόσο χαρακτηρισ της μεσαιωνικής εποχής όσο και σύγχρονους. Οι παίκτες έχουν την ευχέρεια να επιλέξουν την αμφίεση που αντανακλά καλύτερα τις προτιμήσεις και το στυλ τους, προσφέροντας έτσι μια εξατομικευμένη εμπειρία μέσα στο παιχνίδι.

Ο χρήστης μπορεί να περιηγηθεί εύκολα μεταξύ των διαθέσιμων αμφιεσμένων επιλογών, πατώντας το κατάλληλο κουμπί στο μενού του παιχνιδιού. Με αυτόν τον τρόπο, μπορούν να δοκιμάσουν διάφορες εμφανίσεις και να επιλέξουν αυτή που τους ελκύει περισσότερο,

προσφέροντας στον χαρακτήρα τους μοναδικότητα και προσωπικότητα. Η διαδικασία αυτή ενισχύει τη διαδραστικότητα του παιχνιδιού, δίνοντας στους παίκτες τη δυνατότητα να εξερευνήσουν μια πληθώρα επιλογών που καλύπτουν διαφορετικές χρονικές περιόδους και στυλ.



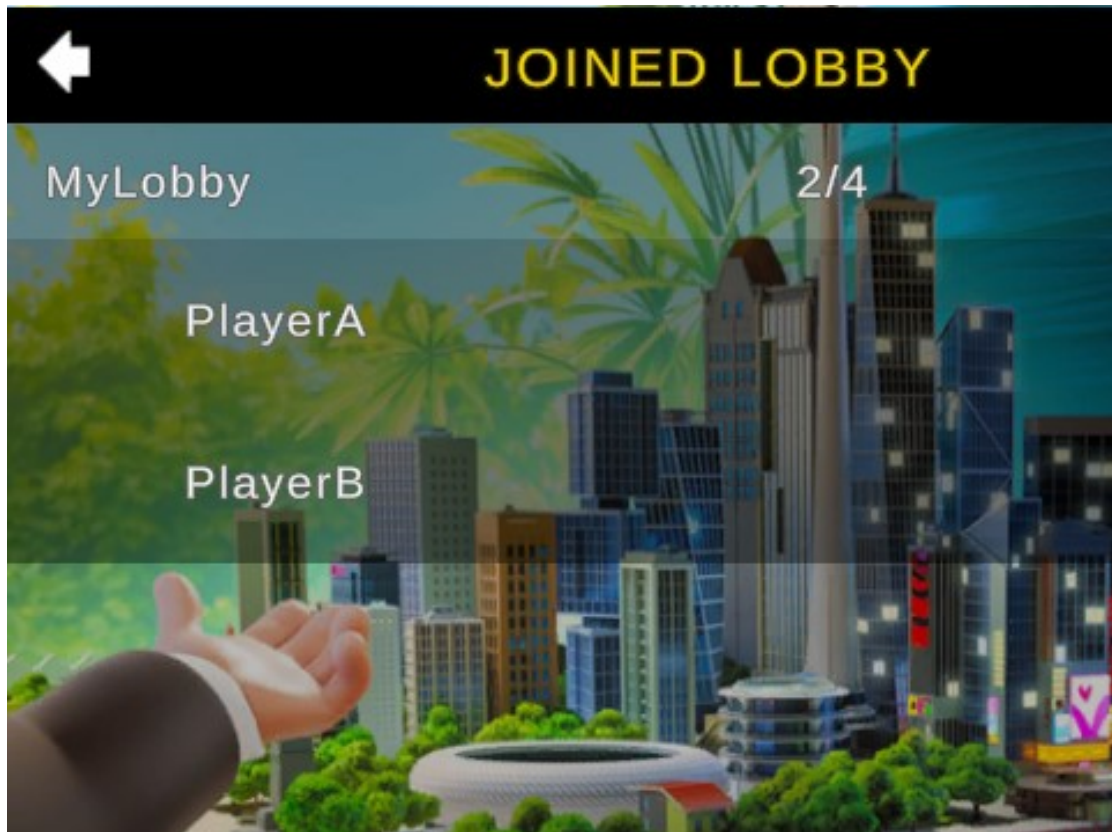
Εικόνα 1: Επιλογή Αμφίεσης

1.2.2.2 Lobby Manager

Ο Lobby Manager είναι υπεύθυνος για τη διαχείριση όλων των λειτουργιών που αφορούν το lobby του παιχνιδιού. Αυτό περιλαμβάνει τη δημιουργία νέων lobby, τη σύνδεση των παικτών σε αυτά, καθώς και την αποχώρησή τους όταν το επιθυμούν. Ο Lobby Manager παρέχει στους χρήστες τη δυνατότητα να επιλέξουν ένα lobby για να παίξουν, εξασφαλίζοντας ότι οι παίκτες μπορούν να εισέλθουν και να αποχωρήσουν από το παιχνίδι με ευκολία και χωρίς προβλήματα. Παράλληλα, διαχειρίζεται την αλληλεπίδραση των παικτών εντός του lobby, όπως η παρακολούθηση των συνδέσεων, και η απομάκρυνση παικτών σε περίπτωση αποσύνδεσης ή άλλων τεχνικών προβλημάτων.

1.2.2.3 LobbyUi

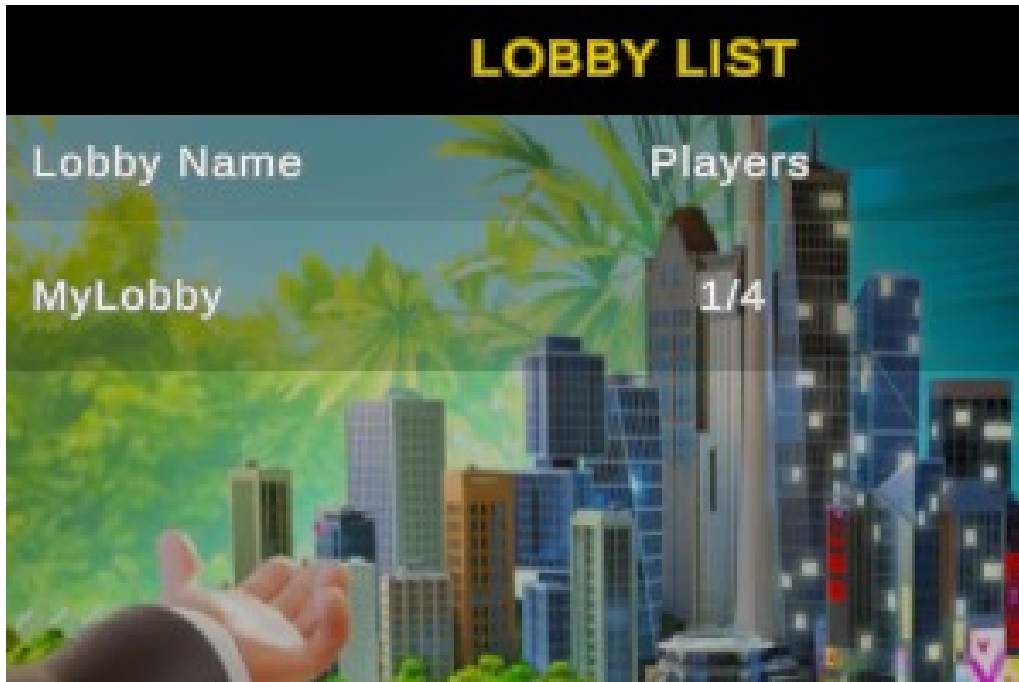
Σε συνεννόηση με τον LobbyManager το lobbyUI είναι υπεύθυνο για την σωστή απεικόνιση των παικτών που βρίσκονται στο κάθε lobby. Γίνεται συγχρονισμός στοιχείων όπως το όνομα του κάθε παίκτη που είναι συνδεδεμένος καθώς και εμφάνιση όλων των παικτών κάθε φορά που ένας παίκτης εισέρχεται ή αποχωρεί από το lobby. Επίσης στον ιδιοκτήτη δίνεται η δυνατότητα αφαίρεσης προβληματικών παικτών από το lobby, καθώς και η διαχείριση της έναρξης του παιχνιδιού. Μόλις συγκεντρωθούν όλοι οι απαιτούμενοι παίκτες, ο δημιουργός μπορεί να ξεκινήσει το παιχνίδι συνδέοντας όλους του παίκτες στο παιχνίδι και στην κατάλληλη σκηνή.



Εικόνα 2: Lobby interface

1.2.2.4 LobbyListUi

Αυτό το script είναι υπεύθυνο για την απεικόνιση των διαθέσιμων lobby. Παρουσιάζονται πληροφορίες όπως το όνομα του lobby και ο αριθμός των παικτών που συμμετέχουν. Ο χρήστης, όταν επιλέξει ένα lobby, εισέρχεται σε αυτό για να συμμετάσχει στο παιχνίδι. Το script λειτουργεί έτσι ώστε να παρέχει στον χρήστη μια εύκολη και γρήγορη επιλογή του lobby στο οποίο επιθυμεί να παίξει, αναδεικνύοντας τις διαθέσιμες επιλογές και διευκολύνοντας τη διαδικασία εισόδου.



Εικόνα 3: Lobby List interface

1.2.2.5 Create_Lobby

Πατώντας το κατάλληλο κουμπί πάνω αριστερά στο LobbyList interface, μπορεί ένας χρήστης να δημιουργήσει το δικό του Lobby. Μπορεί να επιλέξει το όνομα του lobby, το αν είναι δημόσιο ή ιδιωτικό και τον μέγιστο αριθμό παικτών.



Εικόνα 4: Create Lobby interface

1.2.2.6 PlayerManager

Ο PlayerManager είναι υπεύθυνος για την αποθήκευση και διαχείριση των προσωπικών δεδομένων του χρήστη, όπως το όνομα και η αμφίεση του, τα οποία επιλέγει ο χρήστης στο lobby του παιχνιδιού. Αυτά τα δεδομένα αποθηκεύονται κατά την διάρκεια της αλληλεπίδρασης του χρήστη με το σύστημα, διασφαλίζοντας ότι οι προσωπικές του ρυθμίσεις διατηρούνται ακριβώς όπως τις άφησε, χωρίς να χάνονται. Αυτό επιτρέπει στον χρήστη να προσαρμόσει τον χαρακτήρα του με βάση τις προτιμήσεις του και να εξασφαλίσει μια εξατομικευμένη εμπειρία κατά τη διάρκεια του παιχνιδιού.

Ακόμα, ο PlayerManager διατηρεί αυτές τις επιλογές κατά την αλλαγή σκηνής, ώστε όταν ο χρήστης εισέρχεται σε νέες περιοχές του παιχνιδιού ή μεταβαίνει σε διαφορετικά περιβάλλοντα, οι προσωπικές του ρυθμίσεις (όνομα, αμφίεση) να παραμένουν αναλλοίωτες. Αυτό εξασφαλίζει την ομαλή μετάβαση του χρήστη από το lobby στο παιχνίδι και την αδιάλειπτη εμπειρία του, διασφαλίζοντας ότι οι αλλαγές που έκανε κατά την προετοιμασία του δεν χάνονται σε καμία φάση του παιχνιδιού.

1.3 Unity Relay

Υπάρχουν δύο βασικοί τρόποι για τη σύνδεση παικτών σε ένα multiplayer παιχνίδι. Ο πρώτος τρόπος είναι μέσω ενός κεντρικού διακομιστή στον οποίο συνδέονται όλοι οι παίκτες. Ο δεύτερος τρόπος, γνωστός ως **peer-to-peer σύνδεση**, απαιτεί έναν από τους παίκτες να αναλάβει τον ρόλο του διακομιστή (host), ενώ οι υπόλοιποι συνδέονται σε αυτόν. Παρόλο που η δεύτερη μέθοδος είναι πιο οικονομική και δεν απαιτεί έναν μόνιμο διακομιστή, παρουσιάζει σοβαρά μειονεκτήματα. Για να λειτουργήσει, ο χρήστης-διακομιστής πρέπει να ανοίξει θύρες στο ρούτερ του, να απενεργοποιήσει firewalls και άλλες προστασίες στον υπολογιστή του, ενώ παράλληλα πρέπει να μοιραστεί την προσωπική του διεύθυνση IP. Αυτές οι απαιτήσεις όχι μόνο περιπλέκουν τη διαδικασία αλλά δημιουργούν και σημαντικά ζητήματα ασφαλείας.

Το **Unity Relay** έρχεται να εξαλείψει αυτά τα προβλήματα, προσφέροντας μια σύγχρονη και απλοποιημένη λύση για τη σύνδεση παικτών. Το Relay λειτουργεί ως μεσάζοντας που μεταφέρει δεδομένα μεταξύ των παικτών χωρίς να απαιτείται η απευθείας ανταλλαγή IP διευθύνσεων. Με αυτόν τον τρόπο, οι παίκτες συνδέονται μέσω ενός ασφαλούς, αξιόπιστου καναλιού, αποφεύγοντας τις τεχνικές δυσκολίες και τις ανησυχίες για την ασφάλεια.

Ο συνδυασμός του Unity Relay με το **Unity Lobby** παρέχει έναν ολοκληρωμένο μηχανισμό σύνδεσης παικτών. Όταν δημιουργείται ένα lobby, παράλληλα ενεργοποιείται και το Relay, το οποίο εκχωρεί έναν μοναδικό κωδικό (Relay Join Code). Αυτός ο κωδικός αποθηκεύεται στα δεδομένα του lobby και χρησιμοποιείται για τη σύνδεση των παικτών. Όταν ένας νέος παίκτης εισέρχεται στο lobby, λαμβάνει τον κωδικό του Relay και συνδέεται αυτόματα μέσω αυτού. Με την έναρξη του παιχνιδιού από τον host, όλοι οι παίκτες που είναι συνδεδεμένοι στο lobby και το Relay μεταφέρονται στη σκηνή του παιχνιδιού, έτοιμοι να παίξουν χωρίς καθυστερήσεις ή τεχνικά προβλήματα. Η διαδικασία αυτή εξασφαλίζει ομαλή μετάβαση από τη φάση προετοιμασίας στο lobby στη δράση του παιχνιδιού, διατηρώντας ταυτόχρονα υψηλά επίπεδα ασφαλείας και αξιοπιστίας.

2. Το παιχνίδι

2.1 Σκοπός του παιχνιδιού

Σκοπός του παιχνιδιού είναι η συγκέντρωση ενός συνολικού ποσού \$10.000 "networth". Το "networth" υπολογίζεται ως ο συνδυασμός των χρημάτων που διαθέτει ο παίκτης και της αξίας των περιοχών που κατέχει. Οι περιοχές αυτές, οι οποίες μπορεί να είναι διάφοροι χώροι ή εδάφη μέσα στο παιχνίδι, αυξάνουν την αξία του παίκτη με την πάροδο του χρόνου και με την ανάπτυξή τους. Επομένως, οι παίκτες πρέπει να επενδύσουν στρατηγικά τόσο στη συλλογή χρημάτων όσο και στην απόκτηση και αναβάθμιση περιοχών για να ενισχύσουν το "networth" τους και να φτάσουν τον στόχο των 10.000.

Η αξία των περιοχών που κατέχει ο παίκτης είναι σημαντική, καθώς κάθε περιοχή μπορεί να προσφέρει διαφορετική αύξηση στο συνολικό "networth". Ο παίκτης καλείται να βρει την ισορροπία μεταξύ της απόκτησης περιοχών και της διαχείρισης των χρημάτων του, καθώς και να λάβει στρατηγικές αποφάσεις για την ανάπτυξη των περιοχών του. Η τελική επιτυχία εξαρτάται από τη σωστή διαχείριση αυτών των στοιχείων, με στόχο την επίτευξη των 10.000 "networth" και την επικράτηση στο παιχνίδι.

2.2 Τρόποι αύξησης networth

Στο παιχνίδι υπάρχουν 3 βασικοί τρόποι να αυξησει ένας παίκτης το networth του.

2.2.1 Χρήματα

Τα χρήματα αποτελούν το βασικό εργαλείο των παικτών που τους επιτρέπει την αγορά περιοχών. Αν ο παίκτης δεν έχει αρκετά χρήματα για την αγορά μιας περιοχής δεν μπορεί να την αγοράσει. Τα χρήματα του παίκτη μπορούν να πάρουν αρνητικές τιμές, αλλά προφανώς δεν μπορεί να αγοράσει περιοχές.

2.2.1.1 Ενοίκιο

Αν ένας παίκτης καταλήξει σε περιοχή που ανήκει σε άλλον παίκτη τότε πληρώνει αυτόν τον παίκτη ενοίκιο ένα προκαθορισμένο πόσο. Το πόσο αυτό αυξάνεται ανάλογα με το κόστος της περιοχής(2.2.2) και το επίπεδο της(2.2.2)

2.2.1.2 Αφετηρία

Στο παιχνίδι, κάθε φορά που ένας παίκτης περνά από την αφετηρία, λαμβάνει ένα χρηματικό ποσό που εξαρτάται από τη διάρκεια του παιχνιδιού. Η συγκεκριμένη μηχανική όχι μόνο ανταμείβει τους παίκτες για την πρόοδό τους στο ταμπλό, αλλά προσθέτει και μια εξελικτική δυναμική που αντικατοπτρίζει την πρόοδο του παιχνιδιού.

Το αρχικό ποσό που λαμβάνει ένας παίκτης είναι **\$200**. Ωστόσο, αυτό το ποσό δεν παραμένει σταθερό. Καθώς το παιχνίδι συνεχίζεται και η διάρκεια αυξάνεται, το ποσό που λαμβάνει ο παίκτης κάθε φορά που περνά από την αφετηρία επίσης αυξάνεται. Η προοδευτική αυτή αύξηση ενισχύει την ένταση και τη στρατηγική του παιχνιδιού, καθώς τα υψηλότερα ποσά μπορούν να αλλάξουν σημαντικά την οικονομική κατάσταση των παικτών.

2.2.1.3 Τόκοι

Κάθε φορά που ξεκινά η σειρά ενός παίκτη, αυτός λαμβάνει ένα χρηματικό ποσό που βασίζεται στα διαθέσιμα χρήματά του. Αυτή η μηχανική προσφέρει μια ενδιαφέρουσα δυναμική στο παιχνίδι, καθώς οι παίκτες ανταμείβονται για τη διαχείριση των πόρων τους, ενώ ενθαρρύνεται η οικονομική στρατηγική.

Το ποσό που λαμβάνει ο παίκτης υπολογίζεται ως **\$100 για κάθε \$1000 που διαθέτει**, προσφέροντάς του έναν τρόπο να αυξήσει τα κεφάλαιά του. Ωστόσο, υπάρχει ένα ανώτατο όριο στο ποσό που μπορεί να λάβει, το οποίο είναι **\$500 ανά γύρο**. Αυτό σημαίνει ότι ακόμη και αν ο παίκτης έχει περισσότερα από \$5000, η μέγιστη ανταμοιβή που θα λάβει παραμένει σταθερή. Ο μηχανισμός αυτός εξισορροπεί το παιχνίδι, αποτρέποντας τους πλούσιους παίκτες από το να αποκτήσουν υπερβολικά μεγάλο πλεονέκτημα έναντι των υπολοίπων.

Αυτή η δυνατότητα προσθέτει βάθος στη στρατηγική του παιχνιδιού, καθώς οι παίκτες πρέπει να λάβουν υπόψη όχι μόνο την απόκτηση ακινήτων αλλά και τη διατήρηση επαρκών χρημάτων για να επωφεληθούν από τη συγκεκριμένη ανταμοιβή. Οι αποφάσεις τους σχετικά με την επένδυση ή την εξοικονόμηση χρημάτων γίνονται κρίσιμες για την επιτυχία τους. Τέλος, η παροχή αυτής της ανταμοιβής σε κάθε γύρο διατηρεί το παιχνίδι ζωντανό και δίνει στους παίκτες περισσότερες ευκαιρίες να ανακάμψουν ή να ενισχύσουν τη θέση τους.

2.2.1.4 Ειδικές περιοχές

Στον χάρτη του παιχνιδιού υπάρχουν **ειδικές περιοχές** που έχουν σχεδιαστεί για να αλληλεπιδρούν άμεσα με τα χρήματα του παίκτη, προσθέτοντας στρατηγική και δυναμική στην οικονομική διαχείριση. Αυτές οι περιοχές δεν είναι απλώς τμήματα του ταμπλό αλλά κεντρικά στοιχεία που επηρεάζουν τη ροή του παιχνιδιού, δημιουργώντας ευκαιρίες και κινδύνους για τους παίκτες.

2.2.2 Αγορά περιοχών

Όταν ένας παίκτης καταλήγει σε μια διαθέσιμη περιοχή στο παιχνίδι, έχει την επιλογή είτε να την αγοράσει είτε να την παραχωρήσει για πλειστηριασμό. Αυτή η απόφαση αποτελεί ένα σημαντικό στρατηγικό σημείο, καθώς επηρεάζει τόσο τη δική του οικονομική κατάσταση όσο και τη δυναμική του παιχνιδιού για τους υπόλοιπους παίκτες.

Αν ο παίκτης επιλέξει να αγοράσει την περιοχή, το **networth** του αυξάνεται αμέσως κατά το κόστος της περιοχής. Το networth αντιπροσωπεύει τη συνολική αξία του παίκτη, συμπεριλαμβάνοντας τις περιοχές που κατέχει, το ρευστό χρήμα και τυχόν άλλα στοιχεία. Η αγορά μιας περιοχής, επομένως, δεν είναι απλώς μια δαπάνη, αλλά και μια επένδυση που μπορεί να αποδώσει μέσα από ενοίκια που θα πληρώνουν άλλοι παίκτες όταν βρεθούν σε αυτήν.

Στην κάρτα που εμφανίζεται για την περιοχή, παρέχονται όλες οι απαραίτητες πληροφορίες για να βοηθήσουν τον παίκτη να λάβει την απόφασή του. Συγκεκριμένα, η κάρτα περιλαμβάνει:

- **Το κόστος της περιοχής:** Το ποσό που απαιτείται για την αγορά της περιοχής.
- **Το βασικό ενοίκιο:** Το ποσό που θα πρέπει να πληρώσουν άλλοι παίκτες όταν βρεθούν στην περιοχή, εφόσον δεν υπάρχουν αναβαθμίσεις.
- **Το όνομα και το χρώμα της περιοχής:** Αυτές οι πληροφορίες βοηθούν τον παίκτη να κατανοήσει πώς η περιοχή εντάσσεται στον συνολικό χάρτη και αν ανήκει σε μια χρωματική ομάδα που μπορεί να αξιοποιηθεί για μεγαλύτερα κέρδη.

