



Ιόνιο Πανεπιστήμιο
Τμήμα Πληροφορικής

Σχεδίαση και Ανάπτυξη παιχνιδιού εικονικής
πραγματικότητας για φορητές συσκευές

Πτυχιακή Εργασία
του

Διονύσιου-Παύλου Κοντοζίδη με ΑΜ: Π2016178

24 Ιουνίου 2021



Ιόνιο Πανεπιστήμιο

Επιβλέπουσα καθηγήτρια

Ελένη Χριστοπούλου, Εργαστηριακό Διδακτικό Προσωπικό Τμήμα Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Κωνσταντίνος Οικονόμου, Καθηγητής Τμήμα Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Μιχαήλ Στεφανιδάκης, Αναπληρωτής Καθηγητής Τμήμα Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Ελένη Χριστοπούλου, Εργαστηριακό Διδακτικό Προσωπικό Τμήμα Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Περίληψη

Σήμερα, υπάρχουν πολλοί τρόποι για την ανάπτυξη παιχνιδιών στο διαδίκτυο, που είναι δημοφιλής, καθώς χρησιμοποιούνται από ερασιτέχνες, αλλά και από επαγγελματίες. Τεχνολογίες όπως η επαυξημένη πραγματικότητα συνεχίζουν να αναπτύσσονται και χρησιμοποιούνται για να βελτιώσουν την ζωή των χρηστών σε διάφορους τομείς. Η πτυχιακή αυτή εξερευνά την τεχνολογία της επαυξημένης πραγματικότητας και το πως χρησιμοποιείται στις ηλεκτρονικές εφαρμογές που έχουν στόχο την ψυχαγωγία κάποιου ατόμου.

Στην συγκεκριμένη εργασία, ο στόχος είναι η αναφορά σε ένα παράδειγμα πάνω σε μία εφαρμογή του χώρου ηλεκτρονικών παιχνιδιών που εφαρμόζουν την τεχνολογία αυτή και η εμφάνιση των συμπερασματικών και μελλοντικών προσδοκιών που έχουν εκπονηθεί από αυτή την διαδικασία.

Η ψυχαγωγική εφαρμογή που θα δημιουργήσουμε είναι ένα παιχνίδι της κατηγορίας FPS με ζωντανούς νεκρούς ως αντιπάλους. Το παιχνίδι αυτό θα κατασκευαστεί με την χρήση της μηχανής δημιουργίας παιχνιδιών της εταιρείας Unity. Η εφαρμογή αυτή θα αναμιγνύει ψηφιακό περιεχόμενο, που είναι οι αντίπαλοι και το ψηφιακό περιβάλλον με τον φυσικό χώρο του χρήστη, ο οποίος θα λαμβάνεται από την κάμερα της κινητής συσκευής. Επίσης, θα μπορεί να χρησιμοποιηθεί, από συσκευές με το λογισμικό Android αλλά και iOS. Με αυτό τον τρόπο θα είναι πιο ακριβή τα συμπεράσματα, που θα αναφερθούν στο τέλος, μετά από την δοκιμασία της εφαρμογής. Το περιβάλλον θα είναι μετα-αποκαλυπτικό για να υπάρχει μία συνέργεια με το θέμα της ενσωμάτωσης ζωντανών νεκρών ως αντιπάλους.

Το λογισμικό αυτό που θα δημιουργηθεί, μπορεί να βοηθήσει στην καλύτερη εξήγηση του διδακτικού αντικειμένου, που είναι η τεχνολογία που εφαρμόζεται στην συγκεκριμένη εργασία.

Λέξεις κλειδιά

Επαυξημένη πραγματικότητα, FPS, Android, iOS, Unity

Abstract

Today, there are multiple ways to develop games in the internet, that are popular, as they can be used by amateurs, but also professionals. Technologies, like augmented reality continue on advancing, and are used to make the lives of users better on different sectors. This thesis explores Augmented Reality technology and the ways it can be used on virtual apps, that are used for the entertainment of a person.

Inside the particular project, the objective is to discuss an example of a gaming app that utilizes the aforementioned technology and to show the results and future expectations that have been accounted from this process. The entertainment app we will create is a game of the FPS category with living dead as the main opponents. The game will be constructed by using the game making engine called Unity. This app will mix digital content, the opponents and the digital environment, with the physical space of the user, that will be received from the camera of the mobile device. Also, it will be able to be used, from devices with the software Android, but also iOS. By doing this, the results which will be reported in the end of this thesis, will be more accurate after the testing of the app. The environment will be meta-apocalyptic so that there is a synergy with the theme of the living dead as opponents.

The software that will be created, may help in better explaining the subject, that is the technology that is used in this particular work.

Keywords

Augmented Reality, FPS, Android, iOS, Unity

Ευχαριστίες

Θέλω να ευχαριστήσω την κα. Ελένη Χριστοπούλου που με καθοδήγησε και βοήθησε στην διεκπεραίωση της πτυχιακής εργασίας.

Περιεχόμενα

1	Εισαγωγή	3
1.1	Κίνητρο	4
1.2	Στόχος	4
1.3	Σκοπός	5
1.4	Διάρθρωση	5
2	Βιβλιογραφική Επισκόπηση	6
2.1	Επαυξημένη Πραγματικότητα	6
2.2	Εφαρμογές/Χρήσεις	9
2.2.1	Παιχνίδια	9
2.2.2	Διαφήμιση	9
2.2.3	Στρατός	10
2.2.4	Ιατρική	10
2.2.5	Εκμάθηση	11
2.2.6	Σχεδίαση	11
2.3	Σχετικά AR παιχνίδια	12
3	Επαυξημένη Πραγματικότητα	16
3.1	Επισκόπηση	16
3.2	Εξοπλισμός	17
3.3	Προκλήσεις/Τεχνολογίες AR	17
3.4	Εικονική Πραγματικότητα	18
3.4.1	Ομοιότητες	19
3.4.2	Διαφορές	19
4	Εισαγωγή στην Unity	20
4.1	Μηχανή Unity	20
4.2	Πλεονεκτήματα	22
4.3	Μειονεκτήματα	23
4.4	Βασικά Χαρακτηριστικά	24
5	Πριν την κατασκευή	26
5.1	Αρχική σύλληψη ιδέας	26
5.2	Μάθημα Udemy	27
5.3	Ζωντανοί νεκροί	27
6	Κατασκευή παιχνιδιού	29
6.1	Απαιτήσεις	29

6.2 Assets	30
6.3 Μηχανικές - Mechanics	34
6.4 Scripts	34
6.5 Μετατροπή από iOS σε Android	48
7 Συμπεράσματα	49
7.1 Συμπεράσματα	49
7.2 Μελλοντικές προεκτάσεις	49
8 Ακρωνύμια και συντομογραφίες	

Κεφάλαιο 1

Εισαγωγή

Τα παιχνίδια έχουν μεγάλο ρόλο στην ζωή μας, αφού άτομα από διάφορες ηλικίες ασχολούνται με αυτά, είτε για την εκμάθηση μέσω αυτών είτε για την εξάσκηση και βελτίωση της μνήμης είτε για την ενασχόληση στον ελεύθερο χρόνο για ψυχαγωγία. Με την βοήθειά τους γίνεται η εκμάθηση πιο διαδραστική για τον χρήστη και έτσι μπορεί να βοηθήσει τον χρήστη να “βυθιστεί” σε έναν άλλον κόσμο πιο διασκεδαστικό και να μάθει μέσω αυτόν. Παιχνίδια όπως τα κουίζ είναι κυρίαρχα σε αυτή την κατηγορία αφού πρέπει να απαντήσεις σε ερωτήσεις γνώσεις για να αποκτήσεις το έπαθλο ή την νίκη. Επιπλέον, υπάρχουν παιχνίδια όπως τα παζλ ή ταίριασμα όμοιων εικόνων που εξασκούν τον εγκέφαλο, ώστε να μπορεί να αυξήσει την ικανότητα της απομνημόνευσης του.

Από την άλλη μεριά, υπάρχουν παιχνίδια (που δεν είναι τόσο σοβαρά) που ο μόνος στόχος τους είναι να είναι η ψυχαγωγία χωρίς να εμπεριέχεται μέσα σε αυτά το στοιχείο της γνώσης ή απόκτησης δεξιοτήτων. Αυτά, έχουν διάφορες κατηγορίες και συνεπώς προσφέρουν στον χρήστη την ικανότητα, ανά πάσα στιγμή, να διαλέξει αυτό ακριβώς που τον ενδιαφέρει, όπως είναι Δράση, Περιπέτεια, Κουίζ κτλ.

Το γεγονός ότι τα παιχνίδια αναπτύσσονται συνεχώς και είναι αναγκαία στην ζωή μας, οδηγεί στην ενασχόληση με την δημιουργία τους από εταιρείες αλλά και από χρήστες. Ως αποτέλεσμα, εταιρείες ή χρήστες οδηγούνται στην κατασκευή παιχνιδομηχανών, για την εύκολη δημιουργία ενός συστήματος κατασκευής παιχνιδιών που θα ενσωματώνει νέες τεχνολογίες, ενώ ταυτόχρονα θα κάνει ευκολότερη και πιο γρήγορη την διαδικασία δημιουργίας παιχνιδιών σε πολλαπλά συστήματα (όπως H/Y, φορητές συσκευές, κονσόλες κ.α).

Οι εφαρμογές κατασκευής παιχνιδιών, όπως είναι η Unity και η Unreal Engine διαγωνίζονται συνεχώς για να πάρουν το έπαθλο για το ποια είναι η καλύτερη ανάμεσα σε όλες τις μηχανές. Μηχανές όπως αυτές, έχουν γίνει το επίκεντρο στους κατασκευαστές παιχνιδιών, από τον ερασιτέχνη που θέλει να μάθει για την τεχνολογία της δημιουργίας των βιντεοπαιχνιδιών στον επαγγελματία που δημιουργεί παιχνίδι για επαγγελματικούς σκοπούς.

Το καλύτερο με αυτές τις μηχανές είναι ότι η ανάπτυξη τους είχε την διευκόλυνση του χρήστη, κατά νου όποτε δεν απαιτείται γνώση προγραμματιστικής γλώσσας, αφού υπάρχει το σύστημα ψηφιακών σχεδιαγραμμάτων όπως έχει η Unreal Engine εδώ και πολλά χρόνια και η Unity με την τεχνολογία Bolt που ενσωματώθηκε πρόσφατα.

Σήμερα, η τεχνολογία της Επαυξημένης Πραγματικότητας (AR) είναι μία από τις μεγαλύτερες τεχνολογικές τάσεις παγκοσμίως και αναπτύσσεται εκθετικά, εξαιτίας των έξυπνων φορητών συστημάτων που είναι ευρέως διαθέσιμα.

Με την τεχνολογία αυτή, μπορούμε μέσα από την οθόνη του κινητού μας να ανακαλύψουμε μια νέα πραγματικότητα, και αλληλεπιδρώντας με το περιβάλλον, μπορούμε να διαχειριστούμε γραφικά υπολογιστή στο φυσικό περιβάλλον.

Η τεχνολογία της επαυξημένης πραγματικότητας δεν χρησιμοποιείται αποκλειστικά και μόνο για τα παιχνίδια, αλλά μπορεί να χρησιμοποιηθεί και στα καταστήματα επίπλων (όπως IKEA) που έχει δημιουργήσει εφαρμογή για την δοκιμασία επίπλων στο φυσικό περιβάλλον με το πάτημα ενός κουμπιού. Είναι πραγματικά καινοτόμο, διότι συνδυάζει πολλαπλά διαδραστικά στοιχεία και δημιουργεί μια νέα διαφορετική εμπειρία για τον χρήστη. Η ανάπτυξη ενσωματωμένων πακέτων ή εργαλείων από μηχανές όπως η Unity και άλλων μηχανών κατασκευής παιχνιδιών μπορεί να διευκολύνει τον χρήστη στην ενσωμάτωση νέων τεχνολογιών στα παιχνίδια τους, όπως είναι η επαυξημένη πραγματικότητα.

1.1 Κίνητρο

Το κίνητρο της εργασίας ήταν η επαφή με μοντέρνες αναπτυσσόμενες τεχνολογίες και η έρευνα ως προς το πώς μπορούν να εφαρμοστούν στον τομέα των παιχνιδιών.

Όταν μιλάμε για ηλεκτρονικά παιχνίδια αναφερόμαστε σε παιχνίδια που πραγματοποιούνται με την χρήση κάποιας ηλεκτρονικής συσκευής [1] και, όπως ξέρουμε, είναι κατασκευασμένα έτσι ώστε να προσφέρουν ψυχαγωγία στον παίκτη. Η ψυχαγωγία που προσφέρουν τα κάνει ελκυστικά σε διάφορα άτομα, και υπάρχουν τρόποι ακόμη και να χρησιμοποιηθούν σε διάφορους τομείς, όπως είναι και η εκπαίδευση.

Αυτό, μπορεί να παρατηρηθεί ιδιαίτερα όταν ο χρήστης ενασχολείται με παιχνίδια στην κινητή του συσκευή, μία συσκευή που έχει πάντα μαζί του. Η εύκολη πρόσβαση σε αυτό, σε οποιοδήποτε χώρο, κάνει την ανάπτυξη παιχνιδιών για φορητές συσκευές απαραίτητη και έτσι αναπτύσσονται και ενσωματώνονται συνεχώς νέες τεχνολογίες για να ενσωματωθούν μέσα σε αυτό.

Μία από αυτές τις τεχνολογίες, είναι η AR τεχνολογία. Η προβολή AR περιεχομένου με την χρήση κινητής συσκευής, δεν είναι καινούρια, αλλά με τις διάφορες τεχνολογίες και τις εξελίξεις στον χώρο του hardware, έχει γίνει αρκετά δημοφιλής.

1.2 Στόχος

Πολλοί δημιουργοί εφαρμογών πρέπει να ερευνήσουν σχετικά με νέες τεχνολογίες για να προσαρμόσουν κατάλληλα στις εφαρμογές τους. Στο πεδίο των ψυχαγωγικών εφαρμογών, ενσωματώνονται συνεχώς νέες τεχνολογίες, με κάποιες από αυτές να είναι η AR και η VR τεχνολογία. Ο στόχος της πτυχιακής εργασίας είναι η ανάπτυξη ενός παιχνιδιού επαυξημένης πραγματικότητας για φορητές συσκευές. Συγκεκριμένα, θα δημιουργηθεί ένα παιχνίδι που θα ενσωματώνει τεχνολογίες επαυξημένης πραγματικότητας για την ανάπτυξη της εφαρμογής για iOS και Android συσκευές.

Με αυτή την εφαρμογή, θα δείξουμε κάποιες από τις σημερινές μελέτες που έχουν γίνει πάνω στο θέμα της επαυξημένης πραγματικότητας, και θα δείξουμε ένα παράδειγμα εφαρμογής που μπορεί να φανεί χρήσιμο σε νέους, στον τομέα, κατασκευαστές εφαρμογών σχετικά με την πρόοδο της τεχνολογίας.

1.3 Σκοπός

Ο σκοπός της εργασίας είναι η δημιουργία ενός παιχνιδιού που ενσωματώνει μέσα σε αυτό την τεχνολογία επαυξημένης πραγματικότητας και είναι της κατηγορίας FPS. Με αυτόν τον τρόπο μπορούμε να εξετάσουμε την πρόοδο των τεχνολογιών επαυξημένης πραγματικότητας που μπορούν να χρησιμοποιηθούν σήμερα και να δούμε πως αξιοποιούνται μέσα από ένα μικρό παιχνίδι ως παράδειγμα. Οι περισσότερες έρευνες στα παιχνίδια της κατηγορίας αυτής δεν χρησιμοποιούν την μηχανή της Unity για την κατασκευή τους, οπότε θέλουμε να δείξουμε ποια πακέτα μπορούν να χρησιμοποιηθούν για την δημιουργία από τους χρήστες.

1.4 Διάρθρωση

Η δομή της πτυχιακής εργασίας είναι η βιβλιογραφική επισκόπηση όπου θα γίνεται ανάλυση παρόμοιων εργασιών/άρθρων στο κεφάλαιο 2. Επίσης, εξήγηση του θέματος της επαυξημένης πραγματικότητας στο κεφάλαιο 3. Στη συνέχεια, στα κεφάλαια 4 και 5 θα γίνεται μία ανάλυση του θέματος της παιχνιδομηχανής Unity και των εμπλεκόμενων τεχνολογιών. Τέλος, στο κεφάλαιο 6 θα γίνεται μια παρουσίαση του παιχνιδιού και στο κεφάλαιο 7 θα είναι τα συμπεράσματα και πιθανές μελλοντικές προεκτάσεις.

Κεφάλαιο 2

Βιβλιογραφική Επισκόπηση

Στο πεδίο των παιχνιδιών που ενσωματώνουν τεχνολογίες επαυξημένης πραγματικότητας υπάρχουν πολλές σχετικές έρευνες οι οποίες θα συζητηθούν παρακάτω.

2.1 Επαυξημένη Πραγματικότητα

Η τεχνολογία επαυξημένης πραγματικότητας, δημιουργήθηκε από τον Ivan Sutherland [2] μαζί με τον μαθητή του Bob Sproull, όταν δημιούργησαν μαζί μία συσκευή HMD, δηλαδή συσκευή που εξοπλίζεται στην κεφαλή, που πραγματοποιούσε ανάμιξη/επικάλυψη της πραγματικότητας με ψηφιακό περιεχόμενο το 1968.

Η συσκευή που δημιούργησαν πήρε την ονομασία "Δαμόκλεια σπάθη".

Κάποιες πηγές [3] υποστηρίζουν ότι αυτή η ονομασία προήλθε από το γεγονός ότι η μηχανή ήταν πολύ βαριά και ως εξίσου έπρεπε να ήταν κρεμασμένη από την οροφή του εργαστηρίου. Το γεγονός αυτό, σε συνδυασμό με την γιγαντώδης εμφάνιση του αντικειμένου, αποτέλεσε έμπνευση για την ονομασία αυτή.



Σχήμα 2.1: **Το HMD του Ivan Sutherland, αλλιώς γνωστό ως "Δαμόκλεια Σπάθη (Sword of Damocles)" [2]**

Η μηχανή στο σχήμα εξοπλίζονταν μηχανικά στην κεφαλή του χρήστη και η προβολή του περιεχομένου πραγματοποιούνταν χρησιμοποιώντας δύο καθοδικούς σωλήνες (CRT). Το οπτικό σύστημα της μηχανής μεγέθυνε την ψηφιακή εικόνα σε κάθε έναν από τους δύο σωλήνες και στην συνέχεια εμφανίζονταν η εικόνα περίπου 18 ίντσες μπροστά από τον χρήστη. Η μηχανή χρησιμοποιούσε τον ίδιο καθρέφτη που χρησιμοποιείται σήμερα στους καθρέπτες δύο κατευθύνσεων.

Αυτός επέτρεπε τον χρήστη να βλέπει την ψηφιακή εικόνα από τους καθοδικούς σωλήνες, ενώ παράλληλα να παρατηρεί και τα αντικείμενα στο εξωτερικό περιβάλλον του. Με αυτό τον τρόπο μπορούσε να παρατηρεί περιεχόμενο αυτόνομο από την μηχανή, (που δεν έχει σχέση με τον εξωτερικό χώρο) ή να βλέπει ψηφιακές εικόνες που συσχετιζόνταν με κάποιο αντικείμενο στον χώρο.

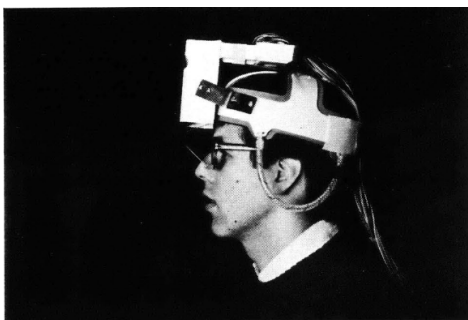
Το HMD του Ivan Sutherland έμοιαζε με μία παλιότερη εφεύρεση που ονομάζεται Telesphere mask. Εκείνη η συσκευή, καταχωρήθηκε ως πατέντα το 1960 και δημιουργήθηκε από τον δημιουργό ταινιών, Morton Heilig, αν και η παλιά εφεύρεση δεν περιείχε καταγραφή κίνησης [4].



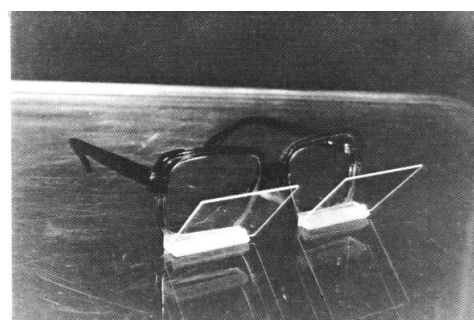
Σχήμα 2.2: **Telesphere Mask, a head-mounted display (HMD), a non-interactive film medium without motion tracking**

Η συσκευή εκείνη ήταν ογκώδης και ήταν φτιαγμένη για χρήση από ένα άτομο, αφού κυκλοφορούσε σε φορητή μορφή για να δώσει την δυνατότητα στον χρήστη να την τοποθετήσει στην κεφαλή του [5]. Αποτελούνταν από οπτικές μονάδες, τηλεοπτικές μονάδες σωλήνων, ακουστικά και ακροφύσια εκκένωσης. Η μάσκα προσομοίωσης αυτή, περιείχε στερεοσκοπική ευρεία προβολή και στερεοφωνικό ήχο.

Το 1983, ο M. Callahan έκανε διδακτορικό στο MIT πάνω σε μία συσκευή που κατασκεύασε, που αν και έμοιαζε με αυτήν του Ivan Sutherland περιείχε ορισμένες νέες τεχνικές και μηχανισμούς [6]. Η μηχανή αυτή χρησιμοποιούσε καθρέφτες και μονόχρωμους CRTs. Η πηγή της εικόνας προέρχονταν από τα CRT που κοιτούσαν προς τα κάτω με τους καθρέφτες μπροστά από τον χρήστη. Η κανονική όραση του χρήστη γίνεται μέσα από τα γυαλιά, ενώ οι εικόνες αντικατοπτρίζονταν μέσα από τους καθρέφτες. Ο χρήστης μπορούσε να "ακουμπήσει" ή να "κουνήσει" την εικόνα με το χέρι του, και να "κοιτάξει γύρω του" στην εικόνα.



(α) Το HMD

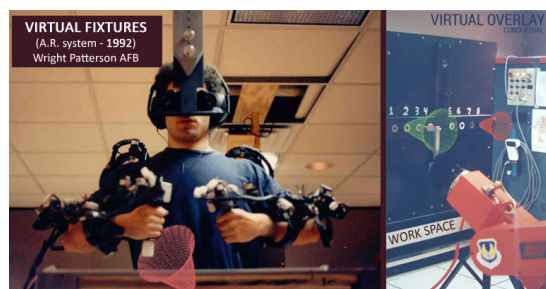


(β) Τα γυαλιά

Σχήμα 2.3: **Η μηχανή που φορούσε ο χρήστης στην κεφαλή και τα γυαλιά με τους καθρέφτες**

Υπήρχαν αρκετές προσπάθειες στην εφεύρεση μίας τέτοιας μηχανής, αλλά ο όρος “Επαυξημένη Πραγματικότητα” δεν εμφανίστηκε μέχρι πιο μετά. Ο όρος “Επαυξημένη Πραγματικότητα” επινοήθηκε το 1990, από τον ερευνητή Boeing αεροπλάνων, Thomas Caudell. Μαζί με τον συνάδελφό του, David Mizell, είχαν ανατεθεί να δημιουργήσουν έναν άλλον τρόπο για να πραγματοποιήσουν μία διαδικασία στο χώρο ενός εργοστασίου [7].

Η διαδικασία αυτή αφορούσε την αλλαγή των διαγραμμάτων και των συσκευών σήμανσης που καθοδηγούσε τους εργάτες, για να αντικαταστήσει τα κόντρα πλακέ με τις σημάνσεις που χρησιμοποιούσαν τότε. Με αυτό τους ήρθε η ιδέα να χρησιμοποιήσουν μια συσκευή HMD που θα φορούσε ο εργάτης και θα έδειχνε τα σχέδια του συγκεκριμένου αεροπλάνου που χρειάζονταν ο εργάτης. Μετά θα τα έβλεπαν ζωντανά στις οανίδες.



Σχήμα 2.4: **Η συσκευή Visual Fixtures, ένα από τα πρώτα ολοκληρωμένα AR συστήματα**

Στη συνέχεια, ο Louis Rosenberg δημιούργησε την συσκευή με την ονομασία Visual Fixtures που ήταν μία καινοτόμα εφεύρεση πάνω στην εικονική και επαυξημένη πραγματικότητα. Η μηχανή αυτή κατασκευάστηκε το 1992 στα εργαστήρια Armstrong της United States Air Force.

Ήταν ένα από τα πρώτα ολοκληρωμένα συστήματα επαυξημένης πραγματικότητας και σύστηνε την άποψη της βύθισης της εφαρμογής στον χρήστη.

Τα 3D γραφικά ήταν πολύ αργά και έτσι χρησιμοποιούσαν δύο ρομπότ που ελέγχονταν από έναν εξωσκελετό [8] που φορούσε ο χρήστης. Χρησιμοποιούσε ειδικούς φακούς ώστε να μεγεθύνει την προβολή του χρήστη και να φαίνεται από την εικόνα του χρήστη ότι τα χέρια του είναι τα μηχανικά, από τα ρομπότ, που είχε πλάι του. Αυτό σημαίνει ότι όταν έπαιζε ο χρήστης, φαινόταν πως στην θέση που βρίσκονταν κανονικά τα χέρια του, ο χρήστης έβλεπε τα ρομποτικά/μεταλλικά ως χέρια του. Η έρευνα του Rosenberg πρότεινε την χρήση της μηχανής, στα συστήματα τηλεπαρουσίας.

Το 1999, η NASA δημιούργησε ένα υβριδικό σύστημα συνθετικής όρασης για το αεροσκάφος X-38. Η μηχανή έκανε χρήση της AR τεχνολογίας για να συνεισφέρει στην καλύτερη πλοήγηση του αεροσκάφους στις δοκιμές που πραγματοποίησαν.



