
ΤΜΗΜΑ ΕΠΙΚΟΙΝΩΝΙΑΣ ΚΑΙ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ
ΣΧΟΛΗ ΚΟΙΝΩΝΙΚΩΝ ΚΑΙ ΑΝΘΡΩΠΙΣΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Ανάπτυξη ενός συνεργατικού 3D παιχνιδιού σε πλατφόρμα
Windows με χρήση της Unity»**

του

Γκίκα Θεοδόση

Επιβλέπων καθηγητής: Επίκουρος καθηγητής Κωνσταντίνος Καρπούζης

Μέλη της Τριμελούς: 1) Αντώνιος Πρωτοψάλτης

2) Μιχαήλ Βρίγκας

Ιούλιος, 2024

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	iii
ΠΕΡΙΛΗΨΗ	vi
ΠΕΡΙΛΗΨΗ ΣΤΑ ΑΓΓΛΙΚΑ	vii
ΕΙΣΑΓΩΓΗ.....	viii
1. Εισαγωγικές έννοιες και ιστορία των βιντεοπαιχνιδιών.....	1
1.1 Τι είναι ένα βιντεοπαιχνίδι	1
1.2 Τι είναι ένα βιντεοπαιχνίδι arcade.....	1
1.3 Τι είναι μια μηχανή παιχνιδιών (game engine)	1
1.4 Τι είναι ένα Game Design Document.....	2
1.4.1 Σχέδιο ανάπτυξης	2
1.4.2 Εργαλείο επικοινωνίας	2
1.4.3 Οδηγός αναφοράς.....	2
1.4.4 Σχεδιασμός και Προϋπολογισμός.....	2
1.4.5 Διαχείριση κινδύνων	2
1.4.6 Παρουσίαση και Μάρκετινγκ.....	2
1.5 Η ιστορία και εξέλιξη των βιντεοπαιχνιδιών.....	3
1.5.1 1950s-1960s: Τα πρώιμα χρόνια	3
1.5.2 1970s-1980s: Η γέννηση των Arcade και η χρυσή εποχή.....	3
1.5.3 1990s: Η άνοδος των τρισδιάστατων παιχνιδιών και κονσολών.....	4
1.5.4 2000s: Η εποχή του διαδικτυακού παιχνιδιού και των χειριστηρίων κίνησης	4
1.5.5 2010s: Η κυριαρχία των παιχνιδιών στα κινητά και η εικονική πραγματικότητα.....	5
1.5.6 2020s: Τα γραφικά Ray-tracing και Cloud Gaming	6
1.6 Η εξέλιξη στην παραγωγή των βιντεοπαιχνιδιών.....	10
1.6.1 Δεκαετία του 1970.....	11
1.6.2 Δεκαετία του 1980-1990	11
1.6.3 Δεκαετία του 2000.....	11
1.6.4 Δεκαετία του 2010-2020	11
2. Unity.....	12
2.1 Unity Software Inc. (Unity Technologies)	12
2.2 Unity (game engine).....	12
2.3 Unity Gaming Services.....	13
2.3.1 Επισκόπηση του Unity Gaming Services	13
2.3.2 Λογαριασμοί (Accounts)	14
2.3.3 Διαχείριση περιεχομένου (Content management)	14
2.3.4 Ανάπτυξη και λειτουργία (DevOps).....	15
2.3.5 Πολλοί παίκτες (Multiplayer).....	15

2.3.6	Αναλυτικά στοιχεία και εμπλοκή παικτών (Analytics and player engagement).....	15
2.3.7	Κοινότητα (Community).....	16
2.3.8	Αναφορά σφαλμάτων (Crash reporting).....	17
2.3.9	Οικονομία παιχνιδιού (Game economy).....	17
2.4	Netcode for GameObjects (NGO).....	17
2.4.1	API υψηλού επιπέδου (High-level API):.....	17
2.4.2	Δικτυωμένα αντικείμενα (Networked Objects):.....	17
2.4.3	Συμπεριφορές δικτύου (Network Behaviors):.....	17
2.4.4	Μεταβλητές δικτύου (Network Variables):.....	18
2.4.5	RPC Κλήσεις απομακρυσμένης διαδικασίας (Remote Procedure Calls):.....	18
2.4.6	Διαχείριση σκηνής (Scene Management):.....	18
2.4.7	Προσαρμογή και Επεκτασιμότητα (Customization and Extensibility):.....	18
2.4.8	Ενσωμάτωση με τις Υπηρεσίες της Unity (Integration with Unity Services):.....	18
3.	Παρουσίαση και ανάλυση του παιχνιδιού.....	19
3.1	Παρουσίαση του παιχνιδιού.....	19
3.2	Game Design Document.....	20
3.3	Εργαλεία που χρησιμοποιήθηκαν.....	23
3.3.1	Unity Editor 2022.....	23
3.3.2	Visual Studio 2022.....	23
3.3.3	Corel Paintshop Pro 2022.....	23
3.3.4	Cool Edit Pro.....	24
3.3.5	Blender 3.6 LTS.....	24
3.3.6	Trello.....	24
3.3.7	LOGO.....	25
3.3.8	OBS Studio.....	25
3.3.9	Εργαλείο δημιουργίας εικονιδίων (IconMaker & RecipeMaker).....	26
3.4	Στοιχεία παιχνιδιού (Game Assets).....	28
3.4.1	Δωρεάν τρισδιάστατα μοντέλα.....	28
3.4.2	Στοιχεία παιχνιδιού από το Unity Asset Store.....	29
3.4.3	Στοιχεία παιχνιδιού που δημιουργήθηκαν.....	29
3.5	Διαδικασία που ακολουθήθηκε για την δημιουργία του παιχνιδιού.....	30
3.5.1	Σύλληψη ιδέας.....	30
3.5.2	Έναρξη διπλωματικής εργασίας.....	31
3.5.3	Εξέλιξη πέρα από τα μαθήματα.....	31
3.5.4	Λειτουργία για πολλούς παίκτες στον ίδιο υπολογιστή.....	32
3.5.5	Εισαγωγή πελατών.....	32
3.5.6	Εισαγωγή νέων μηχανισμών μαγειρέματος.....	33
3.5.7	Εισαγωγή νέων πιάτων φαγητού (συνταγές).....	35

3.5.8	Οθόνη (Σκηνή) επιλογής χαρακτήρα.....	35
3.5.9	Ειδικότητες χαρακτήρων.....	36
3.5.10	Αναβάθμιση μαγαζιού.....	37
3.5.11	Ρύθμιση δυσκολίας του παιχνιδιού.....	39
3.5.12	Μελλοντικά σχέδια.....	40
3.6	Σύνδεση μέσω διαδικτύου.....	41
3.6.1	Χρήση των Gaming Services της Unity.....	41
3.6.2	Relay και Lobby.....	42
3.6.3	Συγχρονισμός αντικειμένων με τη χρήση NGO.....	42
4.	Συμπεράσματα.....	46
	Βιβλιογραφία.....	47
	Παράρτημα.....	49
	Δήλωση Πνευματικών Δικαιωμάτων.....	52

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έχει ως θέμα την υλοποίηση ενός τρισδιάστατου συνεργατικού πραγματικού χρόνου βιντεοπαιχνίδι σε πλατφόρμα Windows (H/Y) με τη χρήση της μηχανής παιχνιδιών Unity. Το συγκεκριμένο βιντεοπαιχνίδι υποστηρίζει έναν ή περισσότερους παίκτες τόσο σε τοπικό (στον ίδιο H/Y) όσο και σε διαδικτυακό επίπεδο. Η υλοποίηση του διαδικτυακού κομματιού πραγματοποιήθηκε με τη χρήση της βιβλιοθήκης “Netcode for GameObjects” της Unity. Ενώ η διασύνδεση μέσω Ίντερνετ πραγματοποιήθηκε με τη χρήση των Relay και Lobby των Gaming Services της Unity. Η εργασία χωρίζεται σε τρία κεφάλαια. Στο πρώτο κεφάλαιο θα αναφερθούν βασικές έννοιες βιντεοπαιχνιδιών, καθώς και θα γίνει μια σύντομη αναδρομή στην ιστορία και στην εξέλιξη τους. Στο δεύτερο κεφάλαιο θα πραγματοποιηθεί ανάλυση της μηχανής παιχνιδιών Unity και στις υπηρεσίες παιχνιδιού (Gaming Services) που παρέχει. Στο τρίτο και τελευταίο κεφάλαιο της διπλωματικής εργασίας, θα γίνει παρουσίαση του παιχνιδιού που δημιουργήθηκε και θα αναλυθούν τα βήματα που έχουν ακολουθηθεί για την δημιουργία του, καθώς και τα προβλήματα που εντοπίστηκαν και επιλύθηκαν.

Λέξεις κλειδιά: βιντεοπαιχνίδι, για πολλούς παίκτες, διαδικτυακό, Unity, Netcode for GameObjects.

ΠΕΡΙΛΗΨΗ ΣΤΑ ΑΓΓΛΙΚΑ

The subject of this thesis is the implementation of a 3D cooperative real-time video game on a Windows platform (PC) using the Unity game engine. This video game supports one or more players both locally (on the same PC) as well as online. The implementation of the online part was carried out using Unity's "Netcode for GameObjects" library. While the connection via the Internet was made using the Relay and Lobby of Unity's Gaming Services. The work is divided into three chapters. In the first chapter, basic concepts of video games will be mentioned, as well as a brief review of their history and development. The second chapter will analyze the Unity game engine and the gaming services it provides. In the third and last chapter of the thesis, there will be a presentation of the game created and the steps followed for its creation will be analyzed, as well as the problems identified and resolved.

Keywords: videogame, multiplayer, online, Unity, Netcode for GameObjects.

ΕΙΣΑΓΩΓΗ

Αντικείμενο της παρούσας εργασίας είναι να δημιουργηθεί και να αναλυθεί η διαδικασία παραγωγής βιντεοπαιχνιδιού με την βοήθεια της μηχανής παιχνιδιού Unity. Τονίζοντας τα σημεία τα οποία τα βρήκα ενδιαφέροντα αλλά και βασικά για τη δομή ενός συνεργατικού παιχνιδιού.

1. Εισαγωγικές έννοιες και ιστορία των βιντεοπαιχνιδιών

Εισαγωγικές έννοιες και σύντομη αναδρομή στην ιστορία και εξέλιξη των βιντεοπαιχνιδιών καθώς λίγα λόγια για την ανάπτυξη τους.

1.1 Τι είναι ένα βιντεοπαιχνίδι

Με τον όρο βιντεοπαιχνίδι (videogame) εννοείται οποιοδήποτε ηλεκτρονικό παιχνίδι που περιλαμβάνει αλληλεπίδραση με μια διεπαφή χρήστη (π.χ. ένα χειριστήριο, ένα πληκτρολόγιο ή μια συσκευή ανίχνευσης κίνησης) για την δημιουργία οπτικής ανατροφοδότησης σε μια συσκευή προβολής βίντεο δύο ή τρεις διαστάσεων, όπως μια οθόνη τηλεόρασης ή υπολογιστή. Τα περισσότερα σύγχρονα βιντεοπαιχνίδια είναι οπτικοακουστικά, με ηχητικό συμπλήρωμα που παρέχεται μέσω ηχείων ή ακουστικών, ακόμα και με άλλους τύπους αισθητηριακής ανάδρασης όπως απτική ανατροφοδότηση (haptic feedback). Οι παίκτες ασχολούνται με τα βιντεοπαιχνίδια για ψυχαγωγικούς ή εκπαιδευτικούς σκοπούς, ακολουθώντας συχνά ένα σύνολο κανόνων για να προχωρήσουν στην ιστορία του παιχνιδιού ή να επιτύχουν έναν συγκεκριμένο στόχο. Τα βιντεοπαιχνίδια διατίθενται σε διάφορα είδη, όπως δράσης, περιπέτειας, παιχνίδι ρόλων, προσομοίωση, παζλ, αθλήματα κ.α.. Επίσης μπορούν να είναι παιχνίδια για έναν παίκτη, για πολλούς παίκτες ή για μαζικά παιχνίδια πολλών παικτών, επιτρέποντας στους παίκτες να αλληλοεπιδρούν μεταξύ τους μέσα σ' έναν εικονικό κόσμο. (Video game - Wikipedia, 2024)

1.2 Τι είναι ένα βιντεοπαιχνίδι arcade

Ένα βιντεοπαιχνίδι arcade λαμβάνει την είσοδο του παίκτη από τα χειριστήρια του, τα επεξεργάζεται μέσω των ηλεκτρικών ή ηλεκτρονικών εξαρτημάτων και εμφανίζει την έξοδο σε μια ηλεκτρονική οθόνη ή παρόμοια οθόνη. Όλα τα βιντεοπαιχνίδια arcade λειτουργούν με κέρματα ή δέχονται και άλλα μέσα πληρωμής, στεγάζονται σε ντουλάπι arcade και βρίσκονται σε στοές ψυχαγωγίας μαζί με άλλα είδη παιχνιδιών arcade. Μέχρι τις αρχές της δεκαετίας του 2000, τα arcade βιντεοπαιχνίδια ήταν το μεγαλύτερο και πιο προηγμένο τεχνολογικά τμήμα της βιομηχανίας βιντεοπαιχνιδιών. (Arcade video game - Wikipedia, 2024)

1.3 Τι είναι μια μηχανή παιχνιδιών (game engine)

Η μηχανή παιχνιδιών είναι ένα λογισμικό σχεδιασμένο για την ανάπτυξη και τη δημιουργία βιντεοπαιχνιδιών, παρέχοντας στους προγραμματιστές εργαλεία για να χειριστούν βασικές λειτουργίες όπως απόδοση γραφικών, προσομοίωση φυσικής, ανίχνευση σύγκρουσης, διαχείριση ήχου και animation. Αυτές οι μηχανές υποστηρίζουν επίσης τη συγγραφή κώδικα (scripting), τη τεχνητή νοημοσύνη (AI), τη δικτύωση (networking) και τη διαχείριση στοιχείων παιχνιδιού (assets management), βελτιστοποιώντας την περίπλοκη διαδικασία ανάπτυξης παιχνιδιών. Στα δημοφιλή

παραδείγματα περιλαμβάνονται η Unity, η Unreal Engine και η Godot, τα οποία βοηθούν τους προγραμματιστές να εστιάζουν περισσότερο στις δημιουργικές πτυχές του σχεδιασμού του παιχνιδιού ελαχιστοποιώντας την ανάγκη προγραμματισμού αυτών των λειτουργιών από την αρχή, κάνοντας έτσι ευκολότερη και πιο αποτελεσματική την παραγωγή παιχνιδιών σε διαφορετικές πλατφόρμες. (Game engine - Wikipedia, 2024) (Game Engines: All You Need to Know - Gameopedia, 2022)

1.4 Τι είναι ένα Game Design Document

Το έγγραφο σχεδιασμού παιχνιδιού (GDD) είναι ζωτικής σημασίας για την δημιουργία ενός βιντεοπαιχνιδιού για διάφορους λόγους.

1.4.1 Σχέδιο ανάπτυξης

Το GDD χρησιμεύει ως προσχέδιο ή ένας χάρτης (roadmap) για ολόκληρη τη διαδικασία ανάπτυξης. Περιγράφει τις βασικές έννοιες, τους μηχανισμούς, τα χαρακτηριστικά, την ιστορία, τους χαρακτήρες και άλλα βασικά στοιχεία του παιχνιδιού. Αυτό το έγγραφο παρέχει μια σαφή κατεύθυνση που πρέπει να ακολουθήσει η ομάδα ανάπτυξης.

1.4.2 Εργαλείο επικοινωνίας

Το GDD λειτουργεί ως εργαλείο επικοινωνίας μεταξύ διαφορετικών μελών της ομάδας ανάπτυξης, συμπεριλαμβανομένων σχεδιαστών, καλλιτεχνών, προγραμματιστών και παραγωγών. Εξασφαλίζει ότι όλοι όσοι συμμετέχουν στο έργο έχουν κοινή κατανόηση του οράματος και των στόχων του παιχνιδιού.

1.4.3 Οδηγός αναφοράς

Σε όλη τη διαδικασία ανάπτυξης, το GDD χρησιμεύει ως οδηγός αναφοράς που μπορούν να συμβουλευτούν οι προγραμματιστές για να διασφαλίσουν ότι παραμένουν πιστοί στο αρχικό όραμα του παιχνιδιού. Βοηθά στην αποφυγή προσθήκης υπερβολικά πολλών δυνατοτήτων (future creep) και διασφαλίζει ότι το τελικό προϊόν ευθυγραμμίζεται με την αρχική ιδέα.

1.4.4 Σχεδιασμός και Προϋπολογισμός

Το GDD βοηθά στον προγραμματισμό της διαδικασίας ανάπτυξης αναλύοντας εργασίες, εκτιμώντας χρονοδιαγράμματα και κατανέμοντας αποτελεσματικά τους πόρους. Βοηθά επίσης στον προϋπολογισμό παρέχοντας μια ολοκληρωμένη επισκόπηση του πεδίου εφαρμογής του έργου.

1.4.5 Διαχείριση κινδύνων

Τεκμηριώνοντας λεπτομερώς τον σχεδιασμό του παιχνιδιού, το GDD βοηθά στον εντοπισμό πιθανών κινδύνων και προκλήσεων νωρίς στη διαδικασία ανάπτυξης. Αυτό επιτρέπει στην ομάδα να αντιμετωπίζει προληπτικά ζητήματα προτού γίνουν σημαντικά εμπόδια.

1.4.6 Παρουσίαση και Μάρκετινγκ

Ένα καλογραμμένο GDD μπορεί να χρησιμοποιηθεί για να προωθήσει το παιχνίδι σε εκδότες, επενδυτές ή ενδιαφερόμενους φορείς. Παρέχει μια σαφή και συναρπαστική επισκόπηση της ιδέας του παιχνιδιού, διευκολύνοντας τη δημιουργία ενδιαφέροντος

και την εξασφάλιση χρηματοδότησης. Επιπλέον, στοιχεία του GDD μπορούν να χρησιμοποιηθούν σε υλικό μάρκετινγκ για την προώθηση του παιχνιδιού σε πιθανούς παίκτες.

Συνολικά, ένα έγγραφο σχεδίασης παιχνιδιού είναι απαραίτητο για τη διασφάλιση ότι η διαδικασία ανάπτυξης εκτελείται ομαλά, η ομάδα παραμένει ευθυγραμμισμένη και το τελικό προϊόν πληροί το επιδιωκόμενο όραμα και τα πρότυπα ποιότητας.

1.5 Η ιστορία και εξέλιξη των βιντεοπαιχνιδιών

1.5.1 1950s-1960s: Τα πρώιμα χρόνια

Από τις αρχές του 1950 οι επιστήμονες υπολογιστών χρησιμοποιούσαν ηλεκτρονικές μηχανές για να κατασκευάσουν απλά συστήματα παιχνιδιών, όπως το Bertie the Brain το 1950 για να παίζεις τρίλιζα [εικόνα 1] ή το Nimrod για να παίζεις Nim. Αυτά τα συστήματα χρησιμοποιούσαν ηλεκτρονικές οθόνες φωτός και η κύρια χρήση τους ήταν ως συστήματα επίδειξης σε μεγάλες εκθέσεις για να επιδείξουν τη δύναμη των υπολογιστών εκείνη την εποχή. Μια άλλη επίδειξη ήταν το Tennis for Two, ένα παιχνίδι που δημιουργήθηκε από τον William Higinbotham στο Brookhaven National Laboratory το 1958, χρησιμοποιώντας έναν αναλογικό υπολογιστή και έναν παλμογράφο για μια οθόνη [εικόνα 2]. Το “Spacewar!” θεωρείται ένα από τα πρώτα αναγνωρισμένα βιντεοπαιχνίδια που είχε ευρύτερη διανομή πίσω από ένα ενιαίο εκθεσιακό σύστημα. Αναπτύχθηκε το 1961 για τον κεντρικό υπολογιστή PDP-1 στο MIT. Το παιχνίδι αυτό επέτρεπε σε δύο παίκτες να προσομοιάσουν μια διαστημική μάχη στην απλοϊκή οθόνη του PDP-1 [εικόνα 3]. Ο πηγαίος κώδικας του παιχνιδιού διαμοιράστηκε και σε άλλα ιδρύματα με PDP-1 καθώς οι φοιτητές του MIT μετακινούνταν, επιτρέποντας το παιχνίδι να αποκτήσει δημοτικότητα Έτσι το “Spacewar!” έγινε το πρώτο γνωστό βιντεοπαιχνίδι που μπορούσε να παιχτεί σε πολλαπλές εγκαταστάσεις υπολογιστών.

1.5.2 1970s-1980s: Η γέννηση των Arcade και η χρυσή εποχή

Την δεκαετία του 70 εμφανίστηκαν τα παιχνίδια arcade όπως το “Pong” (1972) από την Atari, το οποίο έγινε το πρώτο εμπορικά επιτυχημένο βιντεοπαιχνίδι [εικόνα 4]. Το Magnavox Odyssey κυκλοφόρησε το 1972 ως η πρώτη οικιακή κονσόλα βιντεοπαιχνιδιών. Το 1977 με το Atari 2600 και το 1978 με το Magnavox Odyssey 2 έχουμε τις πρώτες κονσόλες με χρήση κασετών για αλλαγή παιχνιδιών. Με την κυκλοφορία του “Space Invaders” (1978) θεωρείται ότι ξεκίνησε η χρυσή εποχή των arcade παιχνιδιών μερικά από τα πιο γνωστά παιχνίδια ήταν το “Pac-Man” (1980) και η συνέχεια του “Ms. Pac-Man” (1982), το “Donkey Kong” (1981) και το “Q*bert” (1982). Εκείνη την περίοδο άρχισαν οι οικιακοί υπολογιστές να είναι πιο φθηνοί και ευρέως κοινοί. Μετά την επιτυχία του Apple II (1977) και του Commodore 64 (1982) πολλοί άνθρωποι άρχισαν να φτιάχνουν τα δικά τους λογισμικά και βιντεοπαιχνίδια στα σπίτια τους. Ένα από τα πιο γνωστά παιχνίδια εκείνης της περιόδου είναι το πρώτο Ultima από τον Richard Garriott το 1981. Τα βιντεοπαιχνίδια κυριαρχούσαν στις βιβλιοθήκες λογισμικού των οικιακών υπολογιστών. Στην δεκαετία του 80 έχουμε τις πρώτες κονσόλες από τις γνωστές εταιρείες Sega και Nintendo. Από την μεριά της Sega οι πιο γνωστές κονσόλες ήταν το Sega Master System (1985/6) και το Sega Genesis

(1988/9) με το εικονικό παιχνίδι τους “Sonic the Hedgehog” (1991). Από την μεριά της Nintendo έχουμε τις κονσόλες Nintendo Entertainment System (NES) και Super Nintendo Entertainment System (SNES) με το γνωστό τους franchise “Super Mario”. Η πρώτη σημαντικότερη φορητή κονσόλα της Nintendo ήταν το Game Boy (1989) που συνοδευόταν μαζί με το γνωστό παιχνίδι “Tetris” (1989). Έπειτα ξεκίνησε ο μεγάλος ανταγωνισμός των δύο αυτών εταιρειών.