Αν ο παίκτης αποφασίσει να μην αγοράσει την περιοχή, αυτή περνάει σε διαδικασία πλειστηριασμού. Αυτό δίνει την ευκαιρία στους υπόλοιπους παίκτες να την αποκτήσουν, δημιουργώντας ένα επιπλέον στοιχείο ανταγωνισμού και στρατηγικής. Η απόφαση να παραχωρηθεί μια περιοχή σε πλειστηριασμό μπορεί να βασίζεται στη στρατηγική του παίκτη να εξοικονομήσει χρήματα ή να αυξήσει τον ανταγωνισμό μεταξύ των άλλων παικτών.

Oriental Value

Cost: 100

Rent: 6

Εικόνα 5: interface Αγοράς περιοχής

2.2.2.1 Πλειστηριασμός

Όταν ένας παίκτης αποφασίσει να μην αγοράσει μια διαθέσιμη περιοχή στο παιχνίδι, αυτή περνάει αυτόματα σε διαδικασία πλειστηριασμού, προσφέροντας μια δυναμική ευκαιρία σε όλους τους παίκτες να την αποκτήσουν. Ο πλειστηριασμός είναι μια συναρπαστική διαδικασία, που όχι μόνο προσθέτει στρατηγική ένταση αλλά και διατηρεί τον ανταγωνισμό μεταξύ των παικτών.

Κατά τη διάρκεια του πλειστηριασμού, όλοι οι παίκτες έχουν τη δυνατότητα να προσφέρουν ένα χρηματικό ποσό που είναι μεγαλύτερο από την τρέχουσα υψηλότερη προσφορά. Αυτή η διαδικασία συνεχίζεται μέχρι να φτάσει η στιγμή που όλοι οι παίκτες επιλέξουν να πατήσουν "Pass", δηλώνοντας ότι δεν επιθυμούν να αυξήσουν την προσφορά τους. Ο τελευταίος παίκτης που πρόσφερε το μεγαλύτερο ποσό είναι αυτός που αποκτά την περιοχή, πληρώνοντας το ποσό που έχει δηλώσει.

Στην κάρτα που εμφανίζεται για τον πλειστηριασμό παρέχονται κρίσιμες πληροφορίες που βοηθούν τους παίκτες να λάβουν αποφάσεις. Αυτές οι πληροφορίες περιλαμβάνουν:

- Το όνομα και το χρώμα της περιοχής που δημοπρατείται.
- Το αρχικό κόστος της περιοχής, που λειτουργεί ως σημείο εκκίνησης για τις προσφορές.
- Το όνομα του παίκτη που έχει σειρά να υποβάλει την επόμενη προσφορά.
- Το όνομα του τελευταίου παίκτη που πρόσφερε το υψηλότερο ποσό, καθώς και το ίδιο το ποσό της προσφοράς.

Αυτή η λειτουργικότητα καθιστά τον πλειστηριασμό διαφανή και δίκαιο για όλους τους συμμετέχοντες, καθώς όλοι οι παίκτες έχουν πλήρη εικόνα της κατάστασης. Η δυνατότητα του κάθε παίκτη να συμμετάσχει ή να αποσυρθεί από τον πλειστηριασμό παρέχει ευελιξία και δημιουργεί ενδιαφέρουσες στρατηγικές επιλογές.

Για παράδειγμα, ένας παίκτης μπορεί να επιλέξει να αυξήσει την προσφορά του για να αποκτήσει μια περιοχή που συμπληρώνει μια χρωματική ομάδα, ή μπορεί να αποσυρθεί από τον πλειστηριασμό για να διαφυλάξει τη ρευστότητά του. Επίσης, το στοιχείο του ανταγωνισμού ενισχύεται όταν δύο ή περισσότεροι παίκτες διεκδικούν σθεναρά την ίδια περιοχή, ανεβάζοντας την τελική τιμή της σε υψηλότερα επίπεδα.

Η διαδικασία του πλειστηριασμού όχι μόνο συμβάλλει στην ισορροπία του παιχνιδιού, αλλά και

προάγει τη στρατηγική σκέψη και την αλληλεπίδραση μεταξύ των παικτών, καθιστώντας το παιχνίδι πιο ζωντανό και διασκεδαστικό.

Reading Railroad

Original Cost: 200

Currently Bidding: Gromp

Last Bidder: BaldMar

Current Bid: 50

Εικόνα 6: Interface πλειστηριασμού

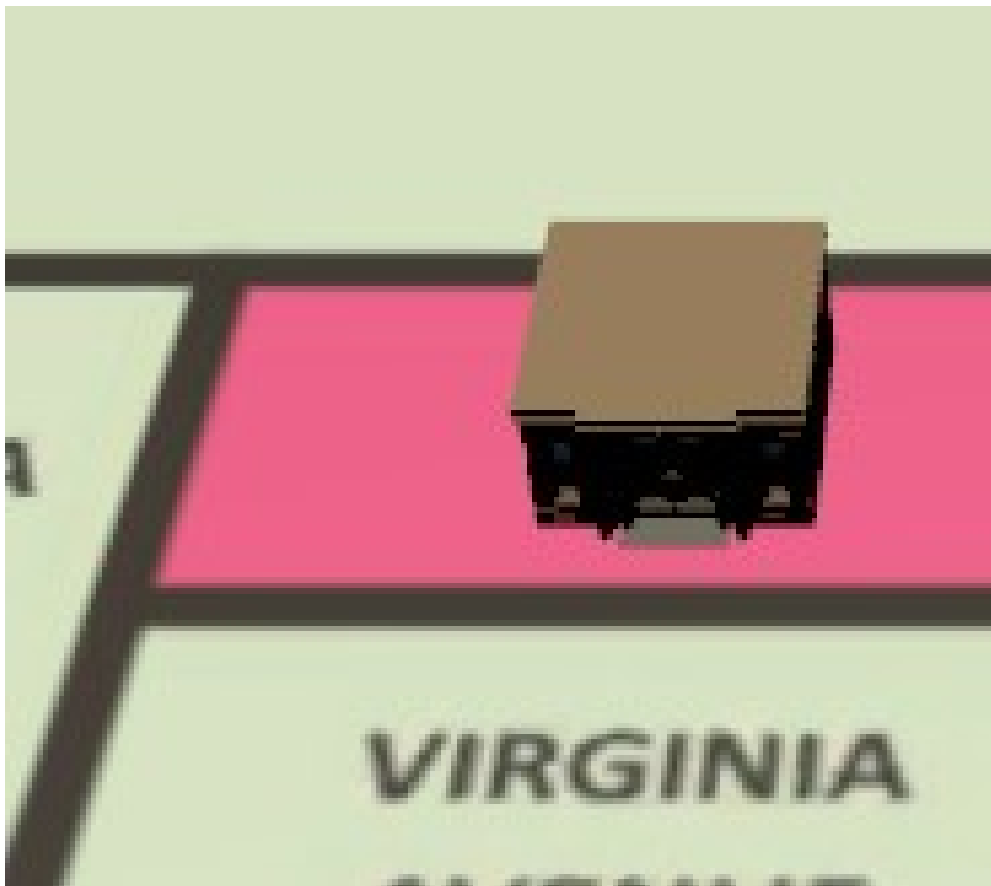
2.2.3 Αναβάθμιση περιοχών

Στο παιχνίδι, οι περιοχές με το ίδιο χρώμα αποτελούν ομάδες που μπορούν να αξιοποιηθούν στρατηγικά για τη μεγιστοποίηση των κερδών ενός παίκτη. Όταν ένας παίκτης καταφέρει να αποκτήσει όλες τις περιοχές μιας συγκεκριμένης χρωματικής ομάδας, αποκτά το δικαίωμα να ξεκινήσει τη διαδικασία αναβάθμισης αυτών των περιοχών. Αυτή η δυνατότητα προσθέτει βάθος στη στρατηγική του παιχνιδιού και επιτρέπει στους παίκτες να αυξήσουν τα έσοδά τους από ενοίκια και να ενισχύσουν τη συνολική τους αξία.

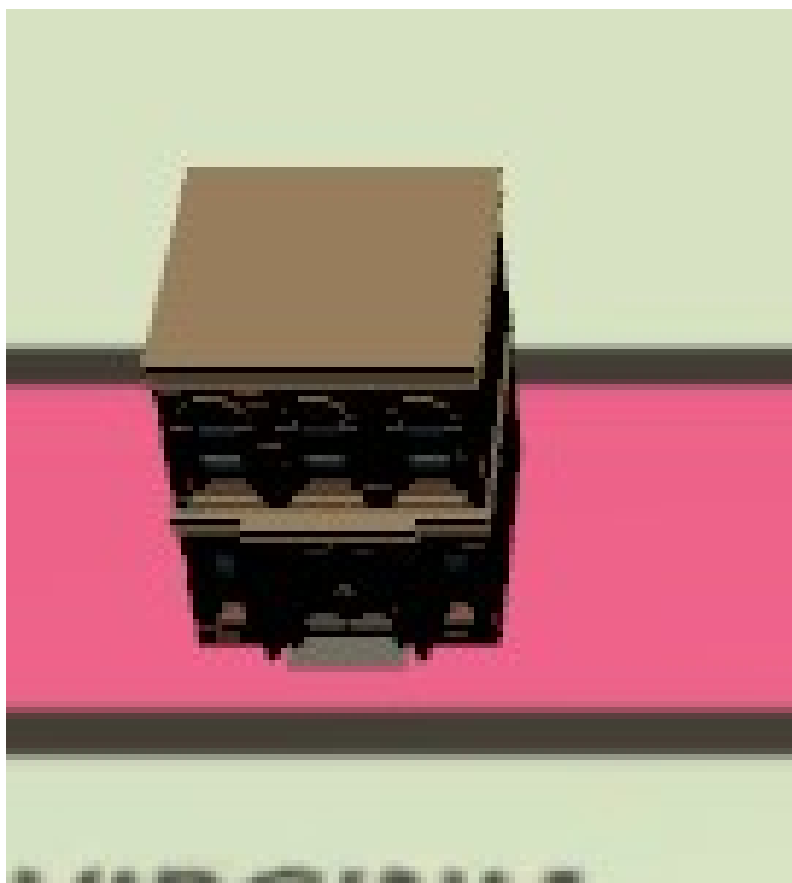
Κάθε αναβάθμιση σε μια περιοχή αυξάνει σημαντικά το ενοίκιο που απαιτείται από άλλους παίκτες όταν περνούν από εκεί, καθιστώντας την ιδιοκτησία ακόμη πιο επικερδή. Παράλληλα, η αναβάθμιση συμβάλλει στην αύξηση του **net worth** του παίκτη, δηλαδή της συνολικής του περιουσίας, η οποία αποτελεί κρίσιμο στοιχείο για τη νίκη στο παιχνίδι. Καθώς τα επίπεδα αναβάθμισης αυξάνονται, το οικονομικό όφελος γίνεται όλο και πιο αισθητό, αλλά και το κόστος για τις αναβαθμίσεις γίνεται υψηλότερο, προσθέτοντας μια στρατηγική διάσταση στην απόφαση του παίκτη για το πότε και πώς να επενδύσει.

Ένα ακόμη χαρακτηριστικό που κάνει την αναβάθμιση των περιοχών πιο ενδιαφέρουσα είναι η αλλαγή της εμφάνισης της κάθε περιοχής με την προσθήκη ενός ξεχωριστού κτιρίου για κάθε επίπεδο αναβάθμισης. Αυτές οι οπτικές αλλαγές προσφέρουν μια άμεση και ευχάριστη ανατροφοδότηση στον παίκτη, υπογραμμίζοντας την πρόοδο και τις στρατηγικές του επιλογές. Τα διαφορετικά κτίρια όχι μόνο προσδίδουν ζωντάνια στον χάρτη αλλά λειτουργούν και ως ένδειξη για τους άλλους παίκτες σχετικά με το πόσο ισχυρή και επικερδής είναι μια συγκεκριμένη περιοχή.

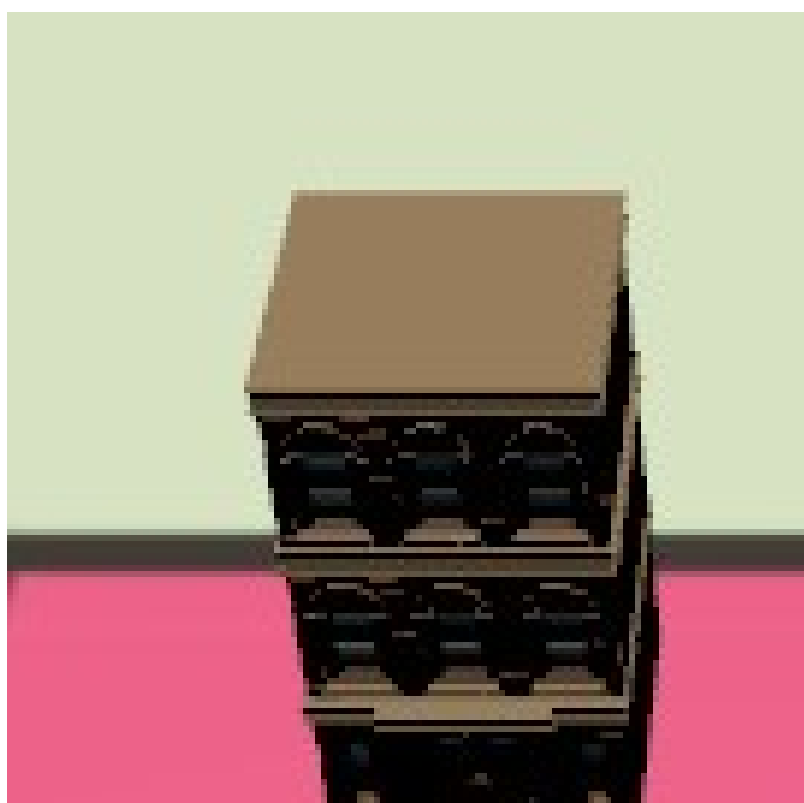
Η διαδικασία αναβάθμισης ενισχύει τη στρατηγική φύση του παιχνιδιού, δίνοντας κίνητρο στους παίκτες να αποκτήσουν ολόκληρες ομάδες περιοχών και να τις αξιοποιήσουν πλήρως. Παράλληλα, εισάγει ένα στοιχείο ρίσκου, καθώς οι επενδύσεις σε αναβαθμίσεις μπορεί να περιορίσουν προσωρινά τη ρευστότητα του παίκτη, αλλά υπόσχονται μακροπρόθεσμα μεγάλα κέρδη. Έτσι, η αναβάθμιση των περιοχών γίνεται ένα από τα πιο κρίσιμα στοιχεία της στρατηγικής που καθορίζει την έκβαση του παιχνιδιού.



Εικόνα 7: Επίπεδο Αναβάθμισης 1



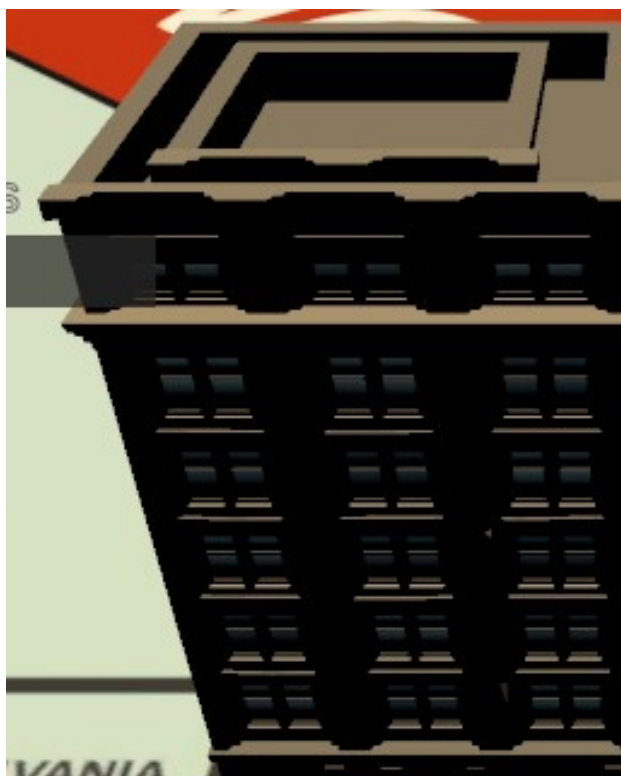
Εικόνα 8: Επίπεδο Αναβάθμισης 2



Εικόνα 9: Επίπεδο Αναβάθμισης 3



Εικόνα 10: Επίπεδο Αναβάθμισης 4



Εικόνα 11: Επίπεδο Αναβάθμισης 5

...

3. Λειτουργίες παίκτη

3.1 Σειρά του παίκτη

Όταν έρθει η σειρά του παίκτη, το οποίο ανακοινώνεται με συγκεκριμένο ήχο καθώς και μήνυμα, ο παίκτης μπορεί να εκτελέσει κάποιες ενέργειες.

3.1.1 Ρήψη ζαριού

Στο παιχνίδι, η ρίψη των ζαριών είναι μια από τις βασικές ενέργειες που καθορίζουν την πρόοδο του κάθε παίκτη στον χάρτη. Πατώντας το πλήκτρο "**R**" στο πληκτρολόγιο, ο παίκτης ρίχνει δύο ζάρια, και η συνολική τιμή που εμφανίζεται καθορίζει τον αριθμό των θέσεων που θα μετακινηθεί η φιγούρα του στον χάρτη. Αυτή η απλή αλλά κρίσιμη ενέργεια συνδέει τη στρατηγική με την τύχη, καθορίζοντας τη θέση του παίκτη και τις πιθανές του επιλογές στη συνέχεια.

Η ρίψη ζαριών παρουσιάζεται με σαφή τρόπο στο παιχνίδι. Το αποτέλεσμα εμφανίζεται στο κάτω αριστερό μέρος της οθόνης, κάτω από την ένδειξη του διαθέσιμου ποσού χρημάτων του παίκτη. Αυτό επιτρέπει στους παίκτες να έχουν πλήρη επίγνωση της τρέχουσας κατάστασής τους και της προόδου τους στον γύρο.

Ένα ιδιαίτερο χαρακτηριστικό της ρίψης είναι η πιθανότητα εμφάνισης "διπλής" (π.χ., 3-3, 5-5), δηλαδή όταν και τα δύο ζάρια δείχνουν τον ίδιο αριθμό. Όταν συμβεί αυτό, ο παίκτης αποκτά το δικαίωμα να παίξει ξανά, προσφέροντάς του ένα σημαντικό πλεονέκτημα για την κίνηση στον χάρτη και τη λήψη στρατηγικών αποφάσεων. Η εμφάνιση της διπλής συνοδεύεται από έναν ειδικό ήχο, που ειδοποιεί τον παίκτη για το γεγονός, καθώς και από την αλλαγή του χρώματος του κειμένου των ζαριών σε χρυσό. Αυτές οι οπτικοακουστικές ενδείξεις εξασφαλίζουν ότι ο παίκτης αντιλαμβάνεται άμεσα τη σημασία της ρίψης και προετοιμάζεται για τη νέα του κίνηση. Η δυνατότητα επαναληπτικής ρίψης όταν εμφανίζεται διπλή, σε συνδυασμό με την οπτική και ηχητική ανατροφοδότηση, προσθέτει μια δυναμική διάσταση στο παιχνίδι. Οι παίκτες ενθαρρύνονται να παρακολουθούν προσεκτικά τις κινήσεις τους, διατηρώντας το ενδιαφέρον και την ένταση σε κάθε γύρο. Η λειτουργία αυτή όχι μόνο ενισχύει τη στρατηγική του παιχνιδιού αλλά και το στοιχείο της διασκέδασης, κάνοντας κάθε ρίψη μοναδική και γεμάτη προσδοκία.

3.1.2 Αναβάθμιση περιοχής

Αν ο χρήστης ελέγχει όλες τις περιοχές με ίδιο χρώμα μπορεί να τις αναβαθμίσει, αυξάνοντας έτσι το ενοίκιο και το networth του.

3.1.3 Πλειστηριασμός

Ο πλειστηριασμός στο παιχνίδι αποτελεί μια κρίσιμη διαδικασία στρατηγικής, κατά την οποία οι παίκτες μπορούν να αποκτήσουν περιοχές που είναι διαθέσιμες προς αγορά. Είναι ένας μηχανισμός που συνδυάζει την τακτική, τη διαχείριση πόρων και τον ανταγωνισμό μεταξύ των παικτών, προσφέροντας παράλληλα ένταση και δυναμική εξέλιξη στο παιχνίδι.

Όταν ξεκινά ένας πλειστηριασμός, οι παίκτες συμμετέχουν με βάση μια προκαθορισμένη σειρά. Κάθε παίκτης, όταν έρθει η σειρά του, έχει τη δυνατότητα να τοποθετήσει χρήματα για να διεκδικήσει την περιοχή. Η προσφορά αυτή πρέπει να είναι μεγαλύτερη από την προηγούμενη, αυξάνοντας έτσι το ποσό που απαιτείται για να κερδίσει κάποιος την περιοχή. Εναλλακτικά, ο παίκτης μπορεί να επιλέξει την επιλογή "Pass", αποσύροντας τον εαυτό του από τη διαδικασία του πλειστηριασμού για την τρέχουσα περιοχή. Αυτή η επιλογή, αν και περιορίζει τις πιθανότητές του να αποκτήσει την περιοχή, μπορεί να είναι στρατηγική κίνηση για την εξοικονόμηση χρημάτων για μελλοντικές ευκαιρίες.

Ο πλειστηριασμός συνεχίζεται μέχρι να απομείνει μόνο ένας παίκτης που έχει κάνει την υψηλότερη προσφορά. Ο παίκτης αυτός κερδίζει την περιοχή, ενώ όλοι οι υπόλοιποι χάνουν τη δυνατότητα διεκδίκησής της. Η επιλογή του "Pass" έχει ιδιαίτερη σημασία, καθώς ο τελευταίος παίκτης που θα πατήσει "Pass" είναι ο νικητής του πλειστηριασμού, αρκεί να έχει καταθέσει την τελευταία προσφορά.

