

Ιόνιο Πανεπιστήμιο

Τμήμα Πληροφορικής



Πτυχιακή Εργασία

**Ανάπτυξη ψυχαγωγικού λογισμικού για άτομα με
Ήπια Γνωστική Διαταραχή**

Του φοιτητή

Τσίπα Κωνσταντίνου

A.M.: Π2015024

Επιβλέπων Καθηγητής

Βλάμος Παναγιώτης

Κέρκυρα 2020 - 2021

Περιεχόμενα

Περίληψη.....σελ. 4	σελ. 4
Εισαγωγή.....σελ. 5	σελ. 5
Κεφάλαιο 1: Ανάλυση Βιβλιογραφίας.....σελ. 6	σελ. 6
Κεφάλαιο 2: Σχεδιασμός.....σελ. 18	σελ. 18
Κεφάλαιο 3: Ανάπτυξη.....σελ. 19	σελ. 19
Κεφάλαιο 4: Projects.....σελ. 22	σελ. 22
Project #1: Cubeathlon.....σελ. 22	σελ. 22
Project #2: Skillxagon.....σελ. 38	σελ. 38
Project #3: Memory Puzzle Game.....σελ. 48	σελ. 48
Πρώτες Εκδόσεις.....σελ. 62	σελ. 62
Μελλοντικές Πιθανές Αναβαθμίσεις.....σελ. 63	σελ. 63
Κεφάλαιο 5: Παρουσίαση των εφαρμογών σε άτομα με ήπιες γνωστικές διαταραχές.....σελ. 65	σελ. 65
Κεφάλαιο 6: Τελικά Συμπεράσματα.....σελ. 72	σελ. 72

Πηγές.....σελ. 74

Περίληψη

Η παρούσα εργασία έχει να κάνει με την συνεισφορά της πληροφορικής και των διαφόρων κλάδων της, στην περίπτωση μας development και βιοπληροφορική, για την καταπολέμηση ή αποτροπή νοητικών παθήσεων όπως έλλειψη συντονισμού και συγχρονισμού, έλλειψη βραχυχρόνιας και μακροχρόνιας μνήμης κλπ. Η εργασία αποτελείται από 3 projects (βιντεοπαιχνίδια). Το καθένα από αυτά είναι αφοσιωμένο σε μία ή περισσότερες νοητικές παθήσεις. Πριν την δημιουργία του κάθε βιντεοπαιχνιδιού πραγματοποιήθηκε αναλυτικός σχεδιασμός όσων αφορά το αντικείμενο με το οποίο θα ασχολείται και το πρόβλημα το οποίο θα έχει σκοπό να καταπολεμά. Στη συνέχεια πραγματοποιήθηκε η ανάπτυξη του καθενός και η ενημέρωση των υπαρχόντων εκδόσεων στα τελικά τους αποτελέσματα.

Η συνεισφορά ενός τέτοιου έργου θα μπορούσε να είναι πολύ μεγάλη εξυψώνοντας την επιστήμη της πληροφορικής ακόμα περισσότερο καθώς μια συνεργασία με διάφορους τομείς της ιατρικής και πιο συγκεκριμένα της νευροεπιστήμης θα μπορούσε να επιφέρει ενδιαφέροντα αποτελέσματα. Τα αποτελέσματα θα μπορούσαν να είναι πρωτοποριακά καθώς ο συνδυασμός παρόμοιων projects με μία σωστή φαρμακευτική αγωγή θα μπορούσε να αποτρέψει πολλές νοητικές παθήσεις πριν φτάσουν σε απροχώρητο σημείο. Από τα 2 άτομα που εθελοντικά έλαβαν μέρος στην παρούσα έρευνα, ένα με διαγνωσμένη έλλειψη συγκέντρωσης και ένα με διαγνωσμένη άνοια, έδειξαν θετικά αποτελέσματα από την δεύτερη κιόλας προσπάθεια στα βιντεοπαιχνίδια αφού τους έγινε οικείο το εικονικό περιβάλλον του εκάστοτε βιντεοπαιχνιδιού την πρώτη τους φορά. Στη συνέχεια, στο πλαίσιο μιας ημερίδας που πραγματοποιήθηκε από την Ένωση Κυριών Δράμας – Σπίτι Ανοιχτής Φιλοξενίας για την παγκόσμια ημέρα κατά της νόσου του Alzheimer's, με προσκάλεσαν να τους παρουσιάσω τα projects μου σε 7 άτομα με Alzheimer's ή γνωστικές διαταραχές παρόμοιες του Alzheimer's και να τα δουν στην πράξη.

Εισαγωγή

Το πρώτο βιντεοπαιχνίδι που δημιουργήθηκε ποτέ ήταν το “Pong” και δημοσιεύτηκε το 1970. Για πολλά χρόνια τα βιντεοπαιχνίδια θεωρούνταν μία απασχόληση για παιδιά για αποκλειστικά ψυχαγωγικούς σκοπούς. Με τα χρόνια τα βιντεοπαιχνίδια έγιναν ένα από τα κύρια μέσα αφήγησης ιστοριών κάνοντας πρωτοποριακά βήματα όσων αφορά την σεναριογραφία, την σκηνογραφία, την ηθοποιία και την παραγωγή. Παιχνίδια όπως το The Last of Us έφεραν σενάρια και ηθοποιούς άξιους του Hollywood στο κοινό των βιντεοπαιχνιδιών. Το «μυστικό όπλο» των βιντεοπαιχνιδιών είναι η διάδραση. Ο παίκτης έχει την ευκαιρία να χειριστεί τον πρωταγωνιστή καθώς και να πάρει αποφάσεις οι οποίες αλλάζουν το αποτέλεσμα της ιστορίας. Πολλές εταιρίες, βλέποντας την σημαντικότητα της διάδρασης, προχώρησαν στην ανάπτυξη πληθώρας εφαρμογών και, σε συνεργασία με τον τομέα της βιοπληροφορικής, είχαν πρωτόγνωρα αποτελέσματα. Πλέον οι χειρουργοί μπορούν να κάνουν εξάσκηση σε εικονικούς ασθενείς για να γίνουν καλύτεροι, οι μελλοντικοί πιλότοι εκπαιδεύονται σε προσομοιωτές για να προσγειώνουν και να απογειώνουν αεροπλάνα, καθώς και να αντιμετωπίζουν δύσκολες συνθήκες χωρίς να χρειαστεί να βάλουν ζωές και εξοπλισμό σε κίνδυνο. Όλα τα παραπάνω είναι παραδείγματα της σημαντικότητας αλλά και του θριάμβου της ανάπτυξης βιντεοπαιχνιδιών και της βιοπληροφορικής.

Με την συνεχή ανάπτυξη της τεχνολογίας ο τομέας αυτός μονίμως εξελίσσεται. Εξού και η ανάπτυξη και συγγραφή της παρούσας εργασίας. Η εργασία έχει ξεκάθαρα τεχνολογικό πλαίσιο. Στόχος είναι να εξακριβώσουμε αν ο τομέας του video game development με αυτόν των bioinformatics μπορούν να επιφέρουν σημαντικά και αξιόπιστα αποτελέσματα και να βοηθήσουν ανθρώπους που αντιμετωπίζουν νοητικές παθήσεις. Δεν υπάρχει αμφισβήτηση για την σημαντικότητα αυτής της μελέτης καθώς θα μπορούσαν να αποτραπούν πολλά περιστατικά που θα έφταναν σε απροχώρητα σημεία για τους ασθενείς. Έχοντας αναλύσει προηγούμενες μελέτες όπως (όλες οι μελέτες) καταλήγουμε στο συμπέρασμα ότι είναι μία πολύ φιλόδοξη μελέτη και άξια ενασχόλησης καθώς τα αποτελέσματα που μπορεί να επιφέρει μπορούν να αλλάξουν δραματικά την μεθοδολογία αντιμετώπισης νοητικών παθήσεων όπως έλλειψης

συγκέντρωσης και επιλεκτικής άνοιας εκτός των άλλων. Στα κεφάλαια που ακολουθούν θα δούμε αναλυτικά την σύλληψη της ιδέας από μελέτες που έγιναν στο παρελθόν, τον προσεκτικό σχεδιασμό του κάθε project (βιντεοπαιχνιδιού), την ανάπτυξη του καθενός με τα διάφορα μέσα που χρησιμοποιήθηκαν (engines, scripts κλπ.), τις διορθώσεις που πραγματοποιήθηκαν μέχρι το τελικό προϊόν, την αναπληροφόρηση και τα αποτελέσματα που παρατηρήθηκαν στο δοκιμαστικό κοινό και φυσικά τα τελικά συμπεράσματα και παρατηρήσεις.