Σχήμα 2.5: Το σύστημα συνθετικής όρασης της NASA [9]

Τα τελευταία χρόνια, με τις αλλαγές του λογισμικού και hardware έχουν ερευνηθεί νέες συσκευές και εφαρμογές, που μπορούν να χρησιμοποιηθούν για να χρησιμοποιήσουν την τεχνολογία αυτή. Αυτό, προήλθε από την ανάγκη των χρηστών και τις εξελίξεις της τεχνολογίας.

2.2 Εφαρμογές/Χρήσεις

Η Επαυξημένη Πραγματικότητα χρησιμοποιείται σε διάφορους τομείς για να βελτιώσει την ζωή των ατόμων. Αυτό συμβαίνει, επειδή μπορεί να προβάλλει ψηφιακό υλικό στο φυσικό περιβάλλον σε πραγματικό χρόνο. Με αυτό τον τρόπο, μπορεί για παράδειγμα ο χρήστης να αντλήσει χρήσιμες πληροφορίες για την πραγματοποίηση κάποιας ανάλυσης, να "αντικαταστήσει" κάποιο αντικείμενο στον φυσικό χώρο με κάποιο ψηφιακό κ.α. Σε αυτό το κεφάλαιο θα μιλήσουμε για κάποιες γνωστές εφαρμογές της τεχνολογίας.

2.2.1 Παιχνίδια

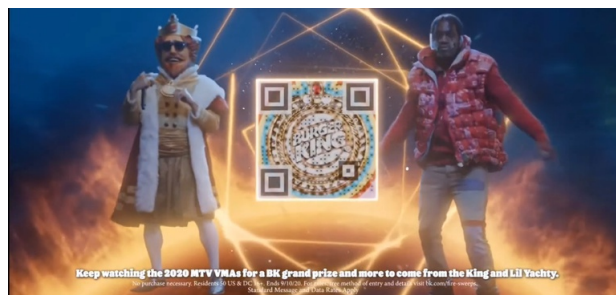
Το δημοφιλές παιχνίδι Pokemon GO [10] αποτελεί ένα πολύ καλό παράδειγμα της χρήσης της τεχνολογίας αυτής στον χώρο των παιχνιδιών [11]. Η ταχύτατη ανάπτυξη της τεχνολογίας βιομηχανίας παιχνιδιών, οδήγησε στην ενσωμάτωση διάφορων νέων τεχνολογιών, με μία από αυτές να είναι και η Επαυξημένη Πραγματικότητα.

Η τεχνολογία αυτή, βοηθάει στο να δημιουργηθούν πιο ελκυστικά παιχνίδια που προσελκύουν τους χρήστες [12], εξαιτίας των ιδιαίτερων χαρακτηριστικών τους. Αυτό συμβαίνει, διότι τα παιχνίδια αυτά απαιτούν τον χρήστη να είναι πιο ενεργός για την ενασχόληση με το παιχνίδι του. Με τον τρόπο αυτό, αυξάνεται η "εμβύθιση" του ατόμου στον κόσμο του παιχνιδιού.

2.2.2 Διαφήμιση

Ο σημερινός κόσμος εξαρτάται πολύ από την τεχνολογία, διότι υπάρχουν περίπου 4.5 δισεκατομμύρια άτομα που χρησιμοποιούν το διαδίκτυο παγκοσμίως [13].

Η διαφήμιση συνεχώς χρησιμοποιεί νέες τεχνολογίες για την προβολή των προϊόντων τους. Το γεγονός αυτό, σε συνδυασμό με την ταχεία ανάπτυξη των έξυπνων κινητών τηλεφώνων, οδηγεί στην εφεύρεση νέων τρόπων εμφάνισης των προϊόντων στους πελάτες. Οι διαφημίσεις αυτές συνήθως περιέχουν έναν QR κωδικό, εικόνες ως σημάδια ή κάποια συγκεκριμένη τοποθεσία (κτήριο) που εντοπίζεται με το σύστημα GPS της κινητής συσκευής.



Σχήμα 2.6: Η διαφήμιση της αλυσίδας Burger King στα VMA [14]

Ένα παράδειγμα χρήσης του QR κωδικού, είναι αυτό της αλυσίδας Burger King στα VMA, όπου ζητούσαν από το κοινό να σαρώσει τον κωδικό για να ενεργοποιήσουν μία AR εμπειρία [14]. Αυτή η εμπειρία περιείχε την μασκότ της αλυσίδας μαζί με έναν γνωστό ράπερ, με την ονομασία Lil Yachty. Αυτή η εμπειρία κέρδιζε κάποιες προσφορές στο κοινό όπως ένα δωρεάν Whopper σάντουιτς και συμμετοχή στην κλήρωση για την νίκη μίας ετήσιας προμήθειας τέτοιων σάντουιτς χωρίς χρέωση.

Ένα άλλο παράδειγμα, είναι αυτό των κατασκευαστικών εταιρειών, όπως είναι οι εταιρείες κατασκευής σπιτιών, που δείχνουν στον χρήστη πως θα είναι το τελειοποιημένο προϊόν πριν την ολοκληρωμένη κατασκευή του. Με αυτό τον τρόπο μπορούν να ελέγξουν για λάθη στην κατασκευή και αφού ολοκληρωθεί το προϊόν, να δουν αν όλα εγκαταστάθηκαν σωστά [15].

2.2.3 Στρατός

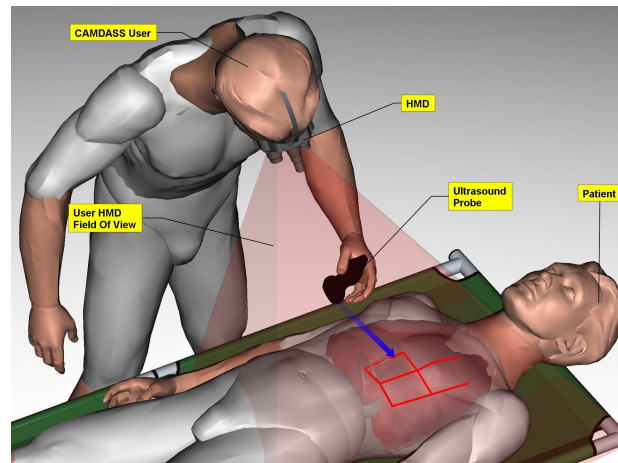
Η τεχνολογία χρησιμοποιείται πολύ για στρατιωτικούς σκοπούς εδώ και πολλά χρόνια. Αρχικά, ένα ενδιαφέρον παράδειγμα προέρχεται από την Rockwell International που δημιούργησε ψηφιακούς χάρτες που επικαλύπτονταν πάνω στους φυσικούς χάρτες και έδειχναν τις δορυφορικές και κυκλοειδείς πορείες κομματιών συντριμμίων για να βοηθήσει στις παρακολουθήσεις των σπιτιών στο Air Force Island Optical System [16]. Στο άρθρο που έγραψαν οι συγγραφείς αναφέρουν την ζωντανή καταγραφή δορυφορικών τροχιών και κυκλοειδών κομματιών από συντρίμια, από αστυνομικούς. Μετά τα δεδομένα αυτά, λαμβάνονταν και γίνονταν ψηφιακοί χάρτες που χρησιμοποιούνταν στα τηλεσκόπια για να μπορούν οι χρήστες να ελέγξουν για επικίνδυνα κομμάτια συντριμμίων και διάφορες απειλές από ασφαλή απόσταση. Οι χάρτες έδειχναν τις τροχιές των αντικειμένων σε γεωγραφικές συντεταγμένες.

Ένα ακόμα αξιόλογο παράδειγμα είναι αυτού των drone. Το 2003 η τεχνολογία AR εφαρμόστηκε από τον αμερικανικό στρατό στο Shadow Pilotless Aerial System, η αλλιώς drones, για την καλύτερη αναγνώριση ατόμων ή σημείων ενδιαφέροντος. Το σύστημα έδειχνε γεωγραφικές συντεταγμένες μαζί με άλλα δεδομένα όπως οι οδοί, τα σημεία ενδιαφέροντος κ.α. σε συνδυασμό με το βίντεο από την κάμερα. Με αυτόν τον τρόπο οι χειριστές των drone είχαν μία πιο καλή επίγνωση του πεδίου.

2.2.4 Ιατρική

Ένα παράδειγμα της χρήσης της τεχνολογίας στην ιατρική είναι αυτό της European Space Agency με την συσκευή με ονομασία CAMDASS: Computer Assisted Medical Diagnosis and Surgery System. Αυτή η μηχανή είναι κατασκευασμένη για την επίλυση ιατρικών θεμάτων στους αστροναύτες.

Είναι ακόμα σε πειραματικό στάδιο και επιτρέπει μόνο εξέταση με υπερήχους προσωρινά. Ο χρήστης εξοπλίζει την συσκευή αυτή στην κεφαλή του και το μηχάνημα τον βοηθάει για την εξέταση με τους υπερήχους με την χρήση της οθόνης, του συστήματος υπερήχου και της κάμερας υπερήχου που περιέχει [17]. Τα δεδομένα του σώματος του ασθενούς εμφανίζονται στην κάμερα, όπου παρέχονται οι απαραίτητες οδηγίες στον χρήστη.



Σχήμα 2.7: Το CAMDASS με παράδειγμα της χρήσης του [17]

Άλλες εφαρμογές της τεχνολογίας στον τομέα αυτόν, είναι η ζωντανή εμφάνιση των δεδομένων των ασθενών, με λειτουργικά βίντεο του, η παροχή δεδομένων στους χειρουργούς σχετικά με το που βρίσκεται το έμβρυο μέσα στην μήτρα [18], η προβολή συμπληρωματικών όγκων και αρτηριών για την χειρουργική [19] κ.α.

2.2.5 Εκμάθηση

Μέσα από την τεχνολογία αυτή μπορεί ο χρήστης να αποκτήσει γνώσεις και δεξιότητες. Αυτό συμβαίνει, διότι η τεχνολογία είναι πιο διαδραστική και ελκύει περισσότερο τον χρήστη. Είναι ένας πιο καινοτόμος τρόπος απόκτησης γνώσης σε σχέση με τον παραδοσιακό τρόπο, δηλαδή την ανάγνωση βιβλίων και την παρακολούθηση μαθημάτων.

Για παράδειγμα, ένας μαθητής μπορεί να μάθει ανατομία χρησιμοποιώντας την τεχνολογία αυτή, παρατηρώντας ζωντανά τα οστά, τους μυς και τις αρτηρίες που υπάρχουν μέσα στο σώμα, και αντλώντας πληροφορίες για το καθένα σε πραγματικό χρόνο [20]. Έτσι γίνεται πιο παραγωγική η εκμάθηση και απορροφάται πιο εύκολα από τους μαθητές.

Η εφαρμογή Complete Anatomy είναι μια εφαρμογή που βοηθάει στην εκπαίδευση [21]. Επιτρέπει τα άτομα να παρατηρήσουν το σώμα μέχρι και σε κυτταρικό επίπεδο και να δουν διάφορα μέρη, όπως τα όργανα σε δράση. Με την εφαρμογή μπορούν να δουν για παράδειγμα την καρδιά πως χτυπάει ζωντανά.

2.2.6 Σχεδίαση

Στην σχεδίαση είναι επίσης πολύ χρήσιμη η τεχνολογία αυτή. Με την χρήση της, μπορεί ένας σχεδιαστής να δει το σχέδιο του να "έρχεται στην ζωή", αφού μπορεί να προβάλλει το CAD μοντέλο του ζωντανά. [22]. Ένα παράδειγμα, μπορεί να είναι μία μηχανή αυτοκινήτου.

Με την προβολή του 3D μοντέλου σε πραγματικό χρόνο μειώνεται η ζήτηση για φυσικά μοντέλα και ως εξίσου μειώνεται το κόστος. Επίσης, μπορεί να χρησιμοποιηθεί για να επιθεωρηθεί ένα αντικείμενο πριν την κατασκευή του για να αντληθούν κριτικές και άλλες γνώμες σχετικά με το σχέδιο.

Η εφαρμογή της τεχνολογίας είναι ενδιαφέρον, επειδή δίνει μια ελευθερία στον σχεδιαστή CAD μοντέλων επιτρέποντας την διάδραση με την αλληλεπίδραση μοντέλου-χρήστη και την προβολή του μοντέλου στον φυσικό χώρο στο κανονικό του μέγεθος.

2.3 Σχετικά AR παιχνίδια

Ένα παιχνίδι που δημιουργήθηκε για την εκμάθηση είναι το TimeWarp [23], ένα υπαίθριο παιχνίδι επαυξημένης πραγματικότητας που παίζεται στην πόλη της Κολωνίας, Γερμανία. Μέσα από την εφαρμογή μπορεί ο παίκτης να εξερευνήσει την ιστορία της πόλης με την χρήση της τεχνολογίας AR. Το παιχνίδι βασίζεται σε έναν γερμανικό μύθο, τον λεγόμενο "Heinzelmannchen" της Γερμανίας. Ο σκοπός του παιχνιδιού είναι η εύρεση κάποιων ξωτικών που βοηθούσαν τους ανθρώπους μέχρι που κάποια μέρα εξαφανίστηκαν. Για την επίτευξη αυτού, ο χρήστης "χρονοταξιδεύει" σε διαφορές χρονικές περιόδους για να τα εντοπίσει, κάτι που κάνει το παιχνίδι ψυχαγωγικό, ενώ επιτρέπει ταυτόχρονα στον χρήστη να μάθει πληροφορίες για την ιστορία της πόλης της Γερμανίας.

Είναι αδύνατον να παραλείψουμε επίσης την αναφορά στο παιχνίδι που έκανε διάσημη την AR τεχνολογία, το Pokemon GO [10]. Το Pokemon GO ήταν ένα από τα πρόσφατα επιτυχημένα παιχνίδια στην αγορά και κυκλοφόρησε το 2016. Είχε κοινωνικά πρόνομια επειδή επέτρεπε στους χρήστες να έρθουν σε επαφή με άλλους χρήστες. Οι χρήστες, επιπλέον, είχαν την δυνατότητα να δημιουργήσουν κοινότητες και να συνδεθούν με άλλα άτομα με κοινά ενδιαφέροντα, ειδικά στα μέρη που είχε πολλά Πόκεμον σε μεγάλες πόλεις. Επιπροσθέτως, είχε μια συμβολή στην υγεία των ατόμων, αφού ενεργοποιούσε τους χρήστες να γυμναστούν ενώ περπατάνε για να προχωρήσουν το παιχνίδι. Ένα άλλο παιχνίδι είναι το ARQuake [24], το οποίο είναι ένα παιχνίδι που προορίζεται για ψυχαγωγία σε εξωτερικούς χώρους. Ουσιαστικά σου δίνει την δυνατότητα να παίζεις ένα First-Person Shooter (FPS) παιχνίδι που δημιουργήθηκε το 1996, το Quake ενσωματώνοντας την επαυξημένη τεχνολογία μέσα σε αυτό. Σε αντίθεση με το πρωτότυπο παιχνίδι που προοριζόταν για υπολογιστή, το παιχνίδι σου επιτρέπει να "βλέπεις" τα τέρατα σε πραγματικούς χώρους με ένα διάφανο HMD (σύστημα όμοιο με αυτό της VR τεχνολογίας που τοποθετείται στην κεφαλή του χρήστη) και ενισχύει την εμπειρία επιτρέποντας την χρήση ακόμα και πλαστικών όπλων με εικονική ανάκρουση για να τα πυροβολήσεις. Φυσικά, αυτό ήταν αδύνατο χωρίς να γίνουν κάποιες τροποποιήσεις στο αρχικό παιχνίδι. Αρχικά, δεν γίνεται να παίζεις άλλο με το ποντίκι και το πληκτρολόγιο. Επίσης, τροποποίησαν κατάλληλα τον κόσμο του Quake ώστε να μην απαιτείται να πετάς ή να κολυμπάς (στις αποστολές) για την ολοκλήρωση του παιχνιδιού.

Τέλος, μείωσαν τους εχθρούς γιατί δεν σου παρέχεται η δυνατότητα να χρησιμοποιήσεις υπερδυνάμεις (για να είναι πιο ρεαλιστικό) και τροποποιήθηκαν οι υφές των δερμάτων των τεράτων για να φαίνονται σωστά μέσα από το διάφανο HMD του χρήστη.

Επιπλέον, υπάρχει και το Human Pac-man [25], ένα ψυχαγωγικό και φορητό παιχνίδι εξωτερικού χώρου το οποίο τοποθετεί τον παίκτη στην θέση του πρωταγωνιστή του κλασικού παιχνιδιού Pac-man. Το Pac-man είναι ένα κλασικό παιχνίδι που κυκλοφόρησε από την Namco το 1980. Κυκλοφόρησε με την ονομασία Pac-man αντί για Puck-man, διότι η ονομασία θα ήταν υβριστική με την αλλαγή ενός γράμματος [26].



Σχήμα 2.8: Το κλασικό παιχνίδι, Pac-Man [26]

Το παιχνίδι αυτό είναι ένα παιχνίδι που περιέχει κυνήγι μέσα σε ένα λαβύρινθο, όπου πρωταγωνιστεί ο γνωστός χαρακτήρας Pac-man και στόχος είναι να φάει όλες τις κουκκίδες, ενώ ταυτόχρονα αποφεύγει τους εχθρούς του. Οι εχθροί του είναι τα 4 φαντάσματα, Blinky (κόκκινο), Pinky (ρόζ), Inky (κυανό) και Clyde (πορτοκαλί). Αν ο Pac-man φάει όλες τις κουκκίδες τότε προχωράει στην επόμενη πίστα, ενώ αν τον προλάβουν τα φαντάσματα χάνει.

Στο Human-Pacman ο παίκτης μπαίνει στον ρόλο του Pac-man και τρώει τις κουκκίδες (μπισκότα), το οποίο παρατηρείται στον πραγματικό κόσμο λόγω της εξαφάνισης των κουκκίδων μετά από την προσπέλασή τους. Αντίθετα με το κλασικό παιχνίδι, στο παιχνίδι δεν παίζει μόνο ένας παίκτης αλλά 8 παίκτες που χωρίζονται σε 2 ομάδες. Στην μία ομάδα είναι Pac-man και 2 βοηθοί και στην άλλη ομάδα είναι 2 φαντάσματα και 2 βοηθοί. Κάθε Pac-man ή φάντασμα έχει και έναν βοηθό, που συνεισφέρει απομακρυσμένα στον παίκτη με χρήση σύνδεσης WIFI από έναν υπολογιστή για να βοηθήσει τον Pac-man να συλλέξει όλα τα μπισκότα και το φάντασμα να εντοπίσει τον Pac-man αντίστοιχα. Ακόμα, υπάρχουν και άλλα ειδικά μπισκότα όπου ο παίκτης αγγίζει αληθινά Bluetooth αντικείμενα που βρίσκονται μέσα στον χώρο του παιχνιδιού. Όταν ο παίκτης βρίσκεται σε εμβέλεια 10 μέτρων και αν έχει τα κριτήρια για την συλλογή του ειδικού αντικειμένου, τότε θα υπάρχει επικοινωνία μεταξύ του φορητού υπολογιστή (που επιτρέπει την AR λειτουργία) και του αντικειμένου, με την μετάδοση ενός μηνύματος στον χρήστη για την συλλογή του.

Φυσικά, υπάρχουν πολλαπλά παιχνίδια που δεν εξαρτώνται από φορητούς υπολογιστές και διάφανα HMD για την εφαρμογή του AR, αλλά από χρήση έξυπνης κινητής συσκευής για την μίξη του πραγματικού κόσμου με γραφικό περιεχόμενο. Ένα παράδειγμα για αυτό μπορεί να θεωρηθεί το ARSoccer [27], ένα παιχνίδι στο οποίο υπάρχει ένα γραφικό τέρμα και πρέπει ο παίκτης να κλωτσήσει την μπάλα μέσα σε αυτό για να σκοράρει.

Ο τρόπος που μπορεί να πραγματοποιηθεί αυτό είναι να τοποθετήσεις το πόδι σου στην θέση που εμφανίζεται μέσα από το κινητό στον χώρο σου και μετά να κλωτσήσεις την εικονική μπάλα στο τέρμα. Μετά από το επιτυχές σκορ του παίκτη, υπάρχει ένα ηχητικό εφέ. Αυτή η εφαρμογή, ενεργοποιεί τον παίκτη να συμμετέχει φυσικά κλωτσώντας μια εικονική μπάλα, κάτι που είναι ενδιαφέρουσα προσθήκη στο χώρο της AR τεχνολογίας. Ένα ακόμα ενδιαφέρον παιχνίδι, είναι το παιχνίδι AR Tennis [28] που επιτρέπει τον χρήστη να παίζει με άλλα άτομα που πραγματοποιείται με επικοινωνία μεταξύ των δύο συσκευών.

Η επικοινωνία εξασφαλίζεται με την χρήση τεχνολογίας bluetooth και κάτι ακόμα που είναι διαφορετικό από τα άλλα παιχνίδια που προαναφέρθηκαν είναι το γεγονός ότι απαιτεί ένα πληκτρολόγιο κινητής συσκευής. Συνεπώς, δεν ακολουθεί το προηγούμενο μοτίβο, δηλαδή την χρήση έξυπνης κινητής συσκευής όπως τα άλλα αλλά τα παλιά κινητά με κουμπιά. Μέσα από την οθόνη του κινητού ο χρήστης μπορεί να δει ένα γήπεδο τένις στο πραγματικό του περιβάλλον και με την επικοινωνία με μια άλλη συσκευή μπορεί να περάσει διασκεδαστικά τον χρόνο του. Σε αυτή την εφαρμογή, ή μόνη εισαγωγή από τον χρήστη είναι το κουμπί 2 στο πληκτρολόγιο του κινητού κατά το σερβίς.

Κάποιες άλλες ενδιαφέρουσες προσπάθειες σε παιχνίδι επαυξημένης πραγματικότητας είναι τα παιχνίδια Kkongalmon, BattleBoard 3D και ARZombie.

Το Kκολγαμον [29] είναι ένα παιχνίδι επαυξημένης πραγματικότητας και χρησιμοποιεί τις γεωγραφικές πληροφορίες του κινητού τηλεφώνου κατά την εκτέλεση του. Το παιχνίδι είναι ένα παιχνίδι που λειτουργεί ως ένας τουριστικός οδηγός για τον χρήστη. Στο παιχνίδι αυτό εντοπίζεις ψηφιακά τέρατα με την χρήση της κάμερας, GPS και γεωγραφικής πληροφορίας ενός τηλεφώνου. Στο παιχνίδι αυτό σου δίνονται πληροφορίες σχετικά με το που θα βρείς το ψηφιακό τέρας στον χάρτη. Η εφαρμογή θυμίζει το παιχνίδι Pokemon GO αφού και τα δύο χρησιμοποιούν GPS τεχνολογία και σε κατεύθυνουν, για την εύρεση ψηφιακών τεράτων.

Το BattleBoard 3D [30] είναι ένα παιχνίδι που δημιουργήθηκε για παιδιά. Στο παιχνίδι αυτό υπάρχουν οι παίκτες χωρίζονται σε 2 ομάδες και ο στόχος είναι η σύλληψη του σεντουκιού του αντιπάλου. Το παιχνίδι αυτό χρησιμοποιεί επαυξημένη τεχνολογία αλλά δεν δημιουργεί στο περιβάλλον κάτι καινούριο αλλά τροποποιεί κάτι που υπάρχει στον χώρο αυτό. Αυτό, σημαίνει ότι χρειάζεται ο παίκτης να έχει κάποια φυσικά κομμάτια. Στην περίπτωση του παιχνιδιού, επειδή το παιχνίδι προορίζεται για παιδιά, οι δημιουργοί χρησιμοποίησαν LEGO τουβλάκια που είναι ήδη διαθέσιμα στα περισσότερα παιδιά. Το παιχνίδι αυτό επειδή περιέχει χρήση της AR τεχνολογίας, επιτρέπει την εισαγωγή 3D μοντέλων animation στα φυσικά κομμάτια. Το παιχνίδι αυτό παίζεται σε μία επιφάνεια που έχει διαστάσεις 6x7. Θυμίζει την σκακiera που χρησιμοποιούμε στο σκάκι, με την διαφορά ότι εκείνη είναι 8x8. Πήρε έμπνευση από το σκάκι και στο παιχνίδι διαχωρίστηκαν τα φυσικά κομμάτια σε 3 επίπεδα. Αυτό σημαίνει ότι το επίπεδο των κομματιών θα επηρεάζει την νίκη. Για παράδειγμα, αν ένας παίκτης έχει το υψηλότερο επίπεδο στο κομμάτι του θα είναι ο νικητής, ενώ αν ο παίκτης έχει το ίδιο επίπεδο φυσικού κομματιού με τον αντίπαλο οι πιθανότητες θα είναι 50/50. Κάθε παίκτης ξεκινάει με 7 κομμάτια, τρία κομμάτια επιπέδου ένα, δύο επιπέδου δύο, ένα επιπέδου τρία και ένα σεντούκι. Τα κομμάτια μπορούν να τοποθετηθούν σε 2 γραμμές (rows) του ταμπλό. Το κάθε κομμάτι μπορεί να μεταβεί μία θέση ανά επίπεδο με αυτά του επιπέδου ένα να περιορίζονται σε μία διεύθυνση, δηλαδή να μπορούν να μεταβούν μόνο οριζόντια. Αυτά που ανήκουν στο επίπεδο 2 και 3 μπορούν να μεταβούν σε οριζόντια αλλά και κάθετη διεύθυνση και τα σεντούκια δεν μπορούν να κουνηθούν.

Είναι αρκετά ενδιαφέρον προσέγγιση αφού χρησιμοποιεί το ARToolkit για την δημιουργία του, ένα εργαλείο που χρησιμοποιεί μία κάμερα για να εντοπίσει ένα σημάδι που έχουμε θέσει προηγουμένως, ένα μαύρο τετράγωνο σχήμα με ένα γραφικό σύμβολο. Το σημάδι αυτό μπορεί να χρησιμοποιηθεί για να τοποθετήσουμε 3D γραφικά σχεδιασμένα από υπολογιστή πάνω σε αυτά που εμφανίζονται στις οθόνες ή στο headset του παίκτη.