1.5.3 1990s: Η άνοδος των τρισδιάστατων παιχνιδιών και κονσολών

Η δεκαετία του 90 ήταν μια δεκαετία σημαντικής καινοτομίας στα βιντεοπαιχνίδια. Εκείνη την χρονική περίοδο έγινε η μετάβαση στα τρισδιάστατα γραφικά και σε πολλά νέα είδη βιντεοπαιχνιδιών, όπως βιντεοπαιχνίδια βολών πρώτου προσώπου (first-person shooter), όπως “Wolfenstein 3D” (1992), βιντεοπαιχνίδια στρατηγικής σε πραγματικό χρόνο (real-time strategy), όπως “Dune II” (1992), και μαζικά διαδικτυακά παιχνίδια (MMO, Massively Multiplayer Online). Μέχρι τότε τα βιντεοπαιχνίδια σε κονσόλες διανεμόταν σε κασέτες ROM ενώ του υπολογιστή σε δισκέτες. Οι δύο μορφές αυτές είχαν περιορισμούς στη χωρητικότητα αποθήκευσης. Οπότε έχουμε και τη μετάβαση σε οπτικά μέσα αποθήκευσης που επωφελήθηκαν πολύ οι υπολογιστές και κάποιες κονσόλες βιντεοπαιχνιδιών, όπως το Sega Saturn (1994/5) η πρώτη 32-bit κονσόλα που χρησιμοποίησε CD αντί για κασέτες. Λίγο μετά την ίδια χρονιά έχουμε άλλη μια 32-bit κονσόλα με CD το PlayStation (1994/5) της Sony. Τον επόμενο χρόνο η Nintendo κυκλοφόρησε το Nintendo 64 (1996) που ήταν 64-bit κονσόλα αλλά χρησιμοποιούσε ακόμα κασέτες. Και οι τρεις αυτές κονσόλες υποστήριζαν πραγματικά τρισδιάστατα γραφικά. Η Sony κυριάρχησε στις πωλήσεις μέχρι το 2000. Η Sega έκανε ακόμα μια προσπάθεια με το Sega Dreamcast (1998/9), που πολλοί θεωρούν ότι ήταν πολύ μπροστά για την εποχή της, ήταν η πρώτη κονσόλα που υποστήριζε διαδικτυακό παιχνίδι, αλλά ήταν μια εμπορική αποτυχία. Οπότε η Sega το 2001 σταμάτησε να ασχολείται με κονσόλες και έγινε πλέον εταιρεία λογισμικού. Την ίδια δεκαετία έχουμε και τον ανταγωνισμό στις φορητές κονσόλες παιχνιδιών: Sega Game Gear, Atari Lynx, Nintendo Game Boy Color. Η Nintendo κέρδισε τον ανταγωνισμό ιδιαίτερα με την έκδοση της σειράς βιντεοπαιχνιδιών “Pokémon”, το οποίο είναι franchise με τις περισσότερες πωλήσεις για την εταιρεία. Στο τέλος της δεκαετίας, τα πρώτα μαζικά διαδικτυακά παιχνίδια ρόλων για πολλούς παίκτες άρχισαν να δημιουργούνται, τα οποία προσάρμοσαν τα νέα τρισδιάστατα γραφικά για να δημιουργήσουν εικονικούς κόσμους στον υπολογιστή, ξεκινώντας με το “Meridian 59” το 1996 και διαδόθηκαν με την επιτυχία του “Ultima Online” το 1997.

1.5.4 2000s: Η εποχή του διαδικτυακού παιχνιδιού και των χειριστηρίων κίνησης

Η Sony ξεκίνησε δυναμικά το 2000 με το PlayStation 2. Το PlayStation 2 ήταν η πρώτη κονσόλα που υποστήριζε DVD παιχνίδια αλλά και ταινίες. Επίσης είχε την δυνατότητα να παίξει τα παιχνίδια της προηγούμενης κονσόλας PlayStation. Η Nintendo το 2001 κυκλοφορεί το GameCube. Η κονσόλα αυτή χρησιμοποιούσε οπτικούς δίσκους, οι οποίοι ήταν ειδικά διαμορφωμένοι για το σύστημα της. Ένας νέος ανταγωνιστής εμφανίζεται στο χώρο των βιντεοπαιχνιδιών, η Microsoft με το Xbox (2001). Ωστόσο φοβήθηκε ότι το PlayStation 2 θα γινόταν το κεντρικό σημείο ηλεκτρονικής ψυχαγωγίας στο σπίτι και θα έχανε κόσμο στο χώρο του υπολογιστή. Το PlayStation 2 παρέμεινε η κορυφαία πλατφόρμα για το πρώτο μέρος της δεκαετίας και παραμένει η κονσόλα με τις περισσότερες πωλήσεις όλων των εποχών. Στα μέσα της δεκαετίας, το

Xbox 360 (2005) της Microsoft, το PlayStation 3 (2006) της Sony και το Wii (2006) της Nintendo ξεκίνησαν τη σύγχρονη εποχή των παιχνιδιών υψηλής ευκρίνειας. Παρόλο που το PlayStation 3 ήταν επιτυχημένο και το μόνο σύστημα εκείνη την εποχή για αναπαραγωγή Blu-ray, αντιμετώπισε σκληρό ανταγωνισμό. Το Xbox 360 με παρόμοιες δυνατότητες γραφικών με το PlayStation 3 επαινέστηκε για το οικοσύστημα διαδικτυακών παιχνιδιών του. Επίσης διέθετε ένα υπερσύγχρονο σύστημα καταγραφής κίνησης που πρόσφερε έναν διαφορετικό τρόπο να παίζεις βιντεοπαιχνίδια, το KINECT. Παρά το γεγονός ότι ήταν τεχνολογικά κατώτερο από τα άλλα δύο συστήματα, το Wii κατατρόπωσε τον ανταγωνισμό του στις πωλήσεις. Τα τηλεχειριστήρια κίνησης του έκαναν το παιχνίδι πιο ενεργό από ποτέ, βοηθώντας το να απευθυνθεί σε μεγαλύτερο κομμάτι του ευρύτερου κοινού, ακόμα και σε άτομα σε οίκους ευγηρίας. Σχετικά με τις φορητές κονσόλες, η Nintendo διατήρησε τη θέση της στην αγορά με το Game Boy Advance το 2001. Η συγκεκριμένη συσκευή είχε υψηλή ανάλυση, έγχρωμη οθόνη LCD και 32-bit επεξεργαστή η οποία επέτρεπε την εύκολη μετατροπή παιχνιδιών από SNES, Nintendo 64 και GameCube στο Game Boy Advance. Τα επόμενα σημαντικά φορητά συστήματα ήταν το Nintendo DS και το PlayStation Portable (PSP) που κυκλοφόρησαν το 2004. Αξίζει να σημειωθεί ότι και τα δύο είχαν ικανότητες ασύρματης δικτύωσης. Το PSP είχε ανώτερα γραφικά και ισχύ σε σύγκριση με το Nintendo DS. Το Nintendo DS με τις δύο οθόνες (με τη μία να είναι οθόνη αφής) ήταν εξαιρετικά δημοφιλείς στους καταναλωτές, ιδιαίτερα στα μικρά παιδιά και στους μεσήλικες παίκτες που τους τράβηξαν την προσοχή τα παιχνίδια “Nintendogs” (2005) and “Brain Age” (2005) αντίστοιχα.

1.5.5 2010s: Η κυριαρχία των παιχνιδιών στα κινητά και η εικονική πραγματικότητα

Η δεκαετία του 2010 πραγματοποιήθηκε η έκρηξη του παιχνιδιού για κινητά, με τα smartphone και τα tablet να γίνονται δημοφιλείς πλατφόρμες παιχνιδιών, οδηγώντας στη δημοτικότητα παιχνιδιών όπως το “Angry Birds” (2009-2010) και το “Candy Crush Saga” (2012). Σχεδόν κάθε άτομο έχει έναν υπολογιστή τσέπης το οποίο υποστηρίζει τη δυνατότητα να παίζει παιχνίδια, ενώ παράλληλα η διαδικτυακή ροή βίντεο (video streaming) τροφοδοτεί την ανάπτυξη των ηλεκτρονικών αθλημάτων (esports). Το κίνημα των indie παιχνιδιών ενηλικιώνεται με την τεράστια δημοτικότητα του Minecraft (2010), του εθιστικού παιχνιδιού κατασκευής τούβλων από τον Σουηδό προγραμματιστή Markus Persson. Το “Skylanders”(2011) γίνεται το πρώτο χτύπημα επαυξημένης πραγματικότητας, επιτρέποντας στους παίκτες να τοποθετούν πλαστικές φιγούρες σε ένα “Portal of Power” για να βάλουν χαρακτήρες στο παιχνίδι. Δύο χρόνια αργότερα, το Disney Infinity εντάσσεται στις τάξεις των υβριδικών παιχνιδιών-βιντεοπαιχνιδιών (toy-video game hybrids). Το 2012 ο ιστότοπος Crowdfunding Kickstarter δίνει τη δυνατότητα στους δημιουργούς παιχνιδιών να συγκεντρώσουν εκατομμύρια δολάρια για να δημιουργήσουν νέες και πειραματικές πλατφόρμες παιχνιδιού, όπως η κονσόλα OUYA (Android-based κονσόλα παιχνιδιών) και το Oculus Rift (VR headset). Το 2013 τα “Gone Home”, “The Last of Us” και “Papers, Please” εισάγουν ένα νέο κύμα ώριμων ιστοριών βιντεοπαιχνιδιών που αντιμετωπίζουν τους παίκτες με δύσκολες συναισθηματικές επιλογές σε ηθικά περίπλοκους κόσμους. Το “Free-to-play” γίνεται κυρίαρχο επιχειρηματικό μοντέλο καθώς μεγάλες παραγωγές όπως το Clash of Clans (2012) και το World of Tanks (2011) επιτυγχάνουν πωλήσεις εκατοντάδων εκατομμυρίων δολαρίων μέσω πληρωμών μικροσυναλλαγών (microtransaction payments) για αντικείμενα εντός παιχνιδιού και περιεχόμενο υψηλής

ποιότητας (premium content). Το 2016 έχουμε τους παίκτες να κυνηγούν εικονικά Pokémon στον πραγματικό κόσμο με τη δωρεάν επιτυχία της Niantic "Pokémon GO", συνδυάζοντας ένα παιχνίδι επαυξημένης πραγματικότητας με την τοποθεσία του παίκτη. Επίσης το 2016 η τεχνολογία εικονικής πραγματικότητας (VR) έκανε σημαντικά βήματα με την κυκλοφορία συσκευών όπως το Oculus Rift CV1 (Consumer Version 1) και το PlayStation VR, προσφέροντας καθηλωτικές εμπειρίες παιχνιδιού. Καθώς τα smartphones και τα tablets γίνονται πιο δημοφιλή ως πλατφόρμες παιχνιδιών, το 2017 η Nintendo κυκλοφόρησε το Nintendo's Switch. Αυτή η συσκευή είναι ένα υβρίδιο που συνδυάζει εμπειρίες κινητών και κονσόλας. Το καινοτόμο σύστημα αυτό επιτρέπει στους παίκτες να παίρνουν μαζί τους παιχνίδια όπως το "Legend of Zelda: Breath of the Wild" (2017) και το "Super Mario Odyssey" (2017) όπου κι αν πάνε.

1.5.6 2020s: Τα γραφικά Ray-tracing και Cloud Gaming

Η δεκαετία αυτή ξεκινά δυναμικά με τις κονσόλες νέας γενιάς PlayStation 5 και Xbox Series X/S (τέλη 2020). Αυτές οι νέες κονσόλες έφεραν σημαντικές βελτιώσεις σε επεξεργαστική ισχύ, γραφικά και χρόνους φόρτωσης, κυρίως χάρη στην τεχνολογία solid-state drive (SSD) και πιο ισχυρές GPUs οι οποίες υποστηρίζουν και ray-tracing γραφικά. Οι νέες κονσόλες τονίζουν επίσης τη συμβατότητα με προηγούμενα παιχνίδια, επιτρέποντας στους παίκτες να απολαύσουν παλαιότερους τίτλους με βελτιωμένη απόδοση. Τα ray-tracing γραφικά είναι μια τεχνική για την απόδοση ρεαλιστικών φωτισμών και σκιών στα γραφικά υπολογιστών, η οποία χρησιμοποιείται κυρίως στα βιντεοπαιχνίδια και στην παραγωγή ταινιών. Η τεχνική αυτή υποστηρίζει τον ρεαλιστικό φωτισμό μοντελοποιώντας τη διαδρομή που ακολουθεί κάθε ακτίνα φωτός όταν εισέρχεται σε μια σκηνή και αλληλοεπιδρά με διάφορες επιφάνειες. Παιχνίδια όπως "Cyberpunk 2077" [εικόνα 6] (Δεκέμβριος 2020), "Minecraft" με RTX [εικόνα 5] (Απρίλιος 2020) και "Control" (Αύγουστος 2019) χρησιμοποιούν το ray-tracing για να προσφέρουν εκπληκτικά ρεαλιστικά γραφικά που ενισχύουν την ατμόσφαιρα και την καθηλωτική εμπειρία του παιχνιδιού. Το ray-tracing συνεχίζει να είναι στο προσκήνιο της τεχνολογίας γραφικών και αναμένεται να εξελιχθεί περαιτέρω καθώς η τεχνολογία γραφικών προχωράει, προσφέροντας ακόμα πιο βελτιωμένες και ρεαλιστικές οπτικές εμπειρίες στους παίκτες. Ταυτόχρονα με όλα αυτά έχουμε και την ωρίμανση των υπηρεσιών Cloud Gaming, όπως το Google Stadia (Νοέμβριος 2019), το NVIDIA GeForce Now (Φεβρουάριος 2020), το Xbox Cloud Gaming (Σεπτέμβριος 2020, μέρος του Xbox Game Pass), και το Amazon Luna (Μάρτιος 2022). Αυτές οι πλατφόρμες επιτρέπουν στους παίκτες να μεταδίδουν παιχνίδια απευθείας στις συσκευές τους χωρίς την ανάγκη για ισχυρό υλικό, καθιστώντας τα παιχνίδια πιο προσβάσιμα σε διάφορες συσκευές, όπως κινητά τηλέφωνα, tablets, smart TVs και υπολογιστές. Εν τω μεταξύ η τεχνολογία της εικονικής πραγματικότητας (VR) συνεχίζει να εξελίσσεται με την κυκλοφορία πιο προηγμένων συσκευών όπως το Oculus Quest 2 (Οκτώβριος 2020, που μετονομάστηκε σε Meta Quest 2), το οποίο προσφέρει μια ανεξάρτητη εμπειρία VR χωρίς την ανάγκη για εξωτερικούς αισθητήρες ή υπολογιστές. Ακόμα έχουμε συσκευές VR που έχουν υποστήριξη για παρακολούθηση ματιών, χεριών και προσώπου. Αξίζει να σημειωθεί η ενισχυμένη πραγματικότητα (AR) που συνεχίζει να αναπτύσσεται, αν και με πιο αργούς ρυθμούς, με εφαρμογές για τα κινητά όπως παιχνίδια και πειράματα που προσφέρουν νέες εμπειρίες παιχνιδιού. Η πανδημία (COVID-19) είχε σημαντικό αντίκτυπο στο χώρο του gaming, με τις καραντίνες να οδηγούν σε αύξηση τόσο του αριθμού των παικτών όσο

και των ωρών που αφιερώνονται στα παιχνίδια. Αυτή η αύξηση επιτάχυνε επίσης την ανάπτυξη της ψηφιακής διανομής και του διαδικτυακού παιχνιδιού. Η ανάπτυξη των παιχνιδιών αντιμετώπισε προκλήσεις, συμπεριλαμβανομένων των καθυστερήσεων και την αλλαγή σε απομακρυσμένη εργασία, η οποία επηρέασε τα προγράμματα κυκλοφορίας και τις συνθήκες εργασίας στον κλάδο.

(History of video games - Wikipedia, 2024), (Video Game History - Timeline & Facts, 2022), (Video Game History Timeline - The Strong National Museum of Play, 2022), (History.com Editors, 2022)



Εικόνα 1 Bertie the Brain https://en.wikipedia.org/wiki/Bertie_the_Brain



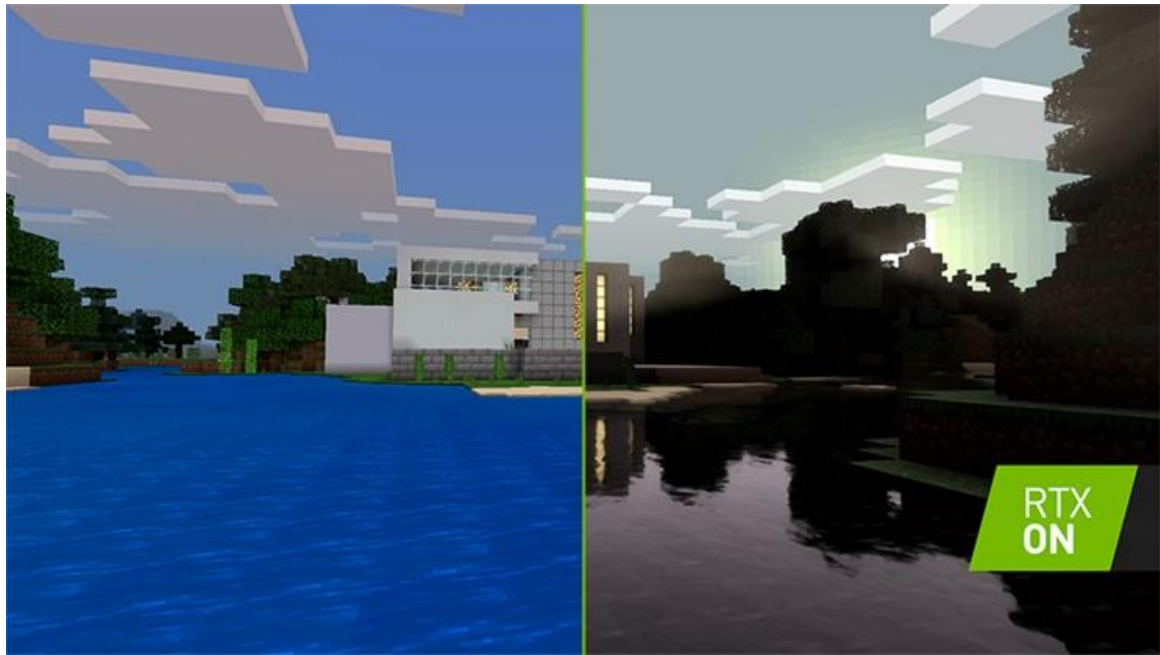
Εικόνα 2 Tennis for Two https://en.wikipedia.org/wiki/Tennis_for_Two



Εικόνα 3 Spacewar! on a PDP-1 <https://en.wikipedia.org/wiki/Spacewar!>

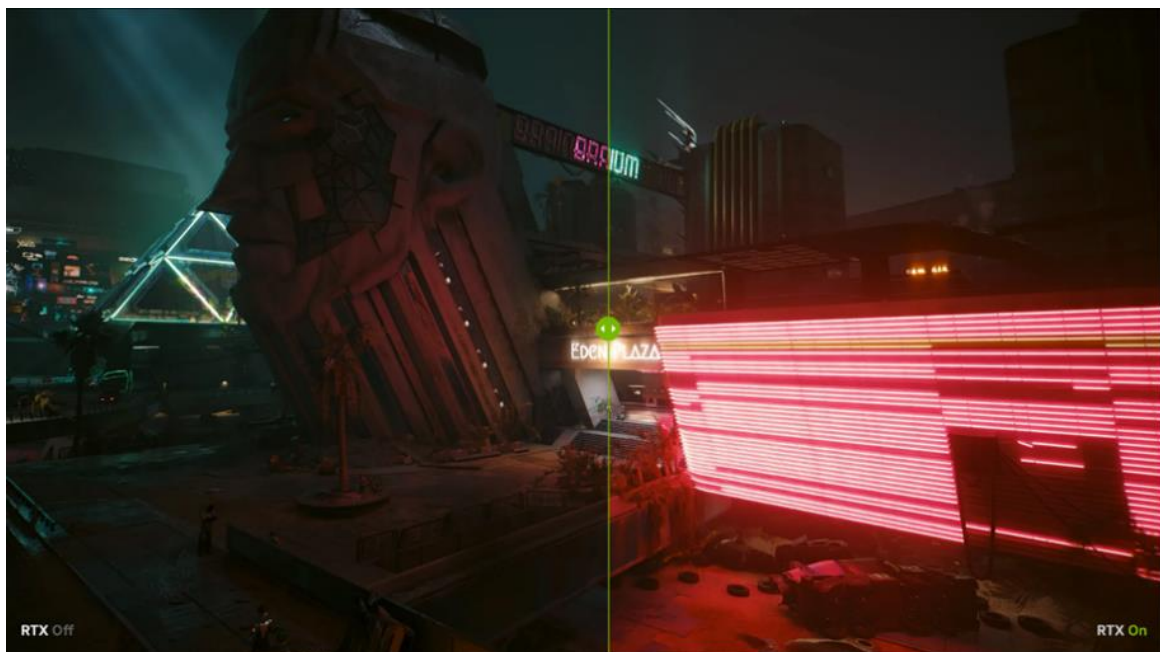


Εικόνα 4 Pong is the first commercially successful arcade video game
https://en.wikipedia.org/wiki/Arcade_video_game



Εικόνα 5 Minecraft με RTX αριστερά και χωρίς δεξιά

<https://www.nvidia.com/en-us/geforce/news/minecraft-rtx-trailer-screenshots-partnership/>



Εικόνα 6 Cyberpunk 2077 με RTX δεξιά και χωρίς αριστερά

<https://www.nvidia.com/en-us/geforce/campaigns/cyberpunk-2077/>

1.6 Η εξέλιξη στην παραγωγή των βιντεοπαιχνιδιών

Η παραγωγή βιντεοπαιχνιδιών έχει υποστεί τεράστιες αλλαγές από τις πρώιμες μέρες της βιομηχανίας μέχρι σήμερα, εξελίχθηκε από απλά προγράμματα σε πολύπλοκα συστήματα με ομάδες πολλαπλών ειδικοτήτων. Παρακάτω θα αναφερθούν μερικές από τις κύριες φάσεις αυτής της εξέλιξης:

1.6.1 Δεκαετία του 1970

Τις πρώτες μέρες τα βιντεοπαιχνίδια αναπτύσσονταν συχνά από έναν ή λίγους προγραμματιστές, οι οποίοι ανέλαβαν πολλαπλούς ρόλους – σχεδιασμός, προγραμματισμός και τεστάρισμα. Υπήρχαν πολλές τεχνικές δυσκολίας διότι η τεχνολογία ήταν πρωτόγονη, και τα παιχνίδια αποτελούνταν κυρίως από απλά γραφικά και μηχανικές. Ο κάθε προγραμματιστής χρειαζόταν να σκεφτεί την υλοποίηση κάθε στοιχείου με τη τεχνολογία της εποχής, καθώς δεν υπήρχαν μηχανές παιχνιδιών όπως έχουμε σήμερα.