Αυτή η διαδικασία όχι μόνο ενισχύει τον ανταγωνισμό αλλά προσθέτει και βάθος στη στρατηγική του παιχνιδιού. Οι παίκτες καλούνται να αποφασίσουν αν θα ρισκάρουν τους πόρους τους για μια περιοχή που μπορεί να είναι κρίσιμη για τη νίκη ή αν θα αποσυρθούν για να επενδύσουν αλλού. Επιπλέον, η συμμετοχή όλων των παικτών, ακόμα και ως παρατηρητές, δημιουργεί ένα διαδραστικό και δυναμικό περιβάλλον που κρατά το ενδιαφέρον σε υψηλά επίπεδα.

Ο πλειστηριασμός, επομένως, δεν είναι απλώς μια διαδικασία απόκτησης περιοχών, αλλά ένας μηχανισμός που δοκιμάζει τη στρατηγική σκέψη και την ικανότητα διαχείρισης των διαθέσιμων πόρων, κάνοντας το παιχνίδι πιο πολυσύνθετο και ενδιαφέρον.

3.2 Γενικές ενέργειες

3.2.1 Χρήση κάμερας

Στο παιχνίδι, ο έλεγχος της κάμερας είναι ένα κρίσιμο στοιχείο για την εμπειρία του χρήστη, καθώς επιτρέπει στον παίκτη να αλληλεπιδρά με το περιβάλλον και να παρακολουθεί τις εξελίξεις του παιχνιδιού. Ο παίκτης έχει τη δυνατότητα να εναλλάσσει μεταξύ δύο διαφορετικών καταστάσεων κάμερας: της **κλειδωμένης** και της **ξεκλειδωμένης** κάμερας, προσφέροντας ευελιξία και στρατηγική στον τρόπο που παρακολουθεί το παιχνίδι.

3.2.1.1 Κλειδωμένη Κάμερα

Όταν η κάμερα είναι κλειδωμένη, η κάμερα ακολουθεί συνεχώς τον παίκτη και κινείται μαζί του σε όλο το παιχνίδι. Αυτή η λειτουργία επιτρέπει στον παίκτη να παρακολουθεί τη δράση από μια σταθερή προοπτική που εστιάζει μόνο στον ίδιο τον χαρακτήρα, εξαλείφοντας την ανάγκη να ελέγχει συνεχώς τη θέση της κάμερας. Αυτός ο τρόπος χρήσης της κάμερας είναι ιδανικός όταν ο παίκτης θέλει να συγκεντρωθεί πλήρως στον χαρακτήρα του και στην άμεση αλληλεπίδραση με το περιβάλλον του.

3.2.1.2 Ξεκλειδωμένη Κάμερα

Όταν ο παίκτης θέλει να έχει μεγαλύτερη ευχέρεια στην κίνηση της κάμερας και να παρακολουθεί ολόκληρο το πεδίο παιχνιδιού ή τις κινήσεις των άλλων παικτών, μπορεί να "ξεκλειδώσει" την κάμερα. Πατώντας το πλήκτρο "C", η κάμερα μεταφέρεται αμέσως πάνω στον χαρακτήρα του παίκτη, δίνοντάς του τον πλήρη έλεγχο. Σε αυτή την κατάσταση, ο παίκτης μπορεί να χρησιμοποιήσει τα πλήκτρα "**WASD**" για να κινεί την κάμερα σε οποιαδήποτε κατεύθυνση, επιτρέποντάς του να παρακολουθεί όχι μόνο τον δικό του χαρακτήρα, αλλά και τις κινήσεις των άλλων παικτών στον χάρτη.

Η ξεκλειδωμένη κάμερα προσφέρει μεγαλύτερη στρατηγική ελευθερία, καθώς επιτρέπει στον παίκτη να παρακολουθεί τη συνολική κατάσταση του παιχνιδιού, τακτικά ή για να βλέπει πώς εξελίσσονται οι κινήσεις των αντιπάλων. Αυτή η κατάσταση είναι χρήσιμη όταν ο παίκτης θέλει να πάρει αποφάσεις για τις επόμενες κινήσεις του βάσει των ενεργειών των άλλων παικτών ή του χάρτη.

3.2.1.3 Εναλλαγή Κάμερας

Η εναλλαγή μεταξύ κλειδωμένης και ξεκλειδωμένης κάμερας γίνεται εύκολα με το πάτημα του πλήκτρου "**C**". Έτσι, ο παίκτης μπορεί γρήγορα να προσαρμόσει την οπτική του γωνία ανάλογα με τις ανάγκες του στο παιχνίδι. Όταν η κατεύθυνση της κάμερας είναι ξεκλειδωμένη, ο παίκτης έχει πλήρη έλεγχο και ευχέρεια κίνησης, αλλά και την ικανότητα να παρακολουθεί τις κινήσεις των άλλων παικτών και να αποκτά καλύτερη κατανόηση της κατάστασης του παιχνιδιού. Αυτή η δυνατότητα προσφέρει έναν συνδυασμό ελέγχου και στρατηγικής παρατήρησης, δίνοντας στον παίκτη την ευχέρεια να επιλέξει την καλύτερη οπτική γωνία ανάλογα με την κατάσταση του παιχνιδιού και τις ανάγκες του.

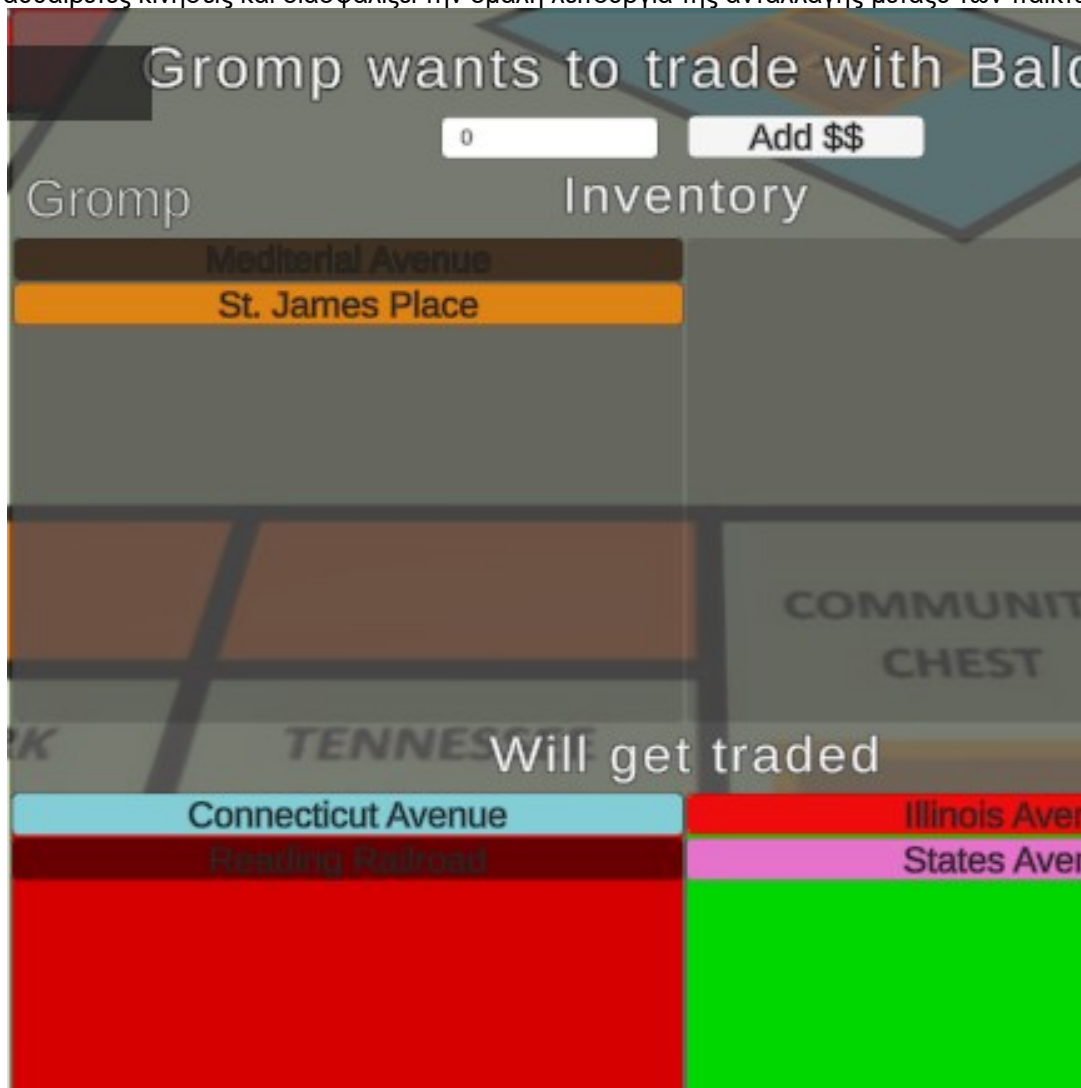
3.2.2 Ανταλλαγή

Στο παιχνίδι, οι παίκτες έχουν τη δυνατότητα να ξεκινήσουν μια διαδικασία ανταλλαγής με άλλους παίκτες μέσω του **Scoreboard**. Για να το κάνουν αυτό, ο παίκτης επιλέγει το όνομα του άλλου παίκτη στο scoreboard, το οποίο ανοίγει το αντίστοιχο **μενού ανταλλαγής**. Αυτή η διαδικασία είναι ανοιχτή για όλους τους παίκτες, επιτρέποντας την παρακολούθηση της διαδικασίας από όλους, χωρίς να έχουν άμεση συμμετοχή, εκτός αν αποφασίσουν να εμπλακούν.

Το **παράθυρο της ανταλλαγής** αποτελείται από διάφορους χώρους που οργανώνουν τις πληροφορίες της ανταλλαγής. Συγκεκριμένα, υπάρχουν δύο βασικοί χώροι: ο πρώτος δείχνει τις περιοχές του κάθε παίκτη (το "Inventory"), ενώ ο δεύτερος δείχνει τα αντικείμενα που προορίζονται να ανταλλαχθούν (το "Will get traded"). Εκτός από τις περιοχές, υπάρχει επίσης ένα πεδίο όπου οι παίκτες μπορούν να εισάγουν χρηματικά ποσά που προγραμματίζουν να ανταλλάξουν.

Η διαδικασία ανταλλαγής είναι εύκολη και διαδραστική. Όταν ο παίκτης κάνει κλικ σε μια περιοχή από το **Inventory**, αυτή μεταφέρεται στον χώρο του **Will get traded**, και αντιστρόφως. Αυτή η αλληλεπίδραση μπορεί να εφαρμοστεί και σε χρηματικά ποσά: αν ο παίκτης κάνει κλικ σε ένα ποσό στον χώρο του "Will get traded", το ποσό αφαιρείται από εκεί. Εφόσον και οι δύο παίκτες συμφωνήσουν με την ανταλλαγή, τα αντικείμενα και τα χρήματα ανταλλάσσονται άμεσα. Σε περίπτωση που οι παίκτες δεν καταλήξουν σε συμφωνία, μπορούν να ακυρώσουν τη

διαδικασία οποιαδήποτε στιγμή πατώντας το κουμπί **"Cancel"**. Αυτό διακόπτει την ανταλλαγή και δεν πραγματοποιούνται καμία αλλαγή στα δεδομένα των παικτών. Ωστόσο, υπάρχει και μια σημαντική λειτουργία που εξασφαλίζει ότι η συμφωνία είναι έγκυρη μόνο αν οι παίκτες τηρούν τις συμφωνίες τους: αν ένας παίκτης αφαιρέσει μια περιοχή από το **"Will get traded"** μετά την αποδοχή της ανταλλαγής από την άλλη πλευρά, τότε η αποδοχή ακυρώνεται και η διαδικασία πρέπει να ξαναρχίσει από την αρχή. Αυτό αποτρέπει κακόβουλες ή αυθαίρετες κινήσεις και διασφαλίζει την ομαλή λειτουργία της ανταλλαγής μεταξύ των παικτών.



Εικόνα 12: Μενού ανταλλαγής

3.2.3 Προβολή περιοχών

Στο παιχνίδι, κάθε παίκτης έχει τη δυνατότητα να αλληλεπιδράσει με τον χάρτη κάνοντας κλικ πάνω σε οποιαδήποτε περιοχή για να δει αναλυτικές πληροφορίες σχετικά με αυτήν. Αυτή η λειτουργία βοηθά τον παίκτη να παρακολουθεί τη δυναμική του παιχνιδιού και να λαμβάνει στρατηγικές αποφάσεις, είτε πρόκειται για την αγορά, την ανάπτυξη ή τη διαχείριση των περιοχών του.

Για τις **διεκδικήσιμες περιοχές**, οι πληροφορίες που εμφανίζονται περιλαμβάνουν το όνομα του **ιδιοκτήτη** της περιοχής, δηλαδή ποιος παίκτης την κατέχει. Αν η περιοχή ανήκει στον ίδιο τον παίκτη που κάνει κλικ, μπορεί να αποφασίσει αν θέλει να αναβαθμίσει την περιοχή ή να την κρατήσει όπως έχει. Επίσης, εμφανίζεται το **κόστος αγοράς** της περιοχής, δηλαδή το χρηματικό ποσό που απαιτείται για να την αγοράσει κάποιος άλλος παίκτης εφόσον δεν έχει ήδη ιδιοκτήτη. Το **επίπεδο** της περιοχής δείχνει την κατάσταση της ανάπτυξης της (όπως πόσα σπίτια ή ξενοδοχεία έχουν τοποθετηθεί εκεί), γεγονός που επηρεάζει την αξία του ενοικίου που απαιτείται από άλλους παίκτες που πέφτουν σε αυτή την περιοχή. Τέλος, το **ενοίκιο** εμφανίζεται

και δείχνει πόσο θα πρέπει να πληρώσει ένας παίκτης στον ιδιοκτήτη εάν πέσει σε αυτήν την περιοχή, κάτι που μπορεί να έχει σημαντικό αντίκτυπο στην οικονομική στρατηγική του παιχνιδιού.

Επιπλέον, αν η περιοχή είναι μία **ειδική περιοχή** (όπως η περιοχή "Go to Jail" ή "Free Parking"), τότε οι πληροφορίες που εμφανίζονται αφορούν τις λειτουργίες και τις ειδικές δυνατότητες αυτών των περιοχών. Για παράδειγμα, η περιοχή "Go to Jail" θα εμφανίσει την πληροφορία ότι ο παίκτης θα πρέπει να μεταφερθεί στην φυλακή, ενώ η περιοχή "Free Parking" θα δείξει το ποσό που έχει συγκεντρωθεί εκεί για τον επόμενο παίκτη που θα περάσει από αυτήν. Αυτή η δυνατότητα επιτρέπει στους παίκτες να έχουν πλήρη εικόνα του χάρτη και των επιλογών τους σε κάθε στιγμή του παιχνιδιού, διευκολύνοντας την λήψη στρατηγικών αποφάσεων.



Εικόνα 13: Προβολή χρωματιστής περιοχής χωρίς Ιδιοκτήτη

Eletric Company

Owner: playerA

Cost: 150

Level: 0

Rent increases if both uti
companies are owned by

Εικόνα 14: Προβολή περιοχής υπηρεσίας

3.3 Πλήκτρα παίκτη

3.3.1 Κάμερα

Κλείδωμα/Ξεκλείδωμα κάμερας – «C»

3.3.2 Ζάρι

Ρήψη ζαριού – «R»

4. Χάρτης

Ο χάρτης του παιχνιδιού αποτελείται από συνολικά σαράντα (40) περιοχές, οι οποίες είναι χωρισμένες σε δύο βασικές κατηγορίες: διεκδικήσιμες περιοχές και ειδικές περιοχές.

4.1 Διεκδικήσασες περιοχές

4.1.1 Χρωματιστές περιοχές

Στο παιχνίδι, 22 από τις περιοχές του ταμπλό είναι χρωματιστές και χωρίζονται σε ομάδες των 2-3 περιοχών, με κάθε ομάδα να αντιπροσωπεύει μια σειρά περιοχών που έχουν παρόμοια χρώματα. Αυτές οι περιοχές έχουν έναν ιδιαίτερο ρόλο στο παιχνίδι, καθώς οι παίκτες μπορούν να τις αγοράσουν, να τις αναπτύξουν και να κερδίσουν έσοδα από τους άλλους παίκτες που πέφτουν πάνω τους.

Η τιμή των περιοχών αυτών αυξάνεται καθώς προχωράμε προς τα μέσα του ταμπλό. Οι περιοχές που βρίσκονται πιο κοντά στην αφετηρία είναι συνήθως φθηνότερες, ενώ οι περιοχές που βρίσκονται πιο κοντά στο τέλος του ταμπλό έχουν υψηλότερες τιμές. Αυτή η διαφορά στην τιμή των περιοχών είναι συνήθως συνδεδεμένη με τη στρατηγική του παιχνιδιού, αφού οι πιο ακριβές περιοχές επιτρέπουν μεγαλύτερα κέρδη όταν αναπτύσσονται (με την προσθήκη σπιτιών ή ξενοδοχείων), αλλά είναι επίσης πιο επικίνδυνες για τους παίκτες που προσπαθούν να τις αγοράσουν.

Αν οι παίκτες καταφέρουν να αποκτήσουν όλες τις περιοχές μιας ομάδας χρωμάτων, αποκτούν το πλεονέκτημα να χρεώνουν υψηλότερα ενοίκια, αυξάνοντας σημαντικά την πίεση στους αντιπάλους τους. Για παράδειγμα, αν κατέχουν όλες τις περιοχές της κόκκινης ή μπλε ομάδας, μπορούν να αρχίσουν να κατασκευάζουν σπίτια και ξενοδοχεία στις περιοχές αυτές, κάνοντάς τις ακόμη πιο ακριβές για τους άλλους παίκτες να προσγειωθούν πάνω τους.

Η ισχυρή στρατηγική περιλαμβάνει την αγορά περιοχών κοντά στην αφετηρία για να εξασφαλίσετε σταθερό εισόδημα από τα πρώτα στάδια του παιχνιδιού, αλλά και την επιδίωξη των πιο ακριβών περιοχών για τη δημιουργία ενός ισχυρού χαρτοφυλακίου, το οποίο θα εξασφαλίσει κέρδη καθώς το παιχνίδι προχωρά και οι παίκτες θα είναι πιο κοντά στο τέλος του ταμπλό.

4.1.2 Σταθμοί

Στο παιχνίδι υπάρχουν 4 σταθμοί, οι οποίοι είναι τοποθετημένοι στη μέση της κάθε πλευράς του ταμπλό. Οι σταθμοί έχουν μια ιδιαίτερη ιδιαιτερότητα, καθώς δεν μπορούν να αναβαθμιστούν, όπως οι άλλες περιοχές, με σπίτια ή ξενοδοχεία. Ωστόσο, η αξία τους αυξάνεται ανάλογα με τον αριθμό των σταθμών που κατέχει ο παίκτης.

Για κάθε σταθμό που κατέχει ένας παίκτης, το ενοίκιο που πρέπει να πληρώσει κάποιος άλλος παίκτης όταν πέσει πάνω στον σταθμό αυτόν αυξάνεται κατά \$50. Δηλαδή, εάν ο παίκτης κατέχει μόνο έναν σταθμό, το ενοίκιο είναι το βασικό ποσό. Αν κατέχει δύο σταθμούς, το ενοίκιο αυξάνεται, και αν κατέχει και τους τέσσερις, το ενοίκιο που απαιτείται φτάνει το μέγιστο ποσό.

Για παράδειγμα, αν ο ιδιοκτήτης κατέχει τρεις από τους τέσσερις σταθμούς, το ενοίκιο για οποιονδήποτε άλλον παίκτη που πέσει σε έναν από αυτούς τους σταθμούς θα είναι \$150. Αυτό δημιουργεί μια ισχυρή στρατηγική για τους παίκτες που επιθυμούν να ελέγχουν πολλούς σταθμούς, καθώς η αύξηση του ενοικίου τους μπορεί να έχει σημαντικό οικονομικό όφελος, ιδίως καθώς το παιχνίδι προχωρά και οι παίκτες πλησιάζουν στο να περάσουν από αυτές τις περιοχές πιο συχνά.

4.1.3 Υπηρεσίες

Η περιοχές των υπηρεσιών στο παιχνίδι προσφέρει έναν ιδιαίτερο μηχανισμό που διαφοροποιεί τη στρατηγική των παικτών σε σχέση με τις υπόλοιπες περιοχές. Οι υπηρεσίες δεν έχουν σταθερό ενοίκιο, όπως οι άλλες περιοχές του παιχνιδιού, αλλά το ποσό που πρέπει να πληρώσει ο παίκτης εξαρτάται από το αποτέλεσμα μιας ζαριάς. Όταν ένας παίκτης πέσει σε μία από αυτές τις περιοχές, πρέπει να ρίξει ένα ζάρι, και το ποσό που θα πληρώσει για την υπηρεσία είναι τέσσερις φορές το αποτέλεσμα του ζαριού.

Αυτός ο μηχανισμός προσθέτει έναν παράγοντα τύχης, καθώς ο παίκτης μπορεί να πληρώσει διαφορετικά ποσά κάθε φορά που πέφτει σε αυτή την περιοχή, ανάλογα με το αποτέλεσμα της ζαριάς. Για παράδειγμα, αν ο παίκτης ρίξει 3, το ποσό που θα πληρώσει θα είναι $4 * 3 = 12\$$. Αν ρίξει 6, το ποσό θα είναι $4 * 6 = 24\$$. Αυτή η τυχαιότητα δημιουργεί αβεβαιότητα και προσθέτει έναν επιπλέον στρατηγικό παράγοντα.

Ωστόσο, αν ο ιδιοκτήτης έχει και τις δύο υπηρεσίες υπό την κατοχή του, η κατάσταση αλλάζει. Σε αυτήν την περίπτωση, ο πολλαπλασιαστής αυξάνεται και το ποσό που πρέπει να

πληρωθεί είναι δεκαπλάσιο του αποτελέσματος της ζαριάς, αντί για τέσσερις φορές. Αυτό σημαίνει ότι αν κάποιος ιδιοκτήτης κατέχει και τις δύο περιοχές υπηρεσιών, το ενοίκιο που θα πληρώσει ο παίκτης που θα πέσει στην περιοχή μπορεί να είναι πολύ υψηλότερο, αυξάνοντας την πίεση στους αντιπάλους και δίνοντας στον ιδιοκτήτη σημαντικό πλεονέκτημα.