Το παιχνίδι ARZombie [31] είναι ένα FPS παιχνίδι παρόμοιο με την παρούσα πτυχιακή, με την διαφορά ότι η πτυχιακή δεν χρησιμοποιεί τεχνολογία face-tracking. Στο παιχνίδι αυτό πρέπει ο παίκτης να σκοτώσει ζωντανούς νεκρούς πυροβολώντας τους στον εγκέφαλο, επειδή ένας ιός διέρρευσε από κάποιο εργαστήριο πανεπιστημίου και μετατράπηκαν οι άρρωστοι αυτοί σε ζωντανούς νεκρούς. Το παιχνίδι χρησιμοποιεί αντικείμενα που υπάρχουν ήδη στο περιβάλλον. Ο στόχος του παιχνιδιού είναι η εξάλειψη των μολυσμένων, που πραγματοποιείται από τον παίκτη, για την συλλογή πόντων. Χρησιμοποιεί την τεχνολογία face-tracking για να εντοπίσει αν κάποιο άτομο στο περιβάλλον είναι μολυσμένος ή όχι. Με την βοήθεια της συσκευής που τρέχει το παιχνίδι, οι χρήστες μπορούν να ελέγχουν τα πρόσωπα των ατόμων για να εξακριβώσουν αν κάποιο άτομο είναι μολυσμένο, δηλαδή αν είναι ζωντανός νεκρός ή όχι. Ο χρήστης έχει 2 επιλογές για τον πυροβολισμό των αντιπάλων, τον αυτόματο πυροβολισμό μετά την επιτυχής εξακρίβωση μολυσμένου ατόμου και τον χειροκίνητο πυροβολισμό μολυσμένου ατόμου με πάτημα στην οθόνη συσκευής. Η εφαρμογή έχει ειδικά εφέ για τα αίματα μετά την επίθεση αντίπαλου και μπόνους που βοηθάνε στην αναπλήρωση ζωής του παίκτη. Ακόμα, υπάρχει η μπάρα ζωής και ο μετρητής διαθέσιμων πυρομαχικών στην διεπαφή του χρήστη, για να έχει την δυνατότητα ο παίκτης ανά πάσα στιγμή να γνωρίζει πόση ζωή και πόσες σφαίρες έχει αντίστοιχα.

Κεφάλαιο 3

Επαυξημένη Πραγματικότητα

Η τεχνολογία της επαυξημένης πραγματικότητας είναι η “ζωντανή” και διαδραστική προβολή ψηφιακού περιεχομένου στον πραγματικό κόσμο. Το περιεχόμενο αυτό παράγεται από υπολογιστή και μπορεί να ανήκει στην μορφή κειμένου, βίντεο, γραφικών και οποιοδήποτε άλλου ψηφιακού μορφοτύπου.

Το περιεχόμενο αυτό με την χρήση της τεχνολογίας, μπορεί να προβληθεί στο φυσικό περιβάλλον κάποιου χρήστη με την χρήση της κάμερας φορητής συσκευής του χρήστη. Χρησιμοποιείται σε διάφορους τομείς με γνωστά παραδείγματα να είναι ψυχαγωγικές εφαρμογές, όπως το Pokemon GO, αλλά και δημοφιλείς κοινωνικά μέσα, όπως είναι το Facebook και το Instagram. Στο Pokemon GO η τεχνολογία χρησιμοποιείται για να προβληθούν οι εχθροί στο φυσικό περιβάλλον, ενώ στα κοινωνικά μέσα για την εφαρμογή φίλτρων, δηλαδή ψηφιακού περιεχομένου στο πρόσωπο του χρήστη για ψυχαγωγικό σκοπό.

Η τεχνολογία μπορεί να βελτιώσει την εμπειρία ενός ατόμου ακόμα και σε απλά πράγματα, όπως είναι η προβολή αξιοθέατων, αφού δίνει την δυνατότητα στον χρήστη να εμφανίσει πληροφορίες για κάποιο μνημείο σε πραγματικό χρόνο από την κινητή του συσκευή.

Σε αυτή την ενότητα, θα γίνει αναφορά πάνω στην κατασκευή της AR τεχνολογίας, του πιθανού εξοπλισμού που θα χρειαστεί για την χρήση της, την σύνδεση της με την πιο γνωστή Εικονική Πραγματικότητα, και τις διαφορές με αυτήν.

3.1 Επισκόπηση

Επαυξημένη Πραγματικότητα ή Μικτή Πραγματικότητα είναι η τεχνολογία που επικαλύπτει εικόνες, κείμενο, ήχους και άλλο ψηφιακό περιεχόμενο στον φυσικό κόσμο γύρω μας. Η εμπειρία που προσφέρει είναι διαδραστική και βοηθάει στην “εμβύθιση” του παίκτη στην εφαρμογή. Η τεχνολογία αυτή αναφέρεται ως την ανάμειξη τριών κύριων χαρακτηριστικών, που είναι η έξυπνη τεχνολογία προβολής, η σωστή εγγραφή 3D περιεχομένου και η έξυπνη τεχνολογία διάδρασης [32]. Η έξυπνη τεχνολογία προβολής, αναφέρεται σε συσκευές, όπως το HMD και τις φορητές συσκευές, που χρησιμοποιούνται για την εμφάνιση του περιεχομένου AR στο περιβάλλον. Η σωστή εγγραφή 3D περιεχομένου, χρησιμοποιείται για την σωστή αναπαραγωγή ψηφιακού περιεχομένου στο φυσικό περιβάλλον.

Η τεχνολογία διάδρασης που χρησιμοποιείται, είναι υπεύθυνη για όλες τις αλληλεπιδράσεις που μπορούν να πραγματοποιηθούν, είτε αυτό αναφέρεται στην αλληλεπίδραση του χρήστη με το ψηφιακό αντικείμενο που έχουμε τοποθετήσει στην σκηνή, είτε στην επικοινωνία τεχνολογιών των συσκευών.

3.2 Εξοπλισμός

Η χρήση της τεχνολογίας AR είναι εύκολη, αφού δεν χρειάζεται κάποια ειδική συσκευή για την χρήση της. Παρόλο, που ο χρήστης μπορεί να χρησιμοποιήσει κάποιο HMD για αυτή την διαδικασία, η συσκευή αυτή δεν είναι απαραίτητη σε αντίθεση με την τεχνολογία VR που είναι κρίσιμη. Αυτό συμβαίνει, διότι για την προβολή AR υλικού, πέρα από την συσκευή HMD, μπορεί να χρησιμοποιηθεί ή κινητή συσκευή του ατόμου. Για αυτό ευθύνεται το γεγονός ότι είναι μερική ή μίξη του φυσικού κόσμου με το γραφικό περιεχόμενο. Οι συσκευές HMD που μπορούν να χρησιμοποιηθούν για το AR είναι γνωστές, όπως τα Google glasses, τα Microsoft Hololens κ.α. Με αυτές τις συσκευές, ο χρήστης μπορεί να παρατηρήσει τον χώρο που βρίσκεται με επιπρόσθετο ψηφιακό περιεχόμενο που μπορεί να χρησιμοποιηθεί για την άντληση πληροφοριών.

3.3 Προκλήσεις/Τεχνολογίες AR

Για την κατασκευή ενός AR παιχνιδιού [33], υπάρχουν ορισμένες προκλήσεις που πρέπει να αντιμετωπιστούν σχετικά με τον προσδιορισμό της θέσης του παίκτη, την ακρίβεια σε κλίμακα εκατοστών ή μέτρου, καθώς και εφαρμογή καταλλήλων τεχνολογιών επικοινωνίας μεταξύ παικτών και υπηρεσιών ή παικτών με άλλους παίκτες. Ένα παράδειγμα για την μεταφορά δεδομένων είναι κατά την μεταφορά κάποιων δεδομένων μετά την πρόοδο ενός παιχνιδιού, για να ειδοποιηθούν οι άλλοι παίκτες για την αλλαγή της κατάστασης του παιχνιδιού αν το παιχνίδι είναι διαδικτυακό. Για να πραγματοποιηθεί αυτό, απαιτείται σύνδεση δικτύου, και προτιμότερα γρήγορη, οπότε γίνεται χρήση δεδομένων (3G ή GRPS). Επιπλέον, υπάρχει το Bluetooth, WIFI και RFID για τοπικές μεταδόσεις.

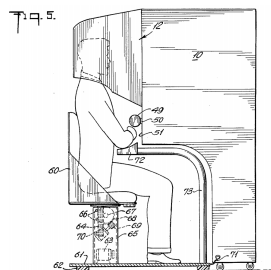
Επίσης, μία άλλη πρόκληση είναι η φορητότητα των συσκευών για την χρήση AR συσκευών. Οι συσκευές πρέπει να είναι ελαφριές, μικρές και φορητές, για να επιτραπεί στον χρήστη η ενασχόληση με AR παιχνίδια σε εξωτερικό χώρο. Αυτό, απαιτεί την χρήση συσκευών σε εξωτερικό χώρο που δεν περιορίζεται από την φορητότητα κάποιας συσκευής. Το πρόβλημα αυτό έχει σε ένα βαθμό αντιμετωπιστεί, αφού η τεχνολογία αυτή σήμερα μπορεί να εφαρμοστεί με την χρήση κινητών συσκευών, φορητών υπολογιστών, ειδικών AR headset, PDAs και tablets. Άλλη μία λιγότερο σημαντική πρόκληση είναι η επίδοση της συσκευής που θα εφαρμοστεί η τεχνολογία αυτή. Αυτό το πρόβλημα είναι λιγότερο σημαντικό, διότι οι συσκευές που ενσωματώνουν AR τεχνολογία δεν έχουν τόσο υψηλές απαιτήσεις όπως αυτές που ενσωματώνουν VR τεχνολογία. Οι VR συσκευές πρέπει να αντικαταστήσουν εξ ολοκλήρου το περιβάλλον με γραφικά που παράγονται από υπολογιστή. Οι AR συσκευές, σε αντίθεση με τις VR συσκευές δεν απαιτούν την ολοκληρωμένη αντικατάσταση του περιβάλλοντος του χρήστη. Αυτό, συμβαίνει γιατί στις AR συσκευές εμφανίζεται το ίδιο περιβάλλον με αυτό του χρήστη με την μόνη διαφορά, ότι δημιουργείται κάποιο αντικείμενο ή τροποποιείται κάποιο αντικείμενο του περιβάλλοντος με computer-generated περιεχόμενο.

Αντίθετα με τις υψηλές προδιαγραφές χαρακτηριστικών συσκευής που χρησιμοποιεί AR τεχνολογία, η συσκευή που χρησιμοποιεί AR τεχνολογία έχει υψηλές απαιτήσεις στον τομέα του tracking και sensing. Ο λόγος είναι το γεγονός ότι δεν αντικαταστίται εντελώς το περιβάλλον, και ως εξίσου η συσκευή πρέπει να εντοπίζει αντικείμενα στον χώρο και να ελέγχει την κίνηση της συσκευής.

3.4 Εικονική Πραγματικότητα

Πριν από την εύρεση της Επαυξημένης Πραγματικότητας (AR), υπήρχε μία άλλη καινοτόμα τεχνολογία, που ονομάζεται Εικονική Πραγματικότητα (VR). Η VR τεχνολογία αναφέρεται στην τεχνολογία αυτή που αντικαταστέι ολοκληρωτικά το πραγματικό περιβάλλον του χρήστη με ένα ψηφιακό. Αυτό, γίνεται με τη χρήση μίας συσκευής που φοριέται στην κεφαλή του χρήστη, ώστε να βλέπει ο χρήστης από αυτήν.

Μία από τις νωρίτερες προσπάθειες στις εικονικές μηχανές ξεκίνησε όταν ο σκηνοθέτης Morton Heilig, εφάρμοσε τις τεχνικές του και δημιούργησε μία μηχανή ονόματι Sensorama [34], έναν εξομοιωτή που διέγειρε τις αισθήσεις του ατόμου.



Σχήμα 3.1: **Sensorama Simulator [34]**

Η εμπειρία της μηχανής στον χρήστη ήταν απίστευτη, όπως περιέγραψε ο Howard Rheingold, που μίλησε για το άτομο όπου σε μία εμπειρία έκανε ποδήλατο (ή μοτοσυκλέτα σε άλλες εκδόσεις) στους δρόμους του Brooklyn.

Η μηχανή ήταν πολύ μπροστά για τον καιρό της αφού μπορούσε να απεικονίσει στερεοσκοπικές τρισδιάστατες εικόνες σε ευρυγώνια προβολή, να προσφέρει σχετική κλίση σώματος (για να είναι πιο ρεαλιστική η εμπειρία), να προσφέρει στερεοφωνικό ήχο και επίσης να επιτρέπει τον χρήστη να βιώνει τις οσμές και τα αρώματα της πόλης, όπου ενεργοποιούνταν κατά την διαδικασία λειτουργίας της μηχανής. Παρόλο που η εμπειρία και η λειτουργία της μηχανής ήταν ικανοποιητική, η μηχανή δεν κατάφερε να κερδίσει στην αγορά, διότι ήταν τεράστια σε μέγεθος και μπορούσε να σπάσει εύκολα. Ωστόσο, αυτό δεν την αποτρέπει από το να λάβει δικαιολογημένα μία θέση στις αξιολογούμενες καινοτόμες εφευρέσεις της εποχής της. Ο ίδιος δημιουργός αυτής της συσκευής, κατασκεύασε επίσης μία άλλη συσκευή, την συσκευή με την ονομασία Telesphere Mask [4]. Οι κατασκευές αυτές ήταν καλές προσπάθειες αλλά η τεχνολογία αυτή δεν είχε κάποια ιδιαίτερη εξέλιξη μέχρι το 1968. Ο Ivan Sutherland που θεωρείται ο πατέρας της Εικονικής Πραγματικότητας, το 1968 δημιούργησε το “Ξίφος του Δαμοκλή”, μια συσκευή που δημιούργησε με την βοήθεια των μαθητών του που αποτέλεσε εξέλιξη-πυλώνα στον τομέα της VR και AR τεχνολογίας.

3.4.1 Ομοιότητες

Η τεχνολογία αυτή έχει κάποιες ομοιότητες με την Επαυξημένη Πραγματικότητα. Οι ομοιότητες είναι:

1. Οι δύο τεχνολογίες αυτές επικαλύπτουν ψηφιακό περιεχόμενο όπως είναι οι ήχοι,βίντεο και εικόνες στην όψη του φυσικού περιβάλλοντος του χρήστη. Αυτή η επικάλυψη γίνεται είτε μερικώς είτε εξόλοκλήρου, δηλαδή υπάρχει πλήρη ανάμιξη ή μερική.
2. Οι δύο τεχνολογίες αυτές μπορούν να προβληθούν με την χρήση μίας συσκευής, που ονομάζεται HMD. Αυτή η συσκευή εξοπλίζεται στην κεφαλή του χρήστη, ώστε να μπορεί ο χρήστης να δει μέσα από αυτήν το ψηφιακό περιεχόμενο στο φυσικό χώρο και να μπορεί να αλληλεπιδράσει με το περιβάλλον.

3.4.2 Διαφορές

Ωστόσο, υπάρχουν και κάποιες διαφορές των δύο τεχνολογιών. Αυτές είναι:

1. Διαφέρουν στην χρήση τους, γιατί στην Εικονική Πραγματικότητα αντικαθίσταται πλήρως το περιβάλλον του χρήστη με ένα ψηφιακό, οπότε ο χρήστης δεν βλέπει το φυσικό περιβάλλον αλλά μόνο το ψηφιακό. Με αυτό τον τρόπο, είναι πιο ρεαλιστική η εμπειρία του χρήστη, δηλαδή ο χρήστης "μεταφέρεται" σε έναν φανταστικό κόσμο της επιλογής του. Αυτή η τεχνολογία συνήθως χρησιμοποιείται για την ψυχαγωγία του χρήστη. Αντίθετα, στην Επαυξημένη Πραγματικότητα προστίθεται η τροποποιείται μερικώς περιεχόμενο στον φυσικό χώρο. Για παράδειγμα, χρησιμοποιώντας την εφαρμογή Instagram, μπορεί ο χρήστης να δει πως είναι το πρόσωπο του σε πραγματικό χρόνο με κάτι ψηφιακό, όπως είναι γυαλιά ή μάσκα.
2. Η Επαυξημένη Πραγματικότητα, αν και ,μπορεί να προβληθεί με την βοήθεια συσκευής HMD, όπως η Εικονική Πραγματικότητα, μπορεί επίσης να προβληθεί και με την χρήση κινητής συσκευής. Συνεπώς, είναι πιο εύκολη η χρήση της τεχνολογίας αυτής επειδή δεν χρειάζεται κάποια ιδιαίτερη συσκευή για την χρήση της.
3. Η επαυξημένη πραγματικότητα χρησιμοποιείται και σε άλλους τομείς, πέρα από την ψυχαγωγία, όπως είναι για στρατιωτικούς σκοπούς, τουρισμό, ιατρικούς σκοπούς κ.α.

Κεφάλαιο 4

Εισαγωγή στην Unity

Οι μηχανές παιχνιδιών είναι τα περιβάλλοντα ανάπτυξης κώδικα που επιτρέπουν στους χρήστες, να αναπτύξουν το δικό τους παιχνίδι [35].

Είναι, ουσιαστικά, εφαρμογές που βοηθούν τον ενδιαφερόμενο χρήστη να κατασκευάσει το δικό τους παιχνίδι, διευκολύνοντας την διαδικασία ανάπτυξης αυτή. Αυτό συμβαίνει, διότι περιέχουν συνήθως έτοιμα μπλοκ κώδικα, κάποιες λειτουργίες και διάφορα εργαλεία που εξοπλίζουν τον χρήστη να κάνει αυτή την διαδικασία εύκολα και γρήγορα.

Για παράδειγμα, αν θέλει ένας χρήστης να δημιουργήσει ένα εκπαιδευτικό παιχνίδι, δεν θα χρειαστεί να ξεκινήσει από την γλώσσα προγραμματισμού να κάνει χειροκίνητα την διαδικασία ανάπτυξης παιχνιδιού, αφού του παρέχεται έτοιμο GUI, δηλαδή γραφικό περιβάλλον, που του παρέχονται όλες αυτές οι λειτουργίες από το κουτί.

Κάποια από τα βασικά συστατικά μίας τέτοιας εφαρμογής είναι ο renderer, η μηχανή φυσικής, η μηχανή ήχου, το κύριο πρόγραμμα και η τεχνητή νοημοσύνη.

Αρχικά, ο renderer βοηθάει στην απόδοση των γραφικών, είτε αυτά είναι 2D είτε 3D. Στη συνέχεια, υπάρχει η μηχανή φυσικής που συνεισφέρει φυσικά για την ρεαλιστική προσομοίωση των νόμων φυσικής μέσα στο περιβάλλον του παιχνιδιού. Μετέπειτα υπάρχει η μηχανή ήχου που χρησιμοποιείται για την επεξεργασία και αναπαραγωγή ήχων στο παιχνίδι. Τέλος, υπάρχει το κύριο πρόγραμμα και η τεχνητή νοημοσύνη, που αναφέρονται στις τεχνολογίες αυτές που χρησιμοποιούνται για την λογική του παιχνιδιού και την βελτιστοποίηση του κώδικα αντίστοιχα.

Σε αυτό το κεφάλαιο, θα αναφερθούμε συγκεκριμένα στην μηχανή Unity, και λειτουργίες που περιέχει αυτή η εφαρμογή.

4.1 Μηχανή Unity

Η Unity είναι μια παιχνιδομηχανή που υποστηρίζει πολλαπλές πλατφόρμες και δημιουργήθηκε από την Unity Technologies Inc..

Η εταιρεία αυτή που δημιούργησε την μηχανή Unity, έχει την έδρα της στην πόλη San Fransisco, αν και ξεκίνησε αρχικά στην Δανία το 2004 με την ονομασία Over the Edge Entertainment (OTEE) και τρία χρόνια αργότερα άλλαξε η ονομασία της στην σημερινή της μορφή.

Η εταιρεία αυτή είναι πιο γνωστή, για την δημιουργία της εφαρμογής Unity, μια μηχανή κατασκευής παιχνιδιών που χρησιμοποιείται για την κατασκευή παιχνιδιών και λοιπών εφαρμογών.

Η ανακοίνωση και η κυκλοφορία της μηχανής έγινε το 2005, όπου κυκλοφόρησε ως μια παιχνιδομηχανή που υποστήριζε αποκλειστικά το λειτουργικό σύστημα Mac OS X της εταιρείας Apple.

Ο στόχος του ήταν η αύξηση της προσβασιμότητας της κατασκευής παιχνιδιών σε περισσότερους δημιουργούς περιεχομένου. Είχε επιτυχία στο έτος κυκλοφορίας και κέρδισε και βραβείο δεύτερης θέσης στην καλύτερη χρήση γραφικών στην πλατφόρμα κυκλοφορίας του, δηλαδή στην Mac OS X. Μέχρι το 2018, η ομάδα ανάπτυξης έχει ήδη επεκτείνει την εμβέλεια της εφαρμογής για να υποστηρίξει περισσότερες από 25 πλατφόρμες.

Είναι μία από τις διασημότερες εφαρμογές δημιουργίας παιχνιδιών, διότι είναι πιο προσιτή για τους αρχάριους από τις άλλες μηχανές. Για παράδειγμα, ένας αρχάριος χρήστης αν θέλει να ξεκινήσει ένα νέο project συνήθως θα σκεφτεί αυτή την μηχανή πρώτα για να πετύχει τον στόχο του. Επιπλέον, είναι δωρεάν και σου δίνει την δυνατότητα ακόμη και να πουλήσεις το παιχνίδι σου σε γνωστά "μαγαζιά" αγοράς παιχνιδιών όπως είναι το Steam, App Store της Apple και το Google Play της ανοικτού κώδικα πλατφόρμας Android.

Αυτό φυσικά έρχεται με κάποιες προϋποθέσεις, διότι αν για παράδειγμα έχεις την δωρεάν έκδοση της εφαρμογής, τότε έχει περιορισμό στον αριθμό των χρημάτων που μπορείς να βγάλεις από αυτό, κάτι που σε παραπέμπει να αγοράσεις την έκδοση με συνδρομή. Εξαιτίας του γεγονότος ότι το Unity έχει τόσο μεγάλη εμβέλεια, υπάρχει μια γιγαντιαία κοινότητα που σε συνδυασμό με τους βοηθητικούς οδηγούς και τις συνδιασκέψεις της εταιρείας, μπορεί να συνεισφέρει τον χρήστη στην εκμάθηση της χρήσης της εφαρμογής, αποκλειστικά και μόνο μέσα από το ίδιο το Unity. Επιπροσθέτως, υπάρχει πάντα ένα Asset Store, που έχει διάφορα χρήσιμα εργαλεία και πακέτα, από 3D μοντέλα, πακέτα ήχων και σκηνικά αντικείμενα σε έτοιμα project που μπορεί να χρησιμοποιήσει ο χρήστης ανάλογα με τις ανάγκες του.

Η μηχανή αυτή είναι κατασκευασμένη με την χρήση C++, αλλά ο χρήστης μπορεί μόνο να κατασκευάσει παιχνίδια με κώδικα γραμμένο σε Java (Unityscript) και C# για την δημιουργία της εφαρμογής του.

Η εφαρμογή αυτή έχει μεγάλη επιτυχία στον τομέα της δημιουργίας παιχνιδιών, και συγκεκριμένα σε κινητές συσκευές, και για αυτό υπάρχει μεγάλη επένδυση σε αυτόν τον τομέα. Προφανώς, αυτό προέρχεται από το γεγονός ότι σήμερα οι κινητές συσκευές είναι μέρος της καθημερινής μας ζωής και είναι πιο προσιτές ως προς την χρήση τους. Είναι πάντα κοντά μας σε αντίθεση με άλλες πιο μεγάλες συσκευές όπως είναι οι επιτραπέζιοι υπολογιστές και έτσι υπάρχει πρόσβαση από οποιαδήποτε σημείο, είτε αυτό είναι το διάλειμμα κατά την άσκηση σε εργασιακό χώρο είτε κατά την χαλάρωση στην έξοδο σε εξωτερικούς χώρους.

Στην δημιουργία ενός νέου project από τον δημιουργό η εφαρμογή προσφέρει δύο επιλογές, όπου η μία έχει να κάνει με την δημιουργία ενός 2D παιχνιδιού και η άλλη με την δημιουργία ενός 3D παιχνιδιού. Φυσικά, οι επιλογές αυτές δεν έχουν μεγάλη διαφορά, με την έννοια ότι ακόμα και αν ο χρήστης επιλέξει την 3D επιλογή δεν αποτρέπει η εφαρμογή τον χρήστη από το να έχει την ελευθερία να δημιουργήσει μία 3D εφαρμογή. Ουσιαστικά, η διαφορά είναι στην όψη της κάμερας και δίνεται η επιλογή για την αλλαγή μεταξύ 2D και 3D προβολής μέσα από την προβολή της σκηνής.

Η 2D επιλογή ωστόσο δεν επιτρέπει τον χρήστη να αλλάξει σε 3D προβολή, κάτι που σημαίνει ότι ο χρήστης θα έχει κλειδωμένη την όψη της κάμερας στις 2 διαστάσεις. Παρόλο που η 2D τεχνολογία είναι πιο ηλικιωμένη σε σχέση με την 3D τεχνολογία, η μηχανή της εταιρείας Unity προσφέρει καλύτερη υποστήριξη κατά την χρήση της 3D τεχνολογίας. Αυτό συμβαίνει διότι η προσθήκη της 2D τεχνολογίας έγινε αργότερα, κάτι που είναι λογικό γιατί τα τελευταία χρόνια, η τεχνολογία αυτή, χρησιμοποιείται σε μικρότερα διαστήματα, σε σχέση με παλιά που ήταν το συνηθισμένο. Παρόλα αυτά, η μηχανή Unity είναι μία αρκετά καλή μηχανή για πολλούς δημιουργούς 2D παιχνιδιών, γιατί υποστηρίζει πολλαπλές πλατφόρμες και κάνει εύκολη την δημιουργία εκδόσεων παιχνιδιών για διάφορες κινητές συσκευές.