1.6.2 Δεκαετία του 1980-1990

Καθώς τα παιχνίδια γίνονταν πιο πολύπλοκα, οι ομάδες ανάπτυξης άρχισαν να μεγαλώνουν, περιλαμβάνοντας εξειδικευμένους προγραμματιστές, σχεδιαστές, καλλιτέχνες και ηχολήπτες. Οι παραγωγές των βιντεοπαιχνιδιών άρχισαν να απαιτούν μεγαλύτερους προϋπολογισμούς, καθιστώντας τα πιο συγκρίσιμα με τις ταινίες.

1.6.3 Δεκαετία του 2000

Η χρήση προηγμένων game engines (όπως η Unity, η Unreal Engine και η Godot) καθιστά την ανάπτυξη παιχνιδιών πιο προσιτή και διευκολύνει τη δημιουργία πολύπλοκων γραφικών και φυσικής. Η δυνατότητα για απομακρυσμένη εργασία και διαδικτυακή συνεργασία επιτρέπει στις ομάδες να εργάζονται από διαφορετικές γεωγραφικές τοποθεσίες.

1.6.4 Δεκαετία του 2010-2020

Πολλές ομάδες ανάπτυξης εφαρμόζουν Agile μεθοδολογίες για να διαχειριστούν τα έργα τους, παρέχοντάς τους μεγαλύτερη ευελιξία και βελτιωμένη ανταπόκριση σε αλλαγές. Οι εταιρείες παιχνιδιών έχουν γίνει παγκόσμιες, με τμήματα ανάπτυξης σε πολλά κράτη, επιτρέποντας την πρόσβαση σε διεθνή ταλέντα. Η ανατροφοδότηση και η συμμετοχή των παικτών μέσω beta testing και early access προγραμμάτων έχουν καταστεί κρίσιμα στοιχεία της διαδικασίας ανάπτυξης. Η τεχνητή νοημοσύνη και τα αυτοματοποιημένα εργαλεία βοηθούν περαιτέρω τον τρόπο παραγωγής παιχνιδιών, από τον αυτόματο σχεδιασμό επιπέδων και μοντέλων μέχρι και τη βελτιστοποίηση του κώδικα.

2. Unity

Λίγα λόγια για την εταιρεία, την μηχανή παιχνιδιού (game engine) και τις online υπηρεσίες που προσφέρει.

2.1 Unity Software Inc. (Unity Technologies)

Η Unity Software Inc. (επιχειρεί ως Unity Technologies) είναι μια αμερικανική εταιρεία ανάπτυξης λογισμικού βιντεοπαιχνιδιών με έδρα το Σαν Φρανσίσκο και γραφεία σε όλο τον κόσμο. Ιδρύθηκε στη Δανία το 2004 ως Over the Edge Entertainment από τους David Helgason, Nicholas Francis και Joachim Ante και έκτοτε εξελίχθηκε σε παγκόσμια δύναμη και άλλαξε το όνομά της το 2007. Η Unity Technologies είναι η εταιρεία πίσω από την πλατφόρμα Unity, μια από τις πιο δημοφιλείς και ευρέως χρησιμοποιούμενες μηχανές ανάπτυξης παιχνιδιών στον κόσμο. Το κορυφαίο προϊόν της εταιρείας, η μηχανή Unity, παρέχει στους προγραμματιστές ένα ολοκληρωμένο σύνολο εργαλείων για τη δημιουργία καθηλωτικών εμπειριών 2D, 3D, AR και VR σε διάφορες πλατφόρμες, συμπεριλαμβανομένων κινητών, υπολογιστών, κονσολών και ιστού. Η Unity Technologies προσφέρει επίσης μια σειρά από υπηρεσίες και λύσεις, όπως τα Unity Gaming Services (που περιέχει Unity Ads, Unity Analytics, Unity Cloud Build, Unity Collaborate κ.α.) και το Unity Asset Store, για την υποστήριξη των προγραμματιστών καθ' όλη τη διάρκεια του κύκλου ζωής της ανάπτυξης του παιχνιδιού. Με έμφαση στη δημοκρατοποίηση της ανάπτυξης παιχνιδιών και την προώθηση της καινοτομίας, η Unity Technologies συνεχίζει να ενθαρρύνει τους δημιουργούς και να προωθεί την πρόοδο της τεχνολογίας της διαδραστικής ψυχαγωγίας. (Unity Technologies - Wikipedia, 2024)

2.2 Unity (game engine)

Η Unity είναι μία ισχυρή και ευέλικτη μηχανή παιχνιδιών που αναπτύχθηκε από τη Unity Technologies. Η πρώτη κυκλοφορία της ήταν το 2005, με στόχο να εκδημοκρατίσει την ανάπτυξη παιχνιδιών καθιστώντας το προσβάσιμο σε περισσότερους προγραμματιστές. Είναι γνωστή για την προσβασιμότητά της, τις δυνατότητες διασυνδεσιμότητας μεταξύ πλατφορμών και το εκτεταμένο σετ χαρακτηριστικών της, καθιστώντας τη μία από τις κορυφαίες επιλογές για τους προγραμματιστές παιχνιδιών όλων των επιπέδων εμπειρίας. Με τη Unity, οι προγραμματιστές μπορούν να δημιουργήσουν εκπληκτικά παιχνίδια 2D και 3D για μια ευρεία γκάμα πλατφορμών, συμπεριλαμβανομένων των κινητών συσκευών, των κονσολών, των Η/Υ και του ιστού. Ένα από τα βασικά πλεονεκτήματα της Unity έγκειται στη φιλική προς το χρήστη διεπαφή και τη ροή εργασίας της. Ο εύχρηστος editor της επιτρέπει στους προγραμματιστές να σχεδιάζουν σκηνές, να εισάγουν στοιχεία και να προσθέτουν διαδραστικότητα χωρίς να χρειάζονται εκτεταμένες γνώσεις κώδικα. Ωστόσο, η Unity υποστηρίζει επίσης προχωρημένο προγραμματισμό σε γλώσσα C#, επιτρέποντας μεγαλύτερη ευελιξία και προσαρμογή. Το κατάστημα

αντικειμένων της Unity είναι ένα άλλο σημαντικό χαρακτηριστικό της, προσφέροντας μια τεράστια βιβλιοθήκη έτοιμων αντικειμένων, σεναρίων και πρόσθετων όπου οι προγραμματιστές μπορούν να ενσωματώσουν στα έργα τους για να επιταχύνουν την ανάπτυξη και να βελτιώσουν τη λειτουργικότητα τους. Επιπλέον, η Unity παρέχει ισχυρά εργαλεία για απόδοση γραφικών, προσομοίωση φυσικής, επεξεργασία ήχου, animation και πολλά άλλα, δίνοντας τη δυνατότητα στους προγραμματιστές να δημιουργήσουν καθηλωτικές εμπειρίες παιχνιδιού με εκπληκτικά γραφικά και ρεαλιστική φυσική. Επιπλέον, η υποστήριξη πολλαπλών πλατφορμών της Unity επιτρέπει στους προγραμματιστές να δημιουργήσουν μία φορά και να αναπτύξουν τα παιχνίδια τους σε πολλές πλατφόρμες με ελάχιστη επιπλέον προσπάθεια, απλοποιώντας τη διαδικασία ανάπτυξης και προσεγγίζοντας ένα ευρύτερο κοινό. Συνολικά, ο συνδυασμός προσβασιμότητας, ισχυρών χαρακτηριστικών και δυνατοτήτων πολλαπλών πλατφορμών της Unity έχει εδραιώσει τη θέση της ως μια από τις κορυφαίες μηχανές παιχνιδιών στον κλάδο, που χρησιμοποιείται τόσο από ανεξάρτητους προγραμματιστές (indie) όσο και από μεγάλα στούντιο (AAA) για τη δημιουργία μιας μεγάλης ποικιλίας επιτυχημένων παιχνιδιών και διαδραστικών εμπειριών. (Real-Time 3D Development Platform & Editor | Unity, 2024) (Unity (game engine) - Wikipedia, 2024)

2.3 Unity Gaming Services

2.3.1 Επισκόπηση του Unity Gaming Services

Το Unity Gaming Services (UGS) είναι μια πλατφόρμα από άκρο σε άκρο (end-to-end) που έχει σχεδιαστεί για να βοηθήσει τους προγραμματιστές να δημιουργήσουν, να εμπλακούν και να αναπτύξουν το παιχνίδι τους. Αυτές οι υπηρεσίες έχουν σχεδιαστεί για να διευκολύνουν διάφορες πτυχές της διαδικασίας ανάπτυξης παιχνιδιών, από τη δημιουργία εσόδων και την ανάλυση έως τη συνεργασία και την ανάπτυξη, δίνοντας τελικά τη δυνατότητα στους προγραμματιστές να δημιουργούν και να διαχειρίζονται επιτυχημένα παιχνίδια πιο αποτελεσματικά. Το UGS απλοποιεί πολλές εργασίες και προκλήσεις ανάπτυξης παιχνιδιών χωρίς να χρειάζεται ανησυχία για την συντήρηση ή την κλιμάκωση της υποδομής back-end. Όλες οι υπηρεσίες που προσφέρει το UGS είναι δωρεάν μέχρι ένα όριο για την κάθε μια ξεχωριστά και μετά υπάρχει χρέωση ανάλογα με τη χρήση της κάθε υπηρεσίας (pay as you go). Μερικά παραδείγματα χρήσης αναφέρονται παρακάτω:

- Προσθήκη λειτουργίας για πολλούς παίκτες και κοινωνικές λειτουργίας στο παιχνίδι.
- Χρήση της λογικής του παιχνιδιού από την πλευρά του διακομιστή (server-side) για να εξασφαλίσετε ίσους όρους ανταγωνισμού.
- Δίνει τη δυνατότητα στους παίκτες να έχουν πρόσβαση στα δεδομένα παιχνιδιού τους σε διάφορες πλατφόρμες παιχνιδιών.
- Εκτέλεση δοκιμών A/B και μέτρηση των δεδομένων παιχνιδιού από διάφορες υπηρεσίες για να βοηθήσει στις αποφάσεις σχεδιασμού.
- Παροχή νέου περιεχομένου χωρίς ενημέρωση της εφαρμογής

- Εκτέλεση προγραμματισμένων εκδηλώσεων και παροχή ποικίλου περιεχομένου στο παιχνίδι κατά την διάρκεια αυτών των εκδηλώσεων.
- Προσέλκυση παικτών με διασκεδαστικά, προοδευτικά συστήματα ανταμοιβής και αφοσίωσης.

Παρακάτω θα αναφερθούν όλες οι υπηρεσίες που παρέχονται από το Unity Gaming Services:

2.3.2 Λογαριασμοί (Accounts)

- Authentication: Οι εφαρμογές συνήθως πρέπει να γνωρίζουν την ταυτότητα ενός χρήστη για να παρέχουν μια ποικιλία λειτουργιών και υπηρεσιών τόσο στους προγραμματιστές παιχνιδιών όσο και στους παίκτες για να διασφαλίζουν ασφάλεια, συνέπεια και ασφάλεια σε κάθε αλληλεπίδραση. Το Unity Authentication παρέχει ανώνυμες και από συγκεκριμένες πλατφόρμες (Google, Apple, Facebook κ.α.) λύσεις ελέγχου ταυτότητας για τις υποστηριζόμενες πλατφόρμες, συμπεριλαμβανομένων των κινητών και των Η/Υ.

2.3.3 Διαχείριση περιεχομένου (Content management)

- Cloud Code: Ένα σύγχρονο παιχνίδι απαιτεί πολλά στοιχεία για να είναι επιτυχημένο, για παράδειγμα οικονομία στο παιχνίδι, αποθήκευση δεδομένων παίκτη και διαμόρφωση δυναμικών ρυθμίσεων. Χρησιμοποιώντας το Cloud Code της Unity για την συγγραφή κώδικα χωρίς κατάσταση από την πλευρά του διακομιστή σε μια πλήρως διαχειριζόμενη υποδομή, μπορείτε να εστιάσετε κυρίως στην ανάπτυξη της λογικής του παιχνιδιού σας. Επειδή ο κώδικας αυτός εκτελείται σε ένα ασφαλές sandbox περιβάλλον, μπορεί να προστατευτεί η λογική του παιχνιδιού από την αντίστροφη μηχανική. Ακόμα δίνει τη δυνατότητα στο χρήστη να αναπτύξει άμεσα κώδικα σε ζωντανά παιχνίδια χωρίς να διαθέσει μια ενημέρωση πελάτη, η οποία θα απαιτούσε χρόνο από τους χρήστες να ενημερώσουν τις εγκαταστάσεις τους.
- Cloud Content Delivery (CCD): Το Cloud Content Delivery (CCD) είναι μια διαχειριζόμενη υπηρεσία cloud που φιλοξενεί και παραδίδει περιεχόμενο στους χρήστες της εφαρμογής σε όλο τον κόσμο χωρίς να χρειάζεται να εγκαταστήσουν ξανά μια νέα έκδοση της εφαρμογής. Η υπηρεσία είναι πλήρως ενσωματωμένη στην πλατφόρμα ανάπτυξης Unity, εξοικονομώντας μήνες από τη δημιουργία και τη διατήρηση μίας παρόμοιας υπηρεσίας. Το CCD είναι πολύτιμο για πλούσια σε περιεχόμενο, ζωντανά παιχνίδια ή εφαρμογές που απαιτούν ενημερώσεις περιεχομένου σε τακτική βάση.
- Cloud Save: Μπορείτε να χρησιμοποιήσετε την υπηρεσία Cloud Save της Unity για να αποθηκεύσετε μόνιμα δεδομένα παίκτη (όπως η πρόοδος του παιχνιδιού) από ένα παιχνίδι στο cloud, καθιστώντας το ανεξάρτητο από τη συσκευή. Δεδομένου ότι βασίζεται στο cloud, τα δεδομένα του παίκτη είναι προσβάσιμα οπουδήποτε και σε όλες τις συσκευές, μειώνοντας την απώλεια δεδομένων όταν ένας παίκτης αλλάζει συσκευές ή επανεγκαθιστά ένα παιχνίδι.
- Remote Config: Το Remote Config είναι μια υπηρεσία cloud η οποία δίνει την δυνατότητα για τον συντονισμό του σχεδιασμού του παιχνιδιού χωρίς την

ανάπτυξη νέων εκδόσεων της εφαρμογής. Αποτελείται από ένα σύνολο παραμέτρων λέξη κλειδί-τιμή (Key-Value) σε χώρο ονομάτων και ο χρήστη μπορεί προαιρετικά να ορίσει ένα σύνολο τιμών που παρακάμπτουν ή προσθέτουν σε αυτές τις παραμέτρους.

- **User Generated Content:** Το περιεχόμενο που δημιουργείται από το χρήστη (User Generated Content) παρέχει μια ολοκληρωμένη λύση για τη δημιουργία και τη διαχείριση περιεχομένου, συμπεριλαμβανομένης της δημοσίευσης, της αποθήκευσης, της επιμέλειας, της εποπτείας και της διανομής.

2.3.4 *Ανάπτυξη και λειτουργία (DevOps)*

- **Build Automation:** Το Unity Build Automation είναι μια λύση συνεχούς ενοποίησης η οποία δημιουργεί αυτόματα εκδόσεις πολλαπλών πλατφορμών στο Cloud μέσα σε λίγα λεπτά. Ο προγραμματιστής μπορεί να κατευθύνει το Build Automation προς το σύστημα ελέγχου έκδοσης για να αυτοματοποιήσει νέες εκδόσεις, να εντοπίσει προβλήματα νωρίτερα και να επαναλάβει τις εκδόσεις του πιο αποτελεσματικά. Το Build Automation υποστηρίζει τα πιο δημοφιλή συστήματα ελέγχου εκδόσεων και μπορεί να δημιουργήσει εκδόσεις για πολλές πλατφόρμες ταυτόχρονα.
- **Unity Version Control (VCS):** Το Unity VCS είναι ένα σύστημα ελέγχου έκδοσης που βοηθά τις ομάδες να επικεντρωθούν στην παράδοση εργασίας, μία εργασία τη φορά. Οι ομάδες μπορούν να συνεργαστούν πιο αποτελεσματικά με ένα αποδοτικό, εύχρηστο σύστημα ελέγχου έκδοσης (VCS) που έχει σχεδιαστεί για προγραμματιστές, καλλιτέχνες και σχεδιαστές.

2.3.5 *Πολλοί παίκτες (Multiplayer)*

- **Game Server Hosting (Multiplay):** Το Game Server Hosting είναι μια αποκλειστική υπηρεσία φιλοξενίας η οποία αφαιρεί την πολυπλοκότητα της εκτέλεσης και της λειτουργίας μιας υποδομής για πολλούς παίκτες σε κλίμακα. Ακόμα η ομάδα ανάπτυξης του παιχνιδιού μπορεί να επικεντρωθεί στη δημιουργία ελκυστικών εμπειριών για τον παίκτη.
- **Lobby:** Το Lobby παρέχει δύο κύριες ροές χρησιμοποιώντας δημόσια λόμπι και ιδιωτικούς αγώνες (matches) για τους παίκτες για τη δημιουργία και την εύρεση συνεδριών παιχνιδιού (game sessions). Επίσης διευκολύνει την ομαδοποίηση των παικτών και τις ρυθμίσεις διαμόρφωσης πριν από τις συνεδρίες παιχνιδιού.
- **Matchmaker:** Το Matchmaker είναι μέρος της αυξανόμενης σουίτας των υπηρεσιών πολλών παικτών της Unity που έχουν σχεδιαστεί για να βοηθούν στη δημιουργία και στο χειρισμό παιχνιδιών για πολλούς παίκτες ανεξάρτητα από τη μηχανή παιχνιδιού που χρησιμοποιείται.
- **Relay:** Το Relay είναι μια λύση δικτύωσης που συνδέει τους παίκτες με τρόπο peer-to-peer χρησιμοποιώντας ένα περιβάλλον διακομιστή-πελάτη, έτσι διευκολύνει τη δημιουργία συνδέσεων μεταξύ των παικτών, διατηρώντας παράλληλα το απόρρητό τους.

2.3.6 *Αναλυτικά στοιχεία και εμπλοκή παικτών (Analytics and player engagement)*

- **Analytics:** Το Unity Analytics παρέχει μια ολοκληρωμένη λύση δεδομένων και ανάλυσης που έχει σχεδιαστεί για να υποστηρίζει ολόκληρο το στούντιο. Το Analytics επιτρέπει στα στούντιο να κατανοούν εύκολα την απόδοση του παιχνιδιού και τις συμπεριφορές των παικτών.
- **Game Overrides:** Ο χρήστης μπορεί να ρυθμίσει τα Game Overrides χρησιμοποιώντας δεδομένα του Unity Analytics, ώστε άλλες υπηρεσίες Unity θα μπορούν να στοχεύουν ένα Κοινό (Audience) με εξατομικευμένη συμπεριφορά εντός του παιχνιδιού. Η λειτουργία Κοινό συγκρίνει το προφίλ ενός παίκτη με καθορισμένα κριτήρια για να δει αν ταιριάζει στο καθορισμένο Κοινό.
- **Push Notifications:** Τα Push Notifications είναι μια δυνατότητα που μπορεί να χρησιμοποιήσει ένας χρήστης για να προγραμματίσει τα εμπλουτισμένα μηνύματα push σε μια επιλογή επιλεγμένων παικτών. Χρησιμοποιώντας τα Push Notifications, οι παίκτες ενημερώνονται για τα πράγματα που συμβαίνουν στο παιχνίδι τους και εκτός του παιχνιδιού. Ο χρήστης μπορεί να μετρήσει πόσοι παίκτες έχουν ανοίξει τα Push Notifications και μετά έχουν ξεκινήσει το παιχνίδι ως αποτέλεσμα.

2.3.7 Κοινότητα (Community)

- **Friends:** Η υπηρεσία Friends επιτρέπει στην ομάδα ανάπτυξης να ενισχύσει την αφοσίωση των παικτών παρέχοντας στους παίκτες ένα κοινωνικό σύστημα που λειτουργεί καλά και με άλλες υπηρεσίες της Unity.
- **Leaderboards:** Τα Leaderboards είναι μια δυνατότητα της Unity Gaming Services που υποστηρίζει την αποθήκευση, την ταξινόμηση και την κατάταξη των βαθμολογιών των παικτών.
- **Moderation:** Το Unity Moderation κάνει τη διαχείριση τοξικότητας προσιτή, αποτελεσματική και διορατική, παρέχοντας τα εργαλεία στον χρήστη που χρειάζεται για να αναπτύξει και να διατηρήσει υγιείς τις κοινότητες στα παιχνίδια.
- **Safe Voice:** Το Safe Voice είναι μια υπηρεσία φωνητικών αναλυτικών στοιχείων που βασίζεται στη μηχανική εκμάθηση. Αυτή καταγράφει τις φωνητικές επικοινωνίες του παίκτη και παρέχει στους χρήστες ανάλυση της ηχογράφησης και της ίδιας της ηχογράφησης για την υποστήριξη της εποπτείας των πελατών.
- **Voice and Text Chat (Vivox):** Το Voice and Text Chat παρέχει τεχνολογία επικοινωνίας συνομιλίας φωνής και κειμένου που μπορεί να ενσωματωθεί και να κλιμακωθεί σε παιχνίδια όλων των μεγεθών. Το Voice and Text Chat είναι μία ανεξάρτητη μηχανή που χρησιμοποιείται και λειτουργεί με όλες τις μεγάλες πλατφόρμες, συμπεριλαμβανομένων των Xbox®, PlayStation®, Windows, Nintendo Switch™, Android και iOS. Πρόσθετες λειτουργίες, όπως μετατροπή κειμένου σε ομιλία, ομιλία σε κείμενο και εγγραφή από την πλευρά του διακομιστή, μπορούν να βοηθήσουν στην καταπολέμηση της τοξικότητας των παικτών και να κάνουν τα παιχνίδια προσβάσιμα σε όλους.

2.3.8 Αναφορά σφαλμάτων (Crash reporting)

- **Cloud Diagnostics:** Το Cloud Diagnostics είναι μια σουίτα εργαλείων με δυνατότητα cloud που βοηθούν στη συλλογή και τον εντοπισμό πιθανών ζητημάτων στο παιχνίδι που έχει δημιουργηθεί στο Unity. Η συλλογή αναφορών σφαλμάτων και εξαιρέσεων (exceptions), καθώς και τα σχόλια των χρηστών βοηθούν στην καλύτερη διάγνωση των προβλημάτων και στην εξασφάλιση μιας ομαλής εμπειρίας παιχνιδιού.
- **Cloud Diagnostics Advanced:** Το Cloud Diagnostics Advanced είναι ένα εργαλείο αναφοράς σφαλμάτων και crash που εντοπίζει και επιλύει ζητήματα. Επίσης παρακολουθεί τη σταθερότητα στα παιχνίδια τα οποία έχουν δημιουργηθεί με τη Unity. Το Cloud Diagnostics Advanced υποστηρίζεται από την Backtrace, μια εταιρεία της Sauce Labs.