Για παράδειγμα, αν ο ιδιοκτήτης κατέχει και τις δύο υπηρεσίες και ο παίκτης ρίξει ζάρι 4, το ποσό που θα πληρώσει θα είναι $4 * 10 = 40\$$, το οποίο είναι σημαντικά υψηλότερο από το $4 * 4 = 16\$$ που θα πλήρωνε αν ο ιδιοκτήτης κατείχε μόνο μία υπηρεσία. Αυτή η δυναμική ενθαρρύνει τους παίκτες να επιδιώκουν να αποκτήσουν και τις δύο υπηρεσίες για να μεγιστοποιήσουν το κέρδος τους από τους αντιπάλους που πέφτουν σε αυτές τις περιοχές.

Η στρατηγική της αγοράς των υπηρεσιών είναι εξαιρετικά κρίσιμη, καθώς οι υπηρεσίες μπορεί να αποτελούν πηγή σημαντικού εσόδου για τους ιδιοκτήτες τους, ειδικά αν καταφέρουν να αποκτήσουν και τις δύο περιοχές. Ωστόσο, η τύχη των ζαριών μπορεί να κάνει τη διαχείριση αυτών των περιοχών πιο αβέβαιη σε σχέση με τις περιοχές που έχουν σταθερό ενοίκιο.

4.2 Ειδικές περιοχές

4.2.1 Φόροι

Όταν ένας παίκτης πέσει σε μια περιοχή φόρου στο παιχνίδι, είναι υποχρεωμένος να πληρώσει το αναγραφόμενο ποσό φόρου, το οποίο αφαιρείται από το συνολικό του υπόλοιπο χρημάτων. Ωστόσο, υπάρχει μια ιδιαίτερη μηχανική στο παιχνίδι που ενισχύει την αλληλεπίδραση μεταξύ των περιοχών φόρου και της περιοχής «Free Parking»: το ποσό που πληρώνεται για φόρους δεν χάνεται εντελώς, αλλά αντίθετα προστίθεται στην ανταμοιβή της περιοχής «Free Parking». (4.2.3).

4.2.2 Community Chest

Ο μηχανισμός της περιοχής «Community Chest» προσθέτει μια σημαντική διάσταση στρατηγικής και τύχης στο παιχνίδι. Όταν ένας παίκτης πέσει στην περιοχή αυτή, το παιχνίδι καθορίζει, μέσω μιας τυχαίας διαδικασίας, αν ο παίκτης θα κερδίσει ή θα χάσει χρήματα. Αυτή η τυχαιότητα ενισχύει την αίσθηση της απρόβλεπτης φύσης του παιχνιδιού και εισάγει την έννοια της αβεβαιότητας στον χειρισμό των χρημάτων και της στρατηγικής.

Η καινοτόμος πτυχή αυτής της περιοχής είναι το γεγονός ότι, αν ο παίκτης χάσει χρήματα αντί να κερδίσει, το ποσό αυτό δεν χάνεται οριστικά από το παιχνίδι. Αντίθετα, προστίθεται στην ανταμοιβή της περιοχής «Free Parking». Αυτό δημιουργεί έναν αμοιβαία επωφέλη μηχανισμό όπου οι παίκτες μπορούν να επωφεληθούν από τις ζημιές που άλλοι παίκτες έχουν υποστεί, αφού το ποσό που χάθηκε προστίθεται στην περιοχή «Free Parking», προσφέροντας έτσι στους παίκτες που πέφτουν σε αυτήν τη περιοχή την ευκαιρία να κερδίσουν το συσσωρευμένο ποσό.

Η προσθήκη αυτής της δυναμικής δημιουργεί μια έντονη στρατηγική παράμετρο: οι παίκτες πρέπει να σταθμίσουν τις κινήσεις τους, καθώς η απώλεια χρημάτων σε μια περιοχή μπορεί να φέρει ένα πολύ μεγαλύτερο κέρδος στην περιοχή «Free Parking» για μελλοντικούς γύρους. Οι παίκτες που γνωρίζουν αυτήν την πιθανότητα μπορεί να αποφύγουν ή να ενθαρρύνουν συγκεκριμένες περιοχές με την ελπίδα ότι θα κερδίσουν το χρηματικό ποσό που έχει συσσωρευτεί στην περιοχή «Free Parking», αυξάνοντας την αβεβαιότητα και την στρατηγική του παιχνιδιού.

Αυτός ο μηχανισμός προσδίδει και μια αίσθηση "τύχης" καθώς, αντί να είναι απλώς ένας μηχανισμός πληρωμής ενοικίων ή πρόστιμων, οι περιοχές αυτές μπορούν να επηρεάσουν την οικονομική δυναμική του παιχνιδιού. Οι αποφάσεις που παίρνουν οι παίκτες εξαρτώνται από την εκτίμηση της πιθανότητας να χάσουν χρήματα και πώς αυτό μπορεί να λειτουργήσει υπέρ τους αν καταφέρουν να πέσουν στην περιοχή «Free Parking».(4.2.3).

4.2.3 Free Parking

Η περιοχή «Free Parking» προσφέρει μια ενδιαφέρουσα και στρατηγικά ωφέλιμη δυνατότητα στους παίκτες. Όταν ένας παίκτης πέσει σε αυτήν την περιοχή, το πρώτο πράγμα που συμβαίνει είναι ότι ο παίκτης λαμβάνει το ποσό που έχει συλλεχθεί εκείνη τη στιγμή. Αυτό το ποσό μπορεί να είναι σημαντικό, ανάλογα με το πόσο καιρό έχει περάσει από την τελευταία φορά που κάποιος είχε πέσει στην περιοχή και είχε συλλέξει χρήματα. Η περιοχή «Free Parking» λειτουργεί ως μια ειδική "θυσιασμένη" περιοχή στην οποία τα χρήματα συσσωρεύονται με την πάροδο του χρόνου, και όταν ένας παίκτης την επισκέπτεται, τα χρήματα που έχει συλλέξει αποδίδονται στον παίκτη και η περιοχή επιστρέφει σε μηδενική κατάσταση. Αυτό το μηχανισμό προσφέρει έναν ακόμα στοιχείο τύχης στο παιχνίδι, καθώς ο παίκτης που πέφτει στην «Free Parking» μπορεί να κερδίσει μια σημαντική χρηματική ανταμοιβή, ανάλογα με

το πόσο χρόνο έχει περάσει χωρίς να χρησιμοποιηθεί η περιοχή. Επιπλέον, η δυνατότητα να εμφανίζεται το ποσό που έχει συλλεχθεί μέχρι εκείνη τη στιγμή προσφέρει διαφάνεια και επιτρέπει στους παίκτες να αξιολογήσουν αν είναι «αξιοποιήσιμη» η περιοχή πριν αποφασίσουν αν θα προσπαθήσουν να την επισκεφθούν. (3.2.3).

Με αυτόν τον τρόπο, η περιοχή «Free Parking» λειτουργεί ως μία πιθανή στρατηγική «ανταμοιβή», δίνοντας στους παίκτες τη δυνατότητα να επωφεληθούν από τα χρήματα που συσσωρεύονται σε αυτήν, προσφέροντας μια ευκαιρία για οικονομική ενίσχυση μέσα στο παιχνίδι.

4.2.4 Chance

Η περιοχή «Chance» στο παιχνίδι προσφέρει έναν απρόβλεπτο μηχανισμό που επηρεάζει σημαντικά τη στρατηγική του παίκτη. Όταν ένας παίκτης πέσει σε αυτήν την περιοχή, μεταφέρεται τυχαία σε μια άλλη περιοχή του ταμπλό. Η περιοχή στην οποία θα καταλήξει μπορεί να είναι οποιαδήποτε περιοχή του παιχνιδιού, είτε είναι άδεια και διαθέσιμη προς αγορά, είτε ανήκει ήδη σε άλλον παίκτη. Αυτό προσθέτει έναν παράγοντα τύχης που μπορεί να μεταβάλει δραστικά την πορεία του παιχνιδιού, καθώς ο παίκτης δεν έχει έλεγχο πάνω στην τελική του θέση.

Η περιοχή στην οποία θα καταλήξει ο παίκτης μετά τη μεταφορά από την περιοχή «Chance» λειτουργεί κανονικά όπως οποιαδήποτε άλλη περιοχή του παιχνιδιού. Εάν η περιοχή είναι διαθέσιμη προς αγορά, ο παίκτης έχει την ευκαιρία να την αποκτήσει. Αν η περιοχή ανήκει σε άλλον παίκτη, τότε ο παίκτης θα πρέπει να πληρώσει το αντίστοιχο ενοίκιο στον ιδιοκτήτη της περιοχής. Το ενοίκιο αυτό μπορεί να διαφέρει ανάλογα με την περιοχή και την κατάσταση της (π.χ., αν υπάρχουν σπίτια ή ξενοδοχεία στην περιοχή). Παράλληλα, αν ο παίκτης περάσει την αφετηρία κατά τη διάρκεια αυτής της μεταφοράς, λαμβάνει το ποσό των 200\$, όπως και σε κάθε άλλη περίπτωση που περνά από την αφετηρία.

Η περιοχή «Chance» εισάγει μια ενδιαφέρουσα στρατηγική διάσταση, καθώς ο παίκτης ποτέ δεν ξέρει ποια περιοχή θα καταλήξει και πώς αυτό θα επηρεάσει την οικονομική του κατάσταση. Η ανατροπή της τύχης μπορεί να αποβεί υπέρ του, αν η περιοχή είναι άδεια ή αν ο παίκτης αποκτήσει μια ισχυρή περιοχή, ή να του προκαλέσει οικονομικές δυσκολίες αν καταλήξει σε μια περιοχή με υψηλό ενοίκιο. Επομένως, η περιοχή «Chance» ενισχύει την αίσθηση της αβεβαιότητας στο παιχνίδι και δίνει τη δυνατότητα για ανατροπές, δημιουργώντας νέες ευκαιρίες ή προκλήσεις για τους παίκτες, προσθέτοντας βάθος στην εμπειρία του παιχνιδιού.

4.2.5 Go to jail

Η περιοχή "Go to Jail" έχει μια ιδιαίτερη λειτουργία: όταν ο παίκτης βρεθεί σε αυτήν, αποστέλλεται άμεσα στη φυλακή χωρίς να περάσει από την αφετηρία (Start). Αυτό σημαίνει ότι δεν λαμβάνει τα 200\$ που συνήθως κερδίζει όταν περνάει από την αφετηρία. Με την άμεση φυλάκιση, ο παίκτης χάνει την ευκαιρία να προχωρήσει κανονικά στον χάρτη και πρέπει να περιμένει μέχρι να μπορέσει να βγει από τη φυλακή, είτε ρίχνοντας διπλή στα ζάρια, είτε αν το άθροισμα των ζαριών του είναι μεγαλύτερο από 7. Αν δεν το καταφέρει, πρέπει να περιμένει μέχρι την επόμενη σειρά του για να προσπαθήσει ξανά.

4.2.6 Jail

Ο παίκτης θεωρείται φυλακισμένος μόνο όταν πέσει στο πεδίο "Go to Jail". Κατά τη διάρκεια που είναι φυλακισμένος, δεν μπορεί να κινηθεί εκτός αν ρίξει διπλή (δύο ζάρια με την ίδια τιμή) ή αν το συνολικό άθροισμα των ζαριών του είναι μεγαλύτερο από 7. Επίσης, ενώ βρίσκεται στη φυλακή, ο παίκτης δεν λαμβάνει τοκους(2.2.1.3), καθώς η φυλάκιση τον εμποδίζει να αλληλεπιδράσει με τον υπόλοιπο κόσμο του παιχνιδιού. Αυτή η μηχανική προσθέτει μια στρατηγική διάσταση, καθώς ο παίκτης πρέπει να προσπαθήσει να βγει από τη φυλακή όσο το δυνατόν πιο γρήγορα για να συνεχίσει να συμμετέχει πλήρως στο παιχνίδι.

4.2.7 Αφετηρία/Go

Όταν ο παίκτης περάσει πάνω από αυτήν την περιοχή, λαμβάνει ένα χρηματικό ποσό που είναι αναλογικό με τη διάρκεια του παιχνιδιού, ξεκινώντας από 200\$. Το ποσό αυτό αυξάνεται καθώς η διάρκεια του παιχνιδιού προχωρά, προσφέροντας στον παίκτη κίνητρο να παραμείνει ενεργός και να συνεχίσει να εξερευνά και να αλληλεπιδρά με τον κόσμο του παιχνιδιού. Αυτός ο μηχανισμός ενισχύει τη δυναμική του παιχνιδιού, καθιστώντας τον παίκτη πιο στρατηγικό και επιβραβεύοντας τον για την πρόοδο του.

5. Συγχρονισμός δεδομένων

5.1 Δικτύωση/ Netcode for Game Objects

Το **Netcode for Game Objects (NGO)** είναι μια από τις κεντρικές βιβλιοθήκες δικτύωσης που παρέχει η Unity για την ανάπτυξη multiplayer παιχνιδιών, προσφέροντας ένα σύνολο εργαλείων για τον συγχρονισμό των δεδομένων μεταξύ των παικτών. Σκοπός του NGO είναι να διευκολύνει την επικοινωνία μεταξύ του server και των clients, διασφαλίζοντας τη συνέπεια και την ακρίβεια των δεδομένων σε όλη την εμπειρία του παιχνιδιού.

Το NGO παρέχει εργαλεία όπως **Network Variables**, οι οποίες επιτρέπουν τον συγχρονισμό κοινών δεδομένων μεταξύ όλων των παικτών, εξασφαλίζοντας ότι οποιαδήποτε αλλαγή σε μία μεταβλητή συγχρονίζεται αυτόματα σε όλους τους χρήστες. Επίσης, τα **Remote Procedure Calls (RPCs)** επιτρέπουν στον server να εκτελεί ενέργειες στους clients και το αντίστροφο, διασφαλίζοντας ότι οι ενέργειες εκτελούνται σωστά σε όλα τα σημεία του δικτύου. Το NGO επιτρέπει επίσης τον συγχρονισμό της θέσης, της στροφής και της κλίμακας των αντικειμένων μεταξύ των παικτών, καθώς και των **animations**, για ομοιόμορφες κινήσεις σε όλα τα μηχανήματα. Το σύστημα βασίζεται σε αρχιτεκτονική client-server, όπου ο server είναι υπεύθυνος για τον έλεγχο και την ενημέρωση των δεδομένων του παιχνιδιού, ενώ οι clients στέλνουν αιτήματα και λαμβάνουν ενημερώσεις. Μέσω αυτών των εργαλείων, το NGO παρέχει μια ισχυρή, αλλά απλοποιημένη υποδομή για παιχνίδια πολλών παικτών, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στο σχεδιασμό του παιχνιδιού, ενώ η βιβλιοθήκη αναλαμβάνει τη διαχείριση της δικτυακής επικοινωνίας και συγχρονισμού.

5.1.1 Network Object

Το **NetworkObject** είναι μια βασική κλάση που χρησιμοποιείται για τη δημιουργία και διαχείριση δικτυακά συγχρονισμένων αντικειμένων σε ένα multiplayer παιχνίδι. Εξοπλίζει τα Unity GameObjects με δυνατότητα μοναδικής δικτυακής αναγνώρισης μέσω ενός μοναδικού αναγνωριστικού (**NetworkObjectId**), το οποίο χρησιμοποιείται για την επικοινωνία και τον συγχρονισμό μεταξύ διακομιστή (server) και πελατών (clients). Το **NetworkObject** επιτρέπει τη διαχείριση ιδιοκτησίας (ownership), καθορίζοντας ποιος πελάτης έχει τον έλεγχο του αντικειμένου, επιτρέποντας στον ιδιοκτήτη να εκτελεί ενέργειες, όπως ο συγχρονισμός θέσης ή η αλληλεπίδραση με άλλους παίκτες.

Επιπλέον, το **NetworkObject** διευκολύνει τη διαδικασία δημιουργίας (spawn) και καταστροφής (despawn) αντικειμένων στο δίκτυο. Όταν ένα αντικείμενο δημιουργείται ή καταστρέφεται στον διακομιστή, αυτές οι αλλαγές συγχρονίζονται αυτόματα με όλους τους συνδεδεμένους πελάτες. Παράλληλα, συνεργάζεται στενά με το **NetworkBehaviour** για την αποστολή και λήψη δικτυακών μηνυμάτων και τη χρήση μεταβλητών που συγχρονίζονται σε πραγματικό χρόνο, όπως οι **NetworkVariables**. Για παράδειγμα, σε ένα multiplayer παιχνίδι, το **NetworkObject** μπορεί να χρησιμοποιηθεί για τη διαχείριση χαρακτήρων, όπου οι ενέργειες ενός παίκτη ενημερώνονται αυτόματα και εμφανίζονται στους υπόλοιπους.

Το **NetworkObject** προσφέρει απλότητα στη διαχείριση δικτυακών αντικειμένων, αυτοματοποιώντας διαδικασίες όπως ο συγχρονισμός και η διανομή δεδομένων. Συνδυάζεται εύκολα με άλλα εργαλεία, όπως τα **NetworkTransform** για συγχρονισμό θέσης, και προσφέρει ευελιξία και επεκτασιμότητα σε περιβάλλοντα multiplayer. Είναι ο θεμέλιος λίθος για τη δημιουργία δικτυακών εμπειριών στο Unity, παρέχοντας έναν ολοκληρωμένο μηχανισμό για τη συγχρονισμένη αλληλεπίδραση αντικειμένων.

Το script το χρησιμοποιούν τόσο οι παίκτες αλλα και διαφορα interfaces που χρειάζεται να εχουν δεδομενα τους συγχρονισμενα μεταξύ τους.

5.1.2 Rpc

Τα **RPCs (Remote Procedure Calls)** αποτελούν το βασικό εργαλείο για τον συγχρονισμό των δεδομένων σε περιβάλλοντα multiplayer παιχνιδιών, εξασφαλίζοντας την ομαλή επικοινωνία και την ακρίβεια της πληροφορίας μεταξύ του διακομιστή (server) και των πελατών (clients). Μέσω των RPCs, ο server έχει τη δυνατότητα να στέλνει δεδομένα στους πελάτες ή να εκτελεί συγκεκριμένες ενέργειες στους clients, διασφαλίζοντας ότι όλα τα δεδομένα είναι συγχρονισμένα και ότι οι αλληλεπιδράσεις των χρηστών είναι ομοιόμορφες σε όλο το παιχνίδι.

Η χρήση RPCs επιτρέπει στον server να στέλνει κρίσιμες πληροφορίες ή εντολές στους clients, και αυτές οι εντολές εκτελούνται ταυτόχρονα και με συνέπεια σε όλους τους πελάτες, εξασφαλίζοντας ότι όλοι οι παίκτες βλέπουν και αλληλεπιδρούν με το ίδιο σύνολο δεδομένων.

Αυτό είναι ιδιαίτερα σημαντικό σε περιβάλλοντα multiplayer, όπου ο συγχρονισμός δεδομένων μεταξύ των χρηστών είναι απαραίτητος για την αποφυγή ασυμβατότητας και λαθών που μπορεί να προκύψουν από την ανεξάρτητη τροποποίηση δεδομένων από τους clients.

Επιπλέον, τα RPCs είναι κρίσιμα για την εφαρμογή αλληλεπιδράσεων που απαιτούν επικοινωνία μεταξύ του server και των clients, όπως η προβολή μενού, η ενημέρωση του interface ή η συγχρονισμένη εμφάνιση των animations. Ο server, ως κεντρικός διαχειριστής, μπορεί να ελέγχει τις αλληλεπιδράσεις και να σιγουρεύεται ότι όλες οι πληροφορίες εμφανίζονται σωστά στους χρήστες, χωρίς να επιτρέπει στους clients να αλλάξουν κρίσιμα δεδομένα που θα μπορούσαν να προκαλέσουν προβλήματα στην ακεραιότητα του παιχνιδιού.

Συνολικά, τα RPCs είναι το θεμέλιο της επικοινωνίας σε πραγματικό χρόνο σε multiplayer παιχνίδια, επιτρέποντας στον server να διατηρεί τη διαχείριση του παιχνιδιού και στους clients να αλληλεπιδρούν με τον κόσμο του παιχνιδιού με συνέπεια και ακεραιότητα.

5.1.2.1 ServerRpc

Το **ServerRpc** είναι μια βασική λειτουργία στο πλαίσιο των παιχνιδιών πολλών παικτών, επιτρέποντας στους clients να επικοινωνούν με τον server για να ζητήσουν άδεια για να εκτελέσουν συγκεκριμένες ενέργειες. Όταν ένας client καλεί μια **ServerRpc** συνάρτηση, ο server αναλαμβάνει να εκτελέσει τον αντίστοιχο κώδικα, να κάνει τις απαραίτητες αλλαγές σε δεδομένα, να ενημερώσει τις τιμές που χρειάζονται και στη συνέχεια να επιστρέψει το αποτέλεσμα στους clients. Αυτή η διαδικασία εξασφαλίζει ότι όλες οι σημαντικές ενέργειες και αλλαγές που επηρεάζουν την κατάσταση του παιχνιδιού εκτελούνται κεντρικά στον server, διασφαλίζοντας έτσι τον έλεγχο και την ασφάλεια.

Ένα από τα σημαντικά πλεονεκτήματα του **ServerRpc** είναι ότι διατηρεί όλα τα δεδομένα και τις κρίσιμες λογικές του παιχνιδιού στον server, εμποδίζοντας τους κακόβουλους χρήστες να προσπαθήσουν να παραβιάσουν ή να αλλοιώσουν τις τιμές του παιχνιδιού από τη δική τους πλευρά. Χωρίς το **ServerRpc**, οι clients θα μπορούσαν να έχουν άμεση πρόσβαση σε σημαντικά δεδομένα και να τα τροποποιούν χωρίς έλεγχο, κάτι που θα οδηγούσε σε προβλήματα με την ακεραιότητα του παιχνιδιού, όπως η παραποίηση των σκορ, των χρημάτων, ή άλλων παραμέτρων του παιχνιδιού. Με την χρήση του **ServerRpc**, η επικοινωνία και η επεξεργασία των δεδομένων γίνεται μόνο μέσω του server, εξασφαλίζοντας ότι οι αλλαγές ελέγχονται και εγκρίνονται από το κεντρικό σύστημα.

Από την άλλη πλευρά, οι clients γνωρίζουν μόνο "εικονικές" πληροφορίες, δηλαδή δεδομένα που δεν επηρεάζουν την πραγματική κατάσταση του παιχνιδιού στον server. Αυτά τα δεδομένα είναι σχεδιασμένα για να βοηθήσουν στην προβολή του interface και στην αλληλεπίδραση με τους παίκτες, χωρίς να έχουν πραγματική αξία για τον server. Για παράδειγμα, το όνομα ενός παίκτη ή η αμφιέσή του μπορεί να είναι πληροφορίες που είναι χρήσιμες για τον client και για τη σωστή εμφάνιση του παιχνιδιού, αλλά δεν έχουν καμία σημασία ή επίπτωση στην πραγματική κατάσταση του παιχνιδιού στον server.