Ακόμα, αν και δεν υπάρχει πολύ μεγάλη ζήτηση αγοράς στα παιχνίδια που περιέχουν VR περιεχόμενο, υπάρχει ένα μικρό μερίδιο σε αυτό τον τομέα εξαιτίας της μηχανής αυτής. Η υποστήριξη σε νέες τεχνολογίες όπως VR και AR επιτρέπει την ανάπτυξη καινοτόμων παιχνιδιών και προσφέρει ένα πλήθος επιλογών για να καλλιεργήσει την φαντασία του ο δημιουργός και να προκαλέσει τα όριά του.

Η μηχανή υποστηρίζει πολλαπλές διαφορετικές γραφικές διεπαφές προγραμματισμού σε διάφορες πλατφόρμες. Υποστηρίζει παλιές τεχνολογίες όπως είναι το DirectX και το OpenGL, αλλά και νεότερες τεχνολογίες όπως είναι το Vulkan, το Metal αλλά και το Direct3D12. Σε αντίθεση με τα DirectX και OpenGL που κυκλοφορούν για πάνω από μια εικοσαετία, προσθήκες όπως οι χαμηλού επιπέδου γραφικές διεπαφές Metal, ύλκαν και Direct3D12 αποδεικνύουν ότι η εταιρεία προσαρμόζεται στις τελευταίες τεχνολογικές εξελίξεις και προσφέρει μία ελευθερία στον προγραμματιστή/κατασκευαστή παιχνιδιών. Η Vulkan τεχνολογία χρησιμοποιείται από τα συστήματα Linux, Windows και Android, η τεχνολογία Metal αποκλειστικά από Apple συσκευές με iOS και macOS λογισμικό και το Direct3D12 από Windows και Xbox ONE συσκευές.

Η Unity προσφέρει κάποιες υπηρεσίες στους κατασκευαστές παιχνιδιών που μπορούν να βελτιστοποιήσουν την εμπειρία τους κατά την χρήση της εφαρμογής.

Κάποια από αυτά είναι τα Unity Ads, Unity Analytics, Unity Collaborate, Unity Cloud Build κτλ. Αυτά είναι χρήσιμα εργαλεία γιατί διευκολύνουν τον χρήστη σε πολλούς τομείς όπως είναι η ευκολία στην εύρεση εκδόσεων ανάμεσα σε συνεργάτες με το Collaborate, η τοποθέτηση διαφημίσεων για ένα συμπληρωματικό εισόδημα για τον κατασκευαστή κ.α.

4.2 Πλεονεκτήματα

Υποστήριξη πολλαπλών πλατφορμών Δεν μπορούμε να μιλήσουμε για τα πλεονεκτήματα χρήσης της εφαρμογής αυτής, δίχως να αναφέρουμε ότι είναι μία από τις εφαρμογές που υποστηρίζει τον μεγαλύτερο αριθμό πλατφορμών.

Παρόλο που μπορεί να γίνει χρήση της, μόνο από τα λειτουργικά συστήματα Windows και OSX, η εφαρμογή επιτρέπει την έκδοση εφαρμογών σε πολλές διαφορετικές πλατφόρμες. Τα πιο γνωστά από αυτά είναι φυσικά τα Windows, Mac OS και Linux σε συστήματα υπολογιστών και τα Android και iOS στα συστήματα κινητών συσκευών. Μετά, υπάρχουν τα παιχνίδια που φτιάχνονται για την διαδικτυακή αναπαραγωγή τους που δημιουργούνται με το Unity Web Player. Ακόμα, προσφέρει την δημιουργία παιχνιδιών για μία ευρεία γκάμα κονσολών, και συγκεκριμένα τις κονσόλες Xbox, PS3, PS4, PS VR και Wii U. Φυσικά, παρέχεται η δυνατότητα κατασκευής παιχνιδιών σε αυτές τις πλατφόρμες, με την προϋπόθεση ότι ο χρήστης έχει λάβει άδεια από την αντίστοιχη εταιρεία κονσολών.

Επίσης, παρέχεται η δυνατότητα ανάπτυξης εφαρμογής σε hardware νέων τεχνολογιών, όπως είναι το Oculus Rift και το Google Cardboard. Αυτές οι τεχνολογίες ενεργοποιούν τον χρήστη, στην αξιοποίηση AR και VR περιεχομένου αντίστοιχα. Τέλος, παρέχεται ακόμα και η επιλογή, για την κατασκευή παιχνιδιών για έξυπνες τηλεοράσεις, στις Android TV και Samsung TV.

Γραφικά Τα γραφικά, αν και συνήθως δεν μετράνε τόσο πολύ, όσο το gameplay και το σενάριο σε ένα παιχνίδι, μπορούν να αλλάξουν σε σημαντικό βαθμό την άποψη ενός παίκτη για κάποιο παιχνίδι.

Αυτό, συμβαίνει διότι οι συνεχείς εξελίξεις της τεχνολογίας έχουν κάνει τον παίκτη, να συνηθίσει σε τίτλους με πολύ υψηλά γραφικά, οπότε είναι δύσκολο να έρθει σε επαφή με παλιότερα παιχνίδια. Για παράδειγμα, ένα άτομο που μπορεί παλιότερα να έπαιζε κάποια παιχνίδια και να του άρεσαν, μπορεί πλέον να παρατηρήσει ότι συνηθίζοντας στους καινούργιους τίτλους και γυρνώντας στα παλιά παιχνίδια, έχει λιγότερη διάθεση να ενασχοληθεί με αυτά, χωρίς αυτό να σημαίνει ότι εκείνα δεν είναι ενδιαφέροντα ή αξιότιμα στο τμήμα του gameplay/σεναρίου..

Τα γραφικά της συγκεκριμένης μηχανής αυτής είναι πολύ καλά, κάτι που κάνει το παιχνίδι να εμφανίζεται πιο όμορφο στον παίκτη. Αυτό, οδηγεί στην πιο αποτελεσματική βύθιση στον φανταστικό χώρο του παιχνιδιού από τον παίκτη.

Ευκολία χρήσης Η μηχανή αυτή κάνει εύκολη την διαδικασία κατασκευής παιχνιδιών και συμπεριλαμβάνει πολλούς οδηγούς και κατάστημα για την μεταφόρτωση αντικειμένων, μοντέλων κ.α. Η ευκολία στην χρήση της, την κάνει ιδανική για διάφορα άτομα, από ερασιτέχνες που ενδιαφέρονται στο αντικείμενο μέχρι και επαγγελματίες που ψάχνουν μια μηχανή που θα χρησιμοποιήσουν για να βγάλουν κέρδος.

Ομαδική Κατασκευή Η εφαρμογή παρέχει εργαλεία για την ομαδική δημιουργία παιχνιδιών και για το "άνέβασμα" των εκδόσεων στο διαδίκτυο. Με αυτό το τρόπο, αν συμπεριληφθούν στην ομάδα από τον χρήστη κάποια άτομα, μπορεί να δει τις εκδόσεις και να ελέγξει τις αλλαγές που έχουν πραγματοποιηθεί. Επίσης, του δίνεται η δυνατότητα να διαγράψει πιθανές αλλαγές που έχουν αρνητικό αντίκτυπο και να ανεβάσει την δική του έκδοση που πιθανότατα είναι γρηγορότερη.

4.3 Μειονεκτήματα

Γραφικά και εφέ Παρόλο που τα γραφικά του παιχνιδιού είναι αρκετά καλά, σε άλλες μηχανές δημιουργίας παιχνιδιών όπως η Unreal Game Engine, τα γραφικά είναι υψηλότερα. Αυτό έχει να κάνει, με τα προβλήματα φωτισμού, που έχει η Unity. Τα γραφικά, μπορούν να αυξηθούν χειροκίνητα, αλλά αυτό δεν συμβάλλει στην διευκόλυνση του χρήστη που υποστηρίζει ότι προσφέρει η εταιρεία.

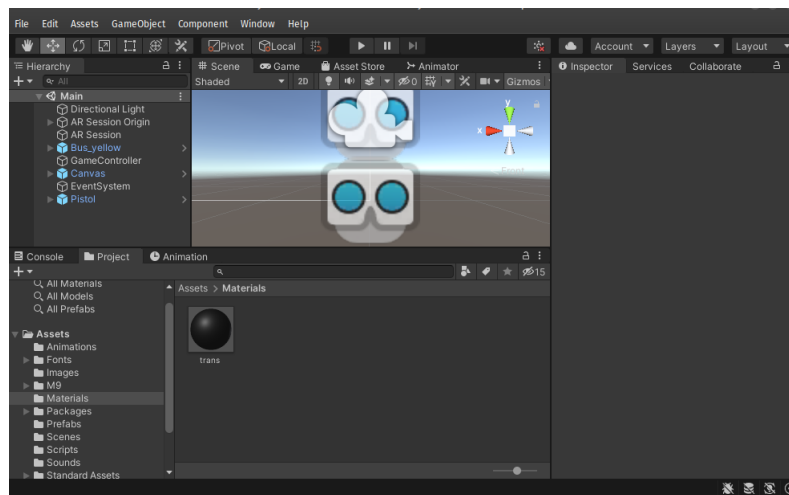
Συνδρομές Σε αντίθεση, με κάποιους ανταγωνιστές της, η μηχανή σύμφωνα με την δωρεάν άδεια χορήγησης της εφαρμογής, δεν επιτρέπει τις περισσότερες από 100.000 πωλήσεις ανά 12 μήνες στους δημιουργούς. Αυτό, σημαίνει ότι αν ο χρήστης θέλει να πουλήσει την εφαρμογή του σε κάποιο διαδικτυακό "μαγαζί", θα πρέπει να αγοράσει την συνδρομή υποχρεωτικά.

Οι συνδρομές αυτές είναι αρκετά ακριβές, που το κάνει αρκετά δύσκολη την πληρωμή τους, ειδικά για τους κατασκευαστές που εργάζονται αυτόνομα.

Επιπλέον, μόνο με την αγορά της συνδρομής θα έχει κάποιος χρήστης πρόσβαση σε προνόμια με γραφικές βελτιώσεις κ.α., κάτι που δυσκολεύει την εργασία του κατασκευαστή που το χρησιμοποιεί χωρίς να πραγματοποιήσει κάποια πληρωμή.

Οδηγοί Μερικοί από τους οδηγούς, που έχει δημιουργήσει η εταιρεία για την καθοδήγηση των νέων χρηστών, είναι είτε παλιοί ή δεν υπάρχουν καθόλου, κάτι που οδηγεί τους χρήστες να πρέπει να ψάξουν σε άλλες σελίδες για την επίλυση του προβλήματός τους.

4.4 Βασικά Χαρακτηριστικά



Σχήμα 4.1: **Βασικά χαρακτηριστικά της οπτικής διεπαφής του χρήστη**

Κάποια από τα βασικά χαρακτηριστικά της παιχνιδομηχανής είναι:

1. Μπάρα ρυθμίσεων στο πάνω μέρος της οθόνης για να είναι εύκολη η αναζήτηση των εργαλείων που μπορούν να βοηθήσουν τον χρήστη κατά την κατασκευή του παιχνιδιού του. Για παράδειγμα στην εικόνα είναι το File, Edit, Assets κτλ.
2. Hierarchy που είναι ουσιαστικά η ιεραρχία των σκηνικών αντικειμένων στην σκηνή και συνήθως βρίσκεται στο αριστερό πάνω μέρος της οθόνης. Όπως λέει και το όνομα, είναι μια ιεραρχία, δηλαδή μια σειριακή λίστα για να τοποθετήσουμε τα πράγματα που θα βρίσκονται μέσα στην σκηνή. Τα αντικείμενα που είναι πιο πάνω θα έχουν πιο μεγάλη βαρύτητα από αυτά που είναι πιο κάτω στην λίστα. Για παράδειγμα στην εικόνα το βλέπουμε, στο ARSession και AR Session Origin που είναι απαραίτητα για την κατασκευή ενός παιχνιδιού που εφαρμόζει τεχνολογία AR.
3. Inspector στο δεξί μέρος της οθόνης για την επιθεώρηση των αντικειμένων του παιχνιδιού. Ουσιαστικά δείχνει τα αντικείμενα με περισσότερες λεπτομέρειες και επιτρέπει τροποποιήσεις σε αυτά. Επίσης, δείχνει ποια άλλα αντικείμενα έχουμε ενώσει με το αντικείμενο αυτό.

Για παράδειγμα στον παίκτη έχουμε κάποιες ηχητικές πηγές για την ακρόαση κάποιων ηχητικών κομματιών, κάποια C# scripts για τον κώδικα γλώσσας C# που χρειάστηκε για την σωστή κίνηση του σώματος του παίκτη κτλ.

4. Scene στο μεσαίο κεντρικό κομμάτι της οθόνης για να μπορεί ο χρήστης να δει την σκηνή του παιχνιδιού. Είναι ένας άδειος χώρος, στον οποίο τοποθετούνται τα αντικείμενα (αν τοποθετήσεις αντικείμενα στον Inspector εμφανίζεται στο Scene και το αντίστροφο). Ένα παιχνίδι μπορεί να περιέχει πολλές σκηνές, δηλαδή πολλαπλούς τέτοιους χώρους, που συνήθως χρησιμοποιούνται για τις πίστες του παιχνιδιού, τα πολλαπλά μενού που θα χρησιμοποιηθούν κτλ.
5. Game οθόνη που είναι η δεύτερη καρτέλα στο μεσαίο κεντρικό κομμάτι της οπτικής διεπαφής για την δοκιμασία του παιχνιδιού χωρίς την κατοχή της απαραίτητης πλατφόρμας ή την δημιουργία εκτελέσιμου αρχείου. Για παράδειγμα αν έχεις το λειτουργικό σύστημα Windows δεν χρειάζεται να περιμένεις για την δημιουργία ενός εκτελέσιμου για κάθε φορά που θα τρέξεις την εφαρμογή. Επίσης, αν έχεις Windows μπορείς να φτιάξεις παιχνίδια για iOS πλατφόρμα χωρίς να έχεις κάποια Apple συσκευή.
6. Λοιπά χαρακτηριστικά, όπως είναι το Console στο κάτω αριστερό μέρος της οθόνης, που εμφανίζει μηνύματα στον παίκτη και σφάλματα που αντιμετωπίστηκαν κατά το τρέξιμο του παιχνιδιού. Κάποια άλλα είναι το Animation, στο κάτω αριστερό μέρος της οθόνης (τρίτη καρτέλα) για την δημιουργία κάποιων κινούμενων σχεδίων. Ένα σχετικό εργαλείο είναι το Animator που βρίσκεται στο κεντρικό πάνω μέρος της οθόνης, όπου μπορείς να τοποθετήσεις αντικείμενα όπως κινήσεις. Για παράδειγμα, οι κινήσεις των ζόμπι τοποθετούνται σε εκείνο το μέρος και με την βοήθεια κάποιων μεταβλητών και του σχετικού C# script, είναι εύκολη η αλλαγή μεταξύ αυτών.

Κεφάλαιο 5

Πριν την κατασκευή

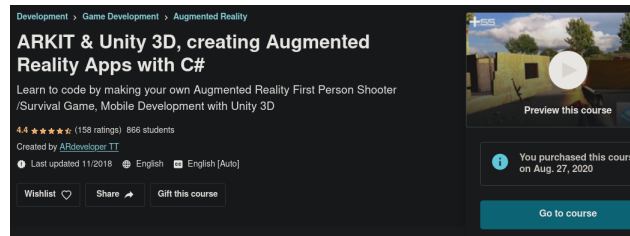
5.1 Αρχική σύλληψη ιδέας

Στην αρχή της κατασκευής του παιχνιδιού έγιναν δοκιμές διαφορετικών πλατφορμών κατασκευής παιχνιδιών, όπου και δοκιμάστηκαν οι δημοφιλέστερες δηλαδή η Unity και η Unreal Engine. Παρόλο που η Unreal Engine προσφέρει καλύτερη ποιότητα γραφικών και απαιτείται (σχετικά) λιγότερος προγραμματισμός, έχει πιο μεγάλη απαίτηση στα χαρακτηριστικά του υπολογιστή και συγκριτικά χειρότερη AR υποστήριξη, κάτι που οδήγησε στην χρήση της Unity μηχανής. Ένα πολύ μεγάλο θετικό σημείο ήταν το γεγονός ότι προμηθεύεται μεγαλύτερη βοήθεια (tutorial, assets) για τον καθημερινό χρήστη σε σχέση με την Unreal που δεν βοηθάει τόσο τον αρχάριο, αλλά τον πιο έμπειρο προγραμματιστή.

Μετά την απόφαση σχετικά με την μηχανή κατασκευής, έγινε η σύλληψη ιδεών, δηλαδή η εύρεση για το τι παιχνίδι θα κατασκευαστεί και πως θα εφαρμοστεί η τεχνολογία AR μέσα σε αυτό. Για την εξακρίβωση αυτού, έγινε αναζήτηση πιθανών βοηθητικών οδηγιών (tutorial) που εφαρμόζουν την τεχνολογία που απαιτείται για την ολοκλήρωση της πτυχιακής εργασίας. Δυστυχώς δεν υπάρχουν πολλοί διαθέσιμοι βοηθητικοί οδηγοί μέσα στην διασημότερη πλατφόρμα προβολής βίντεο Youtube, οπότε έπρεπε να γίνει αναζήτηση και σε άλλες αντισυμβατικές πλατφόρμες όπως είναι το Udemy και το Unity Asset Store. Έπειτα από αναζήτηση σε διάφορους ιστιότοπους έγινε προσπάθεια δημιουργίας παιχνιδιού όμοιου με το δημοφιλές παιχνίδι Pokemon GO, κάτι που δεν μπόρεσε τελικά να ολοκληρωθεί. Η εφαρμογή τεχνολογίας GPS που απαιτείται σε τέτοια παιχνίδια απαιτούσε, τεχνολογία όπως το MapBox (ανοικτού κώδικα πλατφόρμα για δημιουργία χάρτη) ή plugin από το asset store της Unity (GPS χάρτες για παιχνίδια). Αυτά δεν ήταν χρήσιμα για τον σκοπό μας, λόγω της δυσκολίας χρήσης του MapBox και την χρέωση στο GPS plugin του asset store και για αυτό απορρίφθηκαν. Εξαιτίας αυτού, έγινε απόφαση για την αλλαγή της κατεύθυνσης μας και πιο αποτελεσματική αναζήτηση σε Youtube, Udemy και Unity Asset Store, κάτι που έφερε αποτελέσματα μετά την αγορά και δοκιμή κάποιων μαθημάτων στην διαδικτυακή πλατφόρμα εκμάθησης Udemy. Δοκιμάστηκε, επίσης η αγορά ενός πακέτου από την Unity, αλλά χρησιμοποιούσε δική του κάμερα, και δεν ήταν βολικό στην λειτουργία του.

5.2 Μάθημα Udemy

Το μάθημα αυτό είχε να κάνει με ένα First-Person Shooter (FPS) παιχνίδι που ενσωματώνει την τεχνολογία επαυξημένης πραγματικότητας και είναι εύκολο στην δημιουργία του. Χρησιμοποιεί την παιχνιδομηχανή Unity για την δημιουργία του παιχνιδιού και συγκεκριμένα το πακέτο ARKit που απευθύνεται σε χρήστες που σκοπεύουν να δημιουργήσουν παιχνίδια για iOS (Apple) συσκευές.



Σχήμα 5.1: **ARKit μάθημα Udemy για την κατασκευή παιχνιδιού για iOS συσκευές**

Το μάθημα δημιουργήθηκε από τον χρήστη ARDeveloperTT και περιέχει 2.5 ώρες εκπαιδευτικού περιεχομένου για την δημιουργία ενός παιχνιδιού με την χρήση της γλώσσας C#. Αρχικά, έχει μια εισαγωγή 8 λεπτών που αποτελείται από 2 μαθήματα και στην συνέχεια μια εκμάθηση σχετικά με την χρήση της γλώσσας C# στην εφαρμογή Unity που αποτελείται από άλλα 6 μαθήματα. Τέλος, έχει 24 μαθήματα για την διδασκαλία σχετικά με την κατασκευή του που συνολικά βγαίνουν περίπου στις 2 ώρες.

Ο δημιουργός του μαθήματος βοηθάει τον χρήστη να έρθει σε επαφή με τον κόσμο της Unity, και να μάθει να εφαρμόζει την τεχνολογία επαυξημένης πραγματικότητας. Μέσα από το περιεχόμενό του δείχνει πως μπορείς να χρησιμοποιήσεις κώδικα και κάποια σκηνικά αντικείμενα για την κατασκευή ενός ολοκληρωμένου παιχνιδιού.

Επίσης, επισυνάπτει κάποια αρχεία που μπορούν να χρησιμοποιηθούν από τον χρήστη, για την ολοκλήρωση της διαδικασίας αυτής.

5.3 Ζωντανοί νεκροί

Οι ζωντανοί νεκροί, αλλιώς γνωστοί και ως τα "Zombie", είναι πολλοί γνωστοί σε όλους μας, αφού υπάρχουν πολλές ταινίες, βιβλία και άλλα μέσα που τα περιγράφουν. Είναι μυθολογικές αθάνατες φυσικές οντότητες που δημιουργούνται από την "ανάσταση" ενός νεκρού ατόμου. Πολλοί γνωρίζουν για αυτά αλλά δεν ξέρουν από που προέρχεται ο μύθος τους.

Ο μύθος αυτών των οντοτήτων προέρχεται από την λαϊκή κουλτούρα της Αϊτής [36]. Τον 17ο και 18ο αιώνα όταν η χώρα ήταν γνωστή ως "Σαιν-Ντομένγκ" και κυβερνούσε η Γαλλία, είχαν πολλούς σκλάβους που δούλευαν στις φυτείες ζάχαρης. Οι σκλάβοι αυτοί προέρχονταν από την Αφρική και ζούσαν σε τραγικές συνθήκες. Οι μισοί σκλάβοι από αυτούς δεν άντεξαν πάνω από μερικά χρόνια, ένα γεγονός που έκανε πιο σημαντική την εισαγωγή και αγορά άλλων σκλάβων για να συμπληρωθεί ο απαιτούμενος φόρτος εργασίας.

Τα τελευταία χρόνια, με την επιρροή της Αμερικανικής κουλτούρας, ο μύθος αυτός έχει τροποποιηθεί και έχει γίνει αυτό που γνωρίζουμε σήμερα, δηλαδή ότι είναι πλάσματα που τρελαίνονται για την κατανάλωση της σάρκας των ανθρώπων.

Ωστόσο, η αληθινή ιστορία είχε τις οντότητές αυτές, να είναι πλάσματα που τρελαίνονται για την κατανάλωση της δικής τους σάρκας. Φυσικά, αυτή η απεικόνιση των τεράτων αυτών, προέρχεται από την απάνθρωπη συμπεριφορά που έφεραν οι Γάλλοι απέναντί τους, για ένα αρκετά μεγάλο διάστημα.

Οι σκλάβοι εκείνοι θεωρούσαν ότι μόνο ο θάνατος θα τους απελευθέρωνε από την μίζερη ζωή που ζούσαν, αφού θα τους έστελνε στην Αφρική όπου θα ήταν επιτέλους ελεύθεροι. Η αυτοκτονία, ήταν κοινή εκείνον τον καιρό λόγω των συνθηκών, αν και πίστευαν ότι όσοι αυτοκτόνησαν δεν θα είχαν την επιλογή να γυρίσουν στην πατρίδα τους, αλλά θα έμεναν για πάντα να περιτριγυρίζουν τις φυτείες. Θα ήταν, δηλαδή μίζερη, η μετά θάνατον ζωή τους, αφού θα ήταν καταραμένοι να ζήσουν στα ίδια σώματα που απαρνήθηκαν για μια αιωνιότητα.

Μετά την επανάσταση από τους σκλάβους, ο μύθος των ζωντανών νεκρών άλλαξε. Ο μύθος πλέον είχε τις οντότητές αυτές ως κομμάτι της θρησκείας Voodoo, όπου οι ζωντανοί νεκροί ήταν νεκρά σώματα που αναστήθηκαν από σαμάνους και Voodoo ιερείς. Βουντού [37] ονομάζεται η θρησκεία που υπάρχει σε μέρη αφρικανικής κληρονομιάς και αποτελεί μία μορφή μαγικής λατρείας. Πολλοί που ακολουθούν την θρησκεία αυτή, πιστεύουν ότι οι ζωντανοί νεκροί είναι μύθοι, αλλά μερικοί επίσης πιστεύουν ότι αναστήθηκαν με χρήση μαγείας (βουντού) όπως προαναφέραμε. Το άτομο που χρησιμοποίησε Βουντού για την ανάσταση του νεκρού, το αποκαλούσαν bokor [38].

Οι Bokor χρησιμοποιούσαν βότανα, κόκκαλα, μέρη ζώων κ.α. για να δημιουργήσουν σκευάσματα που προκαλούσαν συμπτώματα όμοια με αυτά των ζωντανών νεκρών, όπως είναι η δυσκολία κινήσεων ενός ατόμου, η ψυχική σύγχυση και τα αναπνευστικά προβλήματα.

Παρόλο που οι περισσότεροι πιστεύουν ότι τα πλάσματα αυτά αποτελούν μύθο υπάρχουν κάποιοι που κατέγραψαν μερικά αληθινά τέτοια συμβάντα. Στα συμβάντα, οι νεκροί θάβονταν, διότι κρίνονταν νεκροί κλινικά, αλλά μετά ξαφνικά αποδεικνύονταν ότι ήταν ζωντανοί. Αυτό, πιστεύεται ότι προκαλείται από κάτι τέτοια σκευάσματα που καταναλώνονταν σε μεγάλες ποσότητες και προκαλούσαν προσωρινή παράλυση στα άτομα.

Κεφάλαιο 6

Κατασκευή παιχνιδιού

Σε αυτό το κεφάλαιο αναφέρουμε τα πακέτα, τον κώδικα και γενικότερα την διαδικασία κατασκευής του παιχνιδιού.