2.3.9 Οικονομία παιχνιδιού (Game economy)

- **Economy:** Η υπηρεσία Unity's Economy παρέχει έναν εύκολο τρόπο για τη δημιουργία, τη διαχείριση και τη δημοσίευση ενός συστήματος οικονομίας που θα χρησιμοποιηθεί στο παιχνίδι. Η διαχείριση της οικονομίας του παιχνιδιού από έναν κεντρικό πίνακα ελέγχου διέπει τη δημοσίευση των πόρων του παιχνιδιού και επιτρέπει επίσης την προσθήκη προσαρμοσμένων δεδομένων.
- **In-App Purchasing (IAP):** Το Unity IAP διευκολύνει την υλοποίηση αγορών εντός της εφαρμογής στα πιο δημοφιλή καταστήματα εφαρμογών.

(Unity Gaming Services: Live Game Development Platform, 2024) (Overview of services, 2024)

2.4 Netcode for GameObjects (NGO)

Το Netcode for GameObjects (NGO) είναι η επίσημη βιβλιοθήκη δικτύωσης της Unity για την ανάπτυξη παιχνιδιών για πολλούς παίκτες. Έχει σχεδιαστεί για να παρέχει μια σταθερή βάση για τη δημιουργία δικτυωμένων εμπειριών παιχνιδιού, είτε για μικρά συνεργατικά παιχνίδια είτε για τίτλους πολλών παικτών μεγάλης κλίμακας. Ακολουθούν τα βασικά χαρακτηριστικά του Netcode για GameObjects:

2.4.1 API υψηλού επιπέδου (High-level API):

Το NGO προσφέρει ένα API υψηλού επιπέδου που απλοποιεί κοινές εργασίες δικτύωσης. Αυτό περιλαμβάνει το χειρισμό δημιουργίας δικτυωμένων αντικειμένων (networked objects spawning), το συγχρονισμό μεταβλητών (Network Variables) και τη διαχείριση των συνδέσεων των παικτών.

2.4.2 Δικτυωμένα αντικείμενα (Networked Objects):

Στο NGO επισημαίνονται αντικείμενα ως δικτυωμένα, επιτρέποντάς τους να συγχρονίζονται μεταξύ των πελατών (clients) και του διακομιστή (server). Αυτό περιλαμβάνει τη χρήση στοιχείων NetworkObject που χειρίζονται την κατάσταση και την ιδιοκτησία αυτών των αντικειμένων.

2.4.3 Συμπεριφορές δικτύου (Network Behaviors):

Τα στοιχεία που πρέπει να συγχρονίσουν δεδομένα σε όλο το δίκτυο προέρχονται από την κλάση `NetworkBehaviour`. Αυτό δίνει την δυνατότητα να οριστούν ποιες μεταβλητές και συναρτήσεις θα πρέπει να συγχρονίζονται και πώς θα πρέπει να χειρίζονται.

2.4.4 *Μεταβλητές δικτύου (Network Variables):*

Αυτές είναι δομές δεδομένων που συγχρονίζουν αυτόματα τις τιμές τους σε όλο το δίκτυο. Είναι χρήσιμες για τη διαχείριση απλών τύπων δεδομένων, όπως ακέραιους αριθμούς, δεκαδικούς και προσαρμοσμένες δομές (struct), διασφαλίζοντας ότι όλοι οι πελάτες έχουν τα ίδια δεδομένα.

2.4.5 *RPC Κλήσεις απομακρυσμένης διαδικασίας (Remote Procedure Calls):*

Το NGO υποστηρίζει κλήσεις απομακρυσμένης διαδικασίας και στον πελάτη και στο διακομιστή. Αυτό επιτρέπει να καλούνται λειτουργίες σε άλλους συμμετέχοντες στο δίκτυο, επιτρέποντας πολύπλοκες αλληλεπιδράσεις και μηχανισμούς παιχνιδιού.

2.4.6 *Διαχείριση σκηνής (Scene Management):*

Η βιβλιοθήκη παρέχει εργαλεία για τη διαχείριση των μεταβάσεων σκηνής (scene transitions) και διασφαλίζει ότι όλοι οι παίκτες είναι συγχρονισμένοι όταν αλλάζει η σκηνή. Αυτό είναι ζωτικής σημασίας για τη διασφάλιση μιας συνεπής κατάστασης παιχνιδιού σε όλους τους πελάτες.

2.4.7 *Προσαρμογή και Επεκτασιμότητα (Customization and Extensibility):*

Το NGO παρέχει πολλές λειτουργίες από μόνο του. Ωστόσο έχει σχεδιαστεί για να είναι επεκτάσιμο. Ακόμα παρέχει τη δυνατότητα να προσαρμοστεί η συμπεριφορά του για να ταιριάζει στις συγκεκριμένες ανάγκες του κάθε παιχνιδιού.

2.4.8 *Ενσωμάτωση με τις Υπηρεσίες της Unity (Integration with Unity Services):*

Το NGO έχει σχεδιαστεί για να ενσωματώνεται καλά με το οικοσύστημα της Unity, συμπεριλαμβανομένων των υπηρεσιών Unity Relay και Unity Lobby. Αυτές οι υπηρεσίες βοηθούν στη διαχείριση των περιόδων σύνδεσης για πολλούς παίκτες και την εύρεση παικτών οι οποίοι ψάχνουν να παίξουν (matchmaking).

3. Παρουσίαση και ανάλυση του παιχνιδιού

Εδώ θα παρουσιαστεί η υλοποίηση του παιχνιδιού της παρούσας διπλωματικής εργασίας, καθώς θα αναλυθούν τα βήματα για την κατασκευή ενός τέτοιου παιχνιδιού και τις δυσκολίες που αντιμετωπίστηκαν.

3.1 Παρουσίαση του παιχνιδιού



Το όνομα του παιχνιδιού είναι “GyroRush”. Το παιχνίδι υποστηρίζει έναν ή περισσότερους παίκτες τόσο σε τοπικό όσο και σε διαδικτυακό επίπεδο. Οπότε είναι ένα τρισδιάστατο συνεργατικό παιχνίδι μαγειρικής στο οποίο οι παίκτες λειτουργούν ένα μαγαζί που πουλάει φαγητό. Το κύριο φαγητό του παιχνιδιού είναι τα σουβλάκια και ο γύρος. Ο κάθε παίκτης χειρίζεται έναν χαρακτήρα όπου ο καθένας μπορεί να πάρει παραγγελία, να μαγειρέψει και να σερβίρει φαγητό στους πελάτες που έρχονται στο μαγαζί. Οι παίκτες έχουν συνολικά πέντε «ζωές». Οι παίκτες χάνουν μία «ζωή» όταν ένας πελάτης δεν εξυπηρετηθεί στον προβλεπόμενο χρόνο και αποχωρήσει από το μαγαζί. Υπάρχουν τρεις τύποι πελάτη ο αργός, ο γρήγορος και ο κανονικός με αντίστοιχη υπομονή ο καθένας τους. Το παιχνίδι διαθέτει δύο λειτουργίες. Η πρώτη λειτουργία είναι η ιστορία. Οι παίκτες ξεκινούν το παιχνίδι από μια απλή καντίνα πουλώντας σουβλάκια. Οι πελάτες πληρώνουν για τα σουβλάκια. Αυτά τα χρήματα τα χρησιμοποιούν για να ανεβαίνουν επίπεδο. Αγοράζοντας πράγματα για την αναβάθμιση του μαγαζιού τους. Η δεύτερη λειτουργία είναι η εξάσκηση στο μαγείρεμα. Οι παίκτες μπορούν να εξασκηθούν στη δημιουργία πιάτων/γευμάτων χωρίς την ύπαρξη των πελατών. Επιπλέον και στις δύο λειτουργίες ο παίκτης μπορεί να διαλέξει χαρακτήρες με διαφορετικές ειδικότητες για παράδειγμα μάγειρας και σερβιτόρος. Αυτοί οι χαρακτήρες είναι διαθέσιμοι στον παίκτη αφού τις ξεκλειδώσει δίνοντας χρήματα τα οποία έχει μαζέψει από την εξυπηρέτηση των πελατών κατά τη διάρκεια της ημέρας. Ο κάθε γύρος του παιχνιδιού είναι και μια μέρα στο παιχνίδι. Στο τέλος κάθε ημέρας το παιχνίδι αποθηκεύεται αυτόματα ώστε να μπορεί να συνεχιστεί αργότερα από τον παίκτη.



Εικόνα 7 Στιγμιότυπο οθόνης κατά την ανάπτυξη του παιχνιδιού

3.2 Game Design Document

Για την παρούσα διπλωματική εργασία δημιουργήθηκε ένα απλό έγγραφο σχεδίασης παιχνιδιού (GDD) για να χρησιμοποιηθεί σαν ένα μικρό παράδειγμα αλλά και σαν ένας οδηγός προς την ανάπτυξη του παιχνιδιού.

Επισκόπηση (Overview)

Όνομα παιχνιδιού: GyroRush

Είδος παιχνιδιού: Συνεργατικό (τοπικό και online), μαγειρική, 3D, arcade

Εμπνεύσεις (Inspirations)

Diner Bros από JAYFLGAMES https://store.steampowered.com/app/846800/Diner_Bros/

Overcooked από Ghost Town Games, Team17

<https://store.steampowered.com/app/448510/Overcooked/>

PlateUp! από It's happening <https://store.steampowered.com/app/1599600/PlateUp/>

Διαφορές (Differences)

Σουβλάκια, Γύρος και στοιχεία από παιχνίδια ρόλων (ειδικότητες χαρακτήρων).

Στόχος & Δομή (Objective & Structure)

Ο στόχος του παιχνιδιού είναι η όσον το δυνατόν καλύτερη εξυπηρέτηση των πελατών χωρίς να αποχωρήσουν από το μαγαζί δυσαρεστημένοι. Ο παίκτης θα πρέπει να πάρει παραγγελία από τους πελάτες που έχουν καθίσει, να τους ετοιμάσει το σωστό φαγητό και να τους σερβίρει τον καθένα στο δικό του εύλογο χρονικό διάστημα. Η πίστα θα είναι διαμορφωμένη σαν ένα ψητοπωλείο, με τραπέζια και καρέκλες και μια μεγάλη κουζίνα. Ο στόχος του παίκτη είναι η αναβάθμιση του μαγαζιού του με την συλλογή χρημάτων από τις πωλήσεις φαγητού στους πελάτες και να προχωρήσει σε όσο πιο μεγάλη μέρα παιχνιδιού μπορεί. Κάθε γύρος του παιχνιδιού είναι μια ημέρα η οποία έχει διάρκεια τριών λεπτών. Σε κάθε γύρο όλοι οι παίκτες μαζί έχουν πέντε «ζωές». Οι «ζωές» χάνονται από τους παίκτες όταν δεν προλαβαίνουν να εξυπηρετήσουν έναν πελάτη. Όταν χαθούν όλες οι «ζωές» τελειώνει και ο γύρος του παιχνιδιού.

Εμπειρία παίκτη (Player Experience)

Η εμπειρία του παίκτη αποτελείται από ένα συνδυασμό διαχείρισης χρόνου, στρατηγικού πολλαπλών εργασιών και συνεργατικού παιχνιδιού. Η κλιμακούμενη δυσκολία και οι

ελκυστικοί μηχανισμοί του παιχνιδιού συμβάλλουν στον εθιστικό και ευχάριστο χαρακτήρα του, καθιστώντας το μια καλή επιλογή τόσο για ατομικό όσο και για συνεργατικό παιχνίδι.

Μηχανισμοί (Mechanics)

Οι τρεις κύριοι μηχανισμοί του παιχνιδιού είναι οι παραγγελίες, το μαγείρεμα και το σερβίρισμα του φαγητού. Πιο αναλυτικά:

1. Ένας πελάτης έρχεται ανά τακτά χρονικά διαστήματα στο μαγαζί.
2. Ο πελάτης κάθεται σε μια άδεια καρέκλα ή αν δεν υπάρχει άδεια καρέκλα περιμένει στην ουρά έξω από το μαγαζί. Μόλις αδειάσει μια καρέκλα μπαίνει στο μαγαζί και κάθεται στη καρέκλα.
3. Ο πελάτης περιμένει να έρθει ένας παίκτης για να του πάρει παραγγελία.
4. Ο παίκτης ετοιμάζει την παραγγελία.
5. Ο παίκτης σερβίρει την παραγγελία στον πελάτη.
6. Ο πελάτης πληρώνει αυτόματα και ξεκινάει να τρώει το φαγητό του.
7. Ο πελάτης τελειώνει το φαγητό του και φεύγει από το κατάστημα και αδειάζει η καρέκλα.

Οι δευτερεύων μηχανισμοί είναι η διαχείριση χώρου και χρόνου μέσα στο παιχνίδι και η σωστή συνεννόηση. Πιο αναλυτικά:

- Χώρος: Οι ελεύθεροι πάγκοι μέσα στην χώρο είναι συγκεκριμένοι και ο παίκτης δεν μπορεί να αφήσει τα φαγητά όπου θέλει ούτε να τα αφήνει στον αέρα και να πέσουν κάτω. Μπορεί είτε να τα αφήσει σε έναν άδειο πάγκο είτε να τα πετάξει στον κάδο σκουπιδιών, οπότε δεν μπορεί να ετοιμάζει άσκοπα φαγητά. Αλλά είναι χρήσιμο να υπάρχουν έτοιμα μερικά υλικά φαγητού για να φτιάχνονται γρήγορα τα πιάτα ειδικά όταν ενεργοποιείται ο μηχανισμός «ώρα αιχμής».
- Χρόνος: Ο χρόνος αναμονής κάθε πελάτη είναι διαφορετικός ανά τύπο πελάτη. Επομένως, ο παίκτης είναι καλύτερο να εξυπηρετήσει πρώτα έναν πελάτη που έχει μικρή υπομονή από έναν πελάτη που ήρθε και κάθισε νωρίτερα στο μαγαζί αλλά με μεγαλύτερη υπομονή. Ακόμα, η μέρα χωρίζεται σε τρία κομμάτια. Στο μεσαίο κομμάτι της ημέρας ενεργοποιείται ένας μηχανισμός ο οποίος ονομάζεται «ώρα αιχμής» (rush hour). Σε αυτό το χρονικό διάστημα οι πελάτες έρχονται στο μαγαζί με πιο γρήγορο ρυθμό από ότι συνήθως.
- Οι πελάτες έχουν ξεχωριστή υπομονή για κάθε στάδιο τους. Πιο συγκεκριμένα:
 1. Αναμονή στην ουρά.
 2. Αναμονή για την παραγγελία φαγητού.
 3. Αναμονή για το σερβίρισμα της παραγγελίας τους.

Αν τελειώσει η υπομονή τους σε οποιοδήποτε στάδιο της παραγγελίας τους, οι πελάτες αποχωρούν θυμωμένοι από το μαγαζί και οι παίκτες χάνουν μία «ζωή».

- Συνεννόηση: Στην λειτουργία για πολλούς παίκτες, οι παίκτες συνηθίζεται να διαχωρίζουν τους ρόλους, δηλαδή κάποιοι παίκτες ασχολούνται με το μαγείρεμα και άλλοι παίκτες με το σερβίρισμα του φαγητού στους πελάτες. Οπότε είναι πολύ σημαντικό να υπάρχει σωστή συνεννόηση για τις αρμοδιότητες του κάθε παίκτη. Σε διαφορετική περίπτωση δεν θα εξυπηρετούνται σωστά οι πελάτες και θα αποχωρούν από το μαγαζί.

Επιπλέον στο τέλος κάθε ημέρας/γύρου το παιχνίδι αποθηκεύεται αυτόματα στο δίσκο για να μπορεί να γίνει ανάκτηση του παιχνιδιού για να συνεχιστεί αργότερα ή σε περίπτωση που χαθεί ένας γύρος.

Χειρισμός (Controls)

Ο παίκτης θα έχει τη δυνατότητα να χειριστεί τον χαρακτήρα του είτε με το πληκτρολόγιο είναι με το αγαπημένο του χειριστήριο (gamepad). Επίσης τα κουμπιά θα μπορούν να αλλαχτούν εντός του παιχνιδιού. Αναλυτικά τα κουμπιά του παιχνιδιού είναι τα παρακάτω:

Χειρισμός με πληκτρολόγιο:

- WASD ή τα βελάκια: Για το περπάτημα και την περιστροφή του χαρακτήρα.
- E: Για την πρωτεύων αλληλεπίδραση. Δηλαδή να πάρει ή να αφήσει αντικείμενα (υλικά, πιάτα κ.α.).
- F: Για τη δευτερεύων αλληλεπίδραση. Δηλαδή χρήση αντικειμένων ή συσκευών και αλληλεπίδραση με πελάτες ή άλλους παίκτες. π.χ. κόνιμο υλικού, παράδοση πιάτου στον πελάτη κ.α..
- Spacebar: Για να σταματήσει να περπατάει ο χαρακτήρας και να μπορεί μόνο να περιστρέφεται με τα WASD ή τα βελάκια.

Χειρισμός με χειριστήριο:

- Αριστερός μοχλός: Για το περπάτημα και την περιστροφή του χαρακτήρα.
- Κάτω κουμπί (A Xbox, X PlayStation): Για την πρωτεύων αλληλεπίδραση.
- Αριστερό κουμπί (X Xbox, τετράγωνο PlayStation): Για τη δευτερεύων αλληλεπίδραση.
- Πίσω αριστερή σκανδάλη (Left trigger Xbox, L2 PlayStation): Για να σταματήσει να περπατάει ο χαρακτήρας.

Στυλ (Style)

Γραφικά απλά 3D με λίγα πολύγωνα (low-poly), παρατίθενται οι παρακάτω εικόνες.



Εικόνα 8 Διάφοροι χαρακτήρες (αριστερά) και οι διαφορετικοί πελάτες (δεξιά)

Οι ήχοι θα είναι χαριτωμένοι και διασκεδαστικοί.

Χαρακτήρες (Characters)

Ο παίκτης έχει δυνατότητα να επιλέξει το χρώμα που θα έχει ο χαρακτήρας του, καθώς και να επιλέξει την ειδικότητα του π.χ. απλός, μάγειρας, σερβιτόρος. Ανάλογα με την ειδικότητα του χαρακτήρα θα έχει και άλλες ικανότητες. Αυτές οι ειδικότητες ξεκλειδώνονται με τα χρήματα που μαζεύουν οι παίκτες από τις παραγγελίες των πελατών. Παρακάτω παρουσιάζονται αναλυτικά οι ειδικότητες του κάθε χαρακτήρα:

Ειδικότητα	Ταχύτητα Περπατήματος	Ταχύτητα Κοψίματος Φαγητού	Αντικείμενα που κρατάει στα χέρια
Βασικός	Κανονική	Κανονική	Ένα
Μάγειρας	Αργή	Γρήγορη	Δύο υλικά φαγητού
Σερβιτόρος	Γρήγορη	Μισή	Δύο πιάτα φαγητού
Πολυμήχανος (Multi)	Κανονική	Κανονική	Δύο, είτε υλικά είτε πιάτα φαγητού
Σούπερ Μάγειρας	Λιγότερη Αργή	Διπλάσια	Τρία υλικά φαγητού

Σούπερ Σερβιτόρος	Πιο Γρήγορη	Λιγότερη Κανονική	από	Τρία πιάτα φαγητού
Σούπερ Πολυμήχανος (Super Multi)	Γρηγορότερη από κανονική	Γρηγορότερη κανονική	από	Τρία, είτε υλικά είτε πιάτα φαγητού

Ο κάθε διαφορετικός τύπος πελάτη έχει διαφορετική εμφάνιση για να ξεχωρίζει.

Επίσης διάφοροι πελάτες θα περνάνε τυχαία μέσα από το μαγαζί για να εμποδίζουν και να δυσκολεύουν τους παίκτες.

Ρύθμιση & Θέμα (Setting & Theme)

Το παιχνίδι λαμβάνει χώρα στον τωρινό κόσμο και παρόν χρόνο. Εκφράζει την έντονη αύξηση κίνησης στα ταχυφαγεία και στα εστιατόρια της Αθήνας.

Ιστορία / Επιβράβευση (Story / Reward)

Το παιχνίδι αφηγείται μια μικρή ιστορία. Οποιοσδήποτε μπορεί να ξεκινήσει από μια καντίνα στον δρόμο που πουλάει σουβλάκια και μαζεύει χρήματα, ώστε να καταλήξει να έχει το δικό του μαγαζί. Η ιστορία είναι εμπνευσμένη από το «το καρότσι του γιατρού» στο Ψυρρή (Αθήνα).

3.3 Εργαλεία που χρησιμοποιήθηκαν

Εδώ θα γίνει αναφορά των εργαλείων και υπηρεσιών που χρησιμοποιήθηκαν για την ανάπτυξη του παιχνιδιού.

3.3.1 Unity Editor 2022

Το Unity Editor είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης προσαρμοσμένο για τη δημιουργία διαδραστικών εμπειριών 2D, 3D, επαυξημένης πραγματικότητας (AR) και εικονικής πραγματικότητας (VR). Ένα μεγάλο πλεονέκτημά του είναι η δυνατότητα δημιουργίας του περιεχομένου μια φορά και η ανάπτυξη του σε όλες τις μεγάλες πλατφόρμες AR, VR, κινητών, ηλεκτρονικών υπολογιστών και κονσολών. Η ακριβής έκδοση του Unity Editor που χρησιμοποιήθηκε είναι η 2022.3.18f1. (Real-Time 3D Development Platform & Editor | Unity, 2024)

3.3.2 Visual Studio 2022

Το Visual Studio της Microsoft είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης IDE (Integrated Development Environment) για την επεξεργασία, τον εντοπισμό σφαλμάτων και τη δημιουργία κώδικα. Εκτός από ένα τυπικό επεξεργαστή και ένα εντοπισμό σφαλμάτων που παρέχουν τα περισσότερα IDE, το Visual Studio περιλαμβάνει μεταγλωττιστές, εργαλεία συμπλήρωσης κώδικα, γραφικούς σχεδιαστές και πολλές άλλες δυνατότητες για τη βελτίωση της διαδικασίας ανάπτυξης λογισμικού και την εύκολη ενσωμάτωση του με το Unity Editor. Χρησιμοποιήθηκε για την επεξεργασία των Unity scripts. (Visual Studio: IDE and Code Editor for Software Developers and Teams, 2024)

3.3.3 Corel Paintshop Pro 2022

Το Paintshop Pro της Corel είναι ένα πρόγραμμα επεξεργασίας εικόνων, γραφικών και διανυσματικών γραφικών. Οι χρήστες μπορούν να επωφεληθούν από προηγμένες επιλογές όπως επίπεδα, μάσκες και επεξεργασία αρχείων RAW, καθώς και δημιουργικά εφέ και φίλτρα για να βελτιώσουν τις εικόνες τους. Επιπλέον, το PaintShop Pro περιλαμβάνει εργαλεία για την οργάνωση και τη διαχείριση συλλογών φωτογραφιών. Το εργαλείο αυτό χρησιμοποιήθηκε για τη δημιουργία και τη διόρθωση

του λογότυπου, των εικονιδίων και των γραφικών της διεπαφής του χρήστη. Για παράδειγμα η αλλαγή του μαχαιριού στο λογότυπο και η εισαγωγή της διαφάνειας.