Κεφάλαιο 1: Ανάλυση Βιβλιογραφίας

Η ιδέα για την παρούσα εργασία προήλθε από μια σειρά ερευνών που πραγματοποιήθηκαν κατά την διάρκεια των τελευταίων 10 χρόνων. Πιο συγκεκριμένα, μέσα από μια μελέτη στο Πανεπιστήμιο του Μόντρεαλ, τον Δεκέμβριο του 2017, οι επιστήμονες ανακάλυψαν την σύνδεση μεταξύ βιντεοπαιχνιδιών και μείωση των πιθανοτήτων για Alzheimer's. Πιο αναλυτικά, οι ερευνητές πήραν ένα δείγμα 33 ανθρώπων μεταξύ 55 και 75 ετών και το χώρισαν σε 3 διαφορετικές ομάδες. Μία ομάδα δεν έκανε τίποτα, η δεύτερη ομάδα μάθαινε πώς να παίζει πιάνο με την βοήθεια ενός προγράμματος υπολογιστή και η τρίτη ομάδα έπαιζε το βιντεοπαιχνίδι "Super Mario 64". Μετά το πέρας των 6 μηνών μόνο η ομάδα που έπαιζε το βιντεοπαιχνίδι έδειξε σημαντική αύξηση φαιάς ουσίας στον ιππόκαμπο του εγκεφάλου. Στη συνέχεια, η ομάδα που δεν έκανε τίποτα εμφάνισε έλλειψη φαιάς ουσίας καθώς οι άλλες 2 ομάδες είχαν αύξηση φαιάς ουσίας στην παρεγκεφαλίδα, μέρος του εγκεφάλου που έχει να κάνει με την μηχανική μνήμη (motor movements) και την βραχυχρόνια μνήμη. Ως αποτέλεσμα οι επιστήμονες μπόρεσαν να φτάσουν στην υπόθεση ότι παίζοντας 3D (τρισδιάστατα) βιντεοπαιχνίδια μπορεί να αυξηθεί η παραγωγή φαιάς ουσίας στον εγκέφαλο και να προληφθούν ασθένειες γήρανσης όπως το Alzheimer's. Ο λόγος είναι πολύ απλός. Τα τρισδιάστατα βιντεοπαιχνίδια αναγκάζουν τους παίκτες να είναι χωρικά ενήμεροι καθώς και να χτίζουν γνωστικούς χάρτες που πρέπει να θυμούνται. Είναι μία μεγάλη ανύψωση για τον εγκέφαλο.

Μία άλλη μελέτη από το Τμήμα Βιοϊατρικής του Διεθνές Ινστιτούτου Επιστημών Ζωής της Ινδονησίας, το Κέντρο Έρευνας Έξυπνης Γήρανσης (SARC) του Πανεπιστημίου Tohoku της Ιαπωνίας, το Τμήμα Γνωστικών Επιστημών Υγείας του Ινστιτούτου Ανάπτυξης, Γήρανσης και Καρκίνου (IDAC) του Πανεπιστημίου Tohoku της Ιαπωνίας και το Τμήμα Λειτουργικής Απεικόνισης Εγκεφάλου του Ινστιτούτου Ανάπτυξης, Γήρανσης και Καρκίνου (IDAC) του Πανεπιστημίου Tohoku της Ιαπωνίας το 2019 έδειξε ότι παίζοντας διάφορα βιντεοπαιχνίδια έχουμε ευεργετικά αποτελέσματα στη γνωστική λειτουργία και τον εγκέφαλο, εξετάζοντας διάφορα είδη βιντεοπαιχνιδιών όπως 3D Adventure, First-Person Shooting (FPS), Puzzle, Rhythm Dance και Strategy. Τα αποτελέσματα αυτής της συστηματικής επισκόπησης έδειξαν ότι παίζοντας βιντεοπαιχνίδια μπορεί να είναι επωφέλης για τον εγκέφαλο βοηθώντας σε περιπτώσεις έλλειψης βραχυχρόνιας μνήμης καθώς και έλλειψης συγκέντρωσης. Αυτό βέβαια εξαρτάται και από το είδος του εκάστοτε βιντεοπαιχνιδιού. Τα βιντεοπαιχνίδια αναφέρονται στην εμπειρία της ενασχόλησης με ηλεκτρονικά παιχνίδια, τα οποία ποικίλλουν από ενεργητικά σε παθητικά παιχνίδια, παρουσιάζοντας σε έναν παίκτη σωματικές και ψυχικές προκλήσεις. Το κίνητρο για να παίξει κάποιος βιντεοπαιχνίδια μπορεί να προέρχεται από την εμπειρία της αυτονομίας ή να ανταγωνιστεί με άλλους, κάτι που μπορεί να εξηγήσει γιατί τα βιντεοπαιχνίδια είναι ευχάριστα και εθιστικά. Τα βιντεοπαιχνίδια μπορούν να λειτουργήσουν ως «εκπαιδευτικοί» ανάλογα με τον σκοπό του παιχνιδιού. Τα βιντεοπαιχνίδια έχουν διαφορετικά αποτελέσματα ανάλογα με το είδος του παιχνιδιού. Για παράδειγμα, ένα ενεργό βιντεοπαιχνίδι μπορεί να βελτιώσει τη φυσική κατάσταση, ενώ τα κοινωνικά βιντεοπαιχνίδια μπορούν να βελτιώσουν την κοινωνική συμπεριφορά. Τα πιο ενδιαφέροντα αποτελέσματα δείχνουν ότι με το να παίζει κάποιος βιντεοπαιχνίδια μπορεί να αλλάξει τη γνωστική λειτουργία και τον εγκέφαλο. Παλαιότερες μελέτες έχουν δείξει ότι το παιχνίδι βιντεοπαιχνιδιών μπορεί να ωφελήσει τη γνώση. Μελέτες από διάφορους τομείς έχουν δείξει ότι η εμπειρία του βιντεοπαιχνιδιού σχετίζεται με καλύτερη γνωστική λειτουργία, ειδικά όσον αφορά την οπτική προσοχή και τη βραχυχρόνια μνήμη, τον χρόνο αντίδρασης και τη μνήμη εργασίας. Επιπλέον, ορισμένες τυχαιοποιημένες ελεγχόμενες μελέτες δείχνουν θετικά αποτελέσματα των παρεμβάσεων βιντεοπαιχνιδιών στη γνώση. Πρόσφατες μετα-αναλυτικές μελέτες έχουν επίσης υποστηρίξει τις θετικές επιδράσεις του