6.1 Απαιτήσεις

Για την κατασκευή του παιχνιδιού απαιτήθηκαν κάποια πακέτα του Unity για την εφαρμογή της επαυξημένης τεχνολογίας στο παιχνίδι. Επίσης, επειδή το παιχνίδι δημιουργήθηκε σύμφωνα με το μάθημα Udemyl που είχε να κάνει με παιχνίδι για Apple συσκευές, έπρεπε να τροποποιηθεί καταλλήλως για να επιτρέπεται το τρέξιμο της εφαρμογής σε Android συσκευές που είναι δημοφιλέστερες. Για την επίτευξη αυτού, χρειάστηκαν τα πακέτα ARFoundation, ARKit και ARCore.

Για την κατασκευή ενός Android παιχνιδιού απαιτείται η χρήση του πακέτου ARCore [39] και για την κατασκευή ενός παιχνιδιού για Apple συσκευές απαιτείται η χρήση του πακέτου ARKit [40].

Το ARCore πακέτο είναι μία συλλογή ανάπτυξης λογισμικού (SDK) και κυκλοφόρησε από την εταιρεία Google το 2018 [41]. Αυτό το πακέτο δίνει την δυνατότητα στον χρήστη, να δημιουργήσει εφαρμογές επαυξημένης πραγματικότητας. Χρησιμοποιεί τρεις κύριες τεχνολογίες για να επιτρέπει στον χρήστη την ενσωμάτωση του πραγματικού κόσμου με γραφικό περιεχόμενο μέσα από την κάμερα του κινητού του χρήστη. Αυτά συμπεριλαμβάνουν:

1. Για την κίνηση του χρήστη, την ελευθερία κίνησης ενός άκαμπτου σώματος μέσα σε ένα τρισδιάστατο χώρο (6 βαθμοί ελευθερίας).
2. Την κατανόηση του περιβάλλοντος του χρήστη, έτσι ώστε να βρίσκει την θέση οριζόντιων αντικειμένων, όπως ένα πλακάκι ή ένα γραφείο.
3. Κατανόηση των συνθηκών φωτισμού του χρήστη.

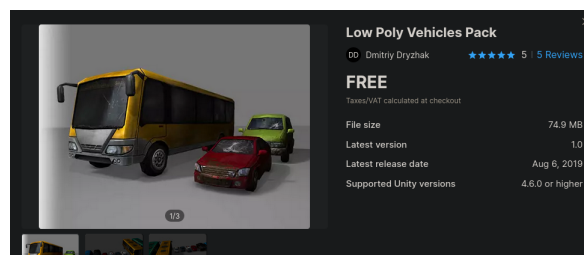
Το ARFoundation πακέτο είναι το πακέτο που ενώνει τα δύο συμπληρωματικά πακέτα ARCore και ARKit για να δημιουργηθούν παιχνίδια που υποστηρίζονται από αυτές τις δύο πλατφόρμες. Πέρα από αυτά τα δύο πακέτα, επίσης, προσφέρει την δυνατότητα να δημιουργήσει περιεχόμενο για τις συσκευές Magic Leap και HoloLens.

Φυσικά, αυτό γίνεται με την προϋπόθεση ότι ο χρήστης έχει εγκαταστήσει τα αντίστοιχα πακέτα, το MagicLeap XR Plugin για το Magic Leap και το Windows XR Plugin για τα HoloLens που τα εισάγει μέσα από τον διαχειριστή πακέτων της μηχανής Unity.

6.2 Assets

Το παιχνίδι ουσιαστικά τοποθετεί τον χρήστη σε ένα μετά-αποκαλυπτικό περιβάλλον, το οποίο δημιουργείται ενώνοντας το πραγματικό περιβάλλον του χρήστη μέσα από την κάμερα με περιεχόμενο που έχει παραχθεί από υπολογιστή. Ο παίκτης πρέπει να σηκώσει το όπλο του και να αντιμετωπίσει ένα κύμα εχθρών, που αποτελείται από τους ζωντανούς νεκρούς.

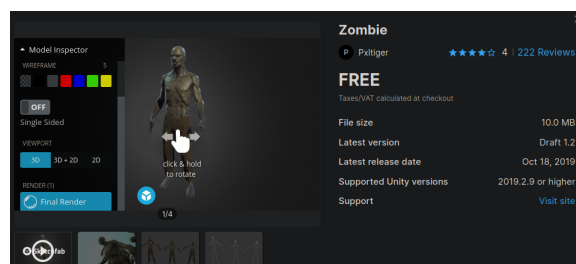
Μετά από την εγκατάστασή των απαραίτητων πακέτων έπρεπε να γίνει η εγκατάσταση των σκηνικών αντικειμένων για να τοποθετηθούν αντικείμενα όπως ζόμπι, ήχοι κτλ. στην πίστα. Αρχικά, κάποια σκηνικά αντικείμενα (assets) που χρησιμοποιήθηκαν για την κατασκευή είναι τα Low Poly Vehicles Pack, Blood Sfx Pack και Zombie Pack. Το Low Poly Vehicles Pack είναι ένα πακέτο που συμπεριλαμβάνει σπασμένα αυτοκίνητα, κάτι το οποίο βοήθησε στο να εμπλουτίσει τον κόσμο του παιχνιδιού, διότι συμμορφώνεται με το μετά-αποκαλυπτικό πρότυπο που έχουμε θέσει στο παιχνίδι.



Σχήμα 6.1: **Low Poly Vehicle Pack**, μία μετά-αποκαλυπτική συλλογή αυτοκινήτων

Η συλλογή αυτή περιλαμβάνει ένα λεωφορείο, ένα σεντάν, ένα μικρό λεωφορείο και ένα αυτοκίνητο αστυνομίας.

Το Zombie pack συμπεριλαμβάνει τους εχθρούς του παιχνιδιού, τους ζωντανούς νεκρούς μαζί με τις κινήσεις τους όπως είναι το αργό περπάτημα, η επίθεση, το περπάτημα μπρος τα πίσω κ.α. Το πακέτο αυτό αναφέρθηκε στο Udemy μάθημα.



Σχήμα 6.2: **Zombie pack**, το πακέτο με τους εχθρούς του παιχνιδιού

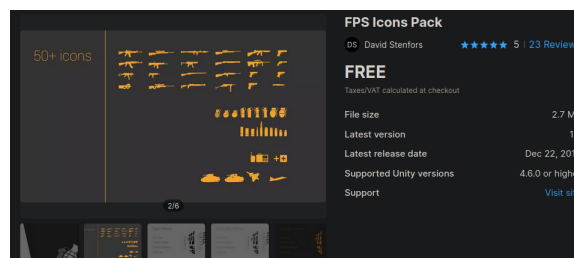
Από αυτές τις κινήσεις χρησιμοποιήσαμε 3, το αργό περπάτημα κατά την διαδικασία εντοπισμού του παίκτη, το τρέξιμο κατά την εύρεση του εχθρού και της επίθεσης κατά την εισαγωγή του ζόμπι μέσα στην κοντινή εμβέλεια με τον παίκτη.

Το Blood SFX πακέτο συμπεριλαμβάνει τους ήχους από τα αίματα που ακούγονται κατά την επιτυχής επίθεση ενός ζόμπι στον παίκτη.



Σχήμα 6.3: **Blood SFX** πακέτο για τα ηχητικά εφέ που ακούγονται κατά την επιτυχής επίθεση

Πολύ σημαντικό είναι και το πακέτο με τα εικονίδια που εμφανίζονται στο UI, δηλαδή στην οπτική διεπαφή του χρήστη.



Σχήμα 6.4: **FPS Icon** πακέτο, που συμπεριλαμβάνει τα εικονίδια του παιχνιδιού

Κάποια παραδείγματα είναι η μπάρα ζωής του παίκτη, το πόσες σφαίρες διαθέτει και έχει την δυνατότητα να καταναλώσει ο παίκτης κτλ. Αυτά είναι απαραίτητα για την σωστή λειτουργία της εφαρμογής. Επιπλέον, σημαντικό είναι να αναφέρουμε το γεγονός ότι είναι τα ίδια που χρησιμοποιεί και ο δημιουργός του μαθήματος Udemy. Έγινε αναζήτηση για άλλα εικονίδια αλλά δεν βρέθηκε κάποια καλύτερη επιλογή για το συγκεκριμένο είδος παιχνιδιού. Ένα ακόμα βασικό μοντέλο που χρησιμοποιήθηκε κατά την κατασκευή του παιχνιδιού είναι το μοντέλο του όπλου του χαρακτήρα παίκτη στο παιχνίδι. Το μοντέλο αυτό είναι ένα όπλο που συμπεριλαμβάνει το χέρι που το κρατάει για περισσότερο ρεαλισμό στο παιχνίδι.

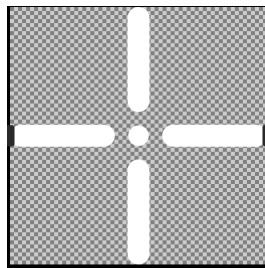
Το μοντέλο του όπλου παρέχεται από τον κατασκευαστή του μαθήματος Udemy και μέσα από την ιστοσελίδα του μαθήματος δίνεται η δυνατότητα για την μεταφόρτωση (download) του αρχείου.



Σχήμα 6.5: **WeaponPack μοντέλο όπλου M1911**

Το περιεχόμενο του συμπιεσμένου αρχείου που παρέχεται από το μάθημα, περιέχει πέρα από το μοντέλο του όπλου (με το χέρι) και το εφέ μετά από τον πυροβολισμό του. Το εφέ αυτό είναι ο φωτισμός του στομίου κάννης (μικρή έκρηξη) κατά την αποδέσμευση της σφαίρας από το όπλο. Αυτό συμβαίνει όταν ο χρήστης πατήσει το κουμπί που βρίσκεται στο κάτω δεξί μέρος της οθόνης.

Τώρα, στον τομέα της διεπαφής του χρήστη χρησιμοποιήθηκαν κάποιες εικόνες για το στόχαστρο, το εφέ για το αίμα μετά την επιτυχή επίθεση του αντιπάλου και το κουμπί για την αποδέσμευση της σφαίρας. Το στόχαστρο χρησιμοποιείται για την στόχευση του αντιπάλου και βρίσκεται στο κεντρικό μέρος της οθόνης ανεξαρτήτως της κίνησης της κάμερας χρήστη.



Σχήμα 6.6: **Στόχαστρο για την ακριβή στόχευση στους αντιπάλους**

Η οθόνη αίματος ενεργοποιείται όταν ο χρήστης βρίσκεται στην ζώνη επίθεσης του αντιπάλου και απενεργοποιείται αυτόματα μετά από ένα πολύ μικρό χρονικό διάστημα (1-2 δευτερόλεπτα).



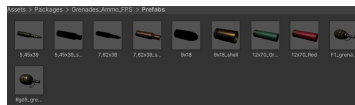
Σχήμα 6.7: **Οθόνη αίματος**

Το κουμπί για την αποδέσμευση της σφαίρας είναι μια απλή εικόνα και βρίσκεται στο κάτω δεξιά σημείο της οθόνης και είναι ουσιαστικά ένας μαύρος-γκρι κύκλος με την αγγλική λέξη "Shoot" μέσα σε αυτό.



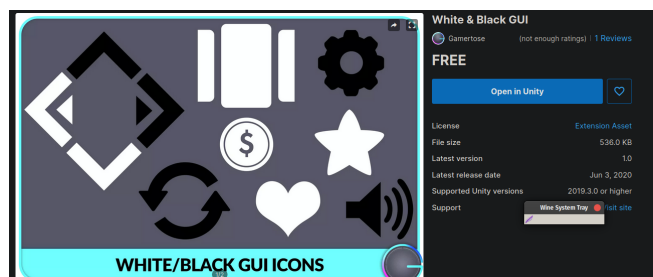
Σχήμα 6.8: **Κουμπί πυροβολισμού της σφαίρας**

Για την σφαίρα χρησιμοποιήθηκε ένα μοντέλο από το Grenade Ammo FPS πακέτο που περιέχει πολλαπλά μοντέλα για σφαίρες και χειροβομβίδες.



Σχήμα 6.9: **Grenade Ammo FPS πακέτο με σφαίρες και χειροβομβίδες**

Ένα άλλο συμπληρωματικό πακέτο που χρησιμοποιήθηκε κατά την κατασκευή του παιχνιδιού είναι το πακέτο White Black GUI. Το πακέτο αυτό περιέχει κάποια εικονίδια που μπορούν να χρησιμοποιηθούν στην γραφική διεπαφή του χρήστη. Στην συγκεκριμένη περίπτωση, δηλαδή στους σκοπούς της εφαρμογής μας, εφαρμόζουμε μερικά εικονίδια από το πακέτο στην δημιουργία της λειτουργίας παύσης του παιχνιδιού.



Σχήμα 6.10: **White & Black GUI** πακέτο με εικονίδια που χρησιμοποιούνται στην γραφική διεπαφή

6.3 Μηχανικές - Mechanics

Το παιχνίδι αυτό έχει κάποιες βασικές μηχανικές όπως όλα τα παιχνίδια. Οι μηχανικές είναι ουσιαστικά όλοι οι κανόνες και τα διαδικασίες για την σωστή λειτουργία ενός παιχνιδιού. Όλα αυτά δείχνουν πως το παιχνίδι θα δουλεύει για τα άτομα που το παίζουν. Αποτελούν τα θεμέλια για ένα παιχνίδι, διότι θέτουν κανόνες για την σωστή λειτουργία του παιχνιδιού.

Οι μηχανικές μπορεί να είναι το σύστημα πόντων που έχουμε ενσωματώσει στο παιχνίδι, οι πίστες που έχουμε προσθέσει, η συμπεριφορά των εχθρών κ.ο.κ.

Στο δικό μας παιχνίδι κάποιες συμπεριφορές που έχουμε θέσει είναι:

1. Όταν σηκώσει το όπλο ο παίκτης θα ενεργοποιηθεί η δυνατότητα να πυροβολήσει για να αντιμετωπίσει τους εχθρούς του
2. Για να πυροβολήσει τους εχθρούς του ο παίκτης θα πρέπει να πατήσει το κουμπί στο δεξί μέρος της οθόνης
3. Για να προχωρήσει στο επόμενο κύμα/στην επόμενη πίστα πρέπει να νικήσει όλους τους εχθρούς

6.4 Scripts

Οι διαδικασίες για την πραγματοποίηση κάποιων μηχανικών στο παιχνίδι μπορούν να ολοκληρωθούν με την χρήση κάποιου script. Τα script είναι γραμμένα σε κάποια προγραμματιστική γλώσσα και έχουν έτοιμα και ολοκληρωμένα προγραμματιστικά μπλοκ για την διευκόλυνση του χρήστη. Τα έτοιμα προγραμματιστικά μπλοκ περιέχουν προετοιμασμένες εντολές έτσι ώστε ο χρήστης να μην χρειάζεται να επαναλαμβάνεται μετά την δημιουργία ενός καινούριου script και φυσικά για να διευκολύνουν τον χρήστη. Ένα παράδειγμα script μπορούμε να δούμε παρακάτω στην φωτογραφία.

```

Assets > Scripts > GameController.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.SceneManagement;
6
7  public class GameController : MonoBehaviour
8  {
9      public GameObject bloodyScreen;
10     public Text healthText;
11     public int health;
12
13     // Start is called before the first frame update
14     void Start()
15     {
16         health = 100;
17     }
18
19     // Update is called once per frame
20     void Update()
21     {
22         if (health <= 0)
23         {
24             SceneManager.LoadScene("GameOver");
25         }
26     }
27 }

```

Σχήμα 6.11: Παράδειγμα προγράμματος - script από το παιχνίδι

Όπως παρατηρείται στο πάνω μέρος του προγράμματος - script φαίνονται όλες οι βιβλιοθήκες που εισήχθησαν για την λειτουργία του προγράμματος. Μερικά είναι βασικά για όλα τα προγράμματα και άλλα είναι μεταβαλλόμενα ανάλογα με την περίπτωση. Για παράδειγμα αν θες να διαχειριστείς την διεπαφή του χρήστη μέσα από το πράγματα πρέπει να εισάγεις την βιβλιοθήκη "UnityEngine.UI", αλλιώς θα εμφανίζονται σφάλματα στην κονσόλα του παιχνιδιού και πιθανώς να μην μπορεί καν να γίνει εκκίνηση της εφαρμογής. Η αποτυχία εκκίνησης προκαλείται επειδή δίχως την προσθήκη της βιβλιοθήκης δεν μπορεί να αναγνωρίσει τα στοιχεία που ανήκουν στην διεπαφή του χρήστη όπως είναι κείμενα, εικόνες, κουμπιά κτλ. Επίσης, με την προσθήκη της βιβλιοθήκης "UnityEngine.SceneManagement" μπορούμε να διαχειριστούμε τις σκηνές του παιχνιδιού. Οι σκηνές είναι σημαντικές διότι μπορούν να χρησιμοποιηθούν για την μετάβαση σε άλλη πίστα, μετάβαση σε μενού κ.α. Στη συγκεκριμένη περίπτωση χρησιμοποιείται για την μετάβαση στην σκηνή αποτυχίας ολοκλήρωσης του παιχνιδιού δηλαδή στην Gameover σκηνή. Αυτό είναι πολύ γνωστό γιατί υπάρχει σε πολλά όμοια παιχνίδια όπου ο παίκτης χάνει όλες τις ζωές του και συνεπώς αποτυγχάνει στην επιτυχημένη ολοκλήρωση του παιχνιδιού.

Παρόμοια με το κείμενο "public class GameController : MonoBehaviour" υπάρχουν σε όλα τα προγράμματα της εφαρμογής Unity. Αυτό σημαίνει ότι έχουμε μια δημόσια κλάση που ονομάζεται GameController, που πάντα θα έχει το ίδιο όνομα με το όνομα του script χωρίς την επέκταση του (.cs), που κληρονομεί από το MonoBehaviour. Το MonoBehaviour είναι το script της Unity όπου κάθε νέο script κληρονομεί κάποιες ιδιότητες από αυτό. Σε κάποιες περιπτώσεις μπορούμε να τοποθετήσουμε κάποιο άλλο δικό μας script σε εκείνο το πεδίο όπως για παράδειγμα στους εχθρούς. Θα μπορούσαμε να έχουμε για παράδειγμα ένα έτοιμο script με όλες τις τιμές (πόντοι ζημιάς, πόντοι ζωής κτλ.) για κάποιους εχθρούς και να θέλουμε να χρησιμοποιήσουμε το ίδιο για κάποιους άλλους εχθρούς με αλλαγή στις τιμές αυτές και προσθήκη κάποιων άλλων ειδικών χαρακτηριστικών. Τα χαρακτηριστικά αυτά μπορούν να είναι κάποιες συναρτήσεις, τιμές, μέθοδοι κ.α. Για να υλοποιηθεί αυτό, απλά θέτουμε ένα script να κληρονομήσει κάποιες ιδιότητες από ένα άλλο script, το οποίο το έχουμε ορίσει ώστε να μην κληρονομεί από το προκαθορισμένο κώδικα της Unity, MonoBehaviour. Για παράδειγμα, αν θέλουμε να πάρουμε μια μέθοδο για τον πυροβολισμό των εχθρών, δεν υπάρχει κανένας λόγος σε νέο script όπου δεν απαιτείται κάποια αλλαγή σε αυτή την μέθοδο, να το γράψουμε όλο από την αρχή.

Οπότε, υποθέτοντας ότι έχουμε ένα script που είναι ο γονέας και ονομάζεται Σκριπ1 και ένα παιδί Script2, τότε απλά δηλώνουμε το Script2 ως `public class Script2 : Script1` για να λάβουμε όλες τις ιδιότητες από το πρόγραμμα Script1, που οδηγεί στην διευκόλυνση του χρήστη.

Στη συνέχεια, όπως βλέπουμε στην εικόνα 18 έχουμε το πεδίο για την δήλωση μεταβλητών. Ένα παράδειγμα αυτού αποτελεί το κείμενο `public GameObject bloodyScreen`. Το `public`, στο κείμενο αυτό, δηλώνει ότι η μέθοδος αυτή είναι προσβάσιμη από άλλες κλάσεις πέρα από την τωρινή, δηλαδή την κλάση που γίνεται μέσα σε αυτήν η δήλωση της μεταβλητής. Αυτό σου δίνει την δυνατότητα να αλλάξεις την τιμή έξω από το script, από το πεδίο στον Inspector της Unity. Έτσι ανά πάσα στιγμή μπορούμε να πειραματιστούμε με τις τιμές αυτές για να βελτιστοποιήσουμε το παιχνίδι. Φυσικά, πολλές φορές δεν χρειάζεται αυτό και μπορούμε να το δηλώσουμε ως `Private`, ώστε μόνο η κλάση που δηλώθηκε η μέθοδος μέσα σε αυτήν θα έχει πρόσβαση σε αυτήν. Παρακάτω, στην εικόνα παρατηρούμε το `void Start` και το `void Update`. Αυτά είναι βασικές συναρτήσεις της μηχανής Unity και υπάρχουν σε όλα τα Script γιατί πραγματοποιούν κάποιες συγκεκριμένες λειτουργίες. Μέσα στην συνάρτηση `void Start` τοποθετούνται όλες οι μέθοδοι που θέλουμε να τρέχουν μετά την αρχική εκτέλεση του προγράμματος και μέσα στο `void Update` τοποθετούνται όλες οι μέθοδοι που θέλουμε να εκτελούνται ανά καρέ του παιχνιδιού. Τα παιχνίδια συνήθως τρέχουν στα 30 ή στα 60 fps, δηλαδή στα 30 ή στα 60 καρέ ανά δευτερόλεπτο.

Συνοπτικά, *mechanics* στα παιχνίδια θεωρούνται πολλές τακτικές ή συμπεριφορές του χρήστη, όπως το *sprinting*, δυνάμεις, αναπλήρωση πόντων ζωής κτλ. Τα *mechanics* στο παιχνίδι που έχουμε δημιουργήσει είναι πολύ απλά, εξαιτίας της κατηγορίας που ανήκει το παιχνίδι. Το παιχνίδι ναί μεν είναι FPS που είναι μία ευρεία κατηγορία, αλλά αντίθετα με τα περισσότερα τέτοια παιχνίδια χρησιμοποιεί το *background* από την κάμερα ως το περιβάλλον και αναπαράγονται νέοι εχθροί μέχρι ένα σημείο, δηλαδή μέχρι να φτάσει στο όριο των εχθρών. Αυτό σημαίνει ότι το παιχνίδι ανήκει στην ψευδο-survival κατηγορία από την φύση του, κάτι που αναφέρεται και παρακάτω στην αναφορά για τις πιθανές μελλοντικές επεκτάσεις. Ουσιαστικά σε αυτό τον τομέα θα μιλήσουμε σχετικά με το πως το παιχνίδι λειτουργεί.

Εχθροί

Οι εχθροί είναι οι ζωντανοί εχθροί όπως έχουμε αναφέρει πολλές φορές προηγουμένως. Σε αυτό τον τομέα θα μιλήσουμε για τα *mechanics* των εχθρών του παιχνιδιού, δηλαδή για την συμπεριφορά τους στην εφαρμογή.

Κίνηση Σε αυτό το πεδίο θα μιλήσουμε για την κίνηση των εχθρών και για το πως αντιδράνε στον "πρωταγωνιστή", δηλαδή την κάμερα. Βασικά, η κίνηση των εχθρών είναι πολύ βασική και δεν απαιτεί κάποια ιδιαίτερη τροποποίηση. Για παράδειγμα θα μπορούσαμε να αυξάνουμε ταχύτητα στους εχθρούς μετά τον εντοπισμό του αντιπάλου τους. Ωστόσο, αυτό θα αυξάνει το επίπεδο δυσκολίας στο παιχνίδι και δεν ταιριάζει πολύ με τον ορισμό ενός ζόμπι. Το ζόμπι υποτίθεται είναι ένας άμυαλος οργανισμός που έχει μολυνθεί από έναν ιό και είναι ουσιαστικά ζωντανός νεκρός. Εμείς τον αφήσαμε απλά να ψάχνει τον αντίπαλο και μετά να έρχεται σταδιακά στον εχθρό.

```
// Update is called once per frame
void Update()
{
    transform.Translate(Vector3.forward * Time.deltaTime * 0.3f);
    transform.LookAt(Camera.main.transform.position);
    transform.eulerAngles = new Vector3(10, transform.eulerAngles.y, 0);
}
```

Σχήμα 6.12: Εντολή για την κίνηση

Είναι μία πολύ απλή εντολή που βρίσκεται μέσα στην συνάρτηση Update, κάτι που πραγματοποιείται σε πολλά άλλα παιχνίδια. Η χρήση εντολών στην συνάρτηση Update γίνεται συνήθως για ελέγχους, διότι η συνάρτηση Update εκτελείται ανά καρέ του παιχνιδιού. Στη συγκεκριμένη περίπτωση είναι πολύ απλές οι εντολές που βρίσκονται μέσα σε αυτή την συνάρτηση όπως μπορούμε να παρατηρήσουμε στην παραπάνω εικόνα. Αυτές οι εντολές χρησιμοποιούν κάποιες βασικές βιβλιοθήκες της Unity.

Το "transform.Translate" είναι πολύ βασικό και μπορεί να ορίσει την κίνηση του παίκτη σε κάποιες συντεταγμένες και σε μία κατεύθυνση που καθορίζεται από τον δημιουργό. Τα transform χρησιμοποιούνται στην Unity για την διαχείριση ενός τρισδιάστατου (3D) χώρου στους άξονες X, Y και Z ή σε ένα δισδιάστατου (2D) χώρο στους άξονες X, Y. Οι άξονες X, Y και Z αντιπροσωπεύουν φυσικά τον οριζόντιο άξονα, τον κάθετο άξονα και τον άξονα του βάθους. Δηλαδή, στον X θα ορίζονται οι τιμές για την οριζόντια κίνηση στο έδαφος του παιχνιδιού, το Y για την κίνηση κάθετα σε ένα ύψος και για την κίνηση σε βάθος που συνήθως είναι ο άξονας που ζητάμε όταν θέλουμε να αλλάξουμε την κίνηση ενός αντικείμενου σε ένα παιχνίδι.