3.3.4 *Cool Edit Pro*

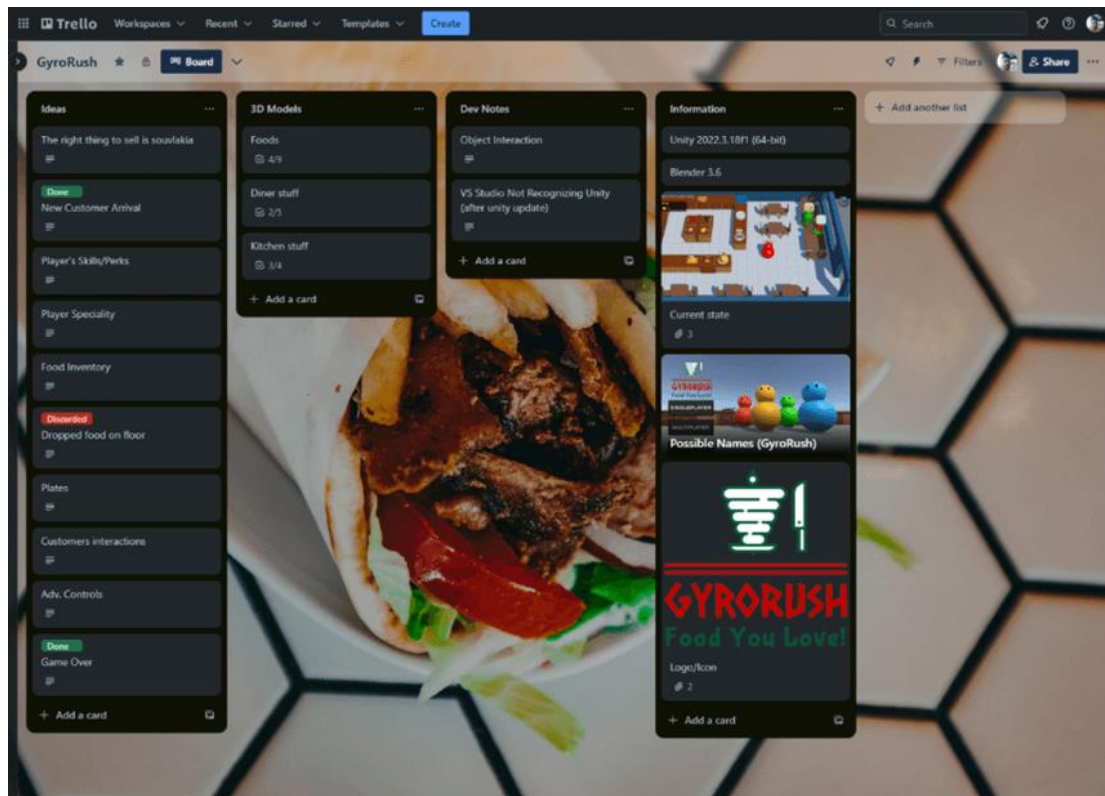
Το Cool Edit Pro της Syntrillium Software είναι ένα πρόγραμμα επεξεργασίας ήχου, που το 2003 το αγόρασε η Adobe και μετονομάστηκε σε Adobe Audition. Χρησιμοποιήθηκε η παλιά έκδοση για την επεξεργασία των ήχων του παιχνιδιού που βρέθηκαν για παράδειγμα επεξεργασία διάρκειας και έντασης ήχου.

3.3.5 *Blender 3.6 LTS*

Το Blender είναι ένα δωρεάν και ανοιχτού κώδικα σουίτα δημιουργίας 3D γνωστή για την ευελιξία και τον ισχυρό συνολικό του εξοπλισμό, που απευθύνεται σε καλλιτέχνες, σχεδιαστές, animator και ερασιτέχνες. Περιλαμβάνει εργαλεία μοντελοποίησης για τη δημιουργία, επεξεργασία και γλυπτική 3D μοντέλων, δυνατότητες animation που περιλαμβάνουν keyframe animation και character rigging, προηγμένους κινητήρες απεικόνισης όπως οι Cycles και Eevee, εργαλεία προσομοίωσης για δυναμική ρευστών (fluid dynamics) και εφέ σωματιδίων (particle effects), ενσωματωμένη σύνθεση και επεξεργασία βίντεο, εκτεταμένο scripting με Python για προσαρμογή και αυτοματισμό, καθώς και μια ζωντανή κοινότητα που παρέχει υποστήριξη και πόρους. Το Blender αποτελεί μια ολοκληρωμένη και ισχυρή λύση για όλες τις πτυχές της δημιουργίας τρισδιάστατου περιεχομένου. Στην παρούσα διπλωματική χρησιμοποιήθηκε για την επεξεργασία υπαρχόντων τρισδιάστατων μοντέλων αλλά και για τη δημιουργία νέων μοντέλων. (blender.org - Home of the Blender project - Free and Open 3D Creation Software, 2024)

3.3.6 *Trello*

Το Trello είναι μια διαδικτυακή εφαρμογή διαχείρισης έργων που χρησιμοποιεί ένα σύστημα πινάκων, λιστών και καρτών για να βοηθήσει άτομα και ομάδες να οργανώσουν και να ιεραρχήσουν εργασίες και έργα. Έχει σχεδιαστεί για να είναι ευέλικτο και εύκολο στη χρήση, καθιστώντας το κατάλληλο για ένα ευρύ φάσμα σκοπών, από προσωπική διαχείριση εργασιών έως ομαδική συνεργασία σε πολύπλοκα έργα. Χρησιμοποιήθηκε για την καταγραφή ιδεών και της προόδου καθ' όλη την διάρκεια της παραγωγής του παιχνιδιού. (Manage Your Team's Projects From Anywhere | Trello, 2024)



Εικόνα 9 Στιγμιότυπο από το Trello του παιχνιδιού που δημιουργήθηκε για την παρόν διπλωματική

3.3.7 LOGO

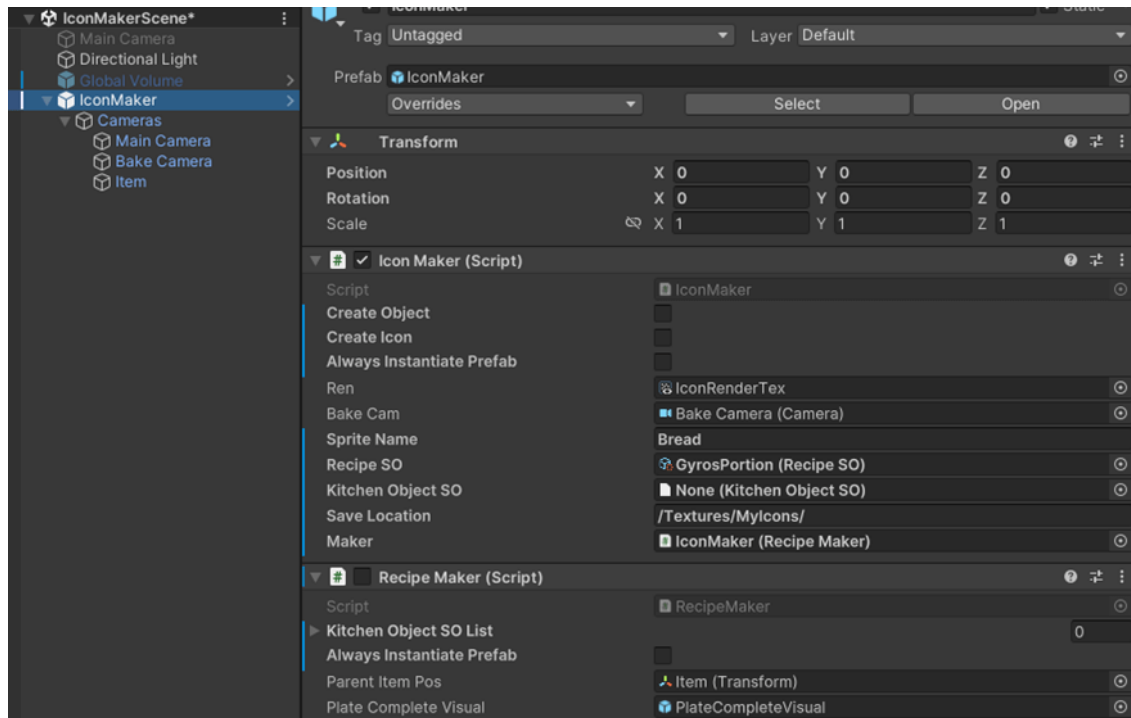
Το LOGO είναι μια διαδικτυακή πλατφόρμα που απλοποιεί τη διαδικασία δημιουργίας επαγγελματικών λογότυπων, προσφέροντας μια φιλική προς το χρήστη διεπαφή εξοπλισμένη με ένα ευρύ φάσμα προσαρμόσιμων προτύπων, γραμματοσειρών και γραφικών. Η πλατφόρμα δίνει τη δυνατότητα δημιουργίας μοναδικών λογότυπων εντελώς δωρεάν. Χρησιμοποιήθηκε για την παραγωγή του λογότυπου του παιχνιδιού. (Free Logo Maker | LOGO.com, 2024)

3.3.8 OBS Studio

Το OBS Studio (Open Broadcaster Software Studio) είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα σχεδιασμένο για εγγραφή βίντεο και ζωντανή ροή (live streaming). Προσφέρει ισχυρές και ευέλικτες δυνατότητες, όπως λήψη και μίξη ήχου/βίντεο σε πραγματικό χρόνο, κωδικοποίηση βίντεο/ήχου υψηλής απόδοσης και προσαρμόσιμες μεταβάσεις (customizable transitions). Οι χρήστες μπορούν να δημιουργήσουν σκηνές από πολλές πηγές, όπως λήψεις παραθύρων (window captures), εικόνες, κείμενο, παράθυρα προγράμματος περιήγησης, κάμερες web και κάρτες λήψης (capture cards). Το OBS Studio υποστηρίζει διάφορες πλατφόρμες ροής όπως το Twitch, το YouTube και το Facebook Live και παρέχει προηγμένες επιλογές διαμόρφωσης για βελτιστοποίηση της απόδοσης και της ποιότητας. Η διαισθητική διεπαφή του, η εκτεταμένη υποστήριξη πρόσθετων και η ενεργή κοινότητα το καθιστούν δημοφιλή επιλογή τόσο για ερασιτέχνες όσο και για επαγγελματίες streamers. Χρησιμοποιήθηκε για την καταγραφή βίντεο του παιχνιδιού ώστε να μεταφορτωθεί στο YouTube (Open Broadcaster Software | OBS, 2024).

3.3.9 Εργαλείο δημιουργίας εικονιδίων (*IconMaker & RecipeMaker*)

Το *IconMaker* μαζί με το *RecipeMaker* είναι εργαλεία που δημιουργήθηκαν για την διευκόλυνση δημιουργίας εικονιδίων για το συγκεκριμένο παιχνίδι. Το *RecipeMaker* δημιουργεί τα αντικείμενα (*GameObject*) της Unity σύμφωνα με μια λίστα αντικειμένων *KitchenObjectSO*. Τα *KitchenObjectSO* (*ScriptableObject*) είναι τα διάφορα υλικά/φαγητά του παιχνιδιού. Το *RecipeMaker* παίρνει σαν είσοδο το “*Plate Complete Visual*” το οποίο είναι ένα prefab της Unity. Αυτό το prefab περιέχει τα *GameObjects* που αντιστοιχούν στα υλικά *KitchenObjectSO* για την εμφάνιση όλων αυτών των υλικών μέσα στο πιάτο. Στην περίπτωση που το συγκεκριμένο *KitchenObjectSO* δεν βρεθεί στο prefab, για παράδειγμα κάποιο υλικό φαγητού που δεν υπάρχει σε συνταγή πιάτου φαγητού, θα δημιουργήσει το *GameObject* σύμφωνα με το prefab που έχει πάνω του το *KitchenObjectSO*. Ακόμα προσθέτει το “*Quick Outline*” στο νέο αντικείμενο (*GameObject*) το οποίο δημιουργεί ένα μαύρο περίγραμμα γύρω από το φαγητό. Το *IconMaker* είναι υπεύθυνο για όλη την διαχείριση και την δημιουργία του εικονιδίου. Σε αυτό επιλέγουμε ποιο υλικό φαγητού (*KitchenObjectSO*) ή συνταγή φαγητού (*RecipeSO*) θέλουμε να φτιάξουμε το εικονίδιο. Στην συνέχεια συμπληρώνουμε το πεδίο “*Save Location*” για το φάκελο που θέλουμε να αποθηκευτεί το εικονίδιο. Έπειτα «*τικάρουμε*» το “*Create Object*” και αυτό δίνει εντολή στο *RecipeMaker* να δημιουργήσει τα αντικείμενα (*GameObject*). Αυτά τα αντικείμενα θα φτιαχτούν κάτω από το *GameObject* “*Item*” [εικόνα 10], μετά πατάμε το *Play* στην Unity και εμφανίζεται το παράθυρο παιχνιδιού στο οποίο μπορούμε να δούμε πως φαίνονται τα αντικείμενα στην κάμερα. Υπάρχουν δύο κάμερες, η “*Main Camera*” για να βλέπουμε τι γίνεται και η “*Bake Camera*” για να κάνει render το texture το οποίο και θα αποθηκευτεί σε μορφή .png. Υπάρχει η δυνατότητα να περιστρέψουμε ή να μετακινήσουμε το νέο *GameObject* που είναι κάτω από το “*Item*” ώστε να βρίσκεται στο κέντρο της κάμερας. Τέλος, «*τικάρουμε*» το κουμπί “*Create Icon*”, το οποίο θα αποθηκεύσει ότι βλέπει η κάμερα “*Bake Camera*” σε ένα αρχείο .png με όνομα αρχείου την τιμή του πεδίου “*Sprite Name*” μέσα στον φάκελο που επιλέξαμε στο πεδίο “*Save Location*”. Το πεδίο “*Sprite Name*” θα πάρει αυτόματα την τιμή του από το όνομα που έχει το *RecipeSO* ή *KitchenObjectSO*.



Εικόνα 10 Ο Inspector της Unity για το IconMaker

Παρακάτω ο κώδικας που καλείται στο πάτημα “Create Icon” από το αρχείο IconMaker.cs

```
private void CreateIcon()
{
    if (recipeSO != null)
        spriteName = recipeSO.recipeName;
    else if (kitchenObjectSO != null)
        spriteName = kitchenObjectSO.objectName;

    if (string.IsNullOrEmpty(spriteName))
        spriteName = "temp";

    string filepath = SaveLocation() + spriteName + ".png";

    bakeCam.targetTexture = new RenderTexture(ren.width, ren.height, 32,
RenderTextureFormat.ARGB32, RenderTextureReadWrite.sRGB);
    bakeCam.targetTexture.Release();
    RenderTexture.active = bakeCam.targetTexture;
    bakeCam.Render();

    Texture2D imgPng = new Texture2D(ren.width, ren.height,
TextureFormat.ARGB32, false, true);
    imgPng.ReadPixels(new Rect(0, 0, ren.width, ren.height), 0, 0);

    var pixels = imgPng.GetPixels();
    for (int i = 0; i < pixels.Length; i++)
    {
        if (pixels[i] == Color.white)
        {
            pixels[i] = Color.clear;
        }
    }
    imgPng.SetPixels(pixels);
}
```

```

RenderTexture.active = null;
byte[] bytesPng = imgPng.EncodeToPNG();
File.WriteAllBytes(filepath, bytesPng);
Debug.Log("Icon Created: " + filepath);
}

```

Κύριος κώδικας στο αρχείο RecipeMaker.cs:

```

private void OnEnable()
{
    foreach (Transform child in ParentItemPos)
    {
        DestroyImmediate(child.gameObject);
    }
    if (KitchenObjectSOList.Count > 0)
    {
        _obj = Instantiate(PlateCompleteVisual, ParentItemPos);
        var outline = _obj.AddComponent<Outline>();
        outline.OutlineColor = Color.black;
        outline.OutlineWidth = 10f;
        outline.OutlineMode = Outline.Mode.OutlineVisible;
#ifdef UNITY_EDITOR
        outline.runInEditMode = true;
#endif
    }

    var plateCompleteVisual = _obj.GetComponent<PlateCompleteVisual>();
    plateCompleteVisual.DisableIngredientVisuals();
    foreach (var kitchenObjectSO in KitchenObjectSOList)
    {
        if (AlwaysInstantiatePrefab)
        {
            Instantiate(kitchenObjectSO.prefab, _obj.transform);
        }
        else
        {
            var added = plateCompleteVisual.SetIngredientVisual(new
PlateKitchenObject.OnIngredientEventArgs() { kitchenObjectSO = kitchenObjectSO
}, true);
            if (!added)
            {
                Instantiate(kitchenObjectSO.prefab, _obj.transform);
            }
        }
    }
    enabled = false;
}

```

3.4 Στοιχεία παιχνιδιού (Game Assets)

Τα κύρια assets που χρησιμοποιήθηκαν είναι από το δωρεάν tutorial του Code Monkey <https://unitycodemonkey.com/kitchenchaoscourse.php>. Χρησιμοποιήθηκαν μερικά δωρεάν τρισδιάστατα μοντέλα από διάφορες ιστοσελίδες. Τα υπόλοιπα στοιχεία παιχνιδιού δημιουργήθηκαν από εμένα. Αναλυτικότερα παρακάτω:

3.4.1 Δωρεάν τρισδιάστατα μοντέλα

Τα παρακάτω μοντέλα είναι δωρεάν και επεξεργάστηκαν στο Blender για να ενσωματωθούν στο παιχνίδι.

- "Pita bread" (<https://skfb.ly/osIFp>) by Dan-Levi Tømta is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).
- "Simple Potato" (<https://skfb.ly/6CVN6>) by Blender3D is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).
- "Chef Hat" <https://www.turbosquid.com/3d-models/3d-chef-hat-1784198> by Kylieuh is licensed under turbosquid's 3D Model License Standard <https://blog.turbosquid.com/turbosquid-3d-model-license/>

3.4.2 Στοιχεία παιχνιδιού από το Unity Asset Store

Τα στοιχεία παιχνιδιού που χρησιμοποιήθηκαν από το Unity Asset Store είναι:

- "Quick-outline" <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488> by Chris Nolet is licensed under Standard Unity Asset Store EULA <https://unity.com/legal/as-terms>
- "Quantum Console" <https://assetstore.unity.com/packages/tools/utilities/quantum-console-211046> by QFSW is licensed under Standard Unity Asset Store EULA <https://unity.com/legal/as-terms>

Το "Quick-outline" είναι ένα εργαλείο περιγράμματος στον χώρο του παιχνιδιού (world-space) το οποίο προσθέτει ένα συμπαγές περίγραμμα σε οποιοδήποτε αντικείμενο. Επίσης το εργαλείο προσφέρει πολλές ρυθμίσεις όπως το πάχος και το χρώμα του περιγράμματος, καθώς και το είδος του περιγράμματος. Για παράδειγμα αν το περίγραμμα θα φαίνεται σε όλο το αντικείμενο ή μόνο στο κρυμμένο μέρος του. Χρησιμοποιήθηκε κυρίως για να προστεθεί περίγραμμα στα πιάτα φαγητού. Έτσι δημιουργείται το εικονίδιο με το εργαλείο IconMaker. Ακόμα χρησιμοποιήθηκε στην αλληλεπίδραση παίκτη με παίκτη για να δείξει το περίγραμμα γύρω από τον παίκτη.

Η "Quantum Console" είναι μια ισχυρή και εύκολη στη χρήση κονσόλα εντολών εντός του παιχνιδιού. Ο προγραμματιστής έχει τη δυνατότητα να προσθέσει τις δικές του εντολές στην κονσόλα απλώς προσθέτοντας το attribute "[Command]" στον κώδικα του. Χρησιμοποιήθηκε κυρίως για δοκιμαστικούς σκοπούς, όπως δοκιμές σε λογική κώδικα πριν δημιουργηθεί η διεπαφή του χρήστη. Επίσης χρησιμοποιήθηκε και για cheats όπως την προσθήκη χρημάτων, την προσθήκη «ζωών», την αλλαγή της ημέρας κ.ά.. Ακόμα υλοποιήθηκε και ένα απλό σύστημα συζήτησης τύπου "chat" μεταξύ των συνδεδεμένων παικτών όπου ο κάθε παίκτης μπορεί να στείλει μήνυμα σε όλους τους υπόλοιπους παίκτες.

3.4.3 Στοιχεία παιχνιδιού που δημιουργήθηκαν

Τα υπόλοιπα τρισδιάστατα μοντέλα δημιουργήθηκαν από εμένα στο Blender. Μερικά από αυτά είναι: Σουβλάκι, Μηχανή Γύρου (Γυριέρα), ποδιά μάγειρα, πουκάμισο σερβιτόρου, φριτέζα λαδιού, κομμένες πατάτες τηγανητές, κομμένος γύρος. Το λογότυπο από το LOGO επεξεργάστηκε στο Corel Paintshop Pro, για παράδειγμα έγινε αλλαγή του μαχαιριού που έχει δίπλα από τον γύρο. Επίσης στο ίδιο πρόγραμμα δημιουργήθηκαν δύο εικονίδια που χρειάστηκαν για την διεπαφή του χρήστη, το εικονίδιο της «ζωής» και το εικονίδιο-κουμπί διαχείρισης των αρχείων αποθήκευσης. Αυτά φτιάχτηκαν κάνοντας τη χρήση των εργαλείων για τα διανύσματα που έχει το

συγκεκριμένο πρόγραμμα. Ακόμα στο ίδιο πρόγραμμα έγινε η σμίκρυνση των υφών (textures) όπου χρειαζόταν λόγω της μεγάλης ανάλυσής τους.



Εικόνα 11 Από αριστερά προς τα δεξιά: λογότυπο παιχνιδιού, εικονίδιο «ζωής», εικονίδιο διαχείρισης αρχείων αποθήκευσης

Τα εικονίδια των πιάτων φαγητού καθώς και των υλικών φαγητού δημιουργήθηκαν με το εργαλείο IconMaker μέσα στο περιβάλλον της Unity.



Εικόνα 12 Εικονίδιο για ψημένα σουβλάκια (αριστερά) και για μερίδα γύρου (δεξιά)

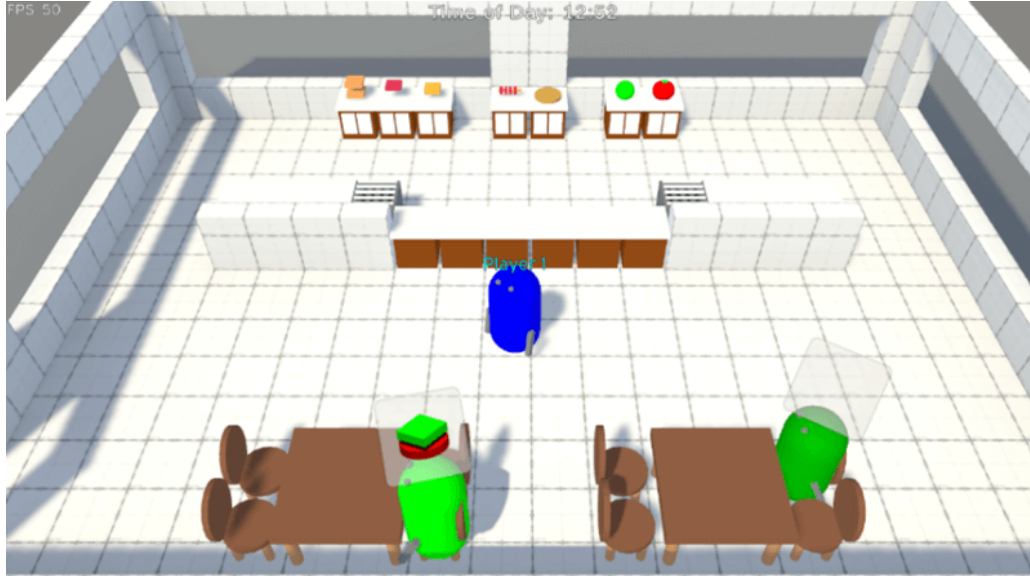
3.5 Διαδικασία που ακολουθήθηκε για την δημιουργία του παιχνιδιού

Στην παρακάτω ενότητα θα πραγματοποιηθεί η ανάλυση των βημάτων από την σύλληψη της ιδέας μέχρι το τέλος της δημιουργίας του παιχνιδιού.