Το **ServerRpc** αποδεικνύεται ιδιαίτερα χρήσιμο για την αποτροπή της παραποίησης δεδομένων, ενώ ταυτόχρονα διασφαλίζει ότι ο server παραμένει ο μόνος υπεύθυνος για τη διαχείριση κρίσιμων στοιχείων του παιχνιδιού, όπως οι κινήσεις των παικτών, η αλλαγή των περιοχών ή των αντικειμένων και οι αλληλεπιδράσεις με άλλους παίκτες. Αυτό όχι μόνο ενισχύει την ασφάλεια του παιχνιδιού, αλλά εξασφαλίζει επίσης ότι η εμπειρία του χρήστη είναι πιο συνεπής και αξιόπιστη, χωρίς απρόβλεπτες ή ανώμαλες καταστάσεις που θα μπορούσαν να προκύψουν αν οι clients είχαν έλεγχο πάνω σε κρίσιμα δεδομένα.

Με τη χρήση **ServerRpc**, ο server έχει τον πλήρη έλεγχο των κρίσιμων μηχανισμών του παιχνιδιού, αποτρέποντας τις παρεμβάσεις που θα μπορούσαν να προκύψουν από κακόβουλες ενέργειες των παικτών. Οι πελάτες παραμένουν υπεύθυνοι για την εμφάνιση του παιχνιδιού και την αλληλεπίδραση του χρήστη, χωρίς να επηρεάζουν τη λειτουργία ή τη δικαιοσύνη του παιχνιδιού σε επίπεδο δεδομένων και λογικής. Αυτός ο διαχωρισμός των ευθυνών μεταξύ server και client είναι ένας από τους βασικούς παράγοντες που καθιστούν το multiplayer παιχνίδι αξιόπιστο και ασφαλές.

5.1.2.2 ClientRpc

Το **ClientRpc** αποτελεί ένα ισχυρό εργαλείο στο πλαίσιο των παιχνιδιών πολλών παικτών, επιτρέποντας στον server να εκτελεί συγκεκριμένο κομμάτι κώδικα σε όλους τους clients ταυτόχρονα. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη για τη συγχρονισμένη ενημέρωση δεδομένων που σχετίζονται με interfaces, προβολή μενού και άλλες σημαντικές αλλαγές στο περιβάλλον του παιχνιδιού. Πρωταρχικός στόχος του **ClientRpc** είναι να διασφαλίσει ότι όλοι οι

παίκτες βλέπουν τις ίδιες πληροφορίες και καταστάσεις σε κάθε δεδομένη στιγμή.

Ας εξετάσουμε ένα χαρακτηριστικό παράδειγμα της χρήσης του **ClientRpc** στο παιχνίδι. Όταν ένας παίκτης εισέρχεται σε ένα μενού πλειστηριασμού ή ανταλλαγής, είναι κρίσιμο να διασφαλιστεί ότι όλες οι διεπαφές (interfaces) εμφανίζονται ταυτόχρονα και με συνέπεια για όλους τους παίκτες. Ανάλογα με το επίπεδο της περιοχής, συγκεκριμένα κτήρια ή αντικείμενα ενδέχεται να πρέπει να εμφανιστούν διαφορετικά για κάθε παίκτη. Χωρίς τη χρήση του **ClientRpc**, κάθε client θα χρειάζεται να διαχειρίζεται το αντίστοιχο μενού ή την αλλαγή στο περιβάλλον του παιχνιδιού τοπικά. Αυτό θα μπορούσε να προκαλέσει ασυμφωνίες και ασυνέπειες, όπως διαφορετικά κτήρια ή διαφορετικά μενού που βλέπουν οι χρήστες, κάτι που υποβαθμίζει την εμπειρία του multiplayer. Με τη χρήση του **ClientRpc**, ο server μπορεί να εκτελέσει αυτόματα τον αντίστοιχο κώδικα σε όλους τους clients. Για παράδειγμα, όταν ο server αποφασίσει ποιο κτήριο πρέπει να εμφανιστεί ανάλογα με το επίπεδο της περιοχής, καλεί το **ClientRpc** για να ενημερώσει όλους τους clients ταυτόχρονα. Έτσι, όλοι οι παίκτες βλέπουν το σωστό κτήριο την ίδια στιγμή, χωρίς καθυστερήσεις ή ασυμφωνίες. Με αυτόν τον τρόπο, το παιχνίδι διατηρεί μια αίσθηση συνοχής, καθώς όλοι οι παίκτες συμμετέχουν σε έναν συνεκτικό κόσμο που ανταποκρίνεται στις επιλογές και τις αποφάσεις του server.

Ένα άλλο χαρακτηριστικό παράδειγμα είναι όταν ένας παίκτης επιλέγει να αλλάξει την αμφιέσή του. Αυτή η αλλαγή επηρεάζει τον σκελετό και τις κινήσεις του χαρακτήρα. Χωρίς το **ClientRpc**, η ενημέρωση της αμφιέσης θα χρειαζόταν χειροκίνητο συγχρονισμό ή, ακόμα χειρότερα, οι άλλοι παίκτες μπορεί να βλέπουν διαφορετική εικόνα από τον επιλεγμένο χαρακτήρα. Με τη χρήση του **ClientRpc**, όταν ένας παίκτης αλλάζει την αμφιέση του χαρακτήρα του, ο server εκτελεί την εντολή μέσω του **ClientRpc**, ενημερώνοντας όλους τους clients ταυτόχρονα ώστε να δουν την ίδια αλλαγή. Αυτό εξασφαλίζει ότι όλοι οι παίκτες βλέπουν τον ίδιο χαρακτήρα με την ίδια αμφιέση, αποφεύγοντας οποιαδήποτε ασυμφωνία στο περιβάλλον του παιχνιδιού.

Ένα σημαντικό πλεονέκτημα του **ClientRpc** είναι η απλότητα και η αποδοτικότητα με την οποία συγχρονίζει δεδομένα σε περιβάλλοντα πολλών παικτών. Όταν μια μεταβλητή ή μια κατάσταση αλλάζει στον server, ο server καλεί το **ClientRpc** και αυτή η αλλαγή συγχρονίζεται ταυτόχρονα σε όλους τους clients, εξοικονομώντας χρόνο και πόρους. Αντί να πρέπει να διαχειρίζεται χειροκίνητα αλλαγές ή να εκτελεί ξεχωριστούς υπολογισμούς για κάθε client, το **ClientRpc** διασφαλίζει ότι κάθε client εφαρμόζει την αλλαγή χωρίς καθυστερήσεις.

Ωστόσο, υπάρχουν και κάποια μειονεκτήματα που πρέπει να ληφθούν υπόψη. Ένα από αυτά είναι ότι το **ClientRpc** μπορεί να προκαλέσει έντονη επιβράδυνση αν χρησιμοποιηθεί σε μεγάλο αριθμό δεδομένων ή σε πολύπλοκες διεργασίες. Αν ο server καλεί επανειλημμένα το **ClientRpc** σε κάθε πελάτη, μπορεί να προκληθεί υπερφόρτωση του δικτύου ή καθυστερήσεις στις απεικονίσεις των δεδομένων. Επομένως, απαιτείται προσεκτικός σχεδιασμός ώστε να αποφεύγεται η υπερβολική χρήση αυτού του μηχανισμού.

Σε τελική ανάλυση, το **ClientRpc** αποδεικνύεται κρίσιμο εργαλείο για τη διατήρηση μιας συνεκτικής και αξιόπιστης εμπειρίας σε multiplayer παιχνίδια. Διασφαλίζει ότι όλοι οι παίκτες έχουν την ίδια εμπειρία παιχνιδιού, επιτρέποντας στο περιβάλλον του παιχνιδιού να προσαρμόζεται με βάση τις αποφάσεις και τις αλλαγές που πραγματοποιούνται από τον server. Παρά τις πιθανές περιορισμένες περιπτώσεις υπερφόρτωσης, η χρήση του **ClientRpc** συνεχίζει να παρέχει αποτελεσματικότητα στη διαχείριση των διεπαφών, των αντικειμένων και των δεδομένων σε πολυπληθή παιχνίδια, ενισχύοντας την εμπειρία παιχνιδιού σε πολλούς χρήστες.

5.1.3 Network Variables

Τα network Variables είναι μεταβλητές που συγχρονίζονται αυτόματα μεταξύ του διακομιστή (server) και όλων των συνδεδεμένων πελατών (clients). Αποτελούν κρίσιμο εργαλείο για τον συγχρονισμό καταστάσεων στο παιχνίδι, καθώς επιτρέπουν τη διατήρηση και ενημέρωση δεδομένων σε πραγματικό χρόνο. Ένα από τα βασικά χαρακτηριστικά των Network Variables είναι ότι υποστηρίζουν την εκτέλεση callback συναρτήσεων κάθε φορά που η τιμή τους αλλάζει, δίνοντας τη δυνατότητα να ενημερώνονται άμεσα στοιχεία του παιχνιδιού ή του interface.

Ένα κλασικό παράδειγμα είναι η διαχείριση της ζωής του κάθε παίκτη. Η ζωή (health) πρέπει να είναι γνωστή σε όλους τους πελάτες, ώστε να υπάρχει συνοχή στην εμπειρία του παιχνιδιού. Αν, για παράδειγμα, ένας παίκτης δεχθεί ζημιά, η αλλαγή στη ζωή του ενημερώνεται στον διακομιστή και αυτή η ενημέρωση διανέμεται σε όλους τους πελάτες. Έτσι, κάθε παίκτης βλέπει την τρέχουσα ζωή του άλλου, τόσο ως τιμή όσο και στο επίπεδο του interface, όπως μία μπάρα υγείας.

5.1.4 Network Variables vs ClientRpc

Οι **Network Variables** και τα **Client RPCs** είναι δύο διαφορετικά εργαλεία που χρησιμοποιούνται στο **Netcode for GameObjects (NGO)** για τον συγχρονισμό και την επικοινωνία μεταξύ διακομιστή (server) και πελατών (clients). Παρόλο που και τα δύο εξυπηρετούν τον ίδιο γενικό στόχο, δηλαδή τη διατήρηση συνοχής στο multiplayer περιβάλλον, έχουν θεμελιώδεις διαφορές στη χρήση, τα πλεονεκτήματα και τους περιορισμούς τους.

Οι **Network Variables** είναι σχεδιασμένες για αυτόματο και συνεχή συγχρονισμό δεδομένων. Αυτές οι μεταβλητές διατηρούνται στον διακομιστή, και οποιαδήποτε αλλαγή στην τιμή τους μεταδίδεται αυτόματα σε όλους τους πελάτες. Είναι ιδανικές για απλά δεδομένα που απαιτούν συχνή ενημέρωση, όπως η ζωή ενός παίκτη, η θέση αντικειμένων ή το σκορ. Ένα σημαντικό πλεονέκτημα των **Network Variables** είναι ότι υποστηρίζουν callbacks, τα οποία ενεργοποιούνται κάθε φορά που αλλάζει η τιμή τους, επιτρέποντας έτσι την άμεση ενημέρωση του interface ή άλλων συστημάτων. Ωστόσο, η χρήση τους περιορίζεται σε σενάρια όπου ο διακομιστής διαχειρίζεται όλες τις αλλαγές, γεγονός που μπορεί να μην είναι κατάλληλο για πιο περίπλοκες αλληλεπιδράσεις.

Αντίθετα, τα **Client RPCs** (Remote Procedure Calls) προσφέρουν μεγαλύτερη ευελιξία, καθώς επιτρέπουν την αποστολή δεδομένων ή την εκτέλεση εντολών απευθείας σε έναν ή περισσότερους πελάτες. Αυτό τα καθιστά ιδανικά για σενάρια που απαιτούν πιο περίπλοκη λογική ή εξειδικευμένη αλληλεπίδραση, όπως η εκκίνηση animations, η αποστολή ειδικών μηνυμάτων σε συγκεκριμένους πελάτες ή η διαχείριση μοναδικών συμβάντων. Παρόλο που τα **Client RPCs** προσφέρουν μεγάλη ευελιξία, απαιτούν επιπλέον κώδικα για τη διαχείριση των δεδομένων και δεν παρέχουν ενσωματωμένο μηχανισμό για τον συγχρονισμό τους, κάτι που αυξάνει τη δυσκολία στη χρήση τους και το φορτίο του δικτύου εάν δεν χρησιμοποιούνται σωστά.

Συνοπτικά, οι **Network Variables** είναι καλύτερες για τη διαχείριση δεδομένων που χρειάζονται συνεχή και απρόσκοπτο συγχρονισμό, ενώ τα **Client RPCs** είναι πιο κατάλληλα για περιστασιακά συμβάντα που απαιτούν μεγαλύτερη ευελιξία και έλεγχο. Ο συνδυασμός τους μπορεί να αποδειχθεί εξαιρετικά χρήσιμος, επιτρέποντας στους προγραμματιστές να διαχειριστούν τόσο τις τακτικές ενημερώσεις δεδομένων όσο και τις πιο σύνθετες λειτουργίες ενός multiplayer παιχνιδιού.

5.2 Συγχρονισμός Αμφιέσεων

Ο συγχρονισμός των αμφιέσεων στους χαρακτήρες του παιχνιδιού αποτέλεσε ένα από τα πιο σύνθετα και απαιτητικά προβλήματα κατά τη διάρκεια της ανάπτυξης. Η δυσκολία έγκειται στο γεγονός ότι τα μοντέλα των χαρακτήρων δεν είναι απλώς απλά δεδομένα, όπως οι αριθμοί ή οι τιμές που μπορούμε εύκολα να συγχρονίσουμε μέσω δικτύου. Αντίθετα, τα μοντέλα περιλαμβάνουν πολύπλοκες πληροφορίες για τη γεωμετρία, τη δομή, τις υφές και τους σκελετούς των χαρακτήρων, πράγμα που τα καθιστά πιο δύσκολα στην αποθήκευση και στον συγχρονισμό τους σε ένα δίκτυο. Ο κάθε χαρακτήρας έχει μια μοναδική αναπαράσταση με τη μορφή ενός σκελετού και μιας σειράς κινήσεων (animations), και η αλλαγή αυτής της αναπαράστασης από τη βασική αμφίεση στην επιλεγμένη αμφίεση απαιτεί εξαιρετική ακρίβεια για να αποφευχθούν προβλήματα, όπως ασυνεπείς κινήσεις ή κακή απεικόνιση των χαρακτήρων.

Η διαδικασία συγχρονισμού γίνεται πιο πολύπλοκη, καθώς οι χαρακτήρες πρέπει να προσαρμόζονται δυναμικά σε νέες αμφιέσεις, οι οποίες ενδέχεται να απαιτούν διαφορετικές δομές σκελετού ή διαφορετικά χαρακτηριστικά. Ο σχεδιασμός του συστήματος, λοιπόν, πρέπει να διασφαλίσει ότι κάθε φορά που ο χρήστης επιλέγει μια νέα αμφίεση, το μοντέλο του χαρακτήρα να αλλάζει χωρίς να προκαλείται σφάλμα ή καθυστέρηση στη διαδικασία του παιχνιδιού. Καθώς η κάθε αμφίεση μπορεί να έχει τον δικό της σκελετό και τις δικές της κινήσεις, πρέπει να διασφαλιστεί ότι οι νέες αμφιέσεις συμβαδίζουν με τις ήδη υπάρχουσες κινήσεις και animations του χαρακτήρα.

Για να επιτευχθεί ο συγχρονισμός, η διαδικασία αρχίζει με την αποθήκευση της επιλεγμένης αμφίεσης στον **PlayerManager**, ένα script που διαχειρίζεται και αποθηκεύει τα δεδομένα του κάθε παίκτη. Αυτός ο διαχειριστής συλλέγει και συντηρεί τις προσωπικές επιλογές του παίκτη, όπως την αμφίεση που έχει επιλέξει. Η τιμή της αμφίεσης αποθηκεύεται ως μια

Network Variable, επιτρέποντας τη συγχρονισμένη αλλαγή σε όλους τους clients. Οι **Network Variables** επιτρέπουν τη διατήρηση δεδομένων που είναι κοινά σε όλους τους πελάτες, διασφαλίζοντας ότι κάθε φορά που ο διακομιστής (server) ενημερώνει μια τέτοια μεταβλητή, όλοι οι πελάτες λαμβάνουν και εφαρμόζουν την ενημέρωση ταυτόχρονα. Με την χρήση αυτής της Network Variable, ο συγχρονισμός γίνεται αυτόματα και σε πραγματικό χρόνο. Όταν η τιμή της μεταβλητής αλλάξει, ενεργοποιείται ένας **callback function**, ο οποίος εφαρμόζει τις αλλαγές της αμφίσεσης για κάθε χρήστη του παιχνιδιού. Αυτή η διαδικασία διασφαλίζει ότι το μοντέλο του χαρακτήρα στον client αλλάζει σε πραγματικό χρόνο και εμφανίζεται στους υπόλοιπους χρήστες με τη σωστή αμφίσεση, χωρίς καθυστερήσεις ή προβλήματα συγχρονισμού. Αυτό σημαίνει ότι αν ένας παίκτης επιλέξει, για παράδειγμα, έναν ιππότη ως αμφίσεση, οι υπόλοιποι παίκτες στο παιχνίδι θα δουν επίσης τον χαρακτήρα αυτόν να φορά την ίδια αμφίσεση ταυτόχρονα και με τη σωστή εμφάνιση.

Η προσέγγιση αυτή με τη χρήση **Network Variables** και **callback functions** μειώνει σημαντικά την πιθανότητα εμφάνισης σφαλμάτων ή ασυμφωνιών στον συγχρονισμό των χαρακτήρων. Εξασφαλίζει ότι όλοι οι παίκτες βλέπουν την ίδια, ακριβή απεικόνιση του χαρακτήρα με τη νέα αμφίσεση, κάτι που συμβάλλει στη συνολική ποιότητα του παιχνιδιού. Ένα ακόμα σημαντικό πλεονέκτημα είναι η ευχρηστία και η ευελιξία που προσφέρει αυτή η μέθοδος. Εφόσον οι αμφίσεσεις ή οι σκελετοί των χαρακτήρων είναι πλήρως διαχειρίσιμοι από τον server, δεν υπάρχει ανάγκη για επιπλέον χειροκίνητο συγχρονισμό σε κάθε client, εξοικονομώντας πόρους και αποφεύγοντας την ανάγκη για σύνθετους υπολογισμούς στο τοπικό επίπεδο του κάθε χρήστη.

Αν και η διαδικασία αυτή είναι τεχνικά απαιτητική, η χρήση του **PlayerManager** και των **Network Variables** σε συνδυασμό με την ενσωμάτωση **callback functions** αποδείχθηκε πολύ αποτελεσματική. Η προσέγγιση αυτή παρέχει μια ευέλικτη και αξιόπιστη λύση για τον συγχρονισμό αμφίσεσεων, ενώ ταυτόχρονα διασφαλίζει μια άριστη εμπειρία για όλους τους χρήστες του παιχνιδιού. Παρά τις προκλήσεις, το σύστημα αυτό προσφέρει μια συνεπή και ομαλή εμπειρία στον τομέα του multiplayer, επιτρέποντας στους παίκτες να απολαύσουν το παιχνίδι χωρίς προβλήματα συγχρονισμού ή ασυμφωνιών στην εμφάνιση των χαρακτήρων τους.

5.3 Συγχρονισμός Animation

Στο παιχνίδι, η αλλαγή της αμφίσεσης ενός χαρακτήρα αποτελεί κρίσιμο σημείο για τη διατήρηση της σωστής λειτουργίας των animations. Κάθε αμφίσεση μπορεί να βασίζεται σε διαφορετικό σκελετό (**skeleton**) ή **Avatar**. Το **Avatar** στην Unity είναι μια συσχέτιση ανάμεσα στον σκελετό ενός χαρακτήρα και τα animations του, που επιτρέπει τη σωστή εφαρμογή των κινήσεων στον χαρακτήρα. Εάν ο σκελετός του χαρακτήρα δεν συμβαδίζει με την αμφίσεση, τα animations δεν θα αποδίδονται σωστά, κάτι που μπορεί να οδηγήσει σε ανεπιθύμητα οπτικά αποτελέσματα, όπως κακές ευθυγραμμίσεις, παραμορφώσεις ή εντελώς αποτυχημένες κινήσεις.

Για την επίλυση αυτού του προβλήματος, κατά την αλλαγή της αμφίσεσης, το σύστημα πρέπει να αντικαθιστά δυναμικά τον τρέχοντα σκελετό του χαρακτήρα με τον κατάλληλο για την επιλεγμένη αμφίσεση. Αυτή η αλλαγή διασφαλίζει ότι τα animations εφαρμόζονται ομαλά και ότι η εμπειρία του παιχνιδιού παραμένει συνεπής. Η διαδικασία αυτή περιλαμβάνει την ανανέωση του **Animator Controller** του χαρακτήρα ώστε να αντιστοιχεί στον νέο σκελετό και στις κινήσεις που υποστηρίζονται από την αμφίσεση.

5.3.1 Χρήση του Network Animator για Συγχρονισμό Animations

Η **Unity** παρέχει το **Network Animator**, ένα εργαλείο που απλοποιεί τον συγχρονισμό των animations σε ένα multiplayer περιβάλλον. Σε μια αρχιτεκτονική τύπου **server-authoritative**, ο διακομιστής διαχειρίζεται όλες τις αλλαγές και ενημερώσεις. Όταν ο διακομιστής ζητά από έναν client να εκτελέσει ένα animation, το **Network Animator** διασφαλίζει ότι το animation εκτελείται σε όλους τους συνδεδεμένους πελάτες.

Για παράδειγμα, όταν ένας παίκτης αλλάξει αμφίσεση και εκτελέσει μια κίνηση (όπως το να τρέξει, να χαιρετίσει ή να πηδήξει), το **Network Animator** συγχρονίζει αυτόματα το animation. Αυτό σημαίνει ότι όλοι οι παίκτες θα δουν τον ίδιο χαρακτήρα να εκτελεί την ίδια κίνηση, με τη σωστή αμφίσεση και σκελετό. Το αποτέλεσμα είναι μια συνεκτική εμπειρία παιχνιδιού, όπου όλες οι αλλαγές και οι ενέργειες εμφανίζονται ομαλά και σε πραγματικό χρόνο.