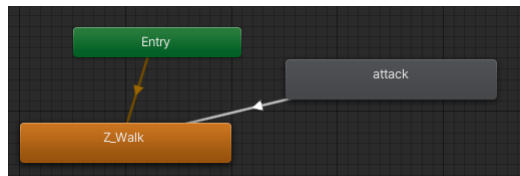
Στην συγκεκριμένη περίπτωση είναι ορισμένο σε ένα πολύ απλό "Vector3.forward * Time.deltaTime * 0.3f". Το "Vector3.forward" είναι μία στενογραφία για να αναφέρουμε τις συντεταγμένες (0, 0, 1). Αυτό σημαίνει ότι δεν θα υπάρχει κάποια αλλαγή στην κίνηση στους X και Y άξονες αλλά θα υπάρχει μία σταθερή κίνηση στον άξονα Y, δηλαδή στο βάθος. Το "Time.deltaTime" είναι ο χρόνος που πέρασε από την ολοκλήρωση του προηγούμενου καρέ σε ένα παιχνίδι. Χρησιμοποιείται όταν θέλουμε κάτι να κινείται σε μία σταθερή ταχύτητα σε ένα παιχνίδι. Για παράδειγμα, στις ταινίες, που παίζουν κάποια σει εικόνων ανά καρέ, όταν ορίζεται ο ρυθμός των καρέ ανά δευτερόλεπτο πχ. 30, αυτό είναι σταθερό. Στα παιχνίδια που δημιουργούνται με την μηχανή Unity πρέπει να χρησιμοποιούμε την εντολή "Time.deltaTime" για να ορίσουμε έναν σταθερό ρυθμό. Το "0.3f" στο τέλος της εντολής αναφέρεται στην ταχύτητα που δίνουμε εμείς στα ζόμπι. Θέλουμε να είναι αρκετά αργά οπότε τοποθετούμε μία ταχύτητα 0.3 που είναι float τύπου.

Το "transform.lookAt" περιστρέφει ένα αντικείμενο για να κοιτάξει σε ένα συγκεκριμένο μέρος που το ορίζουμε εμείς. Στην συγκεκριμένη περίπτωση, το αντικείμενο κοιτάει στο "Camera.main.transform.position" που είναι η θέση της κάμερας, δηλαδή ο πρωταγωνιστής στο παιχνίδι μας.

Τέλος, έχουμε την εντολή "transform.eulerAngles" που χρησιμοποιείται σε πολλά παιχνίδια για την ρύθμιση της περιστροφής. Ο σπουδαίος μαθηματικός Euler σύστησε για πρώτη φορά τις γωνίες Euler, μία ονομασία που προέρχεται από το όνομα του δημιουργού, για την εξέταση της κίνησης περιστροφής σε ένα rigid body (έχουμε στο παιχνίδι) τρισδιάστατου χώρου. Οι γωνίες αυτές είναι απλές και απαιτούν απλά τον ορισμό τριών δεκαδικών (float) τιμών. Ο Unity editor χρησιμοποιεί διανύσματα και γωνίες Euler για να αντιπροσωπεύει την διεύθυνση και κατεύθυνση ενός αντικειμένου.

Στην συγκεκριμένη περίπτωση οι τρεις τιμές των γωνιών Euler παίρνουν τις τιμές 0, `transform.eulerAngles.y` και 0, που σημαίνει ότι η περιστροφή στον άξονα X θα είναι 10, το ύψος, δηλαδή το Y, θα είναι το ίδιο το ύψος στην περιστροφή του αντικειμένου και το Z θα είναι 0.

Όπως παρατηρούμε από το παραπάνω κείμενο, πραγματοποιούνται αλλαγές στην κατεύθυνση και διεύθυνση του αντικειμένου και όχι στην εκτέλεση animation για την αλλαγή της κίνησης του εχθρού. Τα animations των εχθρών, δηλαδή η ομαλή κίνηση με κινήσεις των χαρακτήρων πραγματοποιούνται από τον animator των εχθρών και αλλάζουν μέσα από το script. Στην εικόνα μπορεί να παρατηρηθεί το σχεδιάγραμμα του animator εχθρών μέσα από την εφαρμογή της Unity.



Σχήμα 6.13: **Διάγραμμα animator του zombie χαρακτήρα**

Ο animator της μηχανής Unity είναι η διεπαφή για την διαχείριση του συστήματος Mecanim κινουμένων σχεδίων. Το Mecanim Animation System είναι το σύστημα της Υνιτφ για την εύκολη διαχείριση όλων των αρχείων κινήσεων ενός μοντέλου/χαρακτήρα. Επιτρέπει τον χρήστη να ελέγχει τις κινήσεις ενός χαρακτήρα, να προβάλλει τις κινήσεις και να μπορεί να ελέγχει την αλληλεπίδραση μεταξύ αυτών. Μέσα από αυτό μπορούμε εύκολα να ορίσουμε την ταχύτητα εκτέλεσης ενός animation, να ορίζουμε το πότε θα παίξει κάθε κίνηση και να ελέγχουμε την σχέση μεταξύ κάποιων κινήσεων, όπως για παράδειγμα την εκτέλεση ενός animation μετά τον τερματισμό ενός προηγούμενου animation του μοντέλου. Στην συγκεκριμένη περίπτωση, έχουμε 2 animation και αλλάζουμε μεταξύ τους αυτόματα μέσα από ένα script. Το Z_Walk animation εκτελείται στην κανονική του ταχύτητα απεριόριστα, με την μόνη εξαίρεση όταν "συγκρούεται" ο εχθρός με την κάμερα.

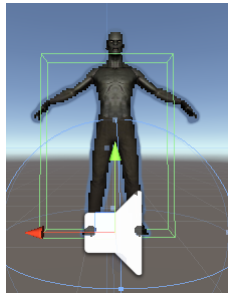
Η σύγκρουση στα παιχνίδια της μηχανής πραγματοποιείται με την ενσωμάτωση collider εργαλείων στα μοντέλα και με την ανίχνευση αυτών μέσα από τα script. Οι πληροφορίες σύγκρουσης μεταβιβάζονται στα συμβάντα `OnCollisionEnter`, `OnCollisionExit` και `OnCollisionStay`. Αρχικά, το `OnCollisionEnter`, όπως λέει και η λέξη, έχει να κάνει με την αρχή της σύγκρουσης μεταξύ κάποιων αντικειμένων, δηλαδή ενεργοποιείται όταν ξεκινάει η σύγκρουση. Το `OnCollisionExit` έχει να κάνει με την λήξη της σύγκρουσης μεταξύ κάποιων αντικειμένων, δηλαδή ενεργοποιείται με το που χωριστούν τα δύο αντικείμενα. Τέλος, έχουμε το `OnCollisionStay` που είναι λιγότερο δημοφιλές και χρησιμοποιείται σε σχετικά πιο σπάνιες περιπτώσεις όταν θέλουμε να συγκρατούμαι πληροφορίες για την συνεχή σύγκρουση κάποιων αντικειμένων. Σε αντίθεση με τα άλλα δύο, δεν λαμβάνει ως είσοδο έναν collider αλλά μία κλάση `Collision`.

Πέρα από αυτά τα συμβάντα αυτά, υπάρχει και το συμβάν `OnTriggerEnter`. Αντίθετα με την μέθοδο `OnCollisionEnter` που ενεργοποιείται μετά την σύγκρουση μεταξύ colliders, το `OnTriggerEnter` εκτελείται όταν ένας collider εισάγεται σε έναν trigger. Ο trigger μπορεί να είναι μία τύπου boolean μεταβλητή που αλλάζει από false σε true την στιγμή σύγκρουσης.

Για την αλλαγή από το Z_Walk animation στο attack animation, έχουμε μία τιμή boolean που αλλάζει τιμή μέσα από έναν OnCollisionEnter και έναν OnCollisionExit αναλόγως. Θα μιλήσουμε πιο αναλυτικά για αυτά στην συνέχεια.

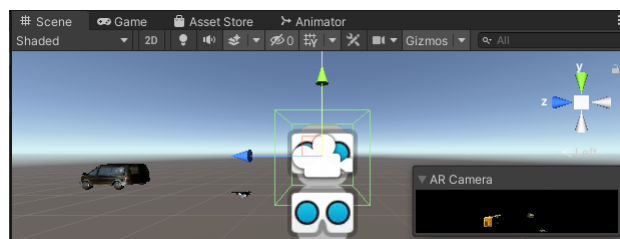
Τα animation αυτά προέρχονται από το πακέτο με τους ζωντανούς νεκρούς που εντοπίσαμε στο μαγαζί της Unity και χρησιμοποιήσαμε για τους σκοπούς της εφαρμογής.

Αλλαγή animation Όπως αναφέραμε προηγουμένως, η κίνηση γίνεται με κάποιες εντολές και με την εκτέλεση κάποιων animation. Η κάμερα και το zombie έχουν collider, εξαρτήματα που καθορίζουν το σχήμα αντικειμένου για σκοπούς φυσικών συγκρούσεων. Η Unity προσφέρει πολλαπλά σχήματα collider και επιτρέπει την αλλαγή του μεγέθους για να ταιριάζει στα μέτρα του μοντέλου.



Σχήμα 6.14: Μοντέλο ζωντανού νεκρού

Στην παραπάνω εικόνα μπορούμε να παρατηρήσουμε το μοντέλο του ζόμπι στο παιχνίδι. Οι πράσινες γραμμές στην εικόνα υποδεικνύουν τον collider του prefab. Όπως βλέπουμε οι γραμμές σχηματίζουν το σχήμα ενός κουτιού που δείχνει ότι χρησιμοποιήσαμε τον Box Collider της μηχανής Unity. Στο μοντέλο, παρατηρούμε επίσης ότι στο κάτω μέρος έχει ένα σχήμα μεγαφώνου, ένα σημάδι ότι προσθέσαμε ακουστική πηγή στο αντικείμενο για την αναπαραγωγή κάποιων ήχων. Η κάμερα έχει τον δικό της εξάρτημα collider για να μπορούμε να ανιχνεύουμε συγκρούσεις μεταξύ του αντικειμένου του εχθρού και της κεντρικής κάμερας της εφαρμογής.



Σχήμα 6.15: Εικόνα κάμερας στην σκηνή

Στην εικόνα αυτή, παρατηρούμε όπως και στο μοντέλο του ζωντανού νεκρού, ότι η κάμερα έχει έναν Box Collider που είναι αρκετά πιο μεγάλος από το μέγεθος της κάμερας και δεν ταιριάζει ακριβώς στις διαστάσεις της. Αυτό συμβαίνει, διότι αντίθετα με τα κανονικά παιχνίδια δεν έχουμε μοντέλο για τον παίκτη και λειτουργεί η κάμερα ως ο χαρακτήρας μας.

Στην περίπτωση όπου ορίζουμε τον collider στις διαστάσεις της κάμερας, ο χρήστης δεν θα μπορούσε να δει καλά τους αντιπάλους, γιατί θα ήταν πολύ κοντά του. Ουσιαστικά, το αντικείμενο θα φαινόταν ότι θα εισάγεται μέσα στην κάμερα οπότε αυξήθηκε σε μέγεθος το εξάρτημα collider για να διορθώνεται αυτό το σφάλμα. Η εικόνα του headset, δηλαδή το "ρομποτάκι" στο κέντρο, υπάρχει γιατί ενσωματώνουμε AR τεχνολογία στο παιχνίδι. Η κάμερα, όπως αναλύσαμε προηγουμένως, προέρχεται από τις βιβλιοθήκες για την ενσωμάτωση AR τεχνολογίας και φαίνεται έτσι, διότι χρησιμοποιήσαμε το AR Session Origin που περιέχει την κάμερα σε αυτό για τους σκοπούς επαυξημένης τεχνολογίας. Στην εικόνα βλέπουμε ότι στο κέντρο έχει μπροστά από το εικονίδιο του headset και ένα σχήμα κάμερας. Αυτό είναι το βασικό σχήμα που έχουν όλες οι κάμερες της Unity.

```
// Update is called once per frame
void Update()
{
    timer += Time.deltaTime;

    if (zombieIsThere && timer >= timeBetweenAttack)
    {
        Attack();
    }
}

void OnCollisionEnter(Collision col)
{
    if (col.gameObject.tag == "MainCamera")
    {
        Debug.Log("Enter");
        zombieIsThere = true;
    }
}

void OnCollisionExit(Collision col)
{
    if (col.gameObject.tag == "MainCamera")
    {
        Debug.Log("Exit");
        zombieIsThere = false;
    }
}
```

Σχήμα 6.16: Εντολές για την διαχείριση των συγκρούσεων

Οι εντολές στην εικόνα δείχνουν την συνάρτηση Update και τα συμβάντα OnCollisionEnter και OnCollisionExit.

Στην Update έχουμε μία μεταβλητή τύπου float που ονομάζεται timer που αυξάνεται συνεχώς για κάθε καρέ του παιχνιδιού. Επίσης, έχουμε την boolean πιο κάτω που ονομάζεται zombieIsThere, που αλλάζει κατά την εκτέλεση και την λήξη συγκρούσεων. Εκτός από αυτές τις δύο έχουμε την timeBetweenAttack που είναι ακέραια τιμή και ρυθμισμένη στην τιμή 2 για την αλλαγή μεταξύ των animation. Η if ελέγχει αν η τιμή zombieIsThere είναι true και αν η τιμή του timer είναι μεγαλύτερη της timeBetweenAttack. Η πρώτη συνθήκη ελέγχει αν το ζόμπι έχει φτάσει στην εμβέλεια του παίκτη, δηλαδή αν συγκρούεται αυτό με την κάμερα. Η δεύτερη συνθήκη ελέγχει αν η τιμή του timer είναι μεγαλύτερη από την τιμή του χρονικού διαστήματος επίθεσης που έχουμε ορίσει. Ουσιαστικά, η δεύτερη περίπτωση ορίζει ότι το animation επίθεσης θα παίζει μόνο για 2 δευτερόλεπτα και όχι παραπάνω. Στην περίπτωση που ισχύουν ταυτόχρονα και οι δύο αυτές συνθήκες τότε εκτελείται η συνάρτηση Attack. Στην attack η τιμή του timer ορίζεται στην τιμή 0, διότι χρησιμοποιούμε την τιμή αυτή στον timer για να ορίσουμε τον χρόνο αναπαραγωγής του animation στα 2 δευτερόλεπτα, δηλαδή στην τιμή του timeBetweenAttack. Το συμβάν OnCollisionEnter λαμβάνει την προκαθορισμένη τιμή για τον collider ως είσοδο και αλλάζει την τιμή της boolean από false σε true. Χρησιμοποιεί την if για να κάνει έλεγχο για το αν η ετικέτα του αντικειμένου που συγκρούστηκε με τον ζωντανό νεκρό έχει την ετικέτα με την ονομασία "MainCamera".

Αυτή η ετικέτα είναι η προκαθορισμένη στην κεντρική κάμερα της μηχανής. Ωστόσο, εμείς έχουμε διαγράψει την κεντρική κάμερα της μηχανής από την σκηνή και την έχουμε αντικαταστήσει με την AR Camera, οπότε αλλάξαμε την ετικέτα της AR κάμερας στην "MainCamera" για αυτήν την διαδικασία. Στο OnCollisionExit τρέξαμε την ίδια διαδικασία με την μόνη αλλαγή ότι αλλάξαμε την τιμή της zombieIsThere από false σε true.

Τέλος, για την αναπαραγωγή των animation, όπως είπαμε και στην Update, χρησιμοποιείται η Attack () συνάρτηση. Χρησιμοποιείται εκεί η εντολή "GetComponent<Animator>().Play("attack")". Η εντολή "GetComponent" χρησιμοποιείται όταν θέλουμε να επιστρέψουμε ένα εξάρτημα που είναι ενωμένο με το αντικείμενο κάποιου τύπου. Στην συγκεκριμένη περίπτωση, λαμβάνουμε το εξάρτημα τύπου Animator και επιλέγουμε την λειτουργία αναπαραγωγής για την εκτέλεση του animation με την ονομασία attack. Μετά από 2 δευτερόλεπτα αυτό απενεργοποιείται και γυρνάει πίσω στο animation που περπατάει με το Z_Walk, αν ο παίκτης δεν συγκρούεται πλέον με τον ζωντανό νεκρό.

```
void Attack()
{
    Debug.Log("Arrived!");
    timer = 0f;
    GetComponent<Animator>().Play("attack");
    gameController.zombieAttack(zombieIsThere);
    attackSound.Play();
}
```

Σχήμα 6.17: Εντολές συνάρτησης/μεθόδου Attack

Όπως βλέπουμε στην εικόνα 24 εκτελούνται οι εντολές που αναφέραμε προηγουμένως και μετά στέλνουμε την τιμή της zombieIsThere στο script με την ονομασία GameController. Τέλος, αναπαράγεται ο ήχος attackSound που είναι ο ήχος επίθεσης. Το Debug.Log που δεν αναφέραμε εσκεμμένα προηγουμένως, δεν είναι ιδιαίτερα σημαντικό και χρησιμοποιείται για την εκτύπωση μηνυμάτων στην κονσόλα της Unity. Για παράδειγμα, όποτε εκτελείται η μέθοδος Attack θα εκτυπώνεται το μήνυμα "Arrived" στην κονσόλα της μηχανής. Επιπλέον, στις OnCollisionEnter και OnCollisionExit θα εκτυπώνονται τα μηνύματα Enter και Exit αντίστοιχα.

Επίθεση Αρχικά, για την επίθεση όπως αναλύσαμε προηγουμένως, εκτελούνται τα animation. Στη συνέχεια, στέλνεται η τιμή της τύπου boolean μεταβλητής στο script GameController. Εκεί, θέλουμε να ενεργοποιήσουμε την εικόνα από τα αίματα που έχουμε τοποθετήσει στην οπτική διεπαφή του χρήστη και μετά από λίγο να την απενεργοποιήσουμε ξανά. Επιπλέον, πρέπει να λάβουμε υπόψιν και την ζημιά που προκαλεί η επίθεση αυτή στον παίκτη και κάπως να την αφαιρέσουμε. Μετά από αυτή την αφαίρεση πόντων ζωής πρέπει κάπως να αλλάξουμε το κείμενο που υπάρχει στην οπτική διεπαφή του χρήστη έτσι ώστε να μπορεί ο χρήστης να ελέγχει τους πόντους ζωής του. Για να πραγματοποιηθούν αυτά απλά τα λαμβάνουμε σαν είσοδο τα game object στο script GameController. Τα game object αναφέρονται ως όλα τα αντικείμενα του παιχνιδιού. Η είσοδος αυτών των αντικειμένων στο script μπορεί να πραγματοποιηθεί με την "public GameObject obj1". Για το παράδειγμα, τοποθετήσαμε obj1 ως το όνομα του αντικειμένου. Επειδή το αντικείμενο είναι δημόσιο, με την εκτέλεση αυτής της εντολής στον Inspector της Unity, θα ανοίξει ένα πεδίο για την τοποθέτηση του αντικειμένου. Μετά μπορούμε να το τοποθετήσουμε μέσα εκεί σέρνοντάς το prefab στο πεδίο αυτό. Εμείς στο script μας λαμβάνουμε ως είσοδο την οθόνη αίματος από το UI.

Επίσης, πρέπει μέσα στο script πρέπει κάπως να ελέγχουμε για το αν η ζωή έχει πέσει σε τιμή κάτω των 100 για να του εμφανίσουμε την οθόνη τερματισμού "GameOver" στην οπτική διεπαφή του χρήστη.

```
public void zombieAttack(bool zombiesThere)
{
    Debug.Log("BloodyScreen");
    bloodyScreen.SetActive(true);
    StartCoroutine(Wait2Secs());
    health -= 5;

    string stringHealth = (health).ToString();
    healthText.text = "" + stringHealth;
}
```

Σχήμα 6.18: Εντολές συνάρτησης/μεθόδου **zombieAttack**

Παραπάνω μπορούμε να παρατηρήσουμε πως υλοποιούνται τα προαναφερόμενα με τις εντολές της γλώσσας προγραμματισμού C#. Σε αυτή την μέθοδο, λαμβάνουμε ως είσοδο την boolean `zombiesThere` που είδαμε στο προηγούμενο script. Στη συνέχεια, εκτυπώνουμε ένα μήνυμα με την εντολή `Debug.Log` για επιβεβαίωση ότι πραγματοποιήθηκαν επιτυχώς. Μετά, ενεργοποιούμε το αντικείμενο με την εντολή `"bloodyScreen.SetActive(true)"`. Επειδή λαμβάνουμε το ίδιο το `gameObject` ως είσοδο δεν απαιτείται να γράψουμε την εντολή `"bloodyScreen.gameObject.SetActive(true)"` αν και θα δούλευε κανονικά και με εκείνην. Μετά ενεργοποιούμε την Coroutine με την εντολή `StartCoroutine(Wait2Secs())`. Οι `yield` είναι συναρτήσεις που μπορούν να κάνουν παύση και συνέχιση της λειτουργίας τους, σε αντίθεση με τις κανονικές συναρτήσεις που πρέπει να τρέχουν σειριακά χωρίς να υπάρχει η δυνατότητα επεξεργασίας της λειτουργίας ανάγνωσής της. Εμείς έχουμε δημιουργήσει την κορουτίνα προηγουμένως. Η `yield` μας ονομάζεται `Wait2Secs()` και ορίζεται με την μέθοδο `IEnumerator Wait2Secs()`. Χρησιμοποιείται διότι θέλουμε να απενεργοποιηθεί μετά από 2 δευτερόλεπτα η οθόνη αίματος. Αυτό γίνεται με την χρήση της εντολής `"yield return new WaitForSeconds(1f)"` και στην συνέχεια με την εντολή που χρησιμοποιήσαμε προηγουμένως, `"bloodyScreen.SetActive(false)"`. Η εντολή `yield` ουσιαστικά παράγει ένα thread, δηλαδή ένα νήμα εκτέλεσης που πρέπει να εκτελεστεί πρώτα για να συνεχίσει στην επόμενη εντολή. Εμείς τις έχουμε δώσει την εντολή `"WaitForSeconds(1f)"` ως είσοδο. Η εντολή αυτή χρησιμοποιείται μόνο με την εντολή `yield` σε κορουτίνες. Με την είσοδο του `1f` σημαίνει ότι θα γίνει αναστολή λειτουργίας της κορουτίνας για 1 δευτερόλεπτο ή για 1 καρέ. Η κορουτίνα εκτελείται μία φορά ανά δευτερόλεπτο οπότε θα εκτελεστεί στο 2ο δευτερόλεπτο. Μετά από αυτά, έχουμε την εντολή `"health -= 5"`. Με αυτή την εντολή παίρνουμε την τιμή της ζωής του παίκτη που έχουμε διαβάσει από την οπτική διεπαφή και την μειώνουμε κατά 5 για κάθε επιτυχής επίθεση που δέχτηκε ο παίκτης. Στη συνέχεια, έχουμε την επεξεργασία της οπτικής διεπαφής για την ανανέωση των πόντων ζωής. Έτσι, ορίζουμε αρχικά μία τύπου string μεταβλητή με την ονομασία `stringHealth` και εκχωρούμε μέσα σε αυτή την τιμή της `health`. Η `"(health).ToString()"` εντολή μετατρέπει την τιμή της ζωής σε τύπο string, δηλαδή σε κείμενο από ακέραια τιμή, γιατί για να ανανεώσουμε το κείμενο της διεπαφής χρειαζόμαστε κείμενο και όχι αριθμό. Τέλος, με την εντολή `"healthText.text"` αλλάζουμε την τιμή του κειμένου ζωής στην οπτική διεπαφή του χρήστη και συγκεκριμένα επεξεργαζόμαστε το πεδίο text αυτού.

Spawn και Death Για την τοποθέτηση των εχθρών στην πίστα χρησιμοποιούμε τυχαία τοποθέτησή τους σε έναν 3D χώρο.

Για την εκτέλεση αυτής της διαδικασίας έχουμε δημιουργήσει ένα script, όπως μας καθοδήγησε και το μάθημα Udemy, που λαμβάνει ένα prefab αντικείμενο zombie και το δημιουργεί στην πίστα. Αυτή η διαδικασία επαναλαμβάνεται για για 5 δευτερόλεπτα με την βοήθεια της μεθόδου Invoke. Η μέθοδος Invoke χρησιμοποιείται στην Unity για την εκτέλεση μίας διαδικασίας σε ένα συγκεκριμένο χρονικό διάστημα. Για εμάς που θέλουμε επανάληψη, μέσα στην μέθοδο αυτή έχουμε πληκτρολογήσει την εντολή "InvokeRepeating ("spawn", 0f, 5f)". Η εντολή αυτή εκτελεί μία μέθοδο στον χρόνο που ορίσουμε εμείς και στη συνέχεια το επαναλαμβάνει για κάποιο συγκεκριμένο χρονικό διάστημα που καθορίζουμε εμείς. Στη συγκεκριμένη περίπτωση, λαμβάνει 3 τιμές ως είσοδο, την μέθοδο που έχουμε δημιουργήσει με την ονομασία spawn, τον χρόνο που θα εκτελεστεί η μέθοδος που στην περίπτωσή μας είναι 0 δηλαδή στην επόμενη εκτέλεση της Update και τον χρόνο επανάληψης της μεθόδου που σε εμάς είναι 5 δευτερόλεπτα, δηλαδή θα επαναληφθεί η μέθοδος αυτή στα επόμενα 2 δευτερόλεπτα. Η διαδικασία για την τοποθέτηση των εχθρών στην πίστα γίνεται μέσα από την μέθοδο spawn που αναφέρθηκε νωρίτερα. Στην spawn ορίζουμε μία τυχαία θέση για την τοποθέτηση των εχθρών και με την θέση αυτή τους εισάγουμε στην πίστα. Η θέση φυσικά είναι σε έναν τρισδιάστατο χώρο οπότε θα λαμβάνει τιμές. Σε εμάς ήταν πολύ απλός ο ορισμός της θέσης του εχθρού. Απλά δημιουργούμε έναν Vector3, που χρησιμοποιείται για την αναπαράσταση 3D διανυσμάτων και σημείων, και στην συνέχεια εισάγουμε τις τιμές αυτές. Όταν δημιουργούμε έναν Vector3 στην Unity μηχανή, που υπάρχει μέσα σε μία μέθοδο, πρέπει να εισάγουμε μέσα σε αυτό ένα διάνυσμα με την βοήθεια της εντολής new Vector3 (X,Y,Z). Για την τυχαία τοποθέτηση απαιτείται η χρήση της εντολής Random.Range της Unity. Η Random.Range λαμβάνει 2 τιμές ως είσοδο και αντιπροσωπεύει ένα εύρος τιμών. Για παράδειγμα αν είχαμε 2 float τύπου τιμές, 0 και 5, και τις τοποθετούσαμε ως είσοδο στη εντολή Random.Range, τότε η τιμή Random.Range θα είχε ως έξοδο οποιαδήποτε τιμή που είναι ανάμεσα από την τιμή 0 και την τιμή 5. Η εντολή με την χρήση αυτών των δύο τιμών θα αλλάξει και θα γίνει Random.Range (0f, 5f).