3.5.1 Σύλληψη ιδέας

Ένα από τα πρώτα συνεργατικά παιχνίδια μαγειρικής που έπαιξα ήταν το “Overcooked”. Η κάθε πίστα ήταν διαφορετική και σχεδιασμένη με διαφορετικές δυσκολίες. Το παιχνίδι αυτό μοιάζει με παζλ και δίνει έμφαση στο χάος. Δεν είχε πελάτες και οι παραγγελίες έφταναν μόνες τους και έπρεπε να σερβίρεις με την σειρά τα πιάτα για έξτρα πόντους. Έπειτα έπαιξα ένα καινούριο παιχνίδι το “Diner Bros” στο οποίο λειτουργούσες το δικό σου εστιατόριο. Σε αυτό το παιχνίδι είχες τη δυνατότητα να αναβαθμίζεις το εστιατόριο σιγά σιγά. Έρχονταν πελάτες τους οποίους έπρεπε να τους εξυπηρετήσεις σύμφωνα με τα ιδιαίτερα χαρακτηριστικά τους (π.χ. κάποιιοι είχαν μικρή υπομονή πριν φύγουν, οπότε είχαν προτεραιότητα στην εξυπηρέτησή τους). Το συγκεκριμένο παιχνίδι δεν μπορείς να το παίζεις μέσω διαδικτύου με άλλους παίκτες. Αυτό είναι πολύ σημαντικό για ένα συνεργατικό παιχνίδι. Έτσι αποφάσισα να δημιουργήσω ένα δικό μου συνεργατικό παιχνίδι με έμπνευση από αυτά τα δύο παιχνίδια. Στο παιχνίδι αυτό ο παίκτης θα έχει το δικό του εστιατόριο το οποίο θα μπορεί να αναβαθμίσει, καθώς και να αναπτύξει το χαρακτήρα του δίνοντας του συγκεκριμένες ειδικότητες ή ικανότητες. Στις αρχές του 2021 ξεκίνησα μόνος μου να υλοποιώ αυτή την ιδέα σιγά σιγά σε Unity [εικόνα 13]. Μια καταγραφή βίντεο παιχνιδιού μπορείτε να δείτε στον επόμενο σύνδεσμο <https://youtu.be/PIyXKygx0nk>. Όμως η έλλειψη γνώσεων για την αρχιτεκτονική του κώδικα μέσα σε ένα παιχνίδι καθιστούσε τη συντήρηση του κώδικα πολύ δύσκολη και γινόταν αδύνατη η συνέχιση του έργου. Το καλοκαίρι του 2022 έμαθα για το συγκεκριμένο μεταπτυχιακό. Εκεί

απέκτησα καινούριες γνώσεις οι οποίες με βοήθησαν να ολοκληρώσω αυτή την διπλωματική εργασία. Η επιλογή θέματος για την διπλωματική εργασία ήταν εύκολη. Ήξερα ότι ήθελα να τελειώσω το συγκεκριμένο παιχνίδι. Όμως έπρεπε να το ξεκινήσω από την αρχή ώστε να φτιάξω μια καλύτερη αρχιτεκτονική στον κώδικα, ώστε να είναι συντηρήσιμος και εύκολα επεκτάσιμος.



Εικόνα 13 Αρχική υλοποίηση παιχνιδιού 2021

3.5.2 Έναρξη διπλωματικής εργασίας

Όταν ξεκίνησα τη διπλωματική εργασία άρχισα να ψάχνω tutorials για τη Unity. Δεν είχα βρει καλά tutorials όσον αφορά καλές πρακτικές στον κώδικα. Βρήκα όμως το Code Monkey <https://unitycodemonkey.com/> που είναι προγραμματιστής και τον ενδιαφέρουν και αυτόν οι καλές πρακτικές και ο καθαρός κώδικας. Η ιστοσελίδα του περιέχει tutorials για κώδικα και για Unity. Εγώ παρακολούθησα ένα ολοκληρωμένο δωρεάν μάθημα με το είδος του παιχνιδιού που ήθελα να φτιάξω (Learn to make a Game with Unity! Beginners and Intermediates - Code Monkey, 2023). Αργότερα παρακολούθησα ένα νέο μάθημά του όπου προσθέτει και τη λειτουργία για πολλούς παίκτες (multiplayer) με την λύση της Unity, “Netcode for Game Objects” (Learn How to Make Multiplayer Games - Code Monkey, 2023). Παράλληλα με την παρακολούθηση των μαθημάτων έφτιαχνα από την αρχή αυτά που έδειχνε με μικρές αλλαγές στον κώδικα για καλύτερη δομή. Το τελικό αποτέλεσμα από τα μαθήματα ήταν: η αυτόματη εμφάνιση των παραγγελιών, ο παίκτης τις ετοίμαζε και τις έβαζε σε έναν πάγκο παραγγελιών. Μέχρι στιγμής τα φαγητά που μπορούσε να φτιάξει ο παίκτης ήταν 3 διαφορετικά μπιφτεκόψωμα και μια σαλάτα. Το παιχνίδι μπορούσες να το παίζεις είτε μόνος σου είτε μέσω διαδικτύου, δεν υποστήριζε την δυνατότητα για πολλούς παίκτες τοπικά.

3.5.3 Εξέλιξη πέρα από τα μαθήματα

Με το πέρασ των μαθημάτων άρχισα να βελτιώνω τον κώδικα αλλά και την ποιότητα του παιχνιδιού, ώστε για να γίνει πιο ευχάριστο. Οι βελτιώσεις αυτές κυρίως αφορούσαν την αλληλεπίδραση με τα πιάτα και τα υλικά. Έπειτα άρχισα να βάζω και δικά μου υλικά φαγητού. Το πρώτο που έβαλα ήταν το σουβλάκι. Το μόνο που

χρειαζόταν ήταν ένα απλό μοντέλο. Τελικά ήταν πιο δύσκολο από ότι φανταζόμουν. Άνοιξα το Blender και ξεκίνησα να φτιάχνω το σουβλάκι. Χρειάστηκε να ψάξω αρκετά για το πως γίνονται διάφορα πράγματα στο πρόγραμμα (Blender). Όπως για παράδειγμα πως βάζεις διαφορετικά υλικά (materials) στο τρισδιάστατο μοντέλο και πως το κάνεις εξαγωγή (export) για Unity. Αφού είχα το μοντέλο, το έκανα εισαγωγή στην Unity και έφτιαξα τα διάφορα υλικά/χρώματα (materials) για τις διάφορες καταστάσεις που έχει το σουβλάκι: ωμό, ψημένο, καμένο. Έπειτα χρειαζόμουν μια συνταγή (recipe) φαγητού για να το σερβίρει ο παίκτης. Η συνταγή ήταν δύο σουβλάκια στο ίδιο πιάτο. Εδώ ξεκίνησε η πρώτη δυσκολία. Έπρεπε να φτιάξω τον κώδικα να υποστηρίξει πολλαπλά ίδια υλικά φαγητού. Αρχικά έπρεπε να προσθέσω στο Prefab του πιάτου (GameObject) όσες φορές χρειάζεται το κάθε υλικό φαγητού. Έπειτα έπρεπε να φτιάξω τον κώδικα ώστε να δουλεύει για πολλαπλά ίδια υλικά φαγητού. Πιο αναλυτικά για τον κώδικα έπρεπε να κρατάει σε μια λίστα τα υλικά που έχουν μπει ώστε να μην τα ξαναβάξει αν δεν πρέπει και μετά για την εμφάνιση του υλικού να ελέγχει αν είναι ήδη ενεργοποιημένο το GameObject πριν το ενεργοποιήσει. Για το εικονίδιο στο σουβλάκι χρησιμοποίησα το IconMaker που είχα φτιάξει στο παλιό μου έργο. Όμως είχε ένα πρόβλημα, δεν αποθήκευε σωστά τα χρώματα, είχαν διαφορετική απόχρωση/φωτεινότητα. Τελικά το έψαξα πιο διεξοδικά και κατάλαβα ότι ήθελε μια αλλαγή σε μια γραμμή κώδικα. Αυτός ο κώδικας έκανε την εικόνα να αποθηκεύεται σε μορφή sRGB. Η αλλαγή αυτή ήταν:

```
bakeCam.targetTexture = new RenderTexture(ren.width, ren.height, 32,
RenderTextureFormat.ARGB32, RenderTextureReadWrite.sRGB);
```

3.5.4 Λειτουργία για πολλούς παίκτες στον ίδιο υπολογιστή

Βασική προϋπόθεση για το παιχνίδι ήταν να παίζει και τοπικά με πολλούς παίκτες αλλά και μέσω διαδικτύου. Στην αρχική μου υλοποίηση (2021) το είχα φτιάξει μόνο με πολλαπλούς παίκτες στον ίδιο υπολογιστή. Είχα χρησιμοποιήσει το νέο Input System της Unity και το νέο στοιχείο “PlayerInputManager” που έκανε την εισαγωγή νέου παίκτη μόνο με το πάτημα ενός πλήκτρου στο χειριστήριο. Στα μαθήματα του Code Monkey είχε χρησιμοποιήσει το νέο Input System αλλά έκανε μόνος του την διαχείριση του παίκτη με τα συμβάντα δικτύου (Network Events) που προσφέρει το Netcode for GameObjects. Αρχικά προσπάθησα να χρησιμοποιήσω και τα δύο μαζί, Player Input Manager και τα συμβάντα δικτύου, αλλά δεν δούλεψε γιατί μπερδεύονταν μεταξύ τους και το ένα καλούσε το άλλο. Η λύση βρέθηκε σε μικρό χρονικό διάστημα. Αυτή η λύση ήταν να μην χρησιμοποιήσω καθόλου το Player Input Manager και να φτιάξω μια δική μου υλοποίηση διαχείρισης τοπικών παικτών πάνω στο υπάρχον κώδικα από του Code Monkey. Τελικά η εισαγωγή του νέου παίκτη γίνεται με το πάτημα του κουμπιού “Add Player” στη διεπαφή του χρήστη και μετά με το πάτημα ενός πλήκτρου στο χειριστήριο που θέλουμε να χρησιμοποιήσουμε.

3.5.5 Εισαγωγή πελατών

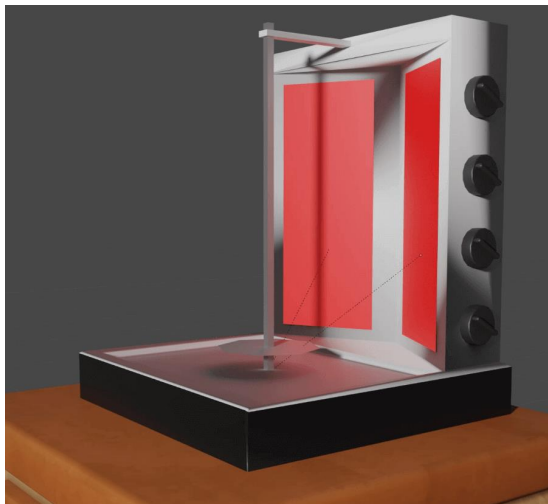
Η προσθήκη των πελατών ήταν ένα απαραίτητο στοιχείο για να προχωρήσει το παιχνίδι στο επόμενο επίπεδο. Οπότε το λογικό ήταν να προσθέσω τους πελάτες από την αρχική υλοποίηση. Έτσι, πρόσθετα τον κώδικα από το παλιό μου έργο στο καινούριο σιγά σιγά και διόρθωνα τα σφάλματα που εμφανίζονταν στην πορεία της υλοποίησης του παιχνιδιού. Επιπλέον, ήθελα το παιχνίδι να λειτουργεί και μέσω δικτύου. Άρα έπρεπε να χρησιμοποιήσω το Netcode for GameObjects. Δηλαδή η λογική του παιχνιδιού

εκτελείται από το server ή το host στο συγκεκριμένο παιχνίδι (server και client μαζί) και για το κομμάτι της εμφάνισης του παιχνιδιού είναι υπεύθυνοι όλοι οι clients. Σε αυτό το σημείο υπήρχαν πολλά προβλήματα. Πιο συγκεκριμένα, υπήρχε ασυμφωνία μεταξύ client και host στον αριθμό των πελατών οι οποίοι εξυπηρετήθηκαν σωστά και στον αριθμό των πελατών οι οποίοι έφυγαν από το μαγαζί δυσαρεστημένοι. Η κύρια πηγή του λάθους ήταν η καθυστέρηση μεταξύ host και client όταν οι παίκτες παίζανε μέσω διαδικτύου. Πριν την εκτέλεση μιας ενέργειας ήταν αναγκαίο να πραγματοποιηθούν έλεγχοι τόσο στο client όσο και στο host. Δηλαδή ο client έπρεπε να «ρωτάει» τον host αν όντως έγινε τελικά αυτή η ενέργεια ή όχι. Ένα συνηθισμένο παράδειγμα ήταν η παράδοση της παραγγελίας σε έναν πελάτη πριν τελειώσει η υπομονή του. Σε αυτό το σημείο κατά την παράδοση της παραγγελίας από το client παίκτη έπρεπε να ελεγχτεί ότι ήταν σωστή η παραγγελία σε client και host αλλά χρειαζόταν και ο host να ελέγξει αν πρόλαβε να δώσει την παραγγελία πριν φύγει ο πελάτης και να ενημερώσει τον client αντίστοιχα. Έτσι είχαν δημιουργηθεί οι πελάτες για το παιχνίδι. Όμως είχα την ιδέα να τους βάλω να περνάνε τυχαία από την αρχική οθόνη του παιχνιδιού, δηλαδή στο κεντρικό μενού. Έτσι έφτιαξα τους περιπλανώμενους πελάτες που αρχίζουν τυχαία από κάποια συγκεκριμένα σημεία και καταλήγουν τυχαία σε ένα από τα σημεία εξόδου. Εκτός από το κεντρικό μενού, τυχαίοι πελάτες εμφανίζονται κατά την διάρκεια του παιχνιδιού. Εισέρχονται στο μαγαζί με σκοπό να δυσκολεύουν τους παίκτες στην μετακίνηση. Τέλος, εμπνεύστηκα από ένα άλλο παιχνίδι που είχα παίξει το “Plate Up!”. Από εκεί πήρα μερικές ιδέες. Μία από αυτές ήταν η ουρά των πελατών. Οι πελάτες θα μπορούσαν να περιμένουν εκτός μαγαζιού όταν όλες οι καρέκλες θα ήταν γεμάτες. Οι πελάτες στην ουρά έχουν υπομονή αν τελειώσει η υπομονή τους αποχωρούν από την ουρά και ο παίκτης χάνει μια «ζωή».

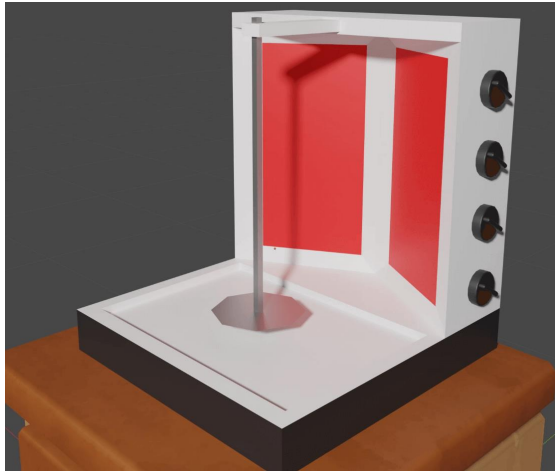
3.5.6 Εισαγωγή νέων μηχανισμών μαγειρέματος

Μέχρι στιγμής το μόνο φαγητό που είχα προσθέσει ήταν τα σουβλάκια. Ακόμα ήθελα να προσθέσω και το γύρος. Οπότε έπρεπε να φτιάξω μια μηχανή γύρου (γυριέρα). Όμως πρώτα έπρεπε να μάθω το Blender καλύτερα. Έτσι, άρχισα να ψάχνω tutorials στο youtube και βρήκα μια πολύ καλή σειρά μαθημάτων από τον “Blender Guru” www.blenderguru.com www.youtube.com/@blenderguru την (Blender 4.0 Beginner Donut Tutorial (NEW) - YouTube, 2023). Η υλοποίηση της μηχανής γύρου ξεκίνησε μετά το πέρας των μαθημάτων. Στο Blender φόρτωσα τον πάγκο με την κουζίνα (counter stove) για να ελέγξω τον όγκο της. Με αυτό το τρόπο αποφάσισα το πόσο μεγάλη έπρεπε να φτιάξω την μηχανή του γύρου. Λανθασμένα ξεκίνησα να φτιάχνω τη μηχανή γύρου τροποποιώντας την κουζίνα. Αφού την έφτιαξα και την φόρτωσα στην Unity παρατήρησα ότι έχει θέματα με τον φωτισμό. Ήμουν σίγουρος ότι το λάθος προήλθε από το μοντέλο που ξεκίνησα δηλαδή την κουζίνα [εικόνα 14]. Έτσι για να το διορθώσω ξεκίνησα από τη αρχή την μηχανή του γύρου αλλά αυτή την φορά από έναν νέο κύβο [εικόνα 15]. Αφού τελείωσα με τον μοντέλο της μηχανής χρειαζόμουν και το μοντέλο του γύρου και σκέφτηκα να επαναχρησιμοποιήσω το μοντέλο από το σουβλάκι. Επομένως, πήρα τα κομμάτια του κρέατος από το σουβλάκι και τα διαμόρφωσα για να μπουν στη μηχανή του γύρου. Έτσι απέκτησα πέντε κομμάτια γύρου διαφορετικού μεγέθους. Ακόμα έφτιαξα το μοντέλο για την μερίδα γύρου, όπου πήρα πάλι τα κομμάτια από το σουβλάκι, τα μίκρυνα, τα πολλαπλασίασα και τα έβαλα τυχαία πάνω στο μοντέλο του πιάτου μέχρι να γεμίσει. Ότι χρειαζόμουν για τον μηχανισμό ψησίματος ήταν έτοιμο. Ο μηχανισμός αυτός είναι: Ο παίκτης θα πρέπει να

πάρει και να τοποθετήσει πολλαπλά κομμάτια γύρου στην μηχανή και αυτά να εμφανίζονται καθώς τα βάζει. Μόλις τα βάλει όλα, η μηχανή του γύρου αρχίζει να γυρνάει και να ψήνει το γύρο. Όταν ψηθεί η πρώτη μερίδα γύρου, ο παίκτης πρέπει να πάει να κόψει το γύρο. Έτσι δημιουργείται η μερίδα στο κάτω μέρος της μηχανής και αρχίζει να ψήνει την επόμενη μερίδα γύρου. Αυτό συνεχίζεται μέχρι να τελειώσει όλος ο γύρος. Καθώς κόβει τον γύρο σε μερίδες, ο γύρος πάνω στο μηχανήμα σιγά σιγά αδειάζει. Έτσι απενεργοποιούνται τα πέντε ξεχωριστά κομμάτια γύρου που είχα φτιάξει πάνω στη μηχανή. Αυτή η διαδικασία γίνεται με έναν έξυπνο τρόπο, ο οποίος υπολογίζει δυναμικά το πότε πρέπει να εμφανίσει ή να εξαφανίσει τα κομμάτια γύρου πάνω από τη μηχανή. Αυτό υπολογίζεται ανάλογα με το πόσα κομμάτια γύρου πρέπει ο παίκτης να τοποθετήσει στη μηχανή και σε πόσες μερίδες αντιστοιχούν αυτά τα κομμάτια. Τα επόμενα φαγητά που πρόσθεσα στο παιχνίδι ήταν οι τηγανητές πατάτες και οι πίτες. Για την δημιουργία της πίτας βρήκα ένα δωρεάν μοντέλο και το έβαλα. Δεν χρειάστηκε να κάνω κάποια αλλαγή στο μοντέλο. Για τις πατάτες ακολούθησα την ίδια διαδικασία με τον γύρο. Βρήκα ένα δωρεάν μοντέλο ολόκληρης πατάτας. Έπειτα έφτιαξα το μοντέλο για τις κομμένες πατάτες τηγανητές. Τέλος έφτιαξα μια φριτέζα στο Blender από την αρχή [εικόνα 16]. Για το καλάθι της φριτέζας χρησιμοποίησα μια εικόνα με διαφάνεια ως texture ώστε να φανεί σαν μεταλλικό δίκτυο το καλάθι χωρίς να χρειαστεί να φτιαχτεί σε τρισδιάστατο μοντέλο. Ο μηχανισμός τηγανίσματος είναι λίγο διαφορετικός από τα υπόλοιπα «αυτόματα» ψησίματα. Εδώ ο παίκτης πρέπει να βάλει από μία κομμένη πατάτα σε καθένα καλάθι φριτέζας (έχει δύο το μοντέλο) και μετά να πατήσει το κουμπί της δευτερεύων αλληλεπίδρασης ώστε να κατέβουν τα καλάθια μέσα στην φριτέζα και μέσα στο λάδι. Αφού ψηθούν οι πατάτες θα πρέπει ο παίκτης να προλάβει και να ξαναπατήσει το ίδιο κουμπί ώστε να ανέβουν τα καλάθια και να μην καούν οι πατάτες. Έτσι ο παίκτης θα μπορεί να βγάλει τις μερίδες από τις τηγανισμένες πατάτες από μέσα. Όταν τελειώσουν οι μερίδες με τις τηγανισμένες πατάτες μπορεί να προσθέσει καινούριες κομμένες πατάτες.