5.4 Συγχρονισμός περιοχών

Τα δεδομένα που αφορούν τις περιοχές αποθηκεύονται αποκλειστικά στον **server**, διασφαλίζοντας έτσι την ακεραιότητα και την ακρίβεια των κρίσιμων πληροφοριών. Οι πελάτες (clients) έχουν πρόσβαση μόνο σε βασικές πληροφορίες, όπως το **όνομα του ιδιοκτήτη**, το **επίπεδο της περιοχής**, το **ενοίκιο** που αποφέρει και το **αν είναι διαθέσιμη ή όχι**. Αυτές οι πληροφορίες παρέχονται αποκλειστικά για σκοπούς εμφάνισης και δεν επιτρέπουν στους πελάτες να επηρεάσουν τη λειτουργικότητα του παιχνιδιού.

Ο σχεδιασμός αυτός είναι σημαντικός, καθώς τα δεδομένα που διαχειρίζεται ο server θεωρούνται αξιόπιστα και προστατεύονται από κακόβουλες ή ακούσιες παρεμβολές. Ακόμα και αν ένας πελάτης επιχειρήσει να τροποποιήσει αυτά τα δεδομένα από την πλευρά του, οι αλλαγές αυτές δεν έχουν κανένα αποτέλεσμα, αφού δεν γίνονται αποδεκτές από τον server. Έτσι, το σύστημα αποτρέπει περιπτώσεις εξαπάτησης (cheating) ή αναντιστοιχίας δεδομένων μεταξύ των παικτών.

Αυτό το μοντέλο λειτουργίας, γνωστό και ως **server-authoritative architecture**, διασφαλίζει τη σταθερότητα του παιχνιδιού και την ορθή διαχείριση κρίσιμων στοιχείων. Παράλληλα, περιορίζει την ποσότητα των δεδομένων που μεταδίδονται στο δίκτυο, εστιάζοντας μόνο σε αυτά που χρειάζονται οι πελάτες για την οπτικοποίηση και την κατανόηση της κατάστασης του παιχνιδιού, όπως η κατάσταση των περιοχών.

5.5 Κίνηση παικτών

Στο παιχνίδι, η κίνηση των παικτών γίνεται αποκλειστικά όταν είναι η σειρά τους και έχουν «ρίξει» το ζάρι, εξασφαλίζοντας την ομαλή και δίκαιη ροή του παιχνιδιού. Το ποιος παίκτης έχει τη σειρά του είναι ένα δεδομένο που διαχειρίζεται αποκλειστικά ο **server**. Αυτός ο σχεδιασμός εξασφαλίζει ότι μόνο ένας παίκτης μπορεί να παίξει κάθε φορά, αποτρέποντας ενδεχόμενες ασυγχρονίες ή παραβιάσεις στους κανόνες του παιχνιδιού.

Όταν ο τρέχων παίκτης πατήσει το κουμπί για να ρίξει το ζάρι, ο **server** αναλαμβάνει να παράγει τυχαία το αποτέλεσμα της ζαριάς, χρησιμοποιώντας έναν αλγόριθμο τυχαιότητας. Στη συνέχεια, ο server υπολογίζει τη νέα θέση του παίκτη, λαμβάνοντας υπόψη τον αριθμό των περιοχών που πρέπει να προχωρήσει, και ενημερώνει τους πελάτες (clients) για την αλλαγή. Έτσι, η θέση του παίκτη συγχρονίζεται σωστά σε όλα τα συστήματα, διασφαλίζοντας ότι όλοι οι παίκτες βλέπουν την ίδια κατάσταση στο παιχνίδι.

Αυτός ο μηχανισμός εγγυάται την ακρίβεια της κίνησης των παικτών, καθώς και τη διατήρηση της σειράς. Επιπλέον, με τον server να ελέγχει αποκλειστικά τη ροή και την τυχαιότητα της ζαριάς, διασφαλίζεται η αξιοπιστία και η ακεραιότητα του παιχνιδιού, αποτρέποντας περιπτώσεις εξαπάτησης ή λαθών στον υπολογισμό της κίνησης.

6. Netcode vs άλλες επιλογές

6.1 Netcode vs Mirror

Η σύγκριση μεταξύ του **Netcode for GameObjects (NGO)** και του **Mirror** για την ανάπτυξη multiplayer παιχνιδιών στην πλατφόρμα Unity αποκαλύπτει σημαντικές διαφορές όσον αφορά την αρχιτεκτονική, την απόδοση, την ευχρηστία, τη δυνατότητα προσαρμογής και τη γενική υποστήριξη. Παρά το γεγονός ότι και τα δύο εργαλεία εξυπηρετούν τον ίδιο σκοπό –την υλοποίηση δικτυακών παιχνιδιών– το καθένα προσφέρει διαφορετική προσέγγιση, ανάλογα με τις ανάγκες του project. Ενώ το **Netcode for GameObjects** είναι μια επίσημη λύση που αναπτύχθηκε από την Unity για να επιτρέψει στους developers να δημιουργήσουν παιχνίδια με εύκολο τρόπο και χωρίς να απαιτείται εξειδικευμένη γνώση της δικτύωσης, το **Mirror** είναι ένα πιο ευέλικτο, ανοιχτού κώδικα εργαλείο που προσφέρει περισσότερη ελευθερία και δυνατότητες προσαρμογής, αλλά μπορεί να απαιτεί μεγαλύτερη εξοικείωση με τις έννοιες δικτύωσης.

6.1.1 Αρχιτεκτονική και Δομή

Η αρχιτεκτονική του **Netcode for GameObjects** επικεντρώνεται στην απλότητα και την εύκολη ενσωμάτωση στην υπάρχουσα Unity πλατφόρμα, επιτρέποντας στους developers να ξεκινήσουν γρήγορα χωρίς να χρειάζεται να εμβαθύνουν σε πολύπλοκες ρυθμίσεις δικτύωσης. Η βασική αρχιτεκτονική του είναι πιο περιορισμένη και ακολουθεί την πεπατημένη πορεία της Unity για τον συγχρονισμό δεδομένων, όπου τα δεδομένα των αντικειμένων και οι καταστάσεις τους συγχρονίζονται μεταξύ του server και των clients μέσω **RPCs (Remote Procedure Calls)** και **Network Variables**. Το Netcode δίνει επίσης τη δυνατότητα να συγχρονίζονται τα GameObjects που αλληλεπιδρούν μεταξύ τους, αλλά με έναν πιο δομημένο και πιο αυστηρό τρόπο. Είναι ιδανικό για παιχνίδια που δεν απαιτούν εξαιρετικά περίπλοκο συγχρονισμό ή απαιτητικές ρυθμίσεις δικτύωσης.

Αντίθετα, το **Mirror** παρέχει μια πολύ πιο ευέλικτη αρχιτεκτονική που επιτρέπει στους developers να έχουν πλήρη έλεγχο των δεδομένων που ανταλλάσσονται μεταξύ server και client. Στο Mirror, οι RPCs και τα NetworkTransform μπορούν να προσαρμοστούν εύκολα για πιο σύνθετες ανάγκες. Το εργαλείο αυτό επιτρέπει την υλοποίηση μεγαλύτερης ποικιλίας αρχιτεκτονικών, όπως peer-to-peer και client-server, με μεγαλύτερη ευχέρεια στον τρόπο που τα δεδομένα συγχρονίζονται και μεταφέρονται. Το Mirror παρέχει επίσης περισσότερες επιλογές για τον χειρισμό της ροής των δεδομένων, καθιστώντας το πιο κατάλληλο για μεγαλύτερης κλίμακας multiplayer παιχνίδια.

6.1.2 Συγχρονισμός και Διαχείριση Δεδομένων

Το **Netcode** παρέχει βασικές δυνατότητες για την ανταλλαγή δεδομένων σε πραγματικό χρόνο, με κύρια εργαλεία το **Network Variable** για την αποθήκευση δεδομένων που συγχρονίζονται μεταξύ των παικτών, και τα **RPCs**, τα οποία επιτρέπουν στους clients να εκτελούν ενέργειες στον server ή και αντίστροφα. Αυτό το μοντέλο δουλεύει καλά για πιο απλά παιχνίδια, αλλά περιορίζει την ευχέρεια και την προσαρμοστικότητα στον χειρισμό των δεδομένων. Ο συγχρονισμός των αντικειμένων, για παράδειγμα, είναι αρκετά ευθύγραμμος και δεν προσφέρει πολλές επιλογές παραμετροποίησης. Αν και αυτό μπορεί να είναι θετικό για απλότητα, μπορεί να γίνει πρόβλημα σε πιο σύνθετα παιχνίδια.

Αντίθετα, το **Mirror** παρέχει περισσότερες δυνατότητες συγχρονισμού δεδομένων, επιτρέποντας στους developers να παραμετροποιούν πώς, πότε και ποια δεδομένα θα συγχρονίζονται. Για παράδειγμα, το Mirror επιτρέπει την παραμετροποίηση του ποια αντικείμενα και δεδομένα θα συγχρονίζονται, την εφαρμογή πιο σύνθετων αλγορίθμων συγχρονισμού και τη διαχείριση πιο περίπλοκων αλληλεπιδράσεων μεταξύ των παικτών. Αυτή η ευελιξία του καθιστά το Mirror εξαιρετικό για παιχνίδια που απαιτούν ακριβή και αξιόπιστο συγχρονισμό δεδομένων σε πραγματικό χρόνο, όπως παιχνίδια δράσης ή παιχνίδια στρατηγικής σε πραγματικό χρόνο (RTS).

6.1.3 Απόδοση και Επεκτασιμότητα

Η απόδοση του **Netcode for GameObjects** είναι εξαιρετική για μικρά και μεσαία multiplayer παιχνίδια, με τη συγχρονισμένη επικοινωνία να είναι εξαιρετικά αποδοτική για τα περισσότερα είδη παιχνιδιών. Ωστόσο, η απόδοση του μπορεί να μειωθεί εάν η εφαρμογή απαιτεί τη διαχείριση πολλών δεδομένων ή πιο περίπλοκες δομές δικτύωσης, καθώς το εργαλείο έχει σχεδιαστεί για να καλύψει βασικές ανάγκες δικτύωσης χωρίς υπερβολική πολυπλοκότητα. Για

πιο απαιτητικά παιχνίδια με πολλούς παίκτες ή πιο σύνθετα σενάρια, η απόδοση του μπορεί να μην είναι τόσο βέλτιστη όσο του Mirror.

Το **Mirror**, ωστόσο, έχει αποδειχθεί εξαιρετικά αποδοτικό και κλιμακούμενο, προσφέροντας τη δυνατότητα να υποστηρίξει πολύ μεγαλύτερους αριθμούς παικτών και πιο σύνθετα δεδομένα χωρίς σημαντική επιβάρυνση στην απόδοση. Ο τρόπος που το Mirror διαχειρίζεται τα δεδομένα και η ευχέρεια παραμετροποίησης καθιστούν τη δυνατότητα επεκτασιμότητας εξαιρετική, ειδικά για παιχνίδια μεγάλης κλίμακας. Αυτή η δυνατότητα του Mirror να διαχειρίζεται μεγαλύτερη κλίμακα χρηστών και δεδομένων είναι ένα από τα κύρια πλεονεκτήματά του σε σχέση με το Netcode.

6.1.4 Ευχρηστία και Υποστήριξη

Το **Netcode** έχει σχεδιαστεί για να ενσωματώνεται εύκολα με την Unity και προσφέρει μια αρκετά απλοποιημένη API που καθιστά τη χρήση του εύκολη για developers με περιορισμένη εμπειρία στον τομέα της δικτύωσης. Η αρχική καμπύλη μάθησης είναι χαμηλή, και το εργαλείο προσφέρει πολλές έτοιμες λύσεις για τις πιο κοινές ανάγκες ενός multiplayer παιχνιδιού. Ωστόσο, για πιο περίπλοκες απαιτήσεις, οι developers ίσως χρειαστεί να εντρυφήσουν περισσότερο στις δυνατότητες του Netcode.

Από την άλλη πλευρά, το **Mirror** απαιτεί περισσότερη εξοικείωση με την έννοια της δικτύωσης, καθώς προσφέρει περισσότερες παραμέτρους που πρέπει να ελέγχονται από τον developer. Η τεκμηρίωση του Mirror είναι εκτενής και καλύπτει όλες τις δυνατότητες του εργαλείου, γεγονός που το καθιστά κατάλληλο για προχωρημένους χρήστες. Παρόλο που το Mirror προσφέρει περισσότερη ευχέρεια, η καμπύλη μάθησης μπορεί να είναι απότομη για νέους χρήστες.

6.1.5 Κοινότητα και Υποστήριξη

Το **Netcode for GameObjects** είναι υποστηριζόμενο από την Unity, κάτι που εξασφαλίζει την επίσημη υποστήριξη και την ανάπτυξή του ως ένα ενσωματωμένο εργαλείο στη Unity. Ωστόσο, η κοινότητά του μπορεί να είναι περιορισμένη σε σύγκριση με το Mirror, καθώς το Netcode έχει κυκλοφορήσει πρόσφατα και είναι ακόμα σε διαδικασία εξέλιξης. Παρ' όλα αυτά, η υποστήριξη από την Unity σημαίνει ότι οι developers μπορούν να είναι σίγουροι για τη συνέχιση της ανάπτυξής του και για την ενσωμάτωσή του σε μελλοντικές εκδόσεις της Unity.

Αντίθετα, το **Mirror** έχει μια ενεργή κοινότητα ανοιχτού κώδικα, με πολλές δημοσιευμένες λύσεις και συμβουλές που βοηθούν τους developers να ξεπεράσουν προβλήματα και να προσαρμόσουν το εργαλείο στις ανάγκες τους. Η κοινότητα του Mirror είναι ιδιαίτερα υποστηρικτική και ανταλλάσσει ιδέες, γεγονός που διευκολύνει τους χρήστες να επιλύσουν προβλήματα και να επεκτείνουν τις δυνατότητες του εργαλείου.

6.1.6 Συμπέρασμα

Η επιλογή μεταξύ **Netcode for GameObjects** και **Mirror** εξαρτάται σε μεγάλο βαθμό από τις ανάγκες του παιχνιδιού που αναπτύσσεται. Το **Netcode for GameObjects** είναι μια εξαιρετική επιλογή για developers που θέλουν μια εύκολη και ενσωματωμένη λύση δικτύωσης με ελάχιστη πολυπλοκότητα και γρήγορη υλοποίηση για μικρής και μεσαίας κλίμακας παιχνίδια. Από την άλλη, το **Mirror** προσφέρει μεγαλύτερη ευελιξία, περισσότερη δύναμη στον έλεγχο των δεδομένων και της υποδομής του δικτύου, καθιστώντας το την καλύτερη επιλογή για πιο απαιτητικά και μεγάλης κλίμακας multiplayer παιχνίδια.

6.2 Netcode vs Fishnet

Η σύγκριση του **Netcode for GameObjects (NGO)** με το **FishNet** αφορά δύο από τα πιο δημοφιλή εργαλεία δικτύωσης για παιχνίδια της Unity. Κάθε ένα από αυτά έχει τις δικές του ισχυρές πλευρές και τις καταλληλότερες εφαρμογές, ανάλογα με τον τύπο του παιχνιδιού και τις απαιτήσεις της ανάπτυξης. Αν και το **Netcode for GameObjects** και το **FishNet** έχουν κοινές δυνατότητες, όπως η απομακρυσμένη εκτέλεση διαδικασιών (RPCs) και η συγχρονισμένη κατάσταση των αντικειμένων, υπάρχουν σημαντικές διαφορές που καθορίζουν ποιο εργαλείο είναι πιο κατάλληλο για κάθε περίπτωση χρήσης.

6.2.1 Αρχιτεκτονική και Βασικές Δυνατότητες

Το **Netcode for GameObjects (NGO)** αναπτύχθηκε από την Unity για να παρέχει μια λύση δικτύωσης ενσωματωμένη απευθείας στο οικοσύστημα της Unity. Η αρχιτεκτονική του είναι απλή και ευέλικτη, προσφέροντας εργαλεία για τον συγχρονισμό δεδομένων, όπως οι **Network Variables** (που επιτρέπουν την απομακρυσμένη συγχρονισμένη μεταβλητότητα), καθώς και τα **Remote Procedure Calls (RPCs)** για την απομακρυσμένη εκτέλεση ενεργειών μεταξύ του

client και του server. Ο στόχος του είναι να παρέχει έναν ευκολότερο και πιο γρήγορο τρόπο για developers να δημιουργούν multiplayer παιχνίδια, χωρίς να απαιτούνται ιδιαίτερες γνώσεις δικτύωσης. Παρόλο που η βασική του δομή είναι αρκετά ισχυρή, το **Netcode for GameObjects** ενδέχεται να παρουσιάσει περιορισμούς σε πιο σύνθετες εφαρμογές ή παιχνίδια με υψηλές απαιτήσεις συγχρονισμού και απόδοσης.

Από την άλλη πλευρά, το **FishNet** είναι ένα εργαλείο δικτύωσης που προσφέρει μια πιο ισχυρή και εξειδικευμένη λύση για την ανάπτυξη παιχνιδιών που απαιτούν υψηλή απόδοση και συγχρονισμό σε πραγματικό χρόνο. Το **FishNet** χρησιμοποιεί ένα πιο προηγμένο μοντέλο δικτύωσης και υποστηρίζει τεχνικές όπως **prediction**, **lag compensation** και **client-side interpolation**, που το καθιστούν κατάλληλο για παιχνίδια σε πραγματικό χρόνο, όπως FPS και παιχνίδια στρατηγικής με πολλές αλληλεπιδράσεις. Σε αντίθεση με το **Netcode for GameObjects**, το FishNet προσφέρει περισσότερο έλεγχο στον προγραμματιστή, επιτρέποντας του να διαχειριστεί την συμπεριφορά της δικτύωσης σε επίπεδο χαμηλότερης αλληλεπίδρασης, κάτι που το καθιστά ιδανικό για μεγάλης κλίμακας και υψηλής απόδοσης παιχνίδια.

6.2.2 Απόδοση και Κλιμάκωση

Το **Netcode for GameObjects** είναι κατάλληλο για παιχνίδια μικρής και μεσαίας κλίμακας, όπου το απαιτούμενο επίπεδο απόδοσης είναι μέτριο. Για παιχνίδια με περιορισμένο αριθμό παικτών ή με λιγότερη ανάγκη για άμεσο συγχρονισμό, το **NGO** παρέχει επαρκή απόδοση και αρκετή απλότητα για να διευκολύνει την ανάπτυξη. Ωστόσο, όσο μεγαλύτερη είναι η κλίμακα του παιχνιδιού, τόσο πιο δύσκολο γίνεται να διαχειριστεί την αυξανόμενη καθυστέρηση, τον όγκο δεδομένων και τις απαιτήσεις συγχρονισμού που προκύπτουν.

Το **FishNet** είναι πολύ πιο αποτελεσματικό για παιχνίδια με μεγαλύτερους αριθμούς παικτών ή για παιχνίδια που απαιτούν γρήγορη και ακριβή ανταλλαγή δεδομένων. Χρησιμοποιεί προηγμένες τεχνικές, όπως την **client-side prediction** (προβλέψεις από τον client για να μειώσει την καθυστέρηση) και την **lag compensation** (αντιστροφή καθυστέρησης), οι οποίες είναι κρίσιμες για παιχνίδια που χρειάζονται άμεσο συγχρονισμό και απαιτούν εξαιρετικά γρήγορες ανταλλαγές πληροφοριών, όπως τα FPS και τα παιχνίδια με πολλούς ταυτόχρονους χρήστες. Η αρχιτεκτονική του FishNet επιτρέπει τη σωστή διαχείριση μεγαλύτερης κλίμακας παιχνιδιών, εξασφαλίζοντας ότι η απόδοση και η ταχύτητα της ανταλλαγής δεδομένων παραμένουν σταθερές, ακόμα και σε παιχνίδια με μεγάλο αριθμό παικτών.

6.2.3 Ευχρηστία και Καμπύλη Εκμάθησης

Όσον αφορά την **ευχρηστία**, το **Netcode for GameObjects** είναι πιο φιλικό προς τους αρχάριους και πολύ πιο εύκολο στην εκμάθηση, χάρη στην απλότητα του και στην ενσωμάτωσή του στο οικοσύστημα της Unity. Οι developers δεν χρειάζονται προηγμένες γνώσεις δικτύωσης για να ξεκινήσουν, καθώς η Unity παρέχει εύκολη και κατανοητή τεκμηρίωση. Το **NGO** είναι κατάλληλο για μικρές και μεσαίες εφαρμογές, όπου η δικτύωση δεν χρειάζεται να είναι εξαιρετικά εξελιγμένη. Η καμπύλη εκμάθησης είναι χαμηλή, και η διαδικασία ανάπτυξης είναι πιο άμεση και γρήγορη.

Το **FishNet** έχει μεγαλύτερη καμπύλη εκμάθησης, καθώς απαιτεί βαθύτερη κατανόηση της δικτύωσης και των τεχνικών υψηλής απόδοσης. Η τεκμηρίωση είναι καλή, αλλά λόγω της πολυπλοκότητας του εργαλείου, οι developers χρειάζονται περισσότερο χρόνο για να μάθουν να χρησιμοποιούν πλήρως το FishNet, ειδικά αν δεν έχουν προηγούμενη εμπειρία με πιο εξειδικευμένα εργαλεία δικτύωσης. Παρά αυτή τη μεγαλύτερη καμπύλη εκμάθησης, το FishNet προσφέρει μεγαλύτερη ευχέρεια και προσαρμοστικότητα για προχωρημένες ανάγκες δικτύωσης.

6.2.4 Κοινότητα και Υποστήριξη

Η **Unity** παρέχει εξαιρετική υποστήριξη και τεκμηρίωση για το **Netcode for GameObjects**, και η κοινότητα είναι τεράστια, αφού το εργαλείο είναι επίσημο και ενσωματωμένο στο οικοσύστημα της Unity. Οι developers μπορούν εύκολα να βρουν παραδείγματα και υποστήριξη από άλλους χρήστες της Unity, κάτι που κάνει το **Netcode** μια ιδανική λύση για γρήγορη ανάπτυξη με λιγότερη ανάγκη από εξειδικευμένη βοήθεια.