Ένα παράδειγμα μετά από την δημιουργία ενός Vector3 για την θέση, είναι η εντολή "Vector3 pos = new Vector3 (Random.Range (0f,2f),Random.Range (0f,4f), Random.Range (2f,5f))". Αυτό δείχνει ότι δημιουργήσαμε ένα τρισδιάστατο διάνυσμα με τιμή στον X άξονα που κυμαίνεται από 0 ως 2, με τιμή στον Y που κυμαίνεται από 0 ως 4 και με τιμή στον Z άξονα που κυμαίνεται από 2 ως 5.

Τώρα, η εισαγωγή των εχθρών στην πίστα μπορεί να υλοποιηθεί με έναν απλό τρόπο χρησιμοποιώντας την εντολή Instantiate της μηχανής. Η εντολή αυτή επιστρέφει έναν υλοποιημένο κλώνο ενός αντικειμένου. και λαμβάνει 3 τιμές ως είσοδο. Αρχικά, λαμβάνει το αντικείμενο που θέλουμε να δημιουργήσουμε στην πίστα, την θέση που θέλουμε να τοποθετήσουμε το αντικείμενο, που στην δική μας περίπτωση θα είναι η pos και η περιστροφή του αντικειμένου. Η περιστροφή αυτή όπως είπαμε και πριν χρησιμοποιεί τις τιμές Euler. Η μόνη διαφορά είναι ότι την προηγούμενη φορά που το αναφέραμε χρησιμοποιήσαμε τις γωνίες Euler ενώ τώρα όχι. Αυτό συμβαίνει γιατί οι γωνίες Euler είναι περισσότερο χρήσιμες σε περιπτώσεις όπου υπάρχει ήδη ένα Quaternion ενώ το απλό Euler χρησιμοποιείται πιο συχνά σε περιπτώσεις όπου δημιουργούμε νέο Quaternion από την αρχή. Το Quaternion ή Τετραδόνιο χρησιμοποιείται από την μηχανή για την αναπαράσταση των περιστροφών. Είναι ένας απλός μαθηματικός συμβολισμός για την αναπαράσταση διαστημικών προσανατολισμών και περιστροφών αντικειμένων σε τρεις διαστάσεις. Η εντολή "Quaternion.Euler" που χρησιμοποιείται έχει 3 τιμές ως είσοδο, την τιμή για την περιστροφή στον X άξονα, την τιμή για την περιστροφή στον Y

άξονα και την τιμή για την περιστροφή στον Z άξονα.

Για τον θάνατο των εχθρών μετά την επιτυχή επίθεση του αντιπάλου πρέπει να λάβουμε υπόψιν την ζημιά που προκάλεσε η σφαίρα στον αντίπαλο και το πως επεξεργαζόμαστε τον εχθρό μετά τον μηδενισμό της ζωής του. Αυτό βασικά είναι πολύ απλό και δεν απαιτεί κάποια ιδιαίτερη πολύπλοκη διαδικασία. Αρχικά, θα έχουμε στους εχθρούς, όπως και στον χαρακτήρα μας, μία μεταβλητή τιμή που περιέχει μέσα σε αυτήν τους πόντους ζωής του. Αυτό είναι το ίδιο με τον χαρακτήρα μας, με την μόνη εξαίρεση ότι δεν ανανεώνονται οι πόντοι ζωής σε κάποια οπτική διεπαφή, όπως απαιτείται στον παίκτη. Μετά θα πρέπει να βρούμε ένα τρόπο να μειώνουμε αυτήν την τιμή και να ελέγχουμε αν η ζωή του αντιπάλου είναι ίση ή μικρότερη του μηδενός για την αφάνιση του παίκτη. Τέλος, θα πρέπει να μπορούμε να σβήσουμε τον αντίπαλο αυτόν από την πίστα αφού έχει μηδενιστεί η ζωή του. Σε εμάς είναι πολύ απλό να γίνει αυτή η διαδικασία. Αρχικά, δημιουργήσαμε μία μέθοδο με δημόσια πρόσβαση σε αυτήν για να είναι προσβάσιμη από άλλα script. Η μέθοδος αυτή έχει μία τιμή ως είσοδο που είναι οι πόντοι ζημιάς, δηλαδή πόση ζημιά έκανε ο παίκτης στον αντίπαλο. Μέσα στην μέθοδο αυτή, μετά την αφαίρεση της ζημιάς, έχουμε μία απλή if για να μπορούμε να ελέγχουμε την τιμή της ζωής του αντιπάλου και να δούμε αν αυτή η τιμή είναι μικρότερη ή ίση του μηδενός. Στη συνέχεια, απλά καλούμε μία άλλη function που έχει μία εντολή για την διαγραφή του αντικείμενου με δύο παραμέτρους ως είσοδο σε αυτήν, που είναι το αντικείμενο που θέλουμε να διαγράψουμε και τον χρόνο που θέλουμε να καθυστερήσουμε για την αφάνιση του αντικειμένου από την σκηνή.

Παίκτης

Κίνηση Η κίνηση στον παίκτη είναι πολύ πιο απλή από τα υπόλοιπα παιχνίδια, επειδή λειτουργεί η κάμερα ως ο κεντρικός χαρακτήρας του παιχνιδιού. Η κάμερα, μαζί με το γονέα αντικείμενο της, ανήκει στο πακέτο που εγκαταστήσαμε με την ονομασία "AR Foundation". Ο γονιός του αντικειμένου είναι το "AR Session Origin" αντικείμενο που ορίσαμε προηγουμένως. Η κάμερα θα κινείται με την κίνηση του κινητού του παίκτη και για αυτό τοποθετήσαμε ένα αντικείμενο μέσα σε αυτήν για να παραμένει σταθερό στην οθόνη. Αυτό είναι το "Weapon" αντικείμενο που περιέχει ως παιδί το "Pistol" αντικείμενο που περιέχει μέσα σε αυτό το μοντέλο το όπλο. Συνεπώς, σε αυτό τον τομέα δεν χρειάζεται να κάνουμε κάποια δημιουργία script για την κίνηση του παίκτη.

Επίθεση Στον τομέα της επίθεσης, δημιουργήσαμε ένα script με κάποιες εντολές για την ομαλή λειτουργία της επίθεσης του παίκτη στον αντίπαλο. Αυτό ήταν σε έναν μικρό βαθμό περίπλοκο, γιατί πέρα από την αλλαγή στην οπτική διεπαφή (UI) και την λειτουργία των εφέ του όπλου, πρέπει να εντοπίσουμε και κάποιον τρόπο για να πραγματοποιηθεί η επίθεση στον αντίπαλο. Για παράδειγμα, θα μπορούσαμε να εισάγουμε το prefab για τις σφαίρες του όπλου σε ένα πρόγραμμα και μετά να το κάνουμε "Instantiate", για την δημιουργία στην πίστα, όπως αναφέραμε προηγουμένως. Εμείς χρησιμοποιήσαμε μία άλλη προσέγγιση, όπου υλοποιούμε την επίθεση με "Raycast". Τα rays ανήκουν στην βιβλιοθήκη των Physics της Unity και χρησιμοποιούνται για να δημιουργήσουμε μία ακτίνα που ξεκινάει από ένα σημείο A στοχεύοντας προς σε ένα σημείο B ενός συγκεκριμένου μήκους Ψ. Αυτά, χρησιμοποιούνται για να μπορούμε να εξετάσουμε για συγκρούσεις με κάποιο αντικείμενο που περιέχει εξάρτημα (component) τύπου collider.

Η υλοποίηση των επιθέσεων με την χρήση raycast, έγινε με τον εξής τρόπο: Αρχικά, δημιουργήσαμε μία μεταβλητή τύπου Raycasthit για που χρησιμοποιείται για το μήκος Ψ που αναφέραμε προηγουμένως. Μετά, δημιουργήσαμε μία if, για να πραγματοποιηθεί ένας έλεγχος, με τρεις τιμές εισόδου, την θέση της κάμερας που μετράει σαν την αρχική θέση της ακτίνας, το σημείο άφιξης δηλαδή το μέρος που στοχεύει η ακτίνα και την μεταβλητή τύπου Raycasthit που δημιουργήσαμε προηγουμένως. Τέλος, εντοπίσαμε αν το αντικείμενο που στοχεύουμε είναι εχθρός, ενεργοποιούμε το ειδικό εφέ αίματος αν είναι πράγματι εχθρός και μετά από αυτό το διαγράφουμε από τον χώρο της σκηνής του παιχνιδιού.

Η συνθήκη της if που αναφέρθηκε προηγουμένως σε εντολές της γλώσσας C# έγινε ως εξής. Χρησιμοποιήσαμε την εντολή "Physics.Raycast (a,b,c)" με a να είναι η θέση της κάμερας, b το σημείο που στοχεύει η κάμερα και c την μεταβλητή που δημιουργήσαμε, όπως έχουν αναφερθεί προηγουμένως. Για να δούμε αν αυτό που χτυπήσαμε ήταν πράγματι ο εχθρός, ελέγχουμε αν το αντικείμενο περιέχει το script με την ονομασία Enemy μέσα σε αυτό. Αυτό το script χρησιμοποιείται μόνο για τους εχθρούς, οπότε δεν υπάρχει περίπτωση η μηχανή να το μπερδέψει με κάποιο λανθασμένο αντικείμενο. Αυτό πραγματοποιείται δημιουργώντας μία Enemy μεταβλητή όπου εκχωρούμε μέσα σε αυτήν την εντολή "hit.transform.GetComponent<Enemy> ()". Αυτή η εντολή λαμβάνει την ακτίνα και ελέγχει αν η θέση της ακτίνας αντιστοιχεί στο αντικείμενο εχθρό, κάνοντας ανάγνωση αν το αντικείμενο που χτυπήθηκε περιέχει κάποιο εξάρτημα Enemy (script) μέσα σε αυτό.

Μετά έχουμε άλλον έναν έλεγχο που πραγματοποιείται με `if` για να δούμε αν αυτή η μεταβλητή που δημιουργήσαμε προηγουμένως είναι διάφορη του μηδενός, δηλαδή πράγματι αν υπάρχει μέσα σε αυτήν κάποιο αντικείμενο με εξάρτημα `Enemy`. Αν ισχύει αυτή η συνθήκη τότε, χρησιμοποιούμε την μέθοδο από το `script` με την ονομασία `Enemy` που διαβάσαμε προηγουμένως για να υπολογίσουμε την ζημιά που προκάλεσε η σφαίρα στον αντίπαλο. Μετά, εισάγουμε το ειδικό εφέ με τα αίματα στην θέση που στοχεύσαμε. Αυτό γίνεται με την ίδια εντολή `Instantiate` που αναφέρθηκε νωρίτερα. Στην παράμετρο εισόδου που τοποθετούμε το μέρος που στοχεύουμε τοποθετούμε τον στόχο με την εντολή `"hit.point"`. Τέλος, καταστρέφουμε το αντικείμενο αυτό με την εντολή `"Destroy"` μετά από κάποια δέκατα του δευτερολέπτου.

Επιπλέον, επειδή θέλουμε να υπάρχει και κάποια δύναμη που ασκείται στον αντίπαλο έχουμε μία ακόμα `if` που ελέγχει αν υπάρχει `rigidbody` στην θέση στόχευσης της ακτίνας. Αν υπάρχει αυτό, τότε με την εντολή `"hit.rigidbody.AddForce"` προσθέτουμε την δύναμη που έχουμε ορίσει προηγουμένως. Η δύναμη αυτή προκαλείται από το σώμα της σφαίρας κατά την σύγκρουση με το σώμα του αντιπάλου. Όπως και στην πραγματική ζωή, κατά τον πυροβολισμό κάποιο αντικείμενου, λόγω της ταχύτητας και της μικρής μετωπικής διατομής, θα ασκούσε μία σφαίρα μεγάλη δύναμη σε κάποιο αντικείμενο. Συνήθως, αυτό θα οδηγούσε στην διαπέραση της σφαίρας μέσα από κάποιο αντικείμενο ή σώμα. Στο παιχνίδι προσπαθούμε να κάνουμε προσωμοίωση ακριβώς αυτής της ιδιότητας με την μόνη διαφορά ότι, επειδή δεν κατέχουμε κάποιο `prefab` για την σφαίρα δεν μπορούμε να αναπαράγουμε την διαπέραση της σφαίρας μέσα από κάποιο στόχο. Αυτό όμως δεν μας αποτρέπει από το να αναπαράγουμε την ισχύ που θα έχει μία σφαίρα σε κάποιο στόχο ώστε να προσφέρουμε λίγο περισσότερο ρεαλισμό στο παιχνίδι. Η τύπου `RaycastHit` μεταβλητή είναι μία δομή που επιστρέφει όλες τις πληροφορίες ενός `raycast`.

Death Στον τομέα της αποτυχημένης ολοκλήρωσης του παιχνιδιού θα εμφανιστεί μία οθόνη `Game Over` οθόνη στην διεπαφή του χρήστη, κάτι που είναι κοινό σε πολλά παιχνίδια. Θεωρητικά, για αυτό χρειάζεται να έχουμε μία μεταβλητή όπου αποθηκεύουμε τους πόντους ζωής μέσα σε αυτήν και να γνωρίζουμε ανά πάσα στιγμή τους πόντους μετά από πιθανές αλλαγές. Αυτό συμβαίνει γιατί ο παίκτης προφανώς δέχεται κάποια ζημιά από τους αντιπάλους κατά την επιτυχή επίθεση που έχει ως αποτέλεσμα την μείωση των πόντων ζωής. Η διαδικασία αυτή γίνεται με έναν πολύ απλό τρόπο προγραμματιστικά. Αρχικά, δημιουργούμε μία δημόσια μεταβλητή που αποθηκεύει τους πόντους ζωής και είναι τύπου `int`. Η μεταβλητή αυτή έγινε δημόσια, διότι επιτρέπει την τροποποίησή της μέσα από τον επιθεωρητή (`Inspector`) της μηχανής `Unity`, και επιλέχθηκε ο τύπος `integer` γιατί οι πόντοι ζημιάς που μπορούν να αφαιρεθούν είναι πάντα ακέραιες τιμές. Εμείς, μετά την δημιουργία της μεταβλητής, μπορούμε να θέσουμε τους πόντους ζωής που θέλουμε να έχει στην αρχή το παιχνίδι. Στην δική μας περίπτωση θέτουμε την τιμή ως 100. Μετά, μέσα στην μέθοδο `Update` της `Unity`, ελέγχουμε συνεχώς αν η τιμή είναι μικρότερη ή ίση του μηδενός. Αν ισχύει αυτή η συνθήκη, απλά μεταφέρουμε την σκηνή του παιχνιδιού στην οθόνη `GameOver` με την χρήση της βιβλιοθήκης `SceneManager` που υπάρχει μέσα στην μηχανή. Η εισαγωγή αυτής της βιβλιοθήκης στο `script` γίνεται με την εντολή `"Using UnityEngine.SceneManagement"` και για να αλλάξουμε σκηνή, χρησιμοποιούμε την εντολή `SceneManager.LoadScene ("GO")`. Σε εμάς το `G` είναι η σκηνή που περιέχει την οθόνη `Game Over` του παιχνιδιού.

Εφέ για σφαίρες Ο κύριος χαρακτήρας του παιχνιδιού κρατάει ένα όπλο και μπορεί να πυροβολήσει με αυτό τους αντιπάλους. Εξαιτίας αυτού, προστέθηκε ένα εφέ για τις σφαίρες που πυροβολούνται από το πυρομαχικό για να προσθέσει ένα layer ρεαλισμού περαιτέρω στο παιχνίδι. Εμείς θέλουμε οι κενοί κάλυκες να πέφτουν από το πυρομαχικό με τυχαία περιστροφή κατά την εκτόξευση της σφαίρας. Αυτό πραγματοποιείται με την τοποθέτηση της σφαίρας στον χώρο του πυροβόλου και με ένα script που έχει συνδεθεί με το prefab της σφαίρας για να διαμορφώνει την τροχιά της σφαίρας. Προφανώς, θα χρησιμοποιήσουμε για το πρώτο βήμα, την μέθοδο που αναφέραμε προηγουμένως που φροντίζει για τον πυροβολισμό του αντιπάλου. Για να γίνει αυτό προγραμματιστικά, πρέπει να εντοπίσουμε ποιες εντολές μπορούμε να εφαρμόσουμε για την εύρεση της θέσης του πυροβόλου, την θέση της τροχιάς της σφαίρας και την τοποθέτηση του πυρομαχικού στην πίστα, και να τα τοποθετήσουμε στην μέθοδο που αναφέραμε προηγουμένως.

Αρχικά, για την εύρεση του όπλου, δημιουργούμε μία μεταβλητή και αποθηκεύουμε μέσα την θέση που πήραμε από το αντικείμενο του πιστολιού. Αυτό πραγματοποιείται με την εντολή `GameObject.FindGameObjectWithTag ().transform.position` που αποθηκεύεται μέσα σε μία τύπου `Vector3` μεταβλητή, δηλαδή ένα διάνυσμα. Μέσα στην παρένθεση τοποθετείται το όνομα της ετικέτας που έχουμε δημιουργήσει για να αναγνωρίζουμε το πυροβόλο. Για παράδειγμα, το όνομα μίας ετικέτας που αντιστοιχεί στο πυροβόλο μπορεί να είναι `Pistol`. Οπότε, η εντολή θα αλλάξει και θα γίνει `...FindGameObjectWithTag ("Pistol")...`.

Το υπόλοιπο μέρος της εντολής χρησιμοποιείται όπως αναφέρθηκε και στον τομέα των εχθρών για τον εντοπισμό της θέσης. Το `transform` δίνει πρόσβαση στα 3 διανύσματα από τον επιθεωρητή της Unity, τα οποία είναι τα `position`, `rotation` και `scale`, και εμείς όπως βλέπουμε χρησιμοποιούμε το διάνυσμα `position`. Δημιουργείται μεταβλητή για να αποθηκευτεί αυτή η μεγάλη εντολή για να μην γεμίσει υπερβολικά το πεδίο των παραμέτρων εισόδου που χρησιμοποιείται αργότερα στην `Instantiate` εντολή. Επίσης, δημιουργείται μία μεταβλητή για την αποθήκευση της περιστροφής των πυρομαχικών που θέλουμε να δημιουργήσουμε στην σκηνή. Αυτό γίνεται προγραμματιστικά δημιουργώντας μία `Quaternion` τύπου μεταβλητή και αποθηκεύοντας μέσα έναν `Euler` με 3 παραμέτρους.

Η εντολή για τις `Euler` τιμές είναι `Quaternion.Euler ()` και μέσα στη παρένθεση τοποθετούνται οι τρεις παράμετροι, δηλαδή της περιστροφής στον οριζόντιο άξονα, περιστροφή στον κάθετο άξονα και περιστροφή ως προς το βάθος. Εμείς τοποθετήσαμε την τιμή `0f` και στους τρεις άξονες για να μην υπάρχει κάποια περιστροφή στο αντικείμενο. Η τιμή `0f` σημαίνει ότι ο τύπος της τιμής είναι `float` και η τιμή του δεκαδικού που εισχωρήσαμε είναι `0`. Για την τοποθέτηση του αντικειμένου στην σκηνή χρησιμοποιήθηκε η εντολή `Instantiate ()` όπου μπαίνουν 3 παραμέτροι, το prefab αντικείμενο, το διάνυσμα της θέσης και το διάνυσμα της περιστροφής. Αυτό σημαίνει ότι, υποθέτωντας ότι το όνομα του διανύσματος της θέσης είναι `pos`, της περιστροφής είναι `rot` και το όνομα αντικειμένου είναι `shell` ότι η μεταβλητή θα άλλαζε και θα γινόνταν `Instantiate (pos, rot, shell)`.

Αυτή η διαδικασία δημιούργησε το αντικείμενο της σφαίρας στη σκηνή. Τώρα, πρέπει να δούμε πως θα κατασκευάσουμε μία τυχαία περιστροφή για την σφαίρα κατά την δημιουργία. Αυτό προφανώς διεκπεραιώνεται μέσα από το script εξάρτημα που είναι συνδεδεμένο με το αντικείμενο. Το script είναι σχετικά απλό. Διαβάζουμε την `Rigidbody` κλάση του αντικειμένου, ξεκινάμε τις `i` και προσθέτουμε κάποιες δυνάμεις για να περιστρέφεται απεριόριστα το αντικείμενο.

Για να πραγματοποιηθούν αυτά, διαβάσαμε την κλάση `Rigidbody` με την εντολή `"GetComponent"()`, όπου μέσα στα σύμβολα `"` τοποθετείται το όνομα της κλάσης-εξαρτήματος που θέλουμε να λάβουμε από το αντικείμενο που είναι συνδεδεμένο το script. Οι κορουτίνες ξεκινάνε με την εντολή `"StartCoroutine"` που είδαμε προηγουμένως. Οι δυνάμεις για την απεριόριστη περιστροφή, φυσικά καθορίζονται μέσα από την μέθοδο `Update` της μηχανής `Unity`. Για να προστεθεί η περιστροφή χρησιμοποιήθηκε η εντολή `"rb.AddForce()"` όπου `rb` είναι η μεταβλητή που αποθηκεύσαμε το εξάρτημα `Rigidbody` και μέσα στην παρένθεση είναι δύναμη που θέλουμε να ορίσουμε στο `rigidbody`. Στην δική μας περίπτωση χρησιμοποιούμε τις εντολές `"transform.right"` και `"transform.up"`, όπου είναι ένα διάνυσμα που κοιτάει στα αριστερά του αντικείμενου και ένα διάνυσμα που κοιτάει πάνω αντίστοιχα. Συνεπώς, χρησιμοποιούμε 2 εντολές `"rb.AddForce()"` και στις παραμέτρους εισόδου τοποθετούμε τις παραπάνω τιμές πολλαπλασιάζοντάς το διάνυσμα με μία ταχύτητα ώστε να δώσει μία ταχύτητα και ως εξίσου να κινηθεί το αντικείμενο σε εκείνη την κατεύθυνση. Επίσης, οι `i` που χρησιμοποιούνται είναι 2. Η μία βοηθάει στην περιστροφή και η άλλη για την αναμονή πριν την εξάλειψη του αντικείμενου. Η περιστροφή ορίζεται με έναν ατέρμων βρόγχο που εκτελείται ανά 0.1 δευτερόλεπτα. Στη μέθοδο αυτή, παίρνουμε τις `Euler` γωνίες του `prefab` και αυξάνουμε την τιμή με ένα διάνυσμα τυχαίας θέσης περιστροφής. Αυτά πραγματοποιούνται με την εντολή `"transform.eulerAngles"` όπου αθροίζουμε την τιμή αυτής της μεταβλητής με μία μεταβλητή `Vector3`. Η εντολή `Vector3` μεταβλητή ορίζεται ως `"Vector3()"` και μέσα στις παραμέτρους για την τυχαία περιστροφή χρησιμοποιείται η μέθοδος `"Random.Range"`. Οι μοίρες ενός κύκλος δεν μπορούν να ξεπεράσουν την τιμή 360, οπότε και στην περιστροφή του παιχνιδιού θα ισχύει το ίδιο. Συνεπώς, για να γίνει τυχαία περιστροφή σε μία από τις τρεις παραμέτρους εισόδου θα ορίζαμε `"Random.Range(-360,360)"`, δηλαδή η περιστροφή θα είναι τυχαία σε μία τιμή του εύρους -360 με 360.

6.5 Μετατροπή από iOS σε Android

Το παιχνίδι που μάθαμε να κατασκευάσουμε στο μάθημα `Udemy` ήταν κατασκευασμένο με το πακέτο `ARKit`, που χρησιμοποιείται για να δουλέψει ένα παιχνίδι ή μία εφαρμογή στις συσκευές με λογισμικό `iOS`. Για αυτό εμείς θέλαμε να μετατρέψουμε το παιχνίδι σε μορφή που υποστηρίζεται από τα πακέτα `ARCore` ή `ARFoundation` της `Unity`. Το `ARCore` χρησιμοποιείται για την δημιουργία λογισμικού για `Android` συσκευές, ενώ το `ARFoundation` χρησιμοποιείται για την δημιουργία και στα δύο λογισμικά, με την χρήση των πακέτων που προαναφέραμε.

Για την επίτευξη αυτού, μεταφορτώσαμε το `ARFoundation` πακέτο σε ένα νέο πρότζεκτ. Με την χρήση αυτού, εισάγαμε ένα `AR Session` και ένα `AR Session Origin` αντικείμενο στην σκηνή μας. Αυτά τα δύο είναι απαραίτητα για την χρήση του `ARFoundation` πακέτου.

Μετά ρυθμίσαμε την κάμερα στην `Android` εφαρμογή για να ταιριάζει με την κάμερα της `iOS` εφαρμογής, και στην επιλογή `"Clear Flags"` της κάμερας, επιλέξαμε `Solid Color`. Προφανώς δεν χρησιμοποιείται το script `ARHitTestExample` που υπάρχει στην έκδοση της `iOS` συσκευής οπότε αφαιρέθηκε. Επειδή υπήρχε πρόβλημα με την έκδοση του `ARCore` στην μηχανή `Unity` μεταφορτώσαμε την προηγούμενη έκδοση.

Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα

Τα συμπεράσματα από την έρευνα ήταν η ανακάλυψη ότι η τεχνολογία έχει αναπτυχθεί αρκετά από την αρχική εμφάνισή της, αν και βρίσκεται ακόμα σε αρχικό στάδιο στον τομέα της ανάπτυξης ηλεκτρονικών παιχνιδιών για κινητές συσκευές. Αυτό παρατηρήθηκε με την ανάπτυξη ενός παιχνιδιού που ενσωματώνει πρόσφατες τεχνολογίες επαυξημένης πραγματικότητας με μία από τις δημοφιλέστερες μηχανές κατασκευής παιχνιδιών, δηλαδή την μηχανή Unity. Παρόλο που το παιχνίδι ολοκληρώθηκε με επιτυχία, είχε ελλείψεις σε μερικά κύρια μέρη. Μολονότι αυτό σε ένα βαθμό οφείλεται στην έλλειψη κατάλληλου τεχνολογικού εξοπλισμού, δεν σημαίνει όμως ότι παύει να ισχύει και εξαιτίας των ίδιων των πακέτων που χρησιμοποιήθηκαν για την κατασκευή της εφαρμογής.

Τα πακέτα αυτά είναι δύσκολο να ανιχνεύσουν μία μεγάλη επιφάνεια για την παραγωγή εχθρών εκτός αν έχει φορτωθεί η βάση δεδομένων με κάποια έτοιμα στοιχεία με την τοποθεσία και το περιβάλλον(άτομα, αντικείμενα περιβάλλοντος) που έχουμε αποφασίσει να καταγράψουμε.

Μετά από δοκιμές της εφαρμογής κάποιων ατόμων, παρατηρήθηκε ότι αν και ήταν διασκεδαστικό το παιχνίδι, ήταν απαραίτητο να υπάρχει σωστός φωτισμός και ευρύχωρος χώρος για να λειτουργήσουν όλα σωστά. Επίσης, πρέπει ο χρήστης να έχει το κινητό ακίνητο κατά την εκκίνηση να δείχνει το έδαφος, αλλιώς μπορεί να καταγραφεί λανθασμένα το περιβάλλον και να αλλοιωθεί η εμπειρία του χρήστη.

Αυτό μας δείχνει ότι τα πακέτα που χρησιμοποιήθηκαν έχουν ακόμα ατέλειες, κάτι που αποδεικνύεται και από τις συνεχείς αναβαθμίσεις των πακέτων που εκδίδονται.

7.2 Μελλοντικές προεκτάσεις

Το παιχνίδι για τον σκοπό της πτυχιακής είναι πολύ απλό και χρησιμοποιείται απλώς ως παράδειγμα, αλλά θα προτείνουμε κάποιες πιθανές επεκτάσεις της εφαρμογής.

Μία μελλοντική επέκταση που θα μπορούσε να ενσωματωθεί στο παιχνίδι για να ανήκει πραγματικά στην κατηγορία FPS Survival είναι η προσθήκη κυμάτων στο παιχνίδι. Κάποιες πιθανές επεκτάσεις μπορούν να είναι:

1. **Κύματα:** Τα κύματα (waves) είναι σαν τις πίστες (levels) στα παιχνίδια και έχουν να κάνουν με την προσθήκη εχθρών σε διαστήματα στην εφαρμογή.

Θα μπορούν να εισάγονται στην πίστα σε κάποια προκαθορισμένα χρονικά διαστήματα ή να δημιουργούνται αυτόματα όταν ο χρήστης θα ολοκληρώνει το προηγούμενο κύμα. Η προσθήκη αυτής της λειτουργίας στο παιχνίδι θα μπορεί να το κάνει πιο ενδιαφέρον για τον παίκτη, διότι αυξάνει το επίπεδο δυσκολίας και προσφέρει μια πρόκληση στον παίκτη. Αυτό, κάνει το παιχνίδι πιο ελκυστικό και σε συνδυασμό με άλλες επεκτάσεις μπορεί να προσφέρει μία φρεσκάδα για τους παίκτες. Η εφαρμογή αυτού του χαρακτηριστικού στο παιχνίδι δεν είναι ιδιαίτερα δύσκολη, ειδικά στην περίπτωση με τα προκαθορισμένα παιχνίδια και θα μπορούσε να υλοποιηθεί στην τωρινή εργασία, απλά δεν προστέθηκε γιατί το παιχνίδι παράγει πολλαπλούς εχθρούς ταυτόχρονα και δεν χρειάζεται να προστεθούν και άλλοι γιατί θα αυξηθεί η δυσκολία σε μεγάλο βαθμό. Για την ενσωμάτωση μπορούμε να θέσουμε μια μεταβλητή με τον χρόνο και χρησιμοποιώντας αυτήν στην συνάρτηση Update με μία if για την εκτέλεση της μεθόδου αναπαραγωγής εχθρών. Για παράδειγμα, αν η μεταβλητή για τον χρόνο είναι ορισμένη σε 180 τότε οι εχθροί θα αναπαράγονται ανά 2 λεπτά στο παιχνίδι. Στη συνέχεια, η τιμή αυτή μπορεί να ελαχιστοποιηθεί ή να αυξηθεί αναλόγως για να γίνεται εισαγωγή εχθρών σε πιο σύντομα ή πιο μεγάλα χρονικά διαστήματα αντίστοιχα.

2. **Πακέτα ζωής:** Τα πακέτα ζωής υπάρχουν σε πολλά παιχνίδια και είναι σημαντικά για τους παίκτες. Χρησιμοποιούνται για να αυξήσουν τους πόντους ζωής όταν είναι κάτω του μέγιστου ορίου. Σε άλλα παιχνίδια όπου ο παίκτης έχει ακέραιο αριθμό ζών και όχι πόντους ζωής, τότε με την απόκτηση ενός τέτοιου πακέτου αυξάνεται ο αριθμός των ζών του παίκτη κατά 1. Στην ολοκληρωτική απώλεια των ζών του παίκτη τότε ο παίκτης χάνει το παιχνίδι και για αυτό οι ζωές είναι σημαντικές στα παιχνίδια αυτά. Στα αγγλικά οι πόντοι ζωής ονομάζονται "hit points", ένας όρος που ορίστηκε αρχικά από τον συν-δημιουργό του δημοφιλές παιχνιδιού Dungeons and Dragons. Στο συγκεκριμένο παιχνίδι, ο παίκτης χάνει ένα ορισμένο πλήθος πόντων ζωής μετά από κάθε επιτυχημένη επίθεση του αντιπάλου. Με την προσθήκη πακέτων ζωής στο παιχνίδι μπορεί να αυξήσει την πιθανότητα επιβίωσης του παίκτη κατά την αντιμετώπιση απέναντι στους εχθρούς. Η προσθήκη πακέτων ζωής είναι εύκολη, διότι απλά χρειάζεται ένα τετράγωνο αντικείμενο με ένα σταυρό στο κέντρο όπου έχουμε την δυνατότητα να το τοποθετήσουμε στην πίστα όποτε θέλουμε. Μπορεί να εισάγεται στην πίστα ανά χρονικά διαστήματα, μπορεί να εισάγεται στην πίστα μετά από κάποιο προκαθορισμένο πλήθος πόντων στον παίκτη κτλ.
3. **Άλλοι εχθροί:** Η χρήση ενός μοντέλου εχθρού μπορεί να προκαλέσει μονοτονία κατά την εμπειρία του χρήστη και να οδηγήσει στην απώλεια ενδιαφέροντος. Η προσθήκη άλλων εχθρών μπορεί να κάνει το παιχνίδι πιο ενδιαφέρον και δημιουργικό. Ωστόσο, αυτό αν και ακούγεται καλό, είναι δύσκολο να υλοποιηθεί. Η προσθήκη άλλων εχθρών με ίδιες συμπεριφορές με τα τωρινά μοντέλα, θα έχει το αντίθετο αποτέλεσμα. Η εύρεση τέτοιων 3D μοντέλων δεν είναι δύσκολη και μπορούμε να εντοπίσουμε μοντέλα μέσα από το Asset Store, δηλαδή το "μαγαζί" της Unity. Το δύσκολο είναι η εφαρμογή άλλων συμπεριφορών για τους αντιπάλους. Μερικοί εχθροί θα είναι εύκολο να εφαρμοστούν, διότι θα έχουν όμοιο animator χωρίς κάποια ιδιαίτερη αλλαγή πέραν από του μοντέλου. Όμως, άλλοι πιο έξυπνοι και πολύπλοκοι εχθροί θα πρέπει να έχουν διαφορετικό Animator με μεγάλες τροποποιήσεις, που θα έχει ως αποτέλεσμα την προσθήκη ή τροποποίηση των περισσότερων script του παιχνιδιού.

Επιπλέον, κατά την εισαγωγή νέων εχθρών θα πρέπει να βρούμε μία συμπεριφορά για να ενσωματώσουμε σε αυτούς όπως για παράδειγμα ή αναπαραγωγή κατά χρονικά διαστήματα. Ακόμα, πρέπει να λάβουμε υπόψιν και το περιβάλλον. Θα είναι δύσκολο να βρούμε χαρακτήρες για να ενσωματώσουμε σε ένα μετά-αποκαλυπτικό περιβάλλον, πέραν του μοντέλου ενός ζόμπι, με διαφορετική συμπεριφορά.

4. **Boss:** Οι boss χαρακτήρες υπάρχουν σε πολλαπλά παιχνίδια και προσφέρουν μία διαφορετική εμπειρία στον χρήστη. Είναι πιο δύσκολοι από τους κανονικούς εχθρούς και ως αποτέλεσμα έχουν πιο πολύπλοκη συμπεριφορά από αυτούς. Η ενσωμάτωσή τους είναι πολύ πιο δύσκολη από τους γενικούς εχθρούς που αναφέρθηκαν παραπάνω εξαιτίας αυτής της συμπεριφοράς και της εφαρμογής του στο παιχνίδι. Ο χαρακτήρας αυτός όπως όλοι οι εχθροί ελέγχεται από τον υπολογιστή και για την επιτυχία στην αντιμετώπιση τους απαιτείται μεγαλύτερη γνώση των μηχανικών του παιχνιδιού. Υπάρχουν κατηγορίες boss χαρακτήρων όπως είναι οι miniboss χαρακτήρες που αναφέρεται σε μειωμένης δυσκολίας boss χαρακτήρες που δεν έχουν τόση σημασία όσο οι κεντρικοί boss χαρακτήρες στο παιχνίδι. Επίσης, υπάρχουν οι superboss και οι final boss χαρακτήρες που αναφέρονται σε εχθρούς που αντιμετωπίζονται πιο δύσκολα από τους υπόλοιπους boss σε ένα παιχνίδι και ως εξίσου είναι πιο δύσκολοι και χαρακτήρες που είναι οι κεντρικοί ανταγωνιστές του πρωταγωνιστή ενός παιχνιδιού αντίστοιχα. Για την εισαγωγή αυτών στην εφαρμογή μας, απαιτείται αυξημένη γνώση της χρήσης της μηχανής Unity. Αυτό συμβαίνει γιατί θα απαιτηθεί εφαρμογή νέου πολύπλοκου animator με διαφορετικά μοτίβα συμπεριφοράς. Τα animation αν και είναι εύκολο να βρεθούν στο διαδίκτυο, θα πρέπει να συνδυαστούν με ένα script και διάφορες μεταβλητές που βρίσκονται μέσα στον Animator, κάτι που είναι δύσκολο. Για την εύρεση τους θα πρέπει να κάνουμε ειδική τροποποίηση animation από εργαλεία όπως το Adobe Mixamo για την ενσωμάτωση στην μηχανή. Αυτό, συμβαίνει γιατί συνήθως υπάρχουν λιγότερα assets που έχουν να κάνουν με boss από εχθρούς στο "μαγαζί" της Unity.
5. **Power ups:** Τα power ups, όπως λέει και η λέξη, προσφέρουν κάτι επιπλέον στον παίκτη, δηλαδή τον ενδυναμώνουν. Ο μόνος περιορισμός στην ενσωμάτωσή τους είναι η φαντασία του δημιουργού, διότι υπάρχουν άπειρες κατηγορίες τους που μπορούν να εφαρμοστούν σε ένα παιχνίδι. Είναι ενδιαφέρον και κάνουν το παιχνίδι πιο ελκυστικό για τον χρήστη. Μπορούν να τοποθετηθούν στην πίστα μετά από την σύλληψη κάποιου συγκεκριμένου πλήθους πόντων ή να τοποθετούνται στην πίστα μετά από κάποιο χρονικό διάστημα. Ένα παράδειγμα μπόνους είναι η προσφορά ασπίδας στον παίκτη που τον κάνει άφθαρτο για κάποιο μικρό χρονικό διάστημα. Άλλα μπορούν να είναι η προσφορά μπόνους πόντων, η προσφορά πόντων ζωής, η αλλαγή των πυρομαχικών κ.α Για την ενσωμάτωσή τους μπορούμε να πάρουμε οποιοδήποτε αντικείμενο και να τα ενσωματώσουμε στο script των εχθρών. Μπορούμε να έχουμε κάποιον μετρητή που μετράει πόσους εχθρούς έχει συνολικά νικήσει ο παίκτης και να το υλοποιούμε στην θέση του νικημένου εχθρού. Αυτό θα υπάρχει σε εκείνο το μέρος για ένα πολύ σύντομο χρονικό διάστημα και μετά θα εξαφανίζεται, δηλαδή θα διαγράφεται από εκείνο το μέρος που ήταν. Το ίδιο θα συμβαίνει και μετά την σύλληψη τους από τον χρήστη όπου και θα προσφέρεται η ικανότητα στον παίκτη και θα διαγράφεται από εκείνο το μέρος.

Για πολλαπλά διαφορετικά power ups, μπορούμε να τα ενσωματώσουμε σε άλλους εχθρούς ή μετά από την ολοκλήρωση κάποιου στόχου του παίκτη.

6. **Στόχοι:** Η επίλυσή κάποιων στόχων στο παιχνίδι για την απόκτηση κάποιων εξτρά πόντων μπορεί να δώσει στον παίκτη ένα κίνητρο και να μειώσει την μονοτονία του μετά από την πολύωρη ενασχόληση στο παιχνίδι. Για την ενσωμάτωση τους στην εφαρμογή, μπορούμε να θέσουμε μία λίστα στόχων στο παιχνίδι και να επιλέξουμε κάποιους που θα αναπαράγονται τυχαία κατά την πρώτη εκτέλεση του παιχνιδιού. Έτσι, θα προσφέρεται μία πρόκληση στον παίκτη που θα απαιτεί την ολοκλήρωση του σε ένα σύντομο χρονικό διάστημα. Όπως και πολλά από τα προηγούμενα, είναι πολλαπλές κατηγορίες που θα μπορούσαν να ανήκουν αυτοί οι στόχοι. Μπορούμε να ενσωματώσουμε κάποια μπόνους που να ανήκουν στην σύλληψη κάποιων μπόνους, όπως κλειδιά, αστέρια, μετάλλια κτλ., που θα είναι κρυμμένα στην πίστα και να περιμένουμε από τον χρήστη να τα συλλέξει. Μετά από την επιτυχημένη σύλληψή τους, θα επιβραβεύεται ο χρήστης με κάποιο έπαθλο ή κάποιους μπόνους πόντους.

7. **Επιπλέον όπλα:** Τα επιπλέον όπλα και τα διαφορετικά πυρομαχικά είναι αξιότιμες προσθήκες σε ένα παιχνίδι FPS και προσφέρουν μία ποικιλία επιλογών στον παίκτη. Τα διαφορετικά όπλα βελτιώνουν την εμπειρία του χρήστη και μπορούν να προσφέρουν κάποιες επιλογές στον χρήστη. Είναι πολλά τα πιθανά σετ όπλων που μπορούν να προστεθούν στο παιχνίδι. Για παράδειγμα μπορεί να προστεθεί πυροβόλο επιθετικό τυφέκιο πίεσεως αερίων όπως είναι το AK-47 όπου θα αυξάνει την ροή των σφαιρών ή μπαζούκα, τον κοινό εκτοξευτή πυραύλων όπου θα εκτοξεύονται πύραυλοι/χειροβομβίδες στους αντιπάλους. Ο εκτοξευτής πυραύλων θα προσφέρεται για ένα περιορισμένο χρονικό διάστημα στον παίκτη για να μην μειώσει την εμπειρία του παίκτη.

Στην περίπτωση που θα υπήρχε για άπειρο χρονικό διάστημα τότε το παιχνίδι θα γίνονταν πολύ εύκολο εκτός και αν οι εχθροί αυξάνονταν, κάτι που θα έκανε το παιχνίδι πολύ περίπλοκο. Η εφαρμογή αυτών στο παιχνίδι θα ήταν δύσκολη εξαιτίας του μηχανισμού χρήσης του παιχνιδιού. Το παιχνίδι λειτουργεί με ένα κουμπί που δεν μπορεί να λειτουργήσει για τον συνεχόμενο πυροβολισμό που απαιτούν όπλα με συνεχόμενο πυροβολισμό. Στην περίπτωση εκείνη θα πρέπει να τροποποιηθεί καταλλήλως το κουμπί πυροβολισμού. Ένα άλλο θέμα είναι οι συμπεριφορές των νέων όπλων. Αυτό το θέμα μπορεί να επιλυθεί με την εύρεση μοντέλων όπλων και την χρήση του ίδιου script με το τωρινό όπλο με αυξημένη ροή πυροβολισμού. Στην περίπτωση με τους πυραύλους θα πρέπει να προστεθεί το κατάλληλο μοντέλο και να δημιουργηθεί νέο script με την τροχιά που θα πρέπει να κινείται η χειροβομβίδα ή στην περίπτωση με τους πυραύλους να γίνει πιο αργή η κίνηση του. Επίσης, θα πρέπει να βρούμε μοντέλα στο μαγαζί της Unity για τους πυραύλους/χειροβομβίδες.

8. **Μαγαζί:** Το μαγαζί θα ήταν πολύ σημαντικό για την ανάπτυξη του παιχνιδιού. Αρχικά, ο παίκτης θα έχει κίνητρο για την συλλογή πόντων. Στη συνέχεια, θα του έδινε την δυνατότητα να μπορεί να αγοράσει κοστούμια, χαρακτήρες εχθρών, όπλα κτλ. Αυτό έχει κάποιες ευκολίες αλλά και κάποιες δυσκολίες ως προς την ενσωμάτωση του στην εφαρμογή. Το εύκολο είναι η δημιουργία του ίδιου του μαγαζιού και η τοποθέτηση κάποιων αντικειμένων σε αυτό. Η επιλογή χαρακτήρων, επειδή είναι απλά αντικείμενα, είναι εύκολο να εισαχθεί στο μαγαζί.

Απλά εγκαταστείς κάποια μοντέλα και μετά τα περνάς στο μαγαζί και μπορείς εύκολα με ένα βασικό script να εναλλάσσεις μεταξύ αυτών. Επιπλέον και τα όπλα με παρόμοιο τρόπο μπορούν να ενσωματωθούν. Ωστόσο, για τα κοστούμια των αντιπάλων, επειδή δεν υπάρχουν συνήθως έτοιμα μοντέλα αντιπάλων με αυτά, θα πρέπει να ετοιμαστούν χειρωνακτικά ή θα πρέπει να εντοπιστούν έτοιμοι χαρακτήρες εχθρών με έτοιμα κοστούμια και να αλλάζεις ανάμεσα με αυτά. Όμως, συνήθως δεν διατίθενται τέτοια μοντέλα ή πρέπει να πληρώσεις μία ακριβή τιμή για αυτά.

Βιβλιογραφία

- [1] “Video game.” [Online]. Available: https://en.wikipedia.org/wiki/Video_game
- [2] I. E. Sutherland, *A head-mounted three dimensional display*, ser. fall joint computer conference. Thompson Books, 1968, pp. 757–764.
- [3] “Ivan sutherland and bob sproull create the first virtual reality head mounted display system.” [Online]. Available: <https://www.historyofinformation.com/detail.php?id=861>
- [4] “The sword of damocles (virtual reality).” [Online]. Available: [https://en.wikipedia.org/wiki/The_Sword_of_Damocles_\(virtual_reality\)](https://en.wikipedia.org/wiki/The_Sword_of_Damocles_(virtual_reality))
- [5] N. Regrebsubla, *Determinants of Diffusion of Virtual Reality*. GRIN Publishing, 2005.
- [6] M. A. Callahan, “A 3-d display head-set for personalized computing,” Ph.D. dissertation, Massachusetts Institute of Technology.
- [7] “The history of augmented reality.” [Online]. Available: <http://sevenmediainc.com/the-history-of-augmented-reality/>
- [8] *Virtual fixtures: Perceptual tools for telerobotic manipulation*.
- [9] “History of vr/ar and impact on our future.” [Online]. Available: <https://www.sutori.com/item/1999-hybrid-synthetic-vision-system-in-1999-nasa-installed-hybrid-synthetic-v>
- [10] R. A. . t. D. M. C. Rauschnabel, P. A., “An adoption framework for mobile augmented reality games: The case of pokémon go.” *Computers in Human Behavior*, pp. 276–286, 2017.
- [11] “Augmented reality.” [Online]. Available: https://en.wikipedia.org/wiki/Augmented_reality#Possible_applications
- [12] N. Garg, “5 ways augmented reality is redefining the gaming industry.” [Online]. Available: <https://e27.co/5-ways-augmented-reality-is-redefining-the-gaming-industry-20190415/>
- [13] G. B., “How augmented reality makes advertising interactive.” [Online]. Available: <https://rubygarage.org/blog/augmented-reality-in-advertising>
- [14] R. Williams, “Burger king delivers immersive ar experience during vmas,” 2020. [Online]. Available: <https://www.marketingdive.com/news/burger-king-delivers-immersive-ar-experience-during-vm-as/584459/>

- [15] A. Morozova, "The applications of augmented reality in construction." [Online]. Available: <https://jasoren.com/augmented-reality-in-construction/>
- [16] H. J. P. M. . L. J. Abernathy, M., "Debris correlation using the rockwell worldview system." *Proceedings of 1993 Space Surveillance Workshop.*, pp. 189–195, 1993.
- [17] E. S. Agency, "Cadmass." [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2012/01/CAMDASS
- [18] "Unc ultrasound/medical augmented reality research." [Online]. Available: <http://www.cs.unc.edu/Research/us/>
- [19] F. J. N. S. S. L. . M. P. W. Mountney, P., "An augmented reality framework for soft tissue surgery," *International Conference on Medical Image Computing and Computer-Assisted Intervention.*, pp. 423–431, 2014.
- [20] Y. T. Rita Layona, Budi Yulianto, "Web based augmented reality for human body anatomy learning," *Procedia Computer Science*, vol. 135, 2018.
- [21] G. Mileva, "How augmented reality is used in teaching anatomy." [Online]. Available: <https://arpost.co/2020/06/10/augmented-reality-teaching-anatomy/>
- [22] E. M. de Sa, "Mobile augmented reality: Exploring design and prototyping techniques," *MobileHCI '12 Proceedings of the 14th international conference on Human-computer interaction with mobile devices and service*, pp. 221–230, 2012.
- [23] B. A. K. M. R. . B. W. Herbst, I., "Timewarp: interactive time travel with a mobile mixed reality game," *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pp. 235–244, 2008.
- [24] W. Piekarski and B. Thomas, "Arquake: the outdoor augmented reality gaming system," *Communications of the ACM*, vol. 1, pp. 36–38, 2002.
- [25] G. K. H. L. W. F. F. F. S. W. T. S. L. . . Y. X. Cheok, A. D., "Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing," *Personal and ubiquitous computing*, vol. 2, pp. 71–81, 2004.
- [26] "Pac-man." [Online]. Available: <https://el.wikipedia.org/wiki/Pac-Man>
- [27] "Foot-based mobile interaction with games." *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, p. 321, 2004.
- [28] M. B. Henrysson, Anders and M. Ollila., "Face to face collaborative ar on mobile phones." *Fourth ieee and acm international symposium on mixed and augmented reality (ismar'05)*, pp. 80–89, 2005.
- [29] S. S. Youm, DongHyun and J.-Y. Kim., "Design and development methodologies of kkongalmon, a location-based augmented reality game using mobile geographic information." *EURASIP Journal on Image and Video Processing 2019*, vol. 1, pp. 1–11, 2019.
- [30] K. S. N. B. W. . G. K. Andersen, T. L., "Designing an augmented reality board game with children: the battleboard 3d experience." *Proceedings of the 2004 conference on Interaction design and children: building a community*, pp. 137–138, 2004.

- [31] N. C. Cordeiro, Diogo and R. Jesus., "Arzombie: A mobile augmented reality game with multimodal interaction." *2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*., pp. 22-31, 2015.
- [32] Y. Chen, "An overview of augmented reality technology."
- [33] O. J. L. I. H. I. . B. A. K. Broll, W., "Meeting technology challenges of pervasive augmented reality games," *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, pp. 28-es, 2006.
- [34] M. L. Heilig, "Sensorama simulator." U.S. Patent 3 050 870, Aug. 1962.
- [35] "Game engine." [Online]. Available: https://en.wikipedia.org/wiki/Game_engine
- [36] M. Mariani, "The tragic, forgotten history of zombies." [Online]. Available: <https://www.theatlantic.com/entertainment/archive/2015/10/how-america-erased-the-tragic-history-of-the-zombie/412264/>
- [37] "History of zombies." [Online]. Available: <https://www.history.com/topics/folklore/history-of-zombies>
- [38] "Zombie." [Online]. Available: <https://en.wikipedia.org/wiki/Zombie>
- [39] "Quickstart for android." [Online]. Available: <https://developers.google.com/ar/develop/unity/quickstart-android>
- [40] "About arkit xr plugin." [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.arkit@4.1/manual/index.htm>
- [41] "Arcore." [Online]. Available: <https://en.wikipedia.org/wiki/ARCore>

Κεφάλαιο 8

Ακρωνύμια και συντομογραφίες

LAN Local Area Network

FPS First-Person Shooter

AR Augmented Reality

HMD Head-Mounted Display

GPS Global Positioning System

OS Operating System

VR Virtual Reality

iOS iPhone operating system

CRT Cathode-ray tube

MIT Massachusetts Institute of Technology

NASA National Aeronautics and Space Administration

WIFI Wireless Fidelity

3G Third Generation

RFID Radio Frequency Identification

GPRS General Packet Radio Service

PDA Personal Digital Assistant

GUI Graphical User Interface

MacOS Macintosh Operating System

OpenGL Open Graphics Library

OSX Operating System Extension

PS PlayStation

TV Television

SDK Software Development Kit

XR X Reality(Cross Reality)

UI User Interface

QR Quick Response

CAD Computer-aided Design