Εικόνα 14 Λάθος μοντέλο μηχανής γύρου, εμφανή προβλήματα φωτισμού (περιβάλλον Blender)



Εικόνα 15 Σωστό μοντέλο μηχανής γύρου, σωστά χρώματα και φωτισμός (περιβάλλον Blender)



Εικόνα 16 Μοντέλο φριτέζας μαζί με τα δύο καλάθια της (περιβάλλον Unity)

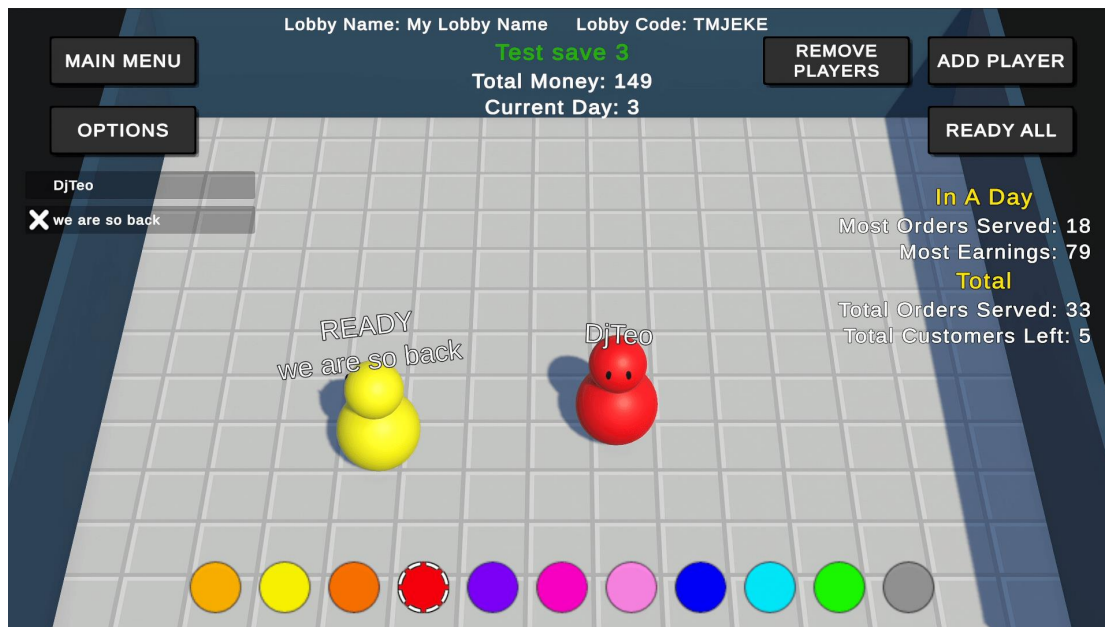
3.5.7 Εισαγωγή νέων πιάτων φαγητού (συνταγές)

Αφού έγινε η προσθήκη των νέων μηχανημάτων για μαγείρεμα και των νέων υλικών φαγητού, έπρεπε να γίνουν αλλαγές στα πιάτα φαγητών. Οπότε έκανα αλλαγές στις συνταγές των πιάτων έβαλα καινούριες και αφαίρεσα όσες δεν ταίριαζαν. Όταν έρχεται ένας πελάτης και θέλει να παραγγείλει φαγητό, επιλέγει τυχαία ένα πιάτο φαγητού από μια λίστα φαγητών. Αργότερα η λίστα αυτή τροποποιήθηκε και προστέθηκαν βάρη, ώστε να μην έχει κάθε πιάτο την ίδια πιθανότητα να επιλεγεί. Με την εισαγωγή ενός νέου φαγητού ή μηχανήματος έπρεπε να προστεθεί και στο εστιατόριο. Οπότε συνέχισα άλλαζα τη διαμόρφωση του χώρου. Τέλος ενδιάμεσα γινόντουσαν δοκιμές (playtesting) για να ρυθμιστεί η ισορροπία του παιχνιδιού. Δηλαδή η υπομονή του πελάτη και το κόστος του φαγητού.

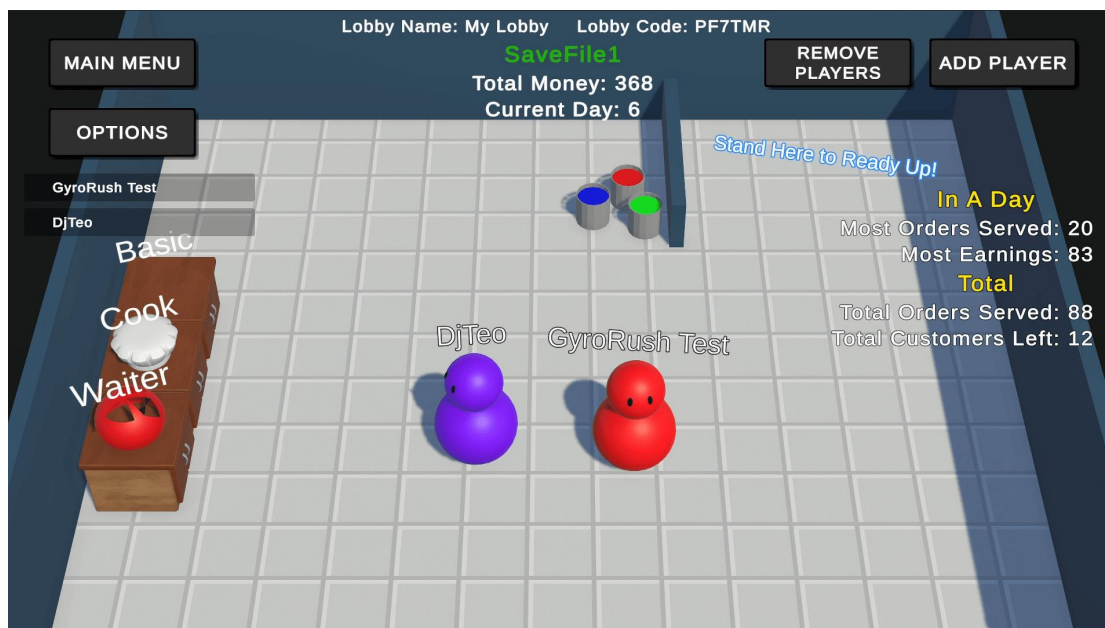
3.5.8 Οθόνη (Σκηνή) επιλογής χαρακτήρα

Μέχρι στιγμής η οθόνη επιλογής χαρακτήρα ήταν μόνο μια διεπαφή και για να αλλάξεις χρώμα ή ο παίκτης να δηλώσει ότι είναι έτοιμος, πατούσες κουμπιά στο χειριστήριο ή με το ποντίκι [εικόνα 17]. Ήταν αναγκαίο να βάλω αντικείμενα στον χώρο για να γίνονται όλα αυτά αντί μέσω της διεπαφή ή των κουμπιών. Εμπνευσμένος από το

παιχνίδι “Plate Up!” έφτιαξα αυτά τα αντικείμενα. Άνοιξα το Blender και έφτιαξα τρεις κουβάδες με μπογιά ώστε να αλλάζει χρώμα ο χαρακτήρας. Έφτιαξα στην σκηνή της Unity έναν ειδικό χώρο για να κάθονται εκεί οι παίκτες όταν είναι έτοιμοι για την έναρξη του γύρου του παιχνιδιού. Ακόμα έφτιαξα και τους πάγκους που επιλέγει ο παίκτης ειδικότητα για τον χαρακτήρα του [εικόνα 18].



Εικόνα 17 Παλιά οθόνη επιλογής χαρακτήρα



Εικόνα 18 Νέα οθόνη επιλογής χαρακτήρα

3.5.9 Ειδικότητες χαρακτήρων

Το παιχνίδι είχε αρχίσει να γίνεται δύσκολο και επαναλαμβανόμενο. Ο παίκτης μάζευε χρήματα από την ορθή εξυπηρέτηση των πελατών και δεν μπορούσε να τα ξοδέψει κάπου. Από την αρχή της ανάπτυξης του παιχνιδιού ήθελα να προσθέσω επιλογές στον παίκτη για τον χαρακτήρα του. Για παράδειγμα ο παίκτης θα μπορούσε να αλλάξει τις

ικανότητες του χαρακτήρα του. Οπότε σκέφτηκα να του δώσω την επιλογή για ειδικότητα. Οι δύο κύριες ειδικότητες που ήθελα να προσθέσω ήταν ο μάγερας και ο σερβιτόρος. Για να το πετύχω αυτό χρειάστηκε να φτιάξω την εμφάνισή τους στο Blender. Έφτιαξα δύο σφαίρες για το σώμα του χαρακτήρα, όπως ακριβώς είναι και μέσα στην Unity. Αρχισα να φτιάχνω τα κοστούμια για τους δύο αυτούς χαρακτήρες. Έφτιαξα μια ποδιά για το μάγειρα και βρήκα ένα καπέλο σεφ δωρεάν. Για τον σερβιτόρο έφτιαξα ένα πουκάμισο και ένα παπιγιόν. Τους έδωσα στον καθένα άλλες ικανότητες, ο μάγειρας περπατάει πιο αργά αλλά κόβει πιο γρηγορά το φαγητό και μπορεί να κρατήσει δύο υλικά φαγητού στα χέρια του ταυτόχρονα. Ενώ ο σερβιτόρος περπατάει πιο γρήγορα αλλά κόβει πιο αργά το φαγητό και μπορεί να κρατήσει δύο πιάτα στα χέρια του ταυτόχρονα. Ο παίκτης μπορούσε να επιλέξει ειδικότητα από την οθόνη επιλογής παικτών. Ο παίκτης μπορεί να ξεκλειδώσει τις ειδικότητες αυτές με χρήματα τα οποία έχει αποκτήσει από την πώληση φαγητών στους πελάτες. Έπειτα πρόσθεσα μια ακόμα ειδικότητα που είναι ένας συνδυασμός αυτών των δύο, όπου ο χαρακτήρας μπορεί να κουβαλήσει είτε δύο υλικά φαγητού, είτε δύο πιάτα στα χέρια του ταυτόχρονα. Πρόσθεσα και από μία αναβάθμιση σε κάθε ειδικότητα. Η αναβάθμιση αυτή τους επέτρεπε να κουβαλάνε τρία αντικείμενα αντίστοιχα ο καθένας τους και με λιγότερα μειονεκτήματα.

3.5.10 Αναβάθμιση μαγαζιού

Το τελευταίο που χρειαζόταν για την ολοκλήρωση του παιχνιδιού ήταν η πρόοδος του παίκτη στο παιχνίδι. Το παιχνίδι μέχρι στιγμής άρχιζε απευθείας από ένα μεγάλο μαγαζί με όλες τις συνταγές ξεκλειδωμένες. Αρχισα να δημιουργώ το κάθε επίπεδο ή αλλιώς αναβάθμιση του μαγαζιού. Ο παίκτης για να ανέβει επίπεδο θα έπρεπε να πληρώσει ένα χρηματικό ποσό. Αρχικά ξεκινάει από μια μικρή καντίνα με λίγες και απλές συνταγές φαγητού. Σε κάθε νέο επίπεδο που θα αγοράζει ο παίκτης θα μεγαλώνει λίγο η καντίνα και θα προστίθενται νέα πιάτα έως ότου φτάσει στο μεγάλο μαγαζί με όλα τα πιάτα ξεκλειδωτά. Αφού ολοκλήρωσα τα πρώτα επίπεδα της καντίνας άρχισα να τα δοκιμάζω αν δουλεύουν σωστά. Χρειάστηκε να κάνω μερικές αλλαγές στην λογική των πελατών. Όσο είναι στην καντίνα δεν υπάρχουν τραπέζια και καρέκλες για τους πελάτες να κάτσουν οπότε είναι take away. Δηλαδή ο πελάτης πάει στο παράθυρο της καντίνας, περιμένει να παραγγείλει και μόλις πάρει την παραγγελία του φεύγει. Επομένως δεν έχει κάπου να κάτσει για να φάει. Μετά έπρεπε να φτιάξω ένα αντικείμενο για τον χώρο της οθόνης επιλογής χαρακτήρων όπου να επιτρέπει την αναβάθμιση της καντίνας/μαγαζιού. Αφού ήταν έτοιμα αυτά, έφτιαξα και τα υπόλοιπα επίπεδα μέχρι το τελικό. Τα επίπεδα εκτός από διαφορετικές συνταγές έχουν και διαφορετικούς συνδυασμούς από τύπους πελατών. Δηλαδή στα αρχικά επίπεδα έχει τους κανονικούς πελάτες μόνο, έπειτα μπαίνουν στο παιχνίδι και οι αργοί πελάτες με την μεγαλύτερη υπομονή και στο τέλος μπαίνουν οι γρήγοροι πελάτες με την μικρή υπομονή. Όσο μεγαλώνει το μαγαζί και περνάνε οι μέρες δυσκολεύει και το παιχνίδι.



Εικόνα 19 Πρώτο επίπεδο καντίνας



Εικόνα 20 Τρίτο επίπεδο καντίνας, εδώ βλέπουμε ότι έχει μεγαλώσει η καντίνα προς τα δεξιά



Εικόνα 21 Τέταρτο επίπεδο καντίνας, εδώ βλέπουμε ότι έχει μεγαλώσει η καντίνα προς τα πάνω



Εικόνα 22 Πρώτο επίπεδο μετά την καντίνα, πλέον μαγαζί με τραπέζια και καρέκλες

3.5.11 Ρύθμιση δυσκολίας του παιχνιδιού

Πλέον έχουν υλοποιηθεί όλα τα σχέδια από το Game Design Document. Θα πρέπει όμως να γίνει και η ισορροπία του παιχνιδιού καλύτερη, δηλαδή το πόσο εύκολο ή δύσκολο πρέπει να είναι το παιχνίδι. Είναι από τις πιο χρονοβόρες διαδικασίες, διότι πρέπει να δοκιμαστεί από την αρχή μέχρι το τέλος αρκετές φορές και με έναν παίκτη αλλά και με πολλούς παίκτες. Σε κάθε δοκιμή πρέπει να γίνουν αλλαγές στις ιδιότητες του κάθε αντικειμένου που συμβάλει στην δυσκολία του παιχνιδιού. Αναλυτικότερα, πρέπει να ρυθμιστεί η ισορροπία αγοράς αναβαθμίσεων είτε του χαρακτήρα είτε του μαγαζιού σε σχέση με τις τιμές των πιάτων φαγητών που πληρώνουν οι πελάτες. Ακόμα πρέπει να ρυθμιστούν οι υπομονές των διάφορων τύπων πελατών, πόση ώρα κάθονται και τρώνε αλλά και η συχνότητα προσέλευση τους κατά τη πάροδο των ημερών. Το

παιχνίδι πρέπει να είναι εύκολο στην αρχή και να δυσκολεύει σταδιακά αλλά να μην είναι και αδύνατον να το κερδίσει ένας καλός παίκτης ή πολύ εύκολο και βαρετό για έναν καλό παίκτη. Συνήθως αυτή η διαδικασία γίνεται προς το τέλος της ανάπτυξης του παιχνιδιού.

3.5.12 Μελλοντικά σχέδια

Θα αναφέρω διάφορες ιδέες για τη βελτίωση του παιχνιδιού. Όλα τα παιχνίδια επιδέχονται βελτιώσεις. Άρα είναι σημαντικό να αποφασίσεις πότε ένα παιχνίδι είναι ολοκληρωμένο και να προχωρήσεις στη δημιουργία του επόμενου παιχνιδιού.

- Μια σημαντική ιδέα είναι η πώληση του παιχνιδιού σε κάποιο κατάστημα όταν ολοκληρωθεί και αποκτήσει καλύτερα γραφικά. Για παράδειγμα η πώληση του παιχνιδιού μέσω ηλεκτρονικών καταστημάτων όπως το Steam.
- Προσθήκη έξτρα φαγητού, για παράδειγμα το τζατζίκι. Για την δημιουργία του θα χρειαστεί ο παίκτης να κόψει το σκόρδο και το αγγούρι, να τα βάλει σε ένα μπολ, να προσθέσει το γιαούρτι και να τα ανακατεύσει. Από το μπολ θα μπορεί να βγάλει αρκετές μερίδες για να φτιάξει νέα πιάτα φαγητού.
- Εισαγωγή επιπλέον δυσκολιών.
 - Θα μπορούσε να περνούσε ένας «κακός» πελάτης ο οποίος θα έκλεβε τυχαία ένα φαγητό από τον πάγκο και θα έφευγε
 - Θα μπορούσε να ερχόταν ένας σκύλος και να έτρωγε τυχαία ένα φαγητό από τον πάγκο, οπότε θα έπρεπε με κάποιον τρόπο να τον σταματήσει ο παίκτης.
 - Όταν ο παίκτης πετάει ένα αντικείμενο στα σκουπίδια, θα αφαιρείτε ένα μικρό ποσό από τα χρήματά του ως τιμωρία. Στην περίπτωση που δεν έχει άλλα χρήματα, δεν θα επιτρέπεται η ρίψη του αντικειμένου στα σκουπίδια.
 - Δημιουργία διαφόρων τυχαίων συμβάντων κατά την διάρκεια ενός γύρου, για παράδειγμα η ταυτόχρονη προσέλευση πολλών πελατών στο μαγαζί.
- Προσθήκη περισσότερων αναβαθμίσεων στο μαγαζί. Για παράδειγμα, αγορά επιπλέον τηγανιού ψησίματος ή πάγκου κοπής ή ακόμα και μηχανής γύρου. Επιπλέον ο παίκτης θα μπορούσε να αναβαθμίσει διάφορες συσκευές μαγειρέματος όπως για παράδειγμα το τηγάνι ή την μηχανή γύρου ώστε να ψήνουν πιο γρήγορα.
- Ακόμα ο πελάτης θα μπορούσε να αφήνει ένα μικρό φιλοδώρημα στον παίκτη ως επιβράβευση όταν εξυπηρετηθεί αρκετά γρήγορα.
- Προσθήκη μηχανής για αναψυκτικά. Οι πελάτες θα ζητάνε αναψυκτικό ανεξάρτητα από την παραγγελία τους, δηλαδή μπορεί να το ζητήσουν κατά την διάρκεια του φαγητού τους.
- Στατιστικά ανά παίκτη και περισσότερα στοιχεία από παιχνίδια ρόλων (RPG). Ο παίκτης θα μπορούσε να αυξήσει κάποια χαρακτηριστικά του χαρακτήρα

του. Υπάρχουν διάφορες ιδέες υλοποίησης. Για παράδειγμα ο παίκτης θα μαζεύει εμπειρία καθώς κάνει πράγματα και θα ανεβαίνει επίπεδο. Έτσι θα κερδίζει έναν πόντο, για να αυξήσει κάποιο χαρακτηριστικό του ή ακόμα και να μπορεί να αγοράζει πόντους με χρήματα. Θα υπήρχε επιλογή ανάμεσα σε διάφορα χαρακτηριστικά όπως η ταχύτητα του περπατήματος, η ταχύτητα της κοπής και η ταχύτητα της περιστροφής του χαρακτήρα.

- Εισαγωγή προκλήσεων ως τρίτη λειτουργία παιχνιδιού. Κατασκευή δύσκολων και διασκεδαστικών σεναρίων όπου οι παίκτες καλούνται να αντιμετωπίσουν κάποιες προκλήσεις. Για παράδειγμα θα μπορούσε να δημιουργηθεί ένα νέο μαγαζί στο οποίο θα έρχονται μόνο οι γρήγοροι πελάτες με μικρή υπομονή ή με συγκεκριμένο μενού. Ακόμα θα μπορούσε να υπάρχει ένα τεράστιο εστιατόριο για να εξυπηρετήσουν με παραπάνω χρόνο ανά γύρο.
- Δημιουργία διακοσμητικών αντικειμένων για μέσα στο μαγαζί αλλά και για έξω στον δρόμο, ώστε να γίνει πιο όμορφο το περιβάλλον του παιχνιδιού. Επίσης οι πελάτες χρειάζονται αλλαγή εμφάνισης για να ταιριάζουν καλύτερα στα γραφικά του υπόλοιπου παιχνιδιού, όπως και για κάποια από τα υλικά φαγητού.
- Μια άλλη ιδέα που θα μπορούσε να γίνει και ένα επιπλέον περιεχόμενο προς λήψη (DLC) είναι η εισαγωγή τυλιχτών συνταγών. Ο παίκτης θα χρησιμοποιούσε αντί για πιάτα λαδόκολλες. Ο παίκτης θα έπαιρνε στα χέρια του τη λαδόκολλα, θα έβαζε τα υλικά που χρειάζονται και πριν το δώσει στον πελάτη θα έπρεπε να το τυλίξει. Μετά το τυλίγμα δεν θα μπορούσε να προσθέσει άλλο υλικό φαγητού.

3.6 Σύνδεση μέσω διαδικτύου

Σε αυτή την ενότητα θα αναλυθεί το διαδικτυακό κομμάτι της εργασίας, για τη χρήση των Relay και Lobby, καθώς και για τη χρήση του Netcode for GameObjects.

3.6.1 Χρήση των Gaming Services της Unity

Για τη χρήση των υπηρεσιών των Gaming Services της Unity πρέπει να γίνουν κάποια βήματα πρώτα. Ο προγραμματιστής του παιχνιδιού χρειάζεται να έχει φτιάξει ένα πρότζεκτ για το παιχνίδι του στον λογαριασμό Unity Cloud. Αν δεν έχει ένα λογαριασμό πρέπει να φτιάξει ένα. Έπειτα το πρότζεκτ της Unity πρέπει να συνδεθεί με τον λογαριασμό, άμα το πρότζεκτ έχει φτιαχτεί μέσα από την Unity λογικά θα έχει γίνει αυτόματα αυτό το βήμα. Ύστερα πρέπει να συνδεθεί στον πίνακα εργαλείων της Unity (Unity Cloud Dashboard) στην παρακάτω ιστοσελίδα <https://cloud.unity.com/>. Από τον πίνακα ελέγχου μπορεί είτε να πλοηγηθεί από την μπάρα αριστερά: Projects - > όνομα έργου/παιχνιδιού -> services και να ενεργοποιήσει όποιο θέλει, είτε να πατήσει στα “Products” και να αναζητήσει την υπηρεσία που θέλει να ενεργοποιήσει. Έπειτα μπορεί να πάει στην Unity και να εγκαταστήσει τα πακέτα που χρειάζονται. Στην συγκεκριμένη εργασία οι υπηρεσίες και τα πακέτα αυτά ήταν το Relay και το Lobby. Όσον αφορά τον κώδικα αυτό που χρειάζεται είναι να γίνει μια αρχικοποίηση των υπηρεσιών αυτών τρέχοντας την εντολή `await UnityServices.InitializeAsync();`. Για τη χρήση κάποιων υπηρεσιών, όπως το Lobby, πρέπει πρώτα ο παίκτης να

πιστοποιηθεί, δηλαδή να συνδεθεί. Τα Gaming Services της Unity υποστηρίζουν πολλούς τρόπους σύνδεσης, για παράδειγμα σύνδεση μέσω Apple, μέσω Google, μέσω Facebook, μέσω Steam και άλλες πλατφόρμες. Στο συγκεκριμένο παιχνίδι χρησιμοποιείται η ανώνυμη σύνδεση του παίκτη με την εντολή `await AuthenticationService.Instance.SignInAnonymouslyAsync();`. Αυτή η εντολή καταχωρεί στον υπολογιστή ένα ID του χρήστη ώστε να χρησιμοποιηθεί το ίδιο ID χρήστη την επόμενη φορά.