Αντίθετα, το **FishNet** είναι ένα εργαλείο ανοικτού κώδικα και η κοινότητα του είναι μικρότερη, αλλά πολύ ενεργή. Παρόλο που η κοινότητα του FishNet είναι μικρότερη σε σχέση με αυτή του **Netcode**, οι χρήστες μπορούν να βρουν βοήθεια μέσω του Discord ή του GitHub, όπου συχνά προσφέρονται λύσεις για πιο εξειδικευμένα ζητήματα. Το FishNet αναπτύσσεται συνεχώς και οι ανανεώσεις του είναι τακτικές, παρέχοντας νέες δυνατότητες και βελτιώσεις που διασφαλίζουν τη συνεχιζόμενη υποστήριξή του.

6.2.5 Συμπέρασμα και Επιλογή Εργαλείου

Η τελική επιλογή μεταξύ του **Netcode for GameObjects** και του **FishNet** εξαρτάται από τις ανάγκες του κάθε project. Αν το παιχνίδι έχει μικρότερο αριθμό παικτών, απαιτεί γρήγορη ανάπτυξη και δεν χρειάζεται υπερβολική συγχρονισμένη αλληλεπίδραση, το **Netcode for GameObjects** είναι μια ιδανική λύση. Είναι εύκολο στη χρήση, αποδοτικό για μικρής και μεσαίας κλίμακας παιχνίδια, και παρέχει τις βασικές δυνατότητες δικτύωσης που απαιτούνται για αυτά τα είδη παιχνιδιών.

Αντιθέτως, το **FishNet** είναι ιδανικό για μεγαλύτερα και πιο περίπλοκα παιχνίδια που απαιτούν εξαιρετική απόδοση και ακριβή συγχρονισμό σε πραγματικό χρόνο. Ειδικά για παιχνίδια FPS ή μεγάλης κλίμακας με υψηλές απαιτήσεις για κλιμάκωση και εξαιρετική απόδοση, το **FishNet** είναι η πιο κατάλληλη λύση, αν και η καμπύλη εκμάθησης του μπορεί να είναι πιο απαιτητική.

6.3 Netcode vs Photon

Η σύγκριση του **Netcode for GameObjects (NGO)** με το **Photon** αφορά δύο από τα πιο δημοφιλή εργαλεία δικτύωσης για multiplayer παιχνίδια στην Unity. Και τα δύο εργαλεία παρέχουν ισχυρές δυνατότητες δικτύωσης, αλλά διαφέρουν σε πολλούς τομείς, όπως η αρχιτεκτονική, οι δυνατότητες, η κλίμακα, η ευχρηστία και οι απαιτήσεις απόδοσης. Ακολουθεί μια σύγκριση του **Netcode for GameObjects** και του **Photon**, εστιάζοντας στις βασικές διαφορές και τα πλεονεκτήματα κάθε εργαλείου.

6.3.1 Αρχιτεκτονική και Βασικές Δυνατότητες

Το **Netcode for GameObjects (NGO)** αναπτύχθηκε από την Unity για να παρέχει μια ενσωματωμένη και ευέλικτη λύση δικτύωσης, η οποία λειτουργεί φυσικά με το οικοσύστημα της Unity. Στο **Netcode**, οι βασικές δυνατότητες περιλαμβάνουν την απομακρυσμένη εκτέλεση διαδικασιών μέσω **Remote Procedure Calls (RPCs)**, τον συγχρονισμό μεταβλητών μέσω **Network Variables**, και τη διαχείριση αντικειμένων και καταστάσεων στον δικτυακό κόσμο. Το **Netcode** είναι ιδανικό για developers που θέλουν να ενσωματώσουν multiplayer gameplay σε Unity χωρίς να απαιτούν πρόσθετες εξωτερικές βιβλιοθήκες και διαχειρίζονται την ίδια αρχιτεκτονική με το υπόλοιπο παιχνίδι.

Από την άλλη πλευρά, το **Photon** είναι ένα πιο εξελιγμένο και έτοιμο προς χρήση εργαλείο για multiplayer παιχνίδια, προσφέροντας μια σειρά από υπηρεσίες, όπως το **Photon Realtime**, το **Photon PUN (Photon Unity Networking)** και το **Photon Fusion**. Το **Photon** παρέχει ένα cloud-based server infrastructure που χειρίζεται τη συγχρονισμένη αλληλεπίδραση παικτών, προσφέροντας μια εξαιρετικά κλιμακούμενη λύση. Ενώ το **Netcode** επικεντρώνεται στην ενσωμάτωση εντός της Unity, το **Photon** προσφέρει μεγαλύτερη ευελιξία όσον αφορά την υποστήριξη για cross-platform multiplayer και την εύκολη κλιμάκωση της υποδομής του.

6.3.2 Κλιμάκωση και Απόδοση

Η **κλίμακα** και η **απόδοση** του παιχνιδιού είναι μια από τις βασικές διαφορές. Το **Netcode for GameObjects** είναι κατάλληλο για παιχνίδια μικρής και μεσαίας κλίμακας. Παρόλο που παρέχει καλή υποστήριξη για μικρούς ή μεσαίους αριθμούς παικτών, η απόδοσή του ενδέχεται να αντιμετωπίσει προβλήματα σε μεγαλύτερες κλίμακες, ειδικά όταν απαιτείται συγχρονισμός μεγάλου αριθμού παικτών ή εντυπωσιακές αλληλεπιδράσεις μεταξύ τους.

Το **Photon**, αντίθετα, έχει μια πιο επαγγελματική υποδομή και είναι φτιαγμένο για να υποστηρίξει μεγαλύτερες κλίμακες. Το **Photon Realtime** και το **Photon PUN** επιτρέπουν τη γρήγορη κλιμάκωση του παιχνιδιού, ενώ το **Photon Fusion** προσφέρει προηγμένες δυνατότητες για συγχρονισμό σε πραγματικό χρόνο και πιο αποδοτικά μοντέλα δικτύωσης, ειδικά για παιχνίδια με πολλούς παίκτες και απαιτήσεις χαμηλής καθυστέρησης. Η υποδομή του **Photon** παρέχει το πλεονέκτημα της αυτόματης κλιμάκωσης, διασφαλίζοντας την αποδοτική επικοινωνία σε πολύ μεγαλύτερες κλίμακες.

6.3.3 Ευχρηστία και Καμπύλη Εκμάθησης

Το **Netcode for GameObjects** είναι συνήθως πιο εύκολο στην εκμάθηση, ειδικά για αρχάριους, λόγω της απλότητας του και της πλήρους ενσωμάτωσης με την Unity. Οι developers που έχουν εμπειρία στην Unity θα βρουν την εφαρμογή του **Netcode** σχετικά εύκολη και γρήγορη για χρήση, καθώς συνδέεται άμεσα με τα εργαλεία που ήδη γνωρίζουν. Η καμπύλη εκμάθησης είναι χαμηλότερη, αλλά ταυτόχρονα, για πιο απαιτητικά παιχνίδια, οι δυνατότητες του **Netcode** μπορεί να είναι περιορισμένες.

Αντίθετα, το **Photon** είναι πιο περίπλοκο, με μεγαλύτερη καμπύλη εκμάθησης, ειδικά αν

κάποιος χρησιμοποιεί υπηρεσίες cloud ή αν χρειάζεται να επεξεργαστεί πιο προηγμένα σενάρια δικτύωσης. Παρόλο που παρέχει εκτενή τεκμηρίωση και υποστήριξη, η εγκατάσταση του Photon και η διαχείριση του multiplayer απαιτεί περισσότερο χρόνο και προηγούμενη εμπειρία. Ωστόσο, για πιο εξελιγμένα παιχνίδια που απαιτούν υψηλή απόδοση και μεγάλο αριθμό παικτών, η ευχέρεια που παρέχει το Photon αποζημιώνει τη μεγαλύτερη καμπύλη εκμάθησης.

6.3.4 Υποστήριξη Cloud και Σχέδιο Κλιμάκωσης

Το **Photon** υπερέρχει με την υποστήριξη του cloud, προσφέροντας ενσωματωμένες υπηρεσίες για τη διαχείριση του δικτύου σε πραγματικό χρόνο. Η υπηρεσία cloud του Photon είναι εξαιρετική για παιχνίδια που απαιτούν παγκόσμια κλίμακα και γεωγραφική κατανομή των servers, χωρίς να χρειάζεται η ανάπτυξη και συντήρηση της υποδομής από τον developer. Αν το παιχνίδι απαιτεί σύνδεση με πολλούς παίκτες από διαφορετικές περιοχές, το Photon αναλαμβάνει τη σύνδεση των παικτών στον πιο κοντινό server, εξασφαλίζοντας καλύτερη απόδοση και χαμηλή καθυστέρηση.

Το **Netcode for GameObjects**, από την άλλη, δεν έχει έτοιμη υποδομή cloud, και απαιτεί από τους developers να δημιουργήσουν ή να χρησιμοποιήσουν εξωτερικούς server providers, όπως το Unity Multiplayer ή άλλες υπηρεσίες, για να καλύψουν τις ανάγκες κλιμάκωσης και συγχρονισμού.

6.3.5 Κοινότητα και Υποστήριξη

Η **Unity** προσφέρει καλή υποστήριξη για το **Netcode for GameObjects** μέσω του επίσημου φόρουμ, της τεκμηρίωσης και των μαθημάτων που είναι ενσωματωμένα στην πλατφόρμα. Το γεγονός ότι είναι ενσωματωμένο στη Unity σημαίνει ότι υπάρχουν αρκετές πηγές υποστήριξης και κοινότητα για να βοηθήσουν στην ανάπτυξη.

Το **Photon**, λόγω της δημοτικότητας του και της διεθνούς κοινότητας, παρέχει μια πιο εκτενή υποστήριξη, με περισσότερες πηγές και φόρουμ από διάφορους προγραμματιστές. Το Photon διαθέτει εξαιρετική τεκμηρίωση, καθώς και εκπαιδευτικά εργαλεία και tutorials για developers που θέλουν να αξιοποιήσουν πλήρως τις δυνατότητες του.

6.3.6 Συμπέρασμα και Επιλογή Εργαλείου

Η τελική επιλογή μεταξύ του **Netcode for GameObjects** και του **Photon** εξαρτάται από τις ανάγκες του κάθε παιχνιδιού. Εάν ο στόχος είναι να δημιουργηθεί ένα μικρό ή μεσαίο multiplayer παιχνίδι, όπου οι ανάγκες κλιμάκωσης δεν είναι πολύ απαιτητικές, το **Netcode for GameObjects** είναι η πιο εύκολη και γρήγορη λύση. Είναι ιδανικό για απλά παιχνίδια που ενσωματώνονται εύκολα στο οικοσύστημα της Unity και δεν απαιτούν πολύπλοκες υποδομές ή εξωτερικές υπηρεσίες.

Ωστόσο, αν το παιχνίδι απαιτεί μεγάλη κλίμακα, προηγμένα χαρακτηριστικά δικτύωσης και υποστήριξη για μεγάλο αριθμό παικτών, το **Photon** είναι η καλύτερη επιλογή. Ειδικά για παιχνίδια που απαιτούν εξαιρετική απόδοση, γεωγραφική κατανομή και ταχύτητα σύνδεσης, το **Photon** παρέχει πιο ισχυρές δυνατότητες και ευχέρεια, καθιστώντας το ιδανικό για παιχνίδια με υψηλές απαιτήσεις.

7. Εξέλιξη και μελλοντικές αλλαγές

Η εξέλιξη ενός παιχνιδιού είναι κρίσιμη για τη διατήρηση της ενδιαφέροντος των παικτών και την προσέλκυση νέων παικτών. Η συνεχιζόμενη ανανέωση, η προσθήκη νέων χαρακτηριστικών και η βελτίωση του παιχνιδιού είναι καθοριστικές για την εξέλιξή του και για τη διατήρηση της ανταγωνιστικότητάς του στην αγορά. Ακολουθεί μια εκτενέστερη ανάλυση των τομέων που απαιτούν βελτίωση και εξέλιξη για το παιχνίδι στο μέλλον.

7.1 Δημιουργία Καινούργιων Χαρτών και Περιβαλλόντων

Η προσθήκη νέων χαρτών αποτελεί τη βασική στρατηγική για να διατηρηθεί η ποικιλία και το ενδιαφέρον στο παιχνίδι. Οι χάρτες πρέπει να είναι δυναμικοί και να προσαρμόζονται στις στρατηγικές του παίκτη, προσφέροντας νέες δυνατότητες και περιοχές που θα αλληλεπιδρούν με τον τρόπο παιχνιδιού. Ορισμένοι χάρτες θα μπορούσαν να προσφέρουν νέες μηχανικές, όπως ειδικές περιοχές που επηρεάζουν τη ροή του παιχνιδιού, περιοχές με κινδύνους ή περιοχές που αυξάνουν τα έσοδα των παικτών. Η δυνατότητα προσαρμογής του χάρτη ανάλογα με την πρόοδο του παιχνιδιού ή ακόμη και την εποχή του έτους θα μπορούσε να προσδώσει νέες στρατηγικές δυνατότητες. Ο χάρτης πρέπει να ενσωματώνει επίσης στοιχεία αλληλεπίδρασης με το περιβάλλον (π.χ., εφέ καιρού ή φυσικά φαινόμενα) που να επηρεάζουν τη στρατηγική των παικτών.

7.2 Αναβάθμιση της Αλληλεπίδρασης Μεταξύ των Παικτών

Η αλληλεπίδραση μεταξύ των παικτών είναι ένας τομέας που απαιτεί συνεχή βελτίωση, προκειμένου να διατηρηθεί η κοινωνική διάσταση του παιχνιδιού. Η δυνατότητα για πιο σύνθετες συνεργασίες, τη δημιουργία συμμαχιών ή την επίλυση διαφορών με διάφορους τρόπους θα εμπλουτίσει την εμπειρία του παιχνιδιού. Επιπλέον, η εισαγωγή εργαλείων που επιτρέπουν στους παίκτες να επικοινωνούν καλύτερα, είτε μέσω φωνητικής συνομιλίας είτε μέσω μηνυμάτων, είναι ένας τρόπος να διευκολυνθεί η στρατηγική επικοινωνία μεταξύ τους. Στη συνέχεια, η εφαρμογή δυναμικών συστημάτων ανταγωνισμού και συνεργασίας (όπως μεταβλητές συμφωνίες ή καταστάσεις που επιτρέπουν ή περιορίζουν τις συμμαχίες) θα ενισχύσουν το κοινωνικό και στρατηγικό κομμάτι του παιχνιδιού.

7.3 Βελτίωση του User Interface (UI)

Ένα φιλικό και ελκυστικό UI είναι κρίσιμο για την απόλαυση των παικτών. Σήμερα, τα παιχνίδια χρειάζονται UI που να είναι όχι μόνο λειτουργικό, αλλά και οπτικά ευχάριστο και εύχρηστο. Ειδικότερα, στο πλαίσιο ενός παιχνιδιού με στρατηγική και διαχείριση πόρων, είναι σημαντικό το UI να προσφέρει γρήγορη και εύκολη πλοήγηση. Η αναβάθμιση του UI μπορεί να περιλαμβάνει πιο μοντέρνα γραφικά, βελτιστοποίηση του layout και τη δημιουργία μιας πιο διαισθητικής πλοήγησης. Η δυνατότητα για προσαρμογή του UI με βάση τις προτιμήσεις του χρήστη και την πρόοδό του στο παιχνίδι (όπως η εμφάνιση νέων πληροφοριών καθώς προχωράει) θα καταστήσει την εμπειρία πιο ελκυστική και πιο ευχάριστη για τον χρήστη.

7.4 Προσαρμοστικότητα και Δυναμική Αντίδραση στις Στρατηγικές του Παίκτη

Η αναβάθμιση της δυναμικής του παιχνιδιού είναι ένα ακόμα σημείο προς βελτίωση. Η προσθήκη μηχανισμών που προσαρμόζονται στις στρατηγικές των παικτών ή την απόδοσή τους μπορεί να ενισχύσει την ένταση και την απόλαυση του παιχνιδιού. Η αλλαγή στο επίπεδο δυσκολίας, η ενσωμάτωση έξυπνης τεχνητής νοημοσύνης που θα ακολουθεί ή θα αντιδρά στις στρατηγικές των παικτών, ή ακόμα και προσαρμοσμένα εφέ και τρόποι παιχνιδιού ανάλογα με την πρόοδο των παικτών, θα ενισχύσουν την αίσθηση της πρόκλησης και θα διατηρήσουν το ενδιαφέρον. Αυτό μπορεί να περιλαμβάνει επίσης την εισαγωγή νέων μηχανισμών διαχείρισης πόρων ή νέες στρατηγικές επιλογές που επιτρέπουν στους παίκτες να παίξουν με διαφορετικούς τρόπους κάθε φορά.

7.5 Διαχείριση Εξωτερικών Πηγών και Διαδικασιών

Η διαχείριση και η εισαγωγή εξωτερικών παραμέτρων που επηρεάζουν το παιχνίδι είναι μια νέα τάση στον σχεδιασμό παιχνιδιών. Αυτές οι παράμετροι μπορεί να περιλαμβάνουν πραγματικούς καιρούς (όπως τον καιρό ή εποχιακές αλλαγές), γεγονότα που να επηρεάζουν την στρατηγική του παιχνιδιού, ή ακόμα και τη σύνδεση με άλλες πηγές όπως εφαρμογές κινητών ή μέσα κοινωνικής δικτύωσης. Η ενσωμάτωση τέτοιων συστημάτων θα προσφέρει στους παίκτες μια πιο εκτεταμένη και διαδραστική εμπειρία, κάνοντάς τους να συνδέονται με το παιχνίδι σε καθημερινή βάση.

7.6 Αναβάθμιση Στρατηγικών Στοιχείων και Αντικειμένων

Η ανάπτυξη νέων στρατηγικών εργαλείων, αντικειμένων και στοιχείων που μπορούν να χρησιμοποιηθούν για να επηρεάσουν τη ροή του παιχνιδιού είναι εξαιρετικά σημαντική. Η ένταξη νέων αντικειμένων, περιοχών και στοιχείων που επηρεάζουν την οικονομία του παιχνιδιού ή τις επιλογές στρατηγικής θα διατηρήσει την πρόκληση και θα προσφέρει νέα σενάρια για τους παίκτες. Για παράδειγμα, η προσθήκη νέων ειδικών αντικειμένων που παρέχουν μοναδικά πλεονεκτήματα ή η αναβάθμιση περιοχών του χάρτη που δημιουργούν νέες στρατηγικές αλληλεπιδράσεις μπορεί να αναζωογονήσει το παιχνίδι και να αυξήσει την ανταγωνιστικότητα.

8. Προβλήματα που πρόεκυψαν

Όπως είναι αναμενόμενο σε κάθε ανάπτυξη λογισμικού, ιδιαίτερα σε ένα multiplayer παιχνίδι, προκύπτουν διάφορα προβλήματα κατά τη διάρκεια της δημιουργίας και της δοκιμής. Ένα από τα πιο κοινά και σημαντικά ζητήματα που αντιμετωπίστηκαν κατά τη διάρκεια της ανάπτυξης του παιχνιδιού ήταν ο συγχρονισμός των δεδομένων μεταξύ των παικτών. Επιπλέον, υπήρξαν διάφορα προβλήματα που σχετίζονται με την αγορά περιοχών και την ανταλλαγή αντικειμένων ή χρημάτων μεταξύ των παικτών.

8.1 Μη συγχρονισμός interface

Ένα συχνό πρόβλημα κατά την ανάπτυξη multiplayer παιχνιδιών είναι ότι το interface δεν μπορεί να εκτελέσει κώδικα που προέρχεται από Server και Client RPC. Αυτό μπορεί να προκαλέσει δυσλειτουργίες, καθώς οι εντολές που αποστέλλονται από τον server προς τους clients ή αντίστροφα δεν εκτελούνται σωστά. Για να επιλυθεί αυτό το πρόβλημα, πραγματοποιήθηκε μια αναζήτηση στο διαδίκτυο και στο επίσημο documentation της βιβλιοθήκης. Από τα αποτελέσματα, προέκυψε ότι για να διασφαλιστεί η σωστή επικοινωνία και εκτέλεση του RPC κώδικα, είναι απαραίτητο το κάθε interface να έχει συσχετισμένο το **NetworkObject Script** και να κληρονομεί την κλάση **NetworkBehavior**.

Η προσθήκη του **NetworkObject Script** σε κάθε interface είναι κρίσιμη, διότι επιτρέπει στο συγκεκριμένο αντικείμενο να συγχρονίζεται σωστά με το δίκτυο, και να αναγνωρίζεται από το σύστημα δικτύωσης. Χωρίς αυτό το script, οι εντολές RPC δεν θα είχαν τρόπο να "προσεγγίσουν" το αντικείμενο, με αποτέλεσμα να μην εκτελούνται. Η κληρονομικότητα από την κλάση **NetworkBehavior** εξασφαλίζει επίσης ότι το interface θα ακολουθεί τις απαραίτητες διαδικασίες δικτύωσης και θα έχει πρόσβαση στις λειτουργίες δικτύωσης που είναι αναγκαίες για την σωστή εκτέλεση των RPC κλήσεων.

Αυτές οι δυο αλλαγές - το **NetworkObject Script** και η κληρονομικότητα από το **NetworkBehavior** - επιτρέπουν στο interface να αποκτήσει τη σωστή υποδομή δικτύωσης και να μπορεί να εκτελεί τις εντολές που προέρχονται από τον server ή τον client. Όταν εφαρμοστούν, οι RPC κλήσεις εκτελούνται σωστά, οι επικοινωνίες μεταξύ των παικτών συγχρονίζονται αποτελεσματικά, και η εμπειρία του χρήστη βελτιώνεται σημαντικά.

8.2 Αγορά Περιοχών

Η αγορά περιοχών αποτέλεσε ένα ακόμη πρόβλημα κατά την ανάπτυξη του παιχνιδιού. Όταν ένας παίκτης αγόραζε μια περιοχή, αυτή η περιοχή κατέληγε πάντα υπό την κατοχή του παίκτη-**host**. Αυτό το πρόβλημα προέκυπτε λόγω του τρόπου που πραγματοποιούνταν η διαδικασία που ανέθετε τον ιδιοκτήτη σε μια περιοχή, δηλαδή το σύστημα δικτύωσης δεν μπορούσε να αντιληφθεί σωστά ποιος παίκτης ήταν ο πραγματικός αγοραστής.