βιντεοπαιχνιδιού στη γνωστική λειτουργία. Αυτές οι μελέτες δείχνουν ότι το παιχνίδι βιντεοπαιχνιδιών παρέχει γνωστικά οφέλη. Τα αποτελέσματα της παρέμβασης βιντεοπαιχνιδιών συζητούνται όλο και περισσότερο μεταξύ των επιστημόνων. Πρέπει να γίνει ανασκόπηση των αποτελεσμάτων και της μεθοδολογικής ποιότητας των πρόσφατα δημοσιευμένων μελετών παρέμβασης. Έχει αναφερθεί μια συστηματική ανασκόπηση των βιντεοπαιχνιδιών και των νευρικών συσχετισμών. Ωστόσο, η τεχνική της νευροαπεικόνισης των αναθεωρημένων μελετών δεν ήταν συγκεκριμένη. Αυτή η συστηματική ανασκόπηση εξέτασε μόνο τις μελέτες μαγνητικής τομογραφίας (MRI) σε αντίθεση με την προηγούμενη συστηματική ανασκόπηση για να επικεντρωθεί στο αποτέλεσμα της νευροπλαστικότητας. Η νευροπλαστικότητα είναι η ικανότητα του εγκεφάλου να προσαρμόζεται για σκοπούς μάθησης, απομνημόνευσης και ανάκτησης. Στην κανονική προσαρμογή, ο εγκέφαλος προσαρμόζεται για να μάθει, να θυμηθεί, να ξεχάσει και να επιδιορθώσει τον εαυτό του. Πρόσφατες μελέτες με χρήση μαγνητικής τομογραφίας για τεχνικές απεικόνισης εγκεφάλου έχουν δείξει αποτελέσματα νευροπλαστικότητας μετά από μια παρέμβαση, η οποία περιλαμβάνει γνωστική άσκηση και μουσική εκπαίδευση σχετικά με την φαιά ουσία και τη λευκή ύλη. Ωστόσο, οι μοριακοί μηχανισμοί της αλλαγής της φαιάς και της λευκής ύλης παραμένουν ασαφείς. Οι προτεινόμενοι μηχανισμοί για την αλλαγή της φαιάς ύλης είναι η νευρογένεση, η γλοιογένεση, η συναπτογένεση και η αγγειογένεση, ενώ αυτοί για την αλλαγή της λευκής ύλης είναι η μοντελοποίηση και ο σχηματισμός μυελίνης, η οργάνωση των ινών και η αγγειογένεση. Πρόσφατες μελέτες που χρησιμοποιούν τεχνική μαγνητικής τομογραφίας για την απεικόνιση του εγκεφάλου έχουν δείξει τις επιδράσεις των βιντεοπαιχνιδιών στη νευροπλαστικότητα. Παλαιότερες μελέτες απεικόνισης που χρησιμοποιούν μεθόδους από διάφορους τομείς έδειξαν ότι η ενασχόληση με βιντεοπαιχνίδια επηρεάζει τη δομή του εγκεφάλου αλλάζοντας την φαιά ουσία, τη λευκή ύλη και τη λειτουργική συνδεσιμότητα. Επιπλέον, μερικές μελέτες παρέμβασης έχουν δείξει η ενασχόληση με βιντεοπαιχνίδια άλλαξε τη δομή και τις λειτουργίες του εγκεφάλου. Η συστηματική ανασκόπηση έχει πολλά συμπεράσματα σχετικά με ευεργετικά αποτελέσματα βιντεοπαιχνιδιών που δεν βασίζονται σε γνωστικά. Πρώτον, το βιντεοπαιχνίδι που δεν βασίζεται στη γνώση μπορεί να χρησιμοποιηθεί σε όλες τις ηλικιακές κατηγορίες ως μέσο βελτίωσης του εγκεφάλου. Ωστόσο, οι επιπτώσεις στα παιδιά παραμένουν

ασαφείς. Δεύτερον, το βιντεοπαιχνίδι που δεν βασίζεται στη γνώση επηρεάζει τόσο τις δομικές όσο και τις λειτουργικές πτυχές του εγκεφάλου. Τρίτον, τα αποτελέσματα ενασχόλησης με βιντεοπαιχνίδια παρατηρήθηκαν μετά από τουλάχιστον 16 ώρες προπόνησης. Τέταρτον, ορισμένα κριτήρια μεθοδολογίας πρέπει να βελτιωθούν για καλύτερη μεθοδολογική ποιότητα. Συμπερασματικά, η συστηματική ενασχόληση με βιντεοπαιχνίδια τουλάχιστον 16 ωρών είναι ευεργετική για τη λειτουργία και τη δομή του εγκεφάλου. Ωστόσο, αποτελέσματα στην περιοχή του εγκεφάλου διαφέρουν ανάλογα με τον τύπο του βιντεοπαιχνιδιού. Μπορείτε να δείτε τα αποτελέσματα των ερευνών στου παρακάτω επίσημους πίνακες.

Gaming for Health in Older Adults Literature Review Matrix

Refere nce	Yea r of stu dy	Theoreti cal foundati ons	Purpose of study	Measures of study	Study design	Setting of study (context)	Conte nt (physi cal, menta l, social)	Description of sample		
								Sample size	Mean age (years)	Gend er (%)
Broere n et al. ⁴⁴	200 8	Not stated	VR system used to assess and promote performa nce in persons affected by stroke	Perception of VR, manual ability (BBT and ABILHAND) , executive function and attention (TMT-B), kinematics (UE test)	Pre-/post- test design with a control group	Facility for commun ity- dwelling persons	Physi cal, menta l	22 (11 control, 11 treatme nt)	68.0	F (78%), M (22%)
Basak et al. ⁵⁵	200 8	Not stated	Assess executive control processe s of older adults	Cognitive ability, executive control, and visuospatia l skills	Pre-/post- test design with a control group	Laborato ry	Mental	39 (20 control, 19 treatme nt)	69.1 contro l, 70.1 treatm ent	F (74%), M (26%)

Gaming for Health in Older Adults Literature Review Matrix

Gaming for Health in Older Adults Literature Review Matrix										
Reference	Year of study	Theoretical foundations	Purpose of study	Measures of study	Study design	Setting of study (context)	Content (physical, mental, social)	Description of sample		
								Sample size	Mean age (years)	Gender (%)
Rosenberg et al. ³⁹	2010	Not stated	Assess feasibility, acceptability, and short-term efficacy and safety of exergames (combined game play with exercise)	Mood (QIDS), health-related QoL (MOS, SF-36), cognitive functioning (RBANS), adherence (personal logs).	Pilot, follow-up study	Residential facility, senior centers	Physical, mental	19 (all treatment participants)	78.7	F (68%), M (32%)
Studencki et al. ⁴²	2010	Not stated	Assess efficacy of adapted interactive dance videogame	Vital signs (pulse, BP, BMI), physical function (performance test), self-reported QoL	Single group, pre-/post-test design	Senior centers	Physical, mental	25 (all treatment participants)	80.1	F (83%), M (17%)
Brem et al. ⁴³	2010	Not stated	Assess cognitive performance due to videogames	Central information processing (Kurtest fur Allgemeine Intelligenz), health-related QoL (12-item short form health survey), mental stability (Neuroticis	10-day longitudinal study	Hospital (bedside)	Mental	32 (16 control, 16 treatment)	66.1 treatment, 68.9 control	F (62%), M (38%)

Gaming for Health in Older Adults Literature Review Matrix

Gaming for Health in Older Adults Literature Review Matrix										
								Description of sample		
Reference	Year of study	Theoretical foundations	Purpose of study	Measures of study	Study design	Setting of study (context)	Content (physical, mental, social)	Sample size	Mean age (years)	Gender (%)
				m-Extroversion-Openness-Five Factor Inventory)						
Hsu et al. ⁴⁶	2011	Not stated	Assess effects of adding simulate game play to standard exercise regimen for residents of long-term care facility	Physical performance (Nursing Home Physical Performance Test), physical activity enjoyment (PACES), pain intensity and pain bothersomeness (NRS), AROM, and perceived change (GPRC)	Randomized, single-blind crossover trial	Long-term care facility	Physical, mental	34 (19 Group I, 15 Group II)	80.0	F (71%), M (29%)
Peretz et al. ⁵⁶	2011	Not stated	Assess whether cognitive training can result in cognitive gains for older adults	Cognitive ability (measured by N-CPC) and Nexade®	Randomized double-blind interventional study	Hospital outpatient facility	Mental	121 (66 control, 55 treatment)	67.8	F (67%), M (33%)

Gaming for Health in Older Adults Literature Review Matrix

Gaming for Health in Older Adults Literature Review Matrix										
Reference	Year of study	Theoretical foundations	Purpose of study	Measures of study	Study design	Setting of study (context)	Content (physical, mental, social)	Description of sample		
								Sample size	Mean age (years)	Gender (%)
Torres ⁵⁷	2011	Not stated	Assess effects of videogames on the elderly pertaining to cognitive ability, self-concept, and QoL	Cognitive ability, cognitive portion of ADAS; self-concept, ICAC; QoL, WHOQOL	Pre-/post-test, 3 groups (experimental, passive control, and active control groups) design	Residential homes for elderly	Mental	43 (15 treatment, 17 passive control, 11 active control)	78.3	F (77%), M (23%)
Bell et al. ³⁸	2011	Not stated	Assess effects of Wii on quality of life, social relationships, and confidence in the ability to prevent falls	QoL, CASP-19; social relationships via Social Provisions Scale; confidence to prevent falls via M-FES	Repeated - measures design	Assisted living facilities	Physical, mental, social	21 (8 Group 1, 6 Group 2, 7 control)	80.8	F (76%), M (24%)
Williams et al. ⁴⁰	2011	Not stated	Assess benefits "Wii Fit" may have on balance for older adults	Balance is measured via the Berg Balance Scale	Single group pre-/post-test design	Independent retirement communities or skilled nursing facilities	Physical	22	83.9	F (82%), M (18%)
Bainbridge et al. ⁴⁶	2011	Not stated	Assess whether "Wii Fit" Balance	Balance, balance confidence, and limits	Prospective, cross-sectional pilot	College sport center	Physical, mental	6	75.0	F (87.5%), M (12.5)

Gaming for Health in Older Adults Literature Review Matrix

Gaming for Health in Older Adults Literature Review Matrix										
Reference	Year of study	Theoretical foundations	Purpose of study	Measures of study	Study design	Setting of study (context)	Content (physical, mental, social)	Description of sample		
								Sample size	Mean age (years)	Gender (%)
			Board leads to improve ments in balance in older adults with perceived balance deficit	of stability were assessed via Berg Balance Scale, Activities-specific Balance Confidence Scale, MDRT and COP excursion measurements.	study, pre-/post-test design		I			%)
Yamaguchi et al. ³⁷	2011	Not stated	Assess effects on cognitive function due to use of enjoyable sports videogames	General cognitive function measured using (HDS-R); visuospatial and constructive function measured using Kohs block-design tests; behavioral changes measured using MOSES	Single group pre-/post-test design	Nursing home	Physical, mental, social	9	88.9	F (66.6%), M (33.3%)
Szturm et al. ⁴¹	2011	Biofeedback training	Assess feasibility and benefits of	Berg Balance Scale, Timed "Up & Go" Test,	Randomized control trial	Geriatric hospital	Physical, mental	27 (13 control, 14 treatment)	NA; median given, inclusion	F (70.4%), M (29.6%)

Gaming for Health in Older Adults Literature Review Matrix

Gaming for Health in Older Adults Literature Review Matrix										
Refere nce	Yea r of stu dy	Theoreti cal foundati ons	Purpose of study	Measures of study	Study design	Setting of study (context)	Conte nt (physi cal, menta l, social)	Description of sample		
								Sample size	Mean age (years)	Gend er (%)

physical therapy via interactiv e videogam e paradigm	Activities- specific Balance Confidence Scale, Clinical Test of Sensory Interaction and Balance, all used to assess reactive balance controls and environme ntal interaction	nt)	on criteri on 65– 85 years	%)
--	---	-----	--	----

ABILHAND, Assessment of Manual Ability; ADAS, Alzheimer Disease Assessment Scale; AROM, active range of motion; BBT, Berg Balance Test; BMI, body mass index; BP, blood pressure; CASP-19, Control, Autonomy, Self-realization, Pleasure-19; COP, Center of Pressure; F, female; GPRC, Global Perceived Rating of Change; HDS-R, Hasegawa's Dementia Scale-revised; ICAC, Clinical Self-Concept Inventory; M, male; MDRT, Multi-Directional Reach Test; M-FES, Modified Falls Efficacy Scale; MOS, Medical Outcomes Study; MOSES, Multidimensional Observation Scale for Elderly Subjects; N, no; NA, not available; N-CPC, Neuropsychological Examination–“CogniFit® Personal Coach”; NRS, Numeric Rating Scale; QIDS, Quick Inventory of Depressive Symptoms; PACES, Physical Activity Enjoyment Scale; QoL, quality of life; RBANS, The Repeatable Battery for Assessment of Neurocognitive Status; SF-36, Short Form 36; TMT-B, The Trail Making Test, part B; UE, upper extremity; VR, virtual reality; WHOQOL, World Health Organization Quality of Life Questionnaire; Y, yes.

Game information							
Game title	Purpose of game	Tailored (Y/N)	Platform	Duration of play	Significant key health outcomes (physical, mental, social)	Duration of intervention (weeks)	Frequency
Not specifically stated	To manipulate virtual objects by using upper extremities	Yes	Phantom Omni	45 minutes	Physical: time and hand-path ratio for VR task ($P<0.05$)	4	3 times/week
"Rise of Nations"	Using strategy to control land	No	PC computers and eMAC	1.5 hours	Mental: ability to quickly switch between two tasks (sessions 2 and 3), $F(1, 34)=6.78$, $P=0.01$; improved focus of attention in the visual short-term memory task, $F(2, 16)=4.02$, $P=0.04$; and memory load cost decreased (session 3 versus session 2), $F(1, 36)=5.49$, $P=0.02$	4–5	15 sessions
"Wii Sports"	Physical movement to simulate actual games ("Bowling," "Baseball," "Tennis," "Golf," and "Boxing")	No	Nintendo Wii	35 minutes	Mental: QIDS-16, $t=3.24$, $P=0.004$, SF-36 Mental Health Composite Score $t=2.57$, $P=0.014$, and Cognitive Functioning (RBANS) $t=2.41$, $P=0.028$	12	3 times/week
"Dancetown"	Pressing correct arrow to the beat of music	Yes	Screen and dance pad	30 minutes	Physical: narrow walk time $P=0.0260$, Short Physical Battery Performance $P=0.0071$. Mental: self-reported health $P=0.02$, SF-36 Mental Component $P=0.0180$, self-reported balance confidence $P=0.0129$	12	2 times/week
"Dr. Kawashima's Brain"	Using cognitive ability to	No	Nintendo DS handheld	45 minutes	Mental: working memory increased $P=0.004$, fluid intelligence	1	7 sessions

Game information							
Game title	Purpose of game	Tailored (Y/N)	Platform	Duration of play	Significant key health outcomes (physical, mental, social)	Duration of intervention (weeks)	Frequency
Training: How Old Is Your Brain?"	correctly perform tasks		console		increased $P=0.006$, information processing increased $P=0.03$		
"Wii Bowling," part of "Wii Sports"	Swing to knock down pins, simulated bowling	No	Nintendo Wii	20 minutes	Physical: Nursing Home Physical Performance Test $P<0.001$, pain bothersomeness $P=0.015$, AROM $P=0.007$. Mental: Physical Activity Enjoyment Scale, enjoyment of activity $P=0.014$	8 (4 using Wii)	2 times/week
"CogniFit Personal Coach"	Complete 21 different training tasks for points	Yes	PC computers	20–30 minutes	Mental: focused attention $P=0.01$, sustained attention $P=0.01$, memory recognition $P=0.02$, mental flexibility $P=0.05$	12	3 times/week
"QBeez," "Super Granny 3," "ZooKeeper," "PenguinPush," "Bricks," and "Pingyn"	Problem solving, using work memory and recognition	No	PC computers	Not specified (unlimited play)	Mental: cognitive portion of ADAS $t(14)=3.505$, $P=0.003$, $r=0.68$	8	1 time/week
"Wii Bowling," part of "Wii Sports"	Simulate bowling game	No	Nintendo Wii	Not specified (until 10 frames completed)	Social: CASP-19, Group 2 item 8, $P=0.030$; Social Provisions Scale, Group 3 item 1, $P=0.030$; Group 2 item 3, $P=0.045$, Group 1 item 14, $P=0.048$. Mental: M-FES, Group 2 item 8, $P=0.047$	8	1 time/week
"Wii Fit" Balance Board	Completing balance activities by moving forward, backward, and	No	Nintendo Wii	20 minutes	Physical: Berg Balance Scale $t(21)=-9.861$, $P<0.01$	4	3 times/week

Game information							
Game title	Purpose of game	Tailored (Y/N)	Platform	Duration of play	Significant key health outcomes (physical, mental, social)	Duration of intervention (weeks)	Frequency
	side to side on Wii Balance Board						
"Wii Fit" Balance Board	Completing balance activities by moving forward, backward, and side to side on Wii Balance Board	No	Nintendo Wii	30 minutes	Physical and mental outcome measures were not found to be statistically significant for Berg Balance Scale and Activities-specific Balance Confidence Scale (3 subjects decreased their fall risks by 6–12%).	6	2 times/week
"XaviX Hot-plus"	Grab coin when it appears to float out of screen or moving legs to music, both to score points	Yes	Television screen and tangible accessories such as soft bowling balls	Not specified	Physical: visuospatial and constructive function measured using Kohs block-design tests, $P=0.02$. Mental: HDS-R, $P=0.002$. Social: MOSES, $P=0.054$	10	1 time/week
"Under Pressure," "CMemory Match," "Balloon Burst"	Move weight to get object directed in right spot, or user shifts weight in order to select correct card, or user shifts weight in order to burst	Yes	FSA Pressure mat connected to laptop computer	45 minutes	Physical: Berg Balance Scale, $P=0.001$; Clinical Test of Sensory Interaction and Balance, Loss of Balance sponge surface, $P=0.007$. Mental: Activities-specific Balance Confidence Scale, $P=0.02$	8	2 times/week

Game information							
Game title	Purpose of game	Tailor ed (Y/N)	Platform	Duration of play	Significant key health outcomes (physical, mental, social)	Duration of intervention (weeks)	Frequen cy
balloon.							

Διαβάζοντας και αναλύοντας τις παραπάνω έρευνες και έχοντας ήδη την τάση ενασχόλησης με την ανάπτυξη ψυχαγωγικού λογισμικού και εικονικών κόσμων κατέληξα στην θεματική της εργασίας η οποία θα αποτελείται από 3 διαφορετικά βιντεοπαιχνίδια, όπου το καθένα θα στοχεύει στην καταπολέμηση μίας ή περισσότερων νοητικών παθήσεων, και φυσικά την γραπτή αναφορά.

Κεφάλαιο 2: Σχεδιασμός

Πριν ξεκινήσει η ανάπτυξη του οποιουδήποτε λογισμικού, πρώτα έπρεπε να γίνει σωστός σχεδιασμός σε όλες τις πτυχές του. Έπρεπε αρχικά να γίνει η επιλογή του είδους του κάθε βιντεοπαιχνιδιού. Να μην είναι ιδιαίτερα πολύπλοκο για νέους χρήστες και άτομα εκτός του κοινού των βιντεοπαιχνιδιών και κυρίως το περιβάλλον να είναι όσο περισσότερο φιλικό σε άτομα που πάσχουν από επιληπτικές κρίσεις. Έτσι λοιπόν κατέληξα σε 3 διαφορετικά σενάρια για το κάθε βιντεοπαιχνίδι. Το πρώτο λογισμικό θα αποτελούσαν από ένα 3D βιντεοπαιχνίδι οδήγησης στο οποίο ο παίκτης καλείται να αποφεύγει τα εμπόδια και να φτάσει στον τερματισμό, αναγκάζοντάς τον να σχεδιάσει έναν γνωστικό χάρτη έτσι ώστε να θυμάται από ποια μέρη μπορεί να αποφύγει τα εμπόδια και από ποια μέρη είναι σχεδόν αδύνατο να περάσει για να φτάσει στον τερματισμό. Το λογισμικό αυτό αφοσιώνεται σε άτομα με έλλειψη συγκέντρωσης και άλλες παθήσεις που αφορούν την έλλειψη συντονισμού και συγχρονισμού καθώς και

για άτομα με παθήσεις που αφορούν την έλλειψη βραχυχρόνιας και μακροχρόνιας μνήμης. Το δεύτερο λογισμικό θα αποτελούνταν από ένα στατικό 2D βιντεοπαιχνίδι στο οποίο ο παίκτης καλείται να περάσει από μία τυχαία οπή μίας πλευράς ενός εξαγώνου χωρίς να συγκρουστεί με της υπόλοιπες πλευρές του καθώς το κάθε εξαγώνο κλείνει όλο και περισσότερο στον παίκτη. Το λογισμικό αυτό είναι σχεδιασμένο για άτομα με έλλειψη συγκέντρωσης και άλλες παθήσεις που αφορούν την έλλειψη συντονισμού και συγχρονισμού. Το τρίτο και τελευταίο λογισμικό θα αποτελούνταν από ένα puzzle παιχνίδι καρτών μνήμης. Ο παίκτης πρέπει να βρει όλα τα κοινά ζεύγη καρτών για να τερματίσει το παιχνίδι. Το λογισμικό αυτό είναι σχεδιασμένο για άτομα με νοητικές παθήσεις που αφορούν την έλλειψη βραχυχρόνιας και μακροχρόνιας μνήμης.

Έχοντας λοιπόν μια γενική εικόνα του κάθε project μπορούσα να προχωρήσω στην ανάπτυξη λογισμικού.

Κεφάλαιο 3: Ανάπτυξη

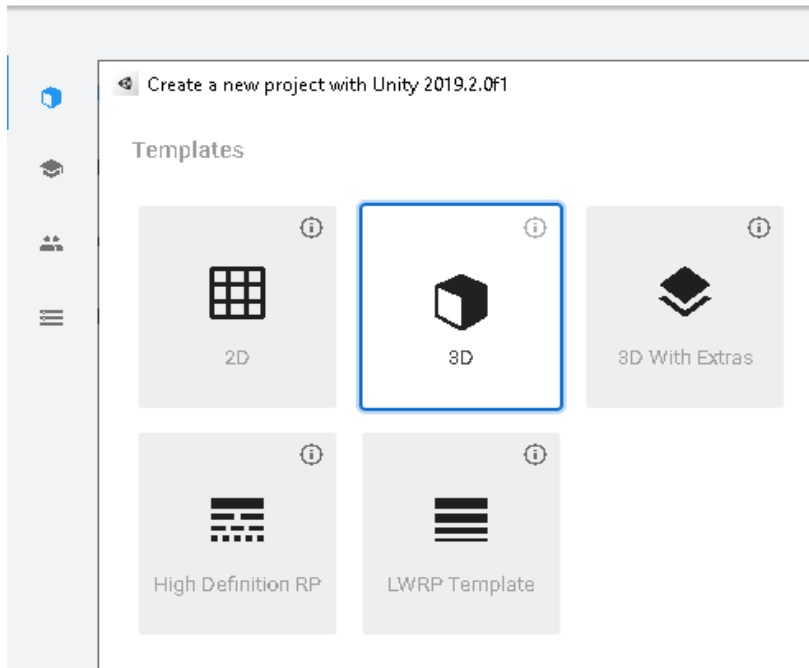
Για την ανάπτυξη του κάθε βιντεοπαιχνιδιού χρησιμοποιήθηκε η Unity Engine, το Microsoft Visual Studio 2019 καθώς και η χρήση scripts σε γλώσσα προγραμματισμού C#.

Η Unity είναι μια μηχανή παιχνιδιών πολλαπλών πλατφορμών που αναπτύχθηκε από την Unity Technologies, η οποία ανακοινώθηκε και κυκλοφόρησε για πρώτη φορά τον Ιούνιο του 2005 στο Worldwide Developers Conference της Apple Inc. ως αποκλειστική μηχανή παιχνιδιών για Mac OS X. Από το 2018, η μηχανή έχει επεκταθεί ώστε να υποστηρίζει περισσότερες από 25 πλατφόρμες. Η μηχανή μπορεί να χρησιμοποιηθεί για τη δημιουργία τρισδιάστατων βιντεοπαιχνιδιών, δισδιάστατων βιντεοπαιχνιδιών, βιντεοπαιχνιδιών εικονικής πραγματικότητας και παιχνιδιών επαυξημένης πραγματικότητας, καθώς και προσομοιώσεων και άλλων εμπειριών. Η μηχανή Unity έχει υιοθετηθεί από βιομηχανίες εκτός βιντεοπαιχνιδιών, όπως ταινιών, αυτοκινήτων, αρχιτεκτονικής, μηχανικής και κατασκευών.

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων υπολογιστών, καθώς και ιστότοπων, εφαρμογών ιστού, υπηρεσιών ιστού και εφαρμογών για κινητά. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft, όπως Windows API, Windows Forms, Windows Presentation Foundation, Windows Store και Microsoft Silverlight. Μπορεί να παράγει τόσο εγγενή κώδικα όσο και διαχειριζόμενο κώδικα. Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει στον code editor και το πρόγραμμα εντοπισμού σφαλμάτων να υποστηρίζει (σε διαφορετικούς βαθμούς) σχεδόν οποιαδήποτε γλώσσα προγραμματισμού, υπό την προϋπόθεση ότι υπάρχει μια υπηρεσία για συγκεκριμένη γλώσσα. Οι ενσωματωμένες γλώσσες περιλαμβάνουν C, C ++, C ++ / CLI, Visual Basic .NET, C #, F #, JavaScript, TypeScript, XML, XSLT, HTML και CSS. Η υποστήριξη για άλλες γλώσσες, όπως Python, Ruby, Node.js και M, μεταξύ άλλων διατίθεται μέσω προσθηκών. Η Java (και J #) υποστηρίχθηκαν στο παρελθόν. Στην προκειμένη περίπτωση χρησιμοποιήσα scripts σε C #.

Η C# (C Sharp) είναι μια γλώσσα προγραμματισμού H/Y. Δημιουργήθηκε από την Microsoft μέσα από την πλατφόρμα .NET και αργότερα αναγνωρίστηκε επισήμως από την ECMA και την ISO. Είναι μια απ τις γλώσσες προγραμματισμού που δημιουργήθηκαν για την Κοινή Υποδομή Γλώσσας (αγγλικά: Common Language Infrastructure). Ο κύριος σκοπός της γλώσσας είναι να είναι απλή αντικειμενοστρεφής γλώσσα για γενική χρήση. Ο διοικητής της ομάδας που διαχειρίζεται την γλώσσα ονομάζεται Άντερς Χάιλσμπεργκ. Στις 15 Αυγούστου 2012 κυκλοφόρησε η έκδοση 5.0 η οποία είναι η πιο πρόσφατη μέχρι σήμερα.

Αρχικά, ανοίγοντας το Unity πρέπει να επιλέξουμε τον τύπο του βιντεοπαιχνιδιού που θέλουμε να αναπτύξουμε αναλόγως βέβαια τις διαστάσεις του αλλά και τις δυνατότητες που θέλουμε να μας παρέχει. Όπως φαίνεται στην παρακάτω εικόνα.



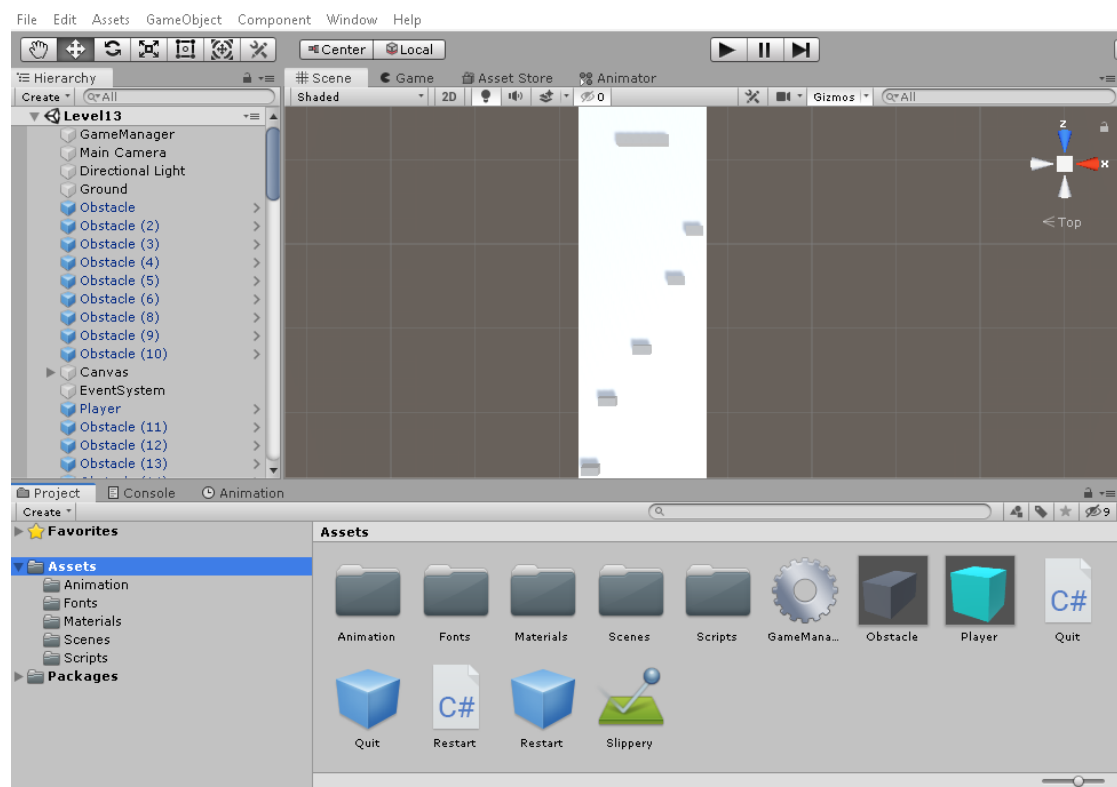
Στη συνέχεια αφού το Unity φορτώσει τα απαραίτητα assets μας μετακινεί σε ένα άδειο περιβάλλον στο οποίο μπορούμε να ξεκινήσουμε να δουλεύουμε το project μας.

Κεφάλαιο 4: Projects

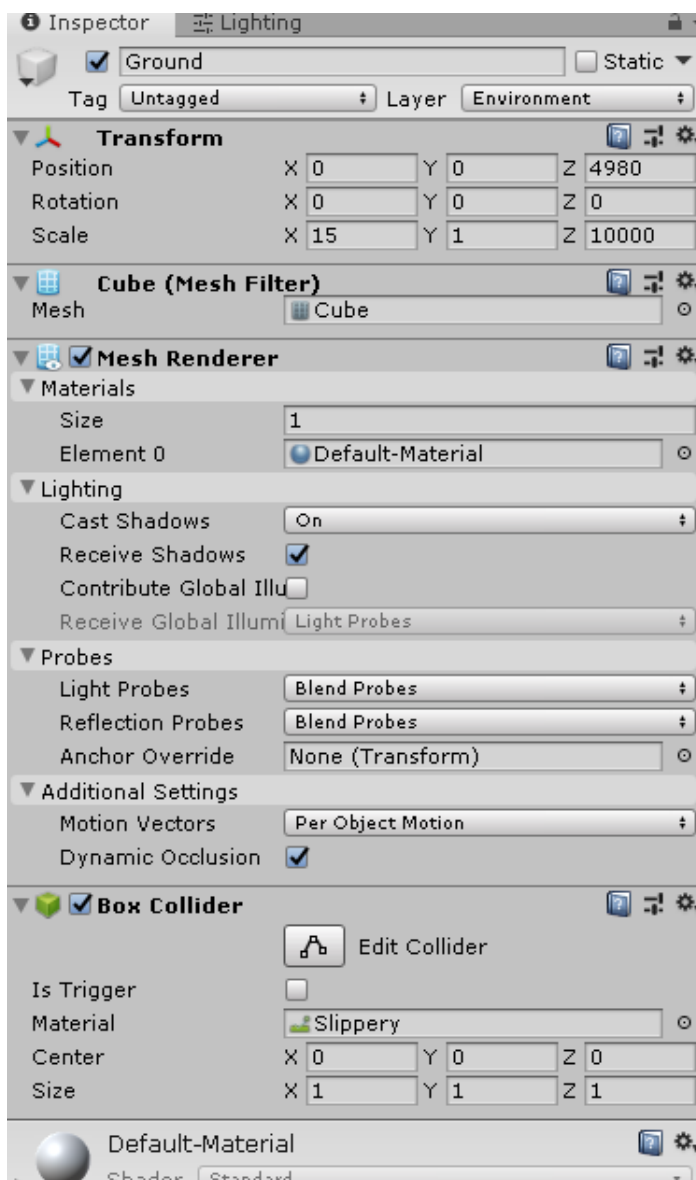
Project #1: Cubeathlon

Το Cubeathlon είναι ένα 3D παιχνίδι. Ο χειρισμός πραγματοποιείται με τα πλήκτρα “a” και “d” για την μετακίνηση του παίκτη αριστερά και δεξιά αντίστοιχα. Ο παίκτης έχει μία αρχική κινητήρια δύναμη και πρέπει να αποφύγει όλα τα εμπόδια χωρίς να φύγει εκτός της πίστας και να φτάσει στον τερματισμό. Αποτελείται από 13 επίπεδα αυξανόμενης δυσκολίας καθώς η απόσταση μεταξύ των εμποδίων μειώνεται. Όπως προαναφέρθηκε η εφαρμογή αυτή απευθύνεται σε άτομα με έλλειψη συγκέντρωσης και συντονισμού καθώς και σε άτομα με παθήσεις που αφορούν την έλλειψη βραχυχρόνιας μνήμης.

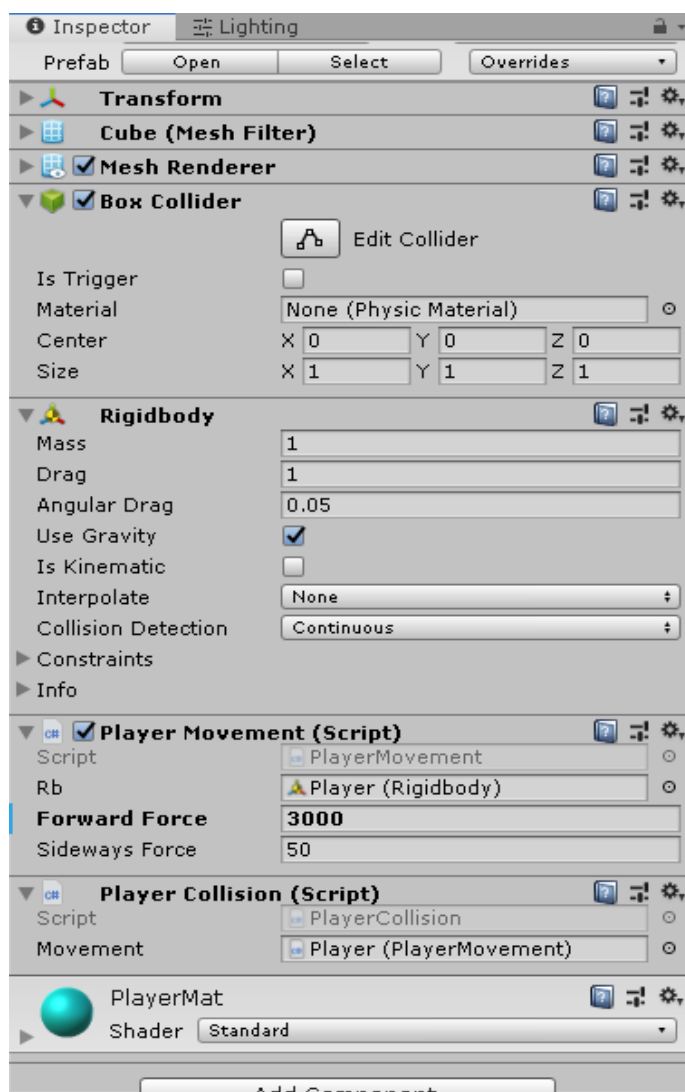
Ας δούμε όμως από τι αποτελείται αυτό το βιντεοπαιχνίδι. Στην παρακάτω εικόνα παρουσιάζονται τα assets από τα οποία αποτελείται καθώς και οι υποφάκελοι που περιέχουν όλα τα scenes του παιχνιδιού, τα scripts, τα animation και τα materials.



Αρχίζοντας από τα βασικά. Το object “Ground” αποτελείται ουσιαστικά από ένα 3D Cube object του οποίου άλλαξα τις διαστάσεις και το μετέτρεψα στο δάπεδο πάνω στο οποίο βρίσκεται ο παίκτης και τα εμπόδια. Όπως μπορείτε να δείτε παρακάτω, οι διαστάσεις του έχουν αλλάξει ως προς τον άξονα Z όσων αφορά την θέση και το μέγεθος, καθώς το μήκος του δαπέδου είναι 15 και το ύψος 1. Στη συνέχεια θα παρατηρήσετε ότι στο πεδίο Box Collider και στην υποκατηγορία Material έχω επιλέξει την επιλογή Slippery. Ο λόγος πίσω από αυτή την επιλογή είναι ότι ο παίκτης έχει μία αρχική ταχύτητα την οποία θέλω να διατηρήσει χωρίς αυξομειώσεις λόγο των physics. Έτσι ο παίκτης πάντα κινείται με μία σταθερή ταχύτητα όπως θα δούμε στη συνέχεια.

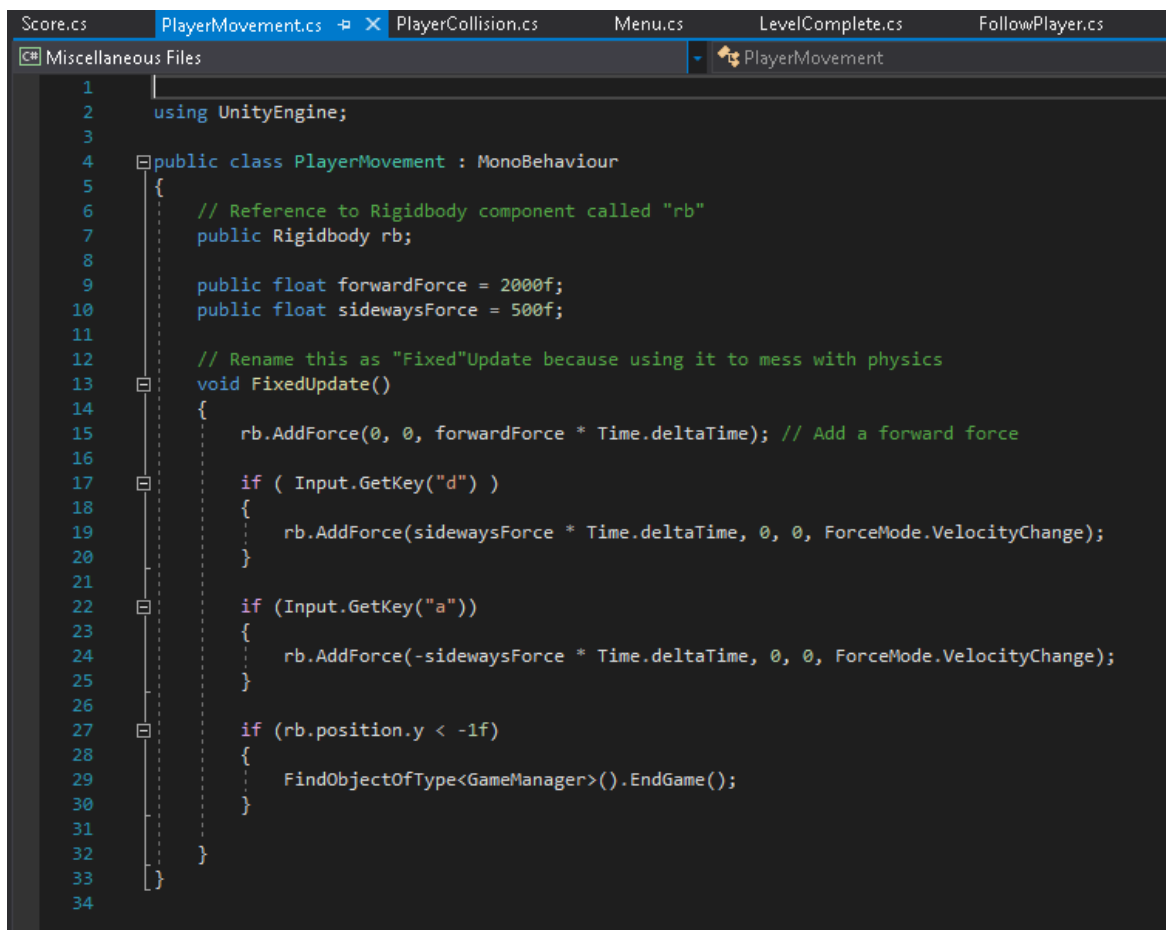


Στη συνέχεια έχουμε το object Player το οποίο είναι ο παίκτης που χειριζόμαστε. Αποτελείται από ένα 3D Cube object σε διαστάσεις 1x1x1. Μπορούμε να δούμε στην παρακάτω εικόνα ότι πέρα από τα αρχικά assets που του δίνει το Unity έχει προστεθεί το Rigidbody το οποίο του δίνει τα απαραίτητα physics για την σύγκρουση με άλλα σώματα, βαρύτητα, μάζα κλπ. καθώς και 2 scripts σε C#, το “Player Movement” και το “Player Collision” τα οποία θα δούμε στη συνέχεια. Επίσης προστέθηκε ένα διαφορετικό material χρώματος σιέλ για να ξεχωρίζει ο παίκτης από το περιβάλλον.



Το script “Player Movement” όπως λέει και το όνομά του αφορά την κίνηση του παίκτη. Αρχικά δημιουργούμε μία αναφορά στο component Rigidbody που αναφέρθηκε παραπάνω και δίνουμε μία εμπρόσθια κινητήρια δύναμη στον παίκτη και στη συνέχεια

μία πλάγια. Στη συνέχεια δημιουργούμε μία void ονομαζόμενη FixedUpdate για να μας δώσει την επιλογή να αλλάξουμε τις δύο δυνάμεις μέσα από το Unity όπως μπορείτε να παρατηρήσετε στην προηγούμενη εικόνα στο “Player Movement (Script)”. Έπειτα πρέπει με την επιλογή Input.GetKey δίνουμε την εντολή με το κουμπί “d” ο παίκτης να κινείται δεξιά και με το κουμπί “a” αριστερά. Τέλος βάζουμε συνθήκη έτσι ώστε αν ο κύβος, δηλαδή ο παίκτης, είναι σε σημείο κάτω απ’το σημείο του object ground, δηλαδή βγει εκτός του δαπέδου, να τελειώνει το παιχνίδι. Παρακάτω μπορείτε να δείτε το script αυτούσιο.



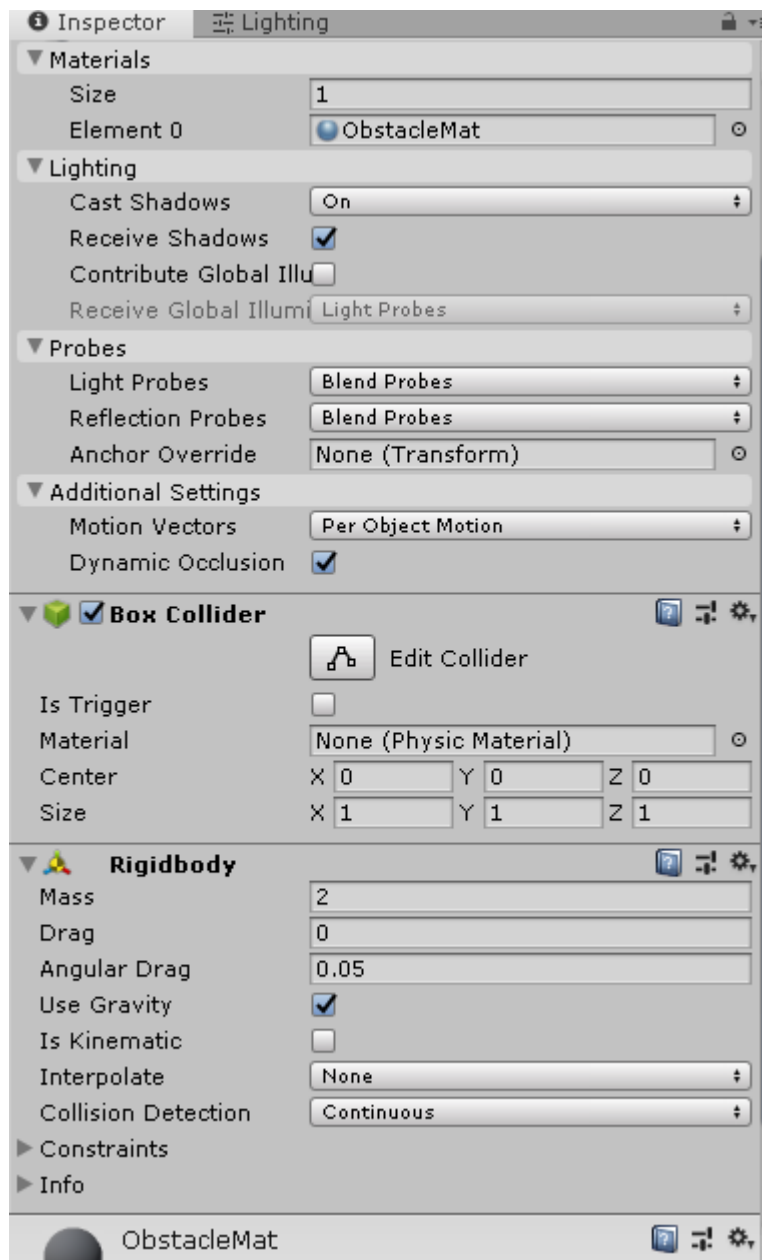
```
1 using UnityEngine;
2
3
4 public class PlayerMovement : MonoBehaviour
5 {
6     // Reference to Rigidbody component called "rb"
7     public Rigidbody rb;
8
9     public float forwardForce = 2000f;
10    public float sidewaysForce = 500f;
11
12    // Rename this as "Fixed"Update because using it to mess with physics
13    void FixedUpdate()
14    {
15        rb.AddForce(0, 0, forwardForce * Time.deltaTime); // Add a forward force
16
17        if ( Input.GetKey("d") )
18        {
19            rb.AddForce(sidewaysForce * Time.deltaTime, 0, 0, ForceMode.VelocityChange);
20        }
21
22        if (Input.GetKey("a"))
23        {
24            rb.AddForce(-sidewaysForce * Time.deltaTime, 0, 0, ForceMode.VelocityChange);
25        }
26
27        if (rb.position.y < -1f)
28        {
29            FindObjectOfType<GameManager>().EndGame();
30        }
31    }
32 }
33
34
```

Το script “Player Collision”, για άλλη μια φορά, όπως λέει και το όνομά του αναφέρεται στην σύγκρουση του παίκτη με τα εμπόδια που υπάρχουν στο παιχνίδι. Αρχικά βάζουμε μια public, δηλαδή μία αναφορά στο προηγούμενο script που αφορούσε την κίνηση του παίκτη και στη συνέχεια μία συνάρτηση που ονομάζουμε OnCollisionEnter η οποία θα