3.6.2 Relay και Lobby

Για την σύνδεση μέσω διαδικτύου χρησιμοποιήθηκαν οι υπηρεσίες της Unity, Relay και Lobby. Το Relay διευκολύνει τη δημιουργία διασύνδεση μεταξύ των παικτών χωρίς να χρειάζεται να ρυθμίσει τίποτα ο παίκτης. Χωρίς το Relay ή κάποια παρόμοια υπηρεσία ο παίκτης ο οποίος θα δημιουργούσε το διακομιστή θα έπρεπε να κάνει προηγμένες ρυθμίσεις στον υπολογιστή του και στο δρομολογητή (router) του. Με αυτό τον τρόπο θα μπορούσε να επιτραπεί οποιαδήποτε εξωτερική σύνδεση να φτάσει στον υπολογιστή του. Το Lobby χρησιμοποιείται ώστε να γίνει εύκολη η αναζήτηση συνεδριών παιχνιδιού. Ο παίκτης μπορεί να χρησιμοποιήσει φίλτρα αναζήτησης για να βρει τη συνεδρία παιχνιδιού που θέλει. Χωρίς το Lobby, ο παίκτης ο οποίος θα δημιουργούσε το διακομιστή θα έπρεπε με κάποιον τρόπο να στείλει τα στοιχεία σύνδεσής του (π.χ. τη διεύθυνση Ίντερνετ του) στους παίκτες που θέλει για να συνδεθούν. Αυτές οι υπηρεσίες έχουν και κάποια αρνητικά. Όλες αυτές οι υπηρεσίες έχουν ένα όριο μέχρι που είναι δωρεάν και μετά χρεώνονται. Επίσης χρησιμοποιώντας το Relay προσθέτει μια μικρή καθυστέρηση διότι η επικοινωνία μεταξύ των πελατών και του διακομιστή γίνεται μέσω ενός Relay Server.

3.6.3 Συγχρονισμός αντικειμένων με τη χρήση NGO

Το πιο σημαντικό στη σύνδεση μέσω διαδικτύου είναι ο σωστός συγχρονισμός των αντικειμένων και να μην γίνεται αισθητή η καθυστέρηση της επικοινωνίας μεταξύ των παικτών. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε η έκδοση 1.7.1 του Netcode for GameObjects. Το NGO προσφέρει πολλούς τρόπους συγχρονισμού αντικειμένων μεταξύ των πελατών και του διακομιστή. Το NGO παρέχει το αντικείμενο “NetworkManager”. Αυτό παρέχει όλες τις λειτουργίες δικτύου που χρειάζεται ένα παιχνίδι. Μια από αυτές είναι ο συγχρονισμός της σκηνής ώστε να αλλάζουν όλοι οι συνδεδεμένοι παίκτες μαζί τη σκηνή όταν χρειάζεται. Ακόμα το NGO έχει το στοιχείο (component) “NetworkObject”. Αυτό το στοιχείο επιτρέπει τον συγχρονισμό των GameObject μεταξύ του διακομιστή και των πελατών. Επίσης έχει μερικές ρυθμίσεις για αυτόν το συγχρονισμό. Επομένως είναι απαραίτητο στοιχείο σε ένα GameObject το οποίο θέλουμε να γίνει οποιοσδήποτε συγχρονισμός μεταξύ των συνδεδεμένων παικτών. Επίσης το NGO μας προσφέρει τα στοιχεία “NetworkTransform” και “NetworkAnimator”. Αυτά τα δύο συγχρονίζουν αυτόματα κάποιες βασικές ιδιότητες ενός GameObject, αναλυτικότερα. Το NetworkTransform επιτρέπει στον προγραμματιστή να επιλέξει ποιες ιδιότητες του GameObject θέλει να συγχρονίζεται, δηλαδή τη τοποθεσία, την περιστροφή και την κλίμακα. Ακόμα του επιτρέπει να επιλέξει το πόσο πολύ πρέπει να αλλάξει μια τιμή ώστε να την συγχρονίσει. Είναι σημαντικό να επιλεγούν μόνο τα απαραίτητα πεδία για τον συγχρονισμό διότι έτσι θα δημιουργηθεί λιγότερη κίνηση δικτύου άρα και λιγότερη καθυστέρηση. Το NetworkAnimator δέχεται ένα στοιχείο “Animator” το οποίο και

συγχρονίζει με τους υπόλοιπους πελάτες και έτσι βλέπουν το ίδιο animation. Εδώ δεν υπάρχουν άλλες ρυθμίσεις. Στο συγκεκριμένο παιχνίδι φτιάχτηκαν δύο νέα στοιχεία από αυτά τα δύο, το “ClientNetworkTransform” και “ClientNetworkAnimator”. Αυτά τα δύο χρειάστηκαν ώστε το NGO να συγχρονίζει τις ιδιότητες του GameObject με τις αλλαγές που γίνονται από τον πελάτη, αλλιώς επιτρέπονται αλλαγές μόνο από το διακομιστή. Ο κώδικας είναι πολύ απλός και περιέχει μόνο μια εντολή όπως φαίνεται παρακάτω.

```
/// <summary>
/// Used for syncing a transform with client-side changes. This includes host.
/// Pure server as owner isn't supported by this. Please use NetworkTransform
/// for transforms that'll always be owned by the server.
/// </summary>
[DisallowMultipleComponent]
public class ClientNetworkTransform : NetworkTransform
{
    /// <summary>
    /// Used to determine who can write to this transform. Owner client only.
    /// This imposes state to the server.
    /// This is putting trust on your clients.
    /// Make sure no security-sensitive features use this transform.
    /// </summary>
    protected override bool OnIsServerAuthoritative()
    {
        return false;
    }
}
```

Ομοίως είναι και ο κώδικας για το “ClientNetworkAnimator”. Ένας τρόπος που προσφέρει το NGO για το συγχρονισμό των μεταβλητών είναι το NetworkVariable, το οποίο δέχεται όλους τους βασικούς τύπους της C# και της Unity. Από προεπιλογή σε αυτές τις μεταβλητές μπορεί να τις αλλάξει μόνο ο διακομιστής αλλά όλοι μπορούν να τις διαβάσουν. Αυτό μπορεί να αλλάξει στην αρχικοποίησή της μεταβλητής. Το NetworkVariable προσφέρει την δυνατότητα εγγραφής στο συμβάν “OnValueChanged” ώστε αν αλλάξει η τιμή της μεταβλητής να καλεστεί μια μέθοδος και να εκτελεστεί η κατάλληλη λογική. Ένας άλλος τρόπος συγχρονισμού είναι μέσω κλήσεων απομακρυσμένης διαδικασίας (RPC). Όπου εδώ έχουμε τις διαδικασίες πελάτη (ClientRpc) και τις διαδικασίες διακομιστή (ServerRpc). Μόνο ο διακομιστής μπορεί να καλέσει μία διαδικασία πελάτη και όλοι οι συνδεδεμένοι πελάτες να την εκτελέσουν. Ενώ μια διαδικασία διακομιστή μπορεί να κληθεί μόνο από έναν πελάτη και θα λαμβάνεται και θα εκτελείται πάντα στο διακομιστή. Επομένως μια διαδικασία διακομιστή παρέχει τη δυνατότητα σε έναν πελάτη να στείλει πληροφορία σε ένα διακομιστή. Η προεπιλεγμένη ρύθμιση του ServerRpc επιτρέπει μόνο στον κάτοχο πελάτη του συγκεκριμένου NetworkObject να την καλέσει. Σε περίπτωση που θέλουμε οποιοσδήποτε πελάτη να έχει δικαίωμα να καλέσει ένα ServerRpc χρειάζεται μόνο να γράψουμε [ServerRpc(RequireOwnership = false)]. Η διαφορά μεταξύ NetworkVariable και RPC είναι ότι το πρώτο χρησιμοποιείται για μόνιμες καταστάσεις, δηλαδή για πληροφορίες που θα διαρκέσουν περισσότερο από μια στιγμή ενώ το δεύτερο χρησιμοποιείται για παροδικά συμβάντα, δηλαδή πληροφορίες χρήσιμες μόνο για μια στιγμή κατά τη λήψη τους. Παρακάτω θα αναλυθεί η λύση του προβλήματος που είχε προαναφερθεί όταν ένας client παίκτης σέρβιρε φαγητό σε έναν πελάτη την στιγμή που τελείωνε η υπομονή του.

Ο παρακάτω κώδικας βρίσκεται στο CustomerAI.cs αρχείο το οποίο τρέχει πάνω στον πελάτη του παιχνιδιού. Η μέθοδος InteractAlt καλείται όταν ο παίκτης πατάει το κουμπί της δευτερεύουσας αλληλεπίδρασης και εκτελείται στον υπολογιστή του ίδιου παίκτη.

```
public void InteractAlt(Player player)
{
    // don't give food if customer haven't order yet
    if (_state.Value == CustomerState.HasOrdered)
    {
        if (player.HasKitchenObject())
        {
            var playersKitchenObjects = player.GetAllKitchenObjects();
            foreach (var kitchenObject in playersKitchenObjects)
            {
                if (kitchenObject != null && kitchenObject.TryGetPlate(out
PlateKitchenObject plateKitchenObject))
                {
                    // Check Give Food
                    if (DeliveryManager.Instance.CheckRecipe(_recipeWaiting,
plateKitchenObject))
                    {
                        DeliverToCustomerServerRpc(player.GetNetworkObject());
                        break;
                    }
                }
            }
        }
        // if customer needs to order take his order
        else if (_state.Value == CustomerState.NeedsToOrder)
            InteractAltCustomerServerRpc();
    }
}
```

Η παραπάνω γραμμή του κώδικα η οποία εκτελείται στο client `DeliveryManager.Instance.CheckRecipe()` ελέγχει αν ο παίκτης (χαρακτήρας) έχει τη σωστή συνταγή. Αν έχει τη σωστή συνταγή, καλεί ένα `ServerRpc` το `DeliverToCustomerServerRpc()` με όρισμα το `NetworkObject` του παίκτη (χαρακτήρα) που έκανε την αλληλεπίδραση. Παρακάτω είναι το `ServerRpc` που εκτελείται μόνο στο διακομιστή. Αυτό έχει `RequireOwnership = false` ώστε να μπορεί να το καλέσει οποιοσδήποτε client.

```
[ServerRpc(RequireOwnership = false)]
private void DeliverToCustomerServerRpc(NetworkObjectReference
kitchenObjectParentNetworkObjectReference)
{
    if (_state.Value == CustomerState.HasOrdered)
    {
        DeliveredToCustomerClientRpc(kitchenObjectParentNetworkObjectReference);
        // is dine in or take away?
        if (_sitAndEat)
            StartEating();
        else
            LeaveWithFood();
    }
    else
    {
        // customer left
        Debug.Log("You didn't catch the customer!");
    }
}
```

Σε αυτό το σημείο ο διακομιστής ελέγχει αν ο πελάτης είναι ακόμα στην κατάσταση ότι περιμένει την παραγγελία του. Άμα ο παίκτης τελικά πρόλαβε να σερβίρει τον πελάτη εγκαίρως, τότε ο διακομιστής θα καλέσει ένα ClientRpc το DeliveredToCustomerClientRpc() με όρισμα το ίδιο NetworkObject που είχε ως είσοδο. Έπειτα ο διακομιστής θα εκτελέσει τη λογική για την επόμενη κατάσταση του πελάτη. Παρακάτω είναι ο κώδικας που θα εκτελεστεί σε κάθε συνδεδεμένο client στην περίπτωση που ο παίκτης πρόλαβε να σερβίρει τον πελάτη.

```
[ClientRpc]
private void DeliveredToCustomerClientRpc(NetworkObjectReference
kitchenObjectParentNetworkObjectReference)
{
    DeliveryManager.Instance.DeliveryCorrectRecipeServerRpc(_customerNumber);

    kitchenObjectParentNetworkObjectReference.TryGet(out NetworkObject
playerNetworkObject);
    var player = playerNetworkObject.GetComponent<Player>();
    var playersKitchenObjects = player.GetAllKitchenObjects();
    foreach (var kitchenObject in playersKitchenObjects)
    {
        if (kitchenObject != null && kitchenObject.TryGetPlate(out
PlateKitchenObject plateKitchenObject))
        {
            // Check Give Food
            if (DeliveryManager.Instance.CheckRecipe(_recipeWaiting,
plateKitchenObject))
            {
                kitchenObject.SetKitchenObjectParent(this);
                break;
            }
        }
    }
}
```

Στην παραπάνω μέθοδο είναι γνωστό ότι ο χαρακτήρας έχει τη σωστή συνταγή που θέλει ο πελάτης. Όμως ο έλεγχος όλων των αντικειμένων του χαρακτήρα πραγματοποιείται πάλι. Αυτό είναι αναγκαίο διότι κάποιες ειδικότητες επιτρέπουν στον χαρακτήρα να κρατάει παραπάνω από ένα πιάτο στα χέρια του. Με αυτό τον τρόπο βρίσκουμε ποιο απ' όλα τα πιάτα είναι το σωστό για να το σερβίρει. Η γραμμή κώδικα kitchenObject.SetKitchenObjectParent(this); αλλάζει τον «πατέρα» του kitchenObject στο "this" το οποίο αναφέρεται στο στοιχείο CustomerAI το οποίο βρίσκεται στον πελάτη.

4. Συμπεράσματα

Στην παρούσα εργασία συμπεραίνουμε ότι η διαδικασία παραγωγής ενός παιχνιδιού είναι χρονοβόρα και κουραστική. Δεν υπάρχουν κανόνες για την σωστή υλοποίηση ενός παιχνιδιού ή συγκεκριμένα βήματα που μπορεί να ακολουθήσει κάποιος. Η εμπειρία είναι πολύτιμη για την δημιουργία ενός παιχνιδιού καθώς ο προγραμματιστής έχει αποκτήσει κάποιες τεχνικές για την υλοποίηση του. Αυτή η εμπειρία θα τον βοηθήσει στην οργάνωσή του και στα βήματα που θα ακολουθήσει. Τέλος προτείνεται η ανάπτυξη ενός βιντεοπαιχνιδιού να γίνεται από ομάδες τουλάχιστον τριών ατόμων. Κάθε άτομο πρέπει να έχει διαφορετικές δεξιότητες για την ανάπτυξη ενός βιντεοπαιχνιδιού.

Βιβλιογραφία

- About Netcode for GameObjects | Unity Multiplayer Networking.* (2024, Ιανουάριος 17). Ανάκτηση από Unity Multiplayer Networking: <https://docs-multiplayer.unity3d.com/netcode/1.7.1/about/>
- Arcade video game - Wikipedia.* (2024, Φεβρουάριος 18). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Arcade_video_game
- Balasubramanian, K. (2022, Σεπτέμβριος 15). *Game Engines: All You Need to Know - Gameopedia.* Ανάκτηση από Gameopedia: <https://www.gameopedia.com/game-engines-all-you-need-to-know-about/>
- Blender 4.0 Beginner Donut Tutorial (NEW) - YouTube.* (2023, Νοέμβριος). Ανάκτηση από YouTube: <https://www.youtube.com/playlist?list=PLjEaoINr3zgEPv5y--4MKpciLaoQYZB1Z>
- Blender Foundation. (2024). *blender.org - Home of the Blender project - Free and Open 3D Creation Software.* Ανάκτηση από Blender: <https://www.blender.org/about/>
- Free Logo Maker | LOGO.com.* (2024). Ανάκτηση από LOGO: <https://logo.com/>
- Game engine - Wikipedia.* (2024, Απρίλιος 13). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Game_engine
- History of video games - Wikipedia.* (2024, Φεβρουάριος 3). Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/History_of_video_games
- History.com Editors. (2022, Οκτώβριος 17). *Video Game History - Timeline & Facts.* Ανάκτηση από HISTORY: <https://www.history.com/topics/inventions/history-of-video-games>
- Kent, S. L. (2001). *The Ultimate History of Video Games.*
- Learn How to Make Multiplayer Games - Code Monkey.* (2023, Μάρτιος). Ανάκτηση από Code Monkey: <https://unitycodemonkey.com/kitchenchaosmultiplayercourse.php>
- Learn to make a Game with Unity! Beginners and Intermediates - Code Monkey.* (2023, Ιανουάριος). Ανάκτηση από Code Monkey: <https://unitycodemonkey.com/kitchenchaoscourse.php>
- Manage Your Team's Projects From Anywhere | Trello.* (2024). Ανάκτηση από Trello: <https://trello.com/>
- Open Broadcaster Software | OBS.* (2024). Ανάκτηση από OBS: <https://obsproject.com/>
- Overview of services.* (2024). Ανάκτηση από Unity Documentation: <https://docs.unity.com/ugs/en-us/manual/overview/manual/unity-gaming-services-home>
- Real-Time 3D Development Platform & Editor | Unity.* (2024). Ανάκτηση από Unity: <https://unity.com/products/unity-engine>
- Unity (game engine) - Wikipedia.* (2024, Απρίλιος). Ανάκτηση από Wikipedia: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- Unity Gaming Services: Live Game Development Platform.* (2024). Ανάκτηση από Unity: <https://unity.com/solutions/gaming-services>

Unity Technologies - Wikipedia. (2024, Μάιος). Ανάκτηση από Wikipedia:
https://en.wikipedia.org/wiki/Unity_Technologies

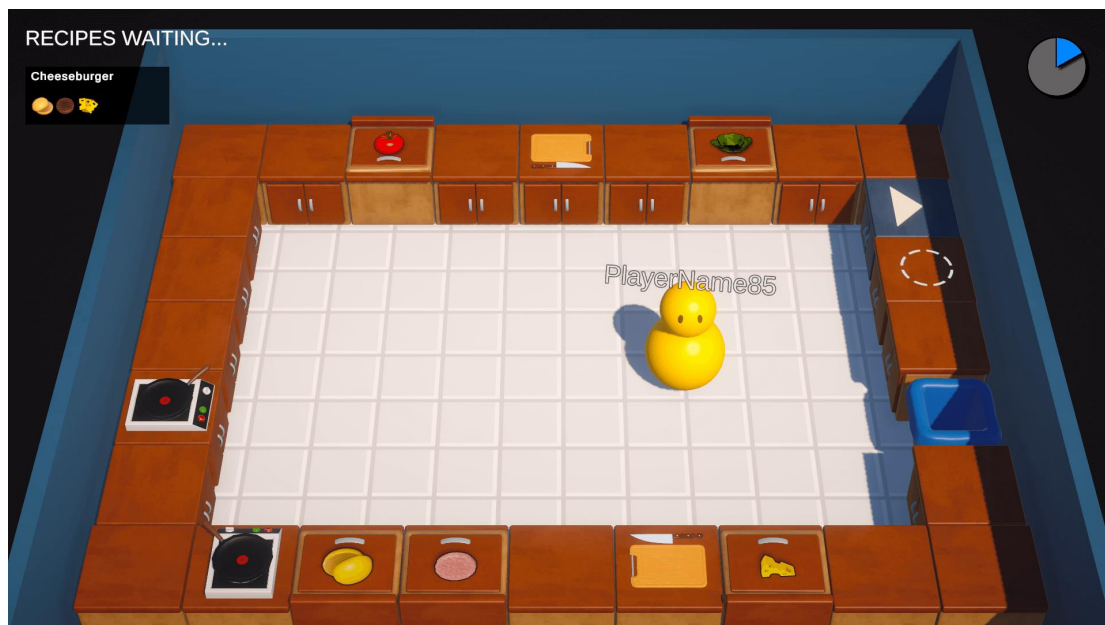
Video game - Wikipedia. (2024, Φεβρουάριος 29). Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/Video_game

Video Game History Timeline - The Strong National Musuem of Play. (2022, Δεκέμβριος 5).
Ανάκτηση από The Strong National Musuem of Play:
<https://www.museumofplay.org/video-game-history-timeline/>

Visual Studio: IDE and Code Editor for Software Developers and Teams. (2024). Ανάκτηση
από Microsoft Visual Studio: <https://visualstudio.microsoft.com/>

Παράρτημα

Παρακάτω ακολουθούν μερικές εικόνες και διαδικτυακοί σύνδεσμοι που δείχνουν την πορεία εξέλιξης του παιχνιδιού πριν μπει η πρόοδος στη λειτουργία ιστορίας.



Εικόνα 23 Πρώτη έκδοση μετά τα μαθήματα του Code Monkey



Εικόνα 24 Πρώτη έκδοση εστιατορίου με πελάτες και σουβλάκια

Πρώτη καταγραφή βίντεο παιχνιδιού με λειτουργία για πολλούς παίκτες:
<https://youtu.be/FnJd9VvHt3A>



Εικόνα 25 Εισαγωγή «ζωών» ανά γύρο παιχνιδιού.

Στον παρακάτω σύνδεσμο καταγραφής βίντεο παιχνιδιού μπορείτε να δείτε εκτός από τις «ζωές» του παιχνιδιού και τους τυχαίους πελάτες να εισέρχονται μέσα στο μαγαζί και να ενοχλούν τους παίκτες <https://youtu.be/UA7pmEpIj5w>.



Εικόνα 26 Εισαγωγή μηχανής γύρου, αλλαγές στο εστιατόριο, καθώς και κάποιους από τους ενοχλητικούς πελάτες (εδώ είναι οι τρεις με το κόκκινο καπέλο)

Στον επόμενο σύνδεσμο καταγραφής βίντεο <https://youtu.be/uaR6zMEKnp8> μπορείτε να δείτε τις αλλαγές στην οθόνη επιλογής χαρακτήρων και τη μηχανή του γύρου σε λειτουργία. Σε αυτό το στάδιο έχει προστεθεί και η ουρά των πελατών εκτός του μαγαζιού αλλά δεν φαίνεται στο βίντεο. Στον παρακάτω σύνδεσμο καταγραφής βίντεο παιχνιδιού <https://youtu.be/O3N0qTrDhJw> μπορείτε να δείτε την προσθήκη των ειδικοτήτων, την εισαγωγή της πίτας και νέων πιάτων φαγητού. Τέλος στην παρακάτω

εικόνα [εικόνα 27] μπορείτε να δείτε το τελικό εστιατόριο με όλα τα υλικά φαγητού που έχουν προστεθεί. Δηλαδή τα σουβλάκια, τον γύρο, τις πίτες και τις πατάτες τηγανητές.



Εικόνα 27 Τελικό εστιατόριο, εδώ βλέπουμε την εισαγωγή της πατάτας και της φριτέζας

Στον παρακάτω σύνδεσμο καταγραφής βίντεο παιχνιδιού <https://youtu.be/SjlheA5TshI> μπορείτε να δείτε την δυνατότητα αναβάθμισης του μαγαζιού καθώς και έναν νέο μηχανισμό του παιχνιδιού. Την ικανότητα των πελατών να σκέφτονται αφού καθίσουν στην καρέκλα και πριν θελήσουν να παραγγείλουν.

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1928, η παρούσα Μεταπτυχιακή Διπλωματική Εργασία με τίτλο:

«Ανάπτυξη ενός συνεργατικού 3D παιχνιδιού σε πλατφόρμα Windows με χρήση της Unity» καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Πρόγραμμα Μεταπτυχιακών Σπουδών «Ανάπτυξη Ψηφιακών Παιχνιδιών και Πολυμεσικών Εφαρμογών» του Τμήματος Επικοινωνίας & Ψηφιακών Μέσων του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του Επίκουρου καθηγητή Κωνσταντίνου Καρπούζης αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

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

Copyright © Γκίκας Θεοδόσιος, Κώστας Καρπούζης, 2024, Αθήνα

Υπογραφή Φοιτητή