Για την επίλυση αυτού του ζητήματος, έγινε μια σημαντική τροποποίηση στο σύστημα του server. Πλέον, μέσα στον server, όταν ένας παίκτης αγοράζει μια περιοχή, το πλήρες αντικείμενο της περιοχής καταχωρείται και συνδέεται με τον παίκτη που πραγματικά αγόρασε την περιοχή. Αυτό διασφαλίζει ότι όλα τα δεδομένα που σχετίζονται με την περιοχή, όπως το όνομα του ιδιοκτήτη, το ενοίκιο και οι πιθανές αναβαθμίσεις, είναι σωστά και αποθηκεύονται στο σωστό αντικείμενο του αγοραστή.

Με αυτή την αλλαγή, αποφεύχθηκαν περιπτώσεις όπου η περιοχή ανήκε λανθασμένα στον host και εξασφαλίστηκε ότι η ιδιοκτησία μιας περιοχής αναγνωρίζεται σωστά για κάθε παίκτη, ανεξάρτητα από το αν είναι ο host ή όχι. Η διαδικασία έγινε πιο αξιόπιστη και επέτρεψε στους παίκτες να αποκτούν περιοχές με την βεβαιότητα ότι θα τις κατέχουν σωστά, ενώ παράλληλα διασφαλίστηκε η σωστή συγχρονισμένη ενημέρωση αυτών των δεδομένων στους άλλους πελάτες του παιχνιδιού.

8.3 Ανταλλαγή

Κατά την διαδικασία της ανταλλαγής χρηματικού ποσού, η εφαρμογή αντιμετώπιζε ένα σοβαρό πρόβλημα που προκαλούσε την εφαρμογή να εισέλθει σε κατάσταση «infinite loop», δηλαδή να κολλήσει σε μια ατέρμονη επανάληψη χωρίς να καταφέρει να ολοκληρώσει την διαδικασία. Αυτό το πρόβλημα παρουσιαζόταν επειδή στην πρώτη εκδοχή του συστήματος, όταν ένα αντικείμενο είχε ανταλλαχθεί επιτυχώς, διαγραφόταν από το σύστημα, κάτι που οδηγούσε τον αλγόριθμο να επανεξετάσει τα δεδομένα και να ξεκινήσει εκ νέου τη διαδικασία ανταλλαγής, χωρίς να φτάνει σε καμία ολοκλήρωση.

Για την επίλυση αυτού του ζητήματος, τροποποιήθηκε η διαδικασία ανταλλαγής. Πλέον, αντί τα αντικείμενα που έχουν ανταλλαχθεί να διαγράφονται, σημαίνονται ως «ανταλλαγμένα». Αυτό

επιτρέπει στο σύστημα να τα αγνοεί στις επόμενες επαναλήψεις του αλγορίθμου και να προχωρά στην επεξεργασία των υπόλοιπων στοιχείων της ανταλλαγής. Έτσι, αποφεύγεται η συνεχιζόμενη επανεξέταση και επανάληψη της ίδιας διαδικασίας, επιλύοντας το πρόβλημα του infinite loop.

Αυτή η τροποποίηση βελτίωσε την σταθερότητα της εφαρμογής και εξασφάλισε ότι η ανταλλαγή χρημάτων και αντικειμένων ολοκληρώνεται με τον σωστό τρόπο χωρίς να μπλοκάρεται σε ατέρμονες επαναλήψεις. Το πρόγραμμα τώρα διαχειρίζεται αποτελεσματικά τις διαδικασίες ανταλλαγής και η ροή του παιχνιδιού συνεχίζεται ομαλά.

8.4 Μη έναρξη του παιχνιδιού

Μερικές φορές όταν ο ιδιοκτήτης ενός lobby ξεκάνει το παιχνίδι, το παιχνίδι δεν ξεκινάει για κάποιους παίκτες. Ενώ έγιναν αρκετές προσπάθειες για την εύρεση και επίλυση του προβλήματος, εξαιτίας της ασυνέπειας του, δεν κατάφερα να εντοπίσω που είναι το πρόβλημα.

9. Αλλαγές από την original monopoly

Με βάση την έρευνα που διεξήχθη μεταξύ γνωστών και φίλων, πραγματοποιήθηκαν σημαντικές αλλαγές για τη βελτίωση της εμπειρίας του παιχνιδιού, λαμβάνοντας υπόψη τα σχόλια των παικτών. Ένα από τα κύρια θέματα δυσaráεσκείας που αναδείχθηκαν ήταν ο μηχανισμός της χρεοκοπίας και των υποθηκών. Συγκεκριμένα, όταν ένας παίκτης ξεμένει από χρήματα και χρειάζοταν να πληρώσει κάποιο ποσό, ήταν υποχρεωμένος να βάλει μια από τις περιοχές του σε υποθήκη. Αν αυτό δεν ήταν αρκετό για να καλύψει το οφειλόμενο ποσό, τότε ο παίκτης αναγκάζονταν να δηλώσει χρεοκοπία και να αποχωρήσει από το παιχνίδι.

Οι ερωτηθέντες εξέφρασαν την άποψη ότι αυτός ο μηχανισμός ήταν υπερβολικά περιοριστικός και στερούσε τη διασκέδαση, καθώς οδηγούσε γρήγορα κάποιους παίκτες στην έξοδο από το παιχνίδι χωρίς ευκαιρία ανάκαμψης. Επιπλέον, ανέδειξαν ότι λόγω του συστήματος χρεοκοπίας, η διάρκεια του παιχνιδιού ήταν εξαιρετικά μεγάλη. Για να υπάρξει νικητής, έπρεπε να χρεοκοπήσουν όλοι οι υπόλοιποι παίκτες, γεγονός που κατέληγε να είναι κουραστικό, ειδικά σε παιχνίδια με πολλούς συμμετέχοντες.

9.1 Μεταφορά σε υπολογιστή

Η μεταφορά ενός φυσικού επιτραπέζιου παιχνιδιού σε μια online μορφή προσφέρει πληθώρα δυνατοτήτων για βελτιώσεις και καινοτομίες που αναβαθμίζουν την εμπειρία του παίκτη. Ο ψηφιακός κόσμος ανοίγει τον δρόμο για τη δημιουργία μηχανισμών που είναι είτε υπερβολικά περίπλοκοι είτε πρακτικά αδύνατο να εφαρμοστούν σε ένα φυσικό περιβάλλον παιχνιδιού. Αυτή η εξέλιξη όχι μόνο κάνει το παιχνίδι πιο πλούσιο, αλλά ενισχύει και τη στρατηγική, την αλληλεπίδραση και τη δυναμική του.

Στο φυσικό παιχνίδι, οι περιορισμοί του χρόνου, της ακρίβειας και των χειροκίνητων υπολογισμών περιορίζουν την πολυπλοκότητα των μηχανισμών. Για παράδειγμα, η παρακολούθηση οικονομικών στοιχείων, η εφαρμογή ειδικών κανόνων ή η εισαγωγή δυναμικών γεγονότων μπορεί να είναι εξαιρετικά χρονοβόρα και επιρρεπή σε λάθη. Αντιθέτως, στην online εκδοχή, ο υπολογιστής αναλαμβάνει να αυτοματοποιήσει αυτές τις διαδικασίες, εξαλείφοντας την ανάγκη για χειροκίνητες ενέργειες και διασφαλίζοντας τη συνεπή εφαρμογή των κανόνων.

Μια άλλη σημαντική πτυχή είναι η δυνατότητα συνεχούς ενημέρωσης και παραμετροποίησης του παιχνιδιού. Οι προγραμματιστές μπορούν εύκολα να προσθέσουν νέους μηχανισμούς, όπως δυναμικές οικονομίες, τοκισμούς ή ειδικές αποστολές, οι οποίες εμπλουτίζουν το gameplay. Ταυτόχρονα, οι παίκτες έχουν την ευκαιρία να προσαρμόσουν το παιχνίδι στις προτιμήσεις τους, κάτι που στο φυσικό παιχνίδι απαιτεί χρονοβόρες αλλαγές στους κανόνες ή στον εξοπλισμό.

9.1.1 Τόκοι

Ένας από τους μηχανισμούς που αναδεικνύει τις δυνατότητες της ψηφιακής μορφής ενός παιχνιδιού είναι η εφαρμογή τόκων. Στο online περιβάλλον, ο υπολογιστής μπορεί να παρακολουθεί αυτόματα τα χρηματικά δεδομένα κάθε παίκτη και να εφαρμόζει τους τόκους σε πραγματικό χρόνο, εξαλείφοντας την ανάγκη για χειροκίνητους υπολογισμούς. Αυτό καθιστά τη διαδικασία απλή, ακριβή και αδιάλειπτη, βελτιώνοντας την εμπειρία του παιχνιδιού. Σε αντίθεση, σε ένα φυσικό επιτραπέζιο παιχνίδι, η εφαρμογή τόκων θα ήταν μια περίπλοκη και χρονοβόρα διαδικασία. Κάθε παίκτης θα έπρεπε να μετρά τα χρήματά του σε κάθε γύρο, να υπολογίζει το επιτόκιο και να διαχειρίζεται την προσθήκη του ποσού, γεγονός που θα επιβράδυνε σημαντικά τον ρυθμό του παιχνιδιού. Επιπλέον, τέτοιες χειροκίνητες διαδικασίες είναι επιρρεπείς σε λάθη, κάτι που θα μπορούσε να οδηγήσει σε παρεξηγήσεις ή να μειώσει την ευχαρίστηση των παικτών.

Η εισαγωγή τόκων στο ψηφιακό παιχνίδι δεν είναι απλώς λειτουργική βελτίωση, αλλά και στρατηγική πρόκληση. Οι παίκτες καλούνται να διαχειρίζονται τον πλούτο τους με στόχο να αξιοποιήσουν στο έπακρο τους τόκους. Αυτό σημαίνει ότι η στρατηγική δεν περιορίζεται μόνο στην απόκτηση περιοχών ή στον έλεγχο του ταμπλό, αλλά επεκτείνεται και στη μακροχρόνια οικονομική τους πολιτική. Οι παίκτες πρέπει να αποφασίζουν πότε είναι σκόπιμο να επενδύσουν ή να διατηρήσουν ρευστότητα, γνωρίζοντας ότι οι τόκοι μπορούν να ενισχύσουν τη χρηματοοικονομική τους κατάσταση ή να δημιουργήσουν νέες ευκαιρίες.

Αυτός ο μηχανισμός προσθέτει βάθος και περιπλοκότητα στο παιχνίδι, κάνοντάς το πιο ενδιαφέρον και απαιτητικό. Οι παίκτες πρέπει να ισορροπούν μεταξύ βραχυπρόθεσμων κινήσεων και μακροπρόθεσμων στόχων, δημιουργώντας έναν διαρκή ανταγωνισμό που διατηρεί το ενδιαφέρον τους. Επομένως, η εφαρμογή των τόκων στο ψηφιακό παιχνίδι δεν είναι απλώς μια τεχνολογική διευκόλυνση, αλλά και μια στρατηγική αναβάθμιση που ενισχύει τη

συνολική εμπειρία.

9.1.2 Αύξηση κοστών κατά την πορεία του παιχνιδιού

Η μεταφορά ενός παιχνιδιού σε υπολογιστή προσφέρει την ευκαιρία για δυναμική προσαρμογή στοιχείων όπως το κόστος των περιοχών και οι ανταμοιβές των παικτών, ενισχύοντας τη μακροχρόνια στρατηγική και τη διατήρηση του ενδιαφέροντος. Σε ένα παραδοσιακό επιτραπέζιο παιχνίδι, οι τιμές των περιοχών και τα ποσά που λαμβάνουν οι παίκτες είναι σταθερά, κάτι που συχνά οδηγεί σε απώλεια της σχετικής τους αξίας καθώς το παιχνίδι προχωρά. Οι παίκτες συσσωρεύουν περισσότερα χρήματα, καθιστώντας τις πρώιμες περιοχές και ανταμοιβές λιγότερο ουσιαστικές και μειώνοντας τη στρατηγική σημασία τους.

Στην ψηφιακή έκδοση, ωστόσο, είναι δυνατή η δυναμική προσαρμογή αυτών των στοιχείων. Το κόστος των περιοχών, καθώς και οι ανταμοιβές που λαμβάνουν οι παίκτες, μπορούν να αυξάνονται σταδιακά όσο το παιχνίδι προχωρά. Αυτό διασφαλίζει ότι ακόμα και στα τελικά στάδια του παιχνιδιού, οι περιοχές παραμένουν κρίσιμες για τη στρατηγική των παικτών. Η αφετηρία, για παράδειγμα, διατηρεί τη σημασία της ως πηγή εισοδήματος, και οι παίκτες εξακολουθούν να λαμβάνουν αποφάσεις σχετικά με το πώς να διαχειριστούν τα αυξημένα ποσά.

Αυτή η δυναμική προσέγγιση ενισχύει τη στρατηγική εμπειρία, καθώς οι παίκτες πρέπει να προσαρμόζονται στις μεταβαλλόμενες συνθήκες του παιχνιδιού. Η αύξηση του κόστους των περιοχών απαιτεί πιο προσεκτικό σχεδιασμό για την απόκτησή τους, ενώ οι αυξημένες ανταμοιβές επιτρέπουν μεγαλύτερες επενδύσεις και ρίσκα. Έτσι, το παιχνίδι παραμένει προκλητικό και ενδιαφέρον, ανεξάρτητα από τη διάρκειά του ή τη φάση στην οποία βρίσκονται οι παίκτες.

9.1.3 Networth

Η μεταφορά του παιχνιδιού στον υπολογιστή επιτρέπει την αντικατάσταση του παραδοσιακού συστήματος χρεοκοπίας και υποθηκών με έναν πιο ευέλικτο και ισορροπημένο τρόπο καθορισμού του νικητή. Με την αυτοματοποίηση των υπολογισμών από τον υπολογιστή, διαδικασίες όπως η αγοραπωλησία περιοχών, οι αναβαθμίσεις και η διαχείριση οικονομικών στοιχείων γίνονται ακριβείς και άμεσες, επιτρέποντας την παρακολούθηση και ενημέρωση του networth κάθε παίκτη σε πραγματικό χρόνο.

Το networth, το οποίο συνυπολογίζει τόσο τα μετρητά όσο και την αξία των περιουσιακών στοιχείων ενός παίκτη, αντικαθιστά την παραδοσιακή έννοια της χρεοκοπίας. Στην περίπτωση που ένας παίκτης εξαντλήσει τα χρήματά του, το σύστημα δεν απαιτεί πλέον την πώληση περιοχών ή την τοποθέτησή τους σε υποθήκη για να συνεχίσει το παιχνίδι. Αντίθετα, η απώλεια χρημάτων αντανάκλαται αρνητικά στο networth του, χωρίς να αφαιρείται η δυνατότητα συμμετοχής του στο παιχνίδι. Παράλληλα, τα χρέη που μεταφέρονται στους αντιπάλους αυξάνουν το δικό τους networth, δημιουργώντας ένα δυναμικό σύστημα που διατηρεί το ενδιαφέρον και τις στρατηγικές επιλογές.

Αυτή η προσέγγιση εξαλείφει την απογοήτευση που προκαλείται από τη σταδιακή εξόντωση των παικτών μέσω της χρεοκοπίας, καθώς τους επιτρέπει να συνεχίζουν να συμμετέχουν ενεργά στο παιχνίδι ανεξάρτητα από τη χρηματική τους κατάσταση. Ταυτόχρονα, το networth γίνεται ο βασικός δείκτης απόδοσης, επιτρέποντας στους παίκτες να ανταγωνίζονται για τη συνολική τους οικονομική ισχύ αντί να επιδιώκουν απλώς να εξουδετερώσουν τους αντιπάλους τους. Αυτό το σύστημα προσφέρει επίσης μεγαλύτερη ευελιξία στον καθορισμό του νικητή. Για παράδειγμα, το παιχνίδι μπορεί να ολοκληρωθεί όταν φτάσει ένας συγκεκριμένος αριθμός γύρων, με νικητή τον παίκτη με το υψηλότερο networth, μειώνοντας σημαντικά τη διάρκεια του παιχνιδιού και αποφεύγοντας την εξαντλητική διαδικασία που απαιτεί την πλήρη εξάλειψη όλων των άλλων παικτών.

Συμπεράσματα

Η δημιουργία ενός multiplayer παιχνιδιού είναι μια εξαιρετικά απαιτητική διαδικασία που προϋποθέτει όχι μόνο την ανάπτυξη της βασικής μηχανής του παιχνιδιού, αλλά και την εφαρμογή ενός αποτελεσματικού δικτύου που να υποστηρίζει την αλληλεπίδραση των παικτών σε πραγματικό χρόνο. Το κυριότερο ζήτημα που πρέπει να αντιμετωπιστεί είναι ο συγχρονισμός των δεδομένων μεταξύ όλων των παικτών, ώστε να διασφαλιστεί ότι όλοι βλέπουν το ίδιο περιβάλλον και τις ίδιες ενέργειες, χωρίς καθυστερήσεις ή αποκλίσεις. Ο συγχρονισμός περιλαμβάνει τη διαχείριση όλων των κινήσεων των παικτών, της κατάστασης του παιχνιδιού και της επικοινωνίας των δεδομένων μεταξύ του server και των clients. Όσο μεγαλύτερη είναι η αρίθμηση των παικτών, τόσο πιο περίπλοκο και απαιτητικό γίνεται το δίκτυο, με αυξημένες ανάγκες σε bandwidth και latency.

Η επιλογή του σωστού δικτύου ή της κατάλληλης βιβλιοθήκης δικτύωσης είναι καθοριστική για την ανάπτυξη του παιχνιδιού. Κάθε λύση δικτύωσης (όπως το Unity Netcode, Photon ή Mirror) έχει τα δικά της πλεονεκτήματα και αδυναμίες και πρέπει να επιλεγεί με βάση τις ανάγκες του παιχνιδιού, τη σύνθεσή του και το πλήθος των παικτών. Για παράδειγμα, το Photon μπορεί να είναι πιο εύχρηστο για μικρές και μεσαίου μεγέθους εφαρμογές, ενώ το Mirror μπορεί να προσφέρει μεγαλύτερη ευελιξία και κλίμακα για μεγαλύτερα παιχνίδια. Επιπλέον, οι λύσεις που στηρίζονται στο client-server μοντέλο, όπως το Netcode, προσφέρουν μεγαλύτερο έλεγχο και ασφάλεια, αλλά απαιτούν πιο απαιτητικούς μηχανισμούς για την αποφυγή σφαλμάτων συγχρονισμού και hacking. Κάθε λύση απαιτεί προσεκτική διαχείριση της επικοινωνίας μεταξύ των παικτών, ώστε να αποφευχθούν προβλήματα όπως lag, desynchronization ή ακόμα και "cheating" από κακόβουλους χρήστες.

Επιπλέον, η εμπειρία του χρήστη και η διατήρηση του ενδιαφέροντος των παικτών σε ένα multiplayer παιχνίδι είναι επίσης εξαιρετικά σημαντικά θέματα. Για να παραμείνει το παιχνίδι διασκεδαστικό και ελκυστικό, πρέπει να ενσωματωθούν σωστά μηχανισμοί αλληλεπίδρασης, όπως η δυνατότητα να ανταγωνίζονται ή να συνεργάζονται οι παίκτες, καθώς και να παρέχονται διαρκώς νέες προκλήσεις και ανταμοιβές. Η αποδοτικότητα και η αποτελεσματικότητα της δικτύωσης επηρεάζουν άμεσα την ποιότητα της εμπειρίας του χρήστη, με την ανάγκη για γρήγορη ανταλλαγή δεδομένων να είναι πιο επιτακτική από ποτέ. Όλοι οι παίκτες πρέπει να αισθάνονται ότι το παιχνίδι ανταποκρίνεται άμεσα στις ενέργειές τους, χωρίς να υπάρχουν καθυστερήσεις ή προβλήματα συγχρονισμού. Για να επιτευχθεί αυτό, οι προγραμματιστές πρέπει να παρακολουθούν στενά την απόδοση του δικτύου και να πραγματοποιούν συνεχιζόμενες βελτιώσεις στην υποδομή τους.

Η δημιουργία ενός multiplayer παιχνιδιού είναι επίσης πολύ πιο χρονοβόρα και δύσκολη από τη δημιουργία ενός single-player παιχνιδιού. Η προσθήκη περισσότερων παικτών και η ανάγκη για συγχρονισμό σε πραγματικό χρόνο καθιστούν τη διαδικασία ανάπτυξης πιο απαιτητική και επηρεάζουν σημαντικά τη συνολική απόδοση του παιχνιδιού. Για παράδειγμα, οι pre-emptive συγχρονισμένοι μηχανισμοί, όπως το buffering των δεδομένων για να μειωθούν οι καθυστερήσεις, απαιτούν αρκετή επεξεργαστική ισχύ και μνήμη, αυξάνοντας το κόστος ανάπτυξης και απαιτώντας περισσότερους πόρους. Παρά τις δυσκολίες, η δυνατότητα να προσφερθεί μια εμπειρία πολλών παικτών που μπορεί να συνδεθεί με άλλους παίκτες σε παγκόσμιο επίπεδο προσφέρει έναν ιδιαίτερα ελκυστικό και διαρκώς εξελισσόμενο κόσμο, που ενισχύει την αξία του παιχνιδιού και την αφοσίωση των χρηστών.

Συνολικά, η ανάπτυξη ενός multiplayer παιχνιδιού είναι μια σύνθετη, πολύπλευρη διαδικασία που απαιτεί ισχυρό σχεδιασμό και συνεχιζόμενη βελτίωση, ιδιαίτερα όσον αφορά τη δικτύωση. Το σωστό σύστημα δικτύωσης, ο έλεγχος των δεδομένων και η άριστη εμπειρία χρήστη είναι καθοριστικοί παράγοντες για την επιτυχία ενός τέτοιου παιχνιδιού.

Βιβλιογραφία

Unity Asset Store: <https://assetstore.unity.com/>

Netcode for GameObjects:

<https://docs.unity3d.com/6000.0/Documentation/Manual/com.unity.netcode.gameobjects.html>

Mirror: <https://mirror-networking.gitbook.io/docs>

Photon: <https://doc.photonengine.com/fusion/current/fusion-intro>

FishNet : <https://fish-networking.gitbook.io/docs>

Unity Relay: <https://docs.unity.com/ugs/manual/relay/manual/introduction>

Unity Lobby: <https://docs.unity.com/ugs/manual/lobby/manual/unity-lobby-service>

Netcode Tutorial : <https://www.youtube.com/watch?v=3yuBOB3VrCk&t=3031s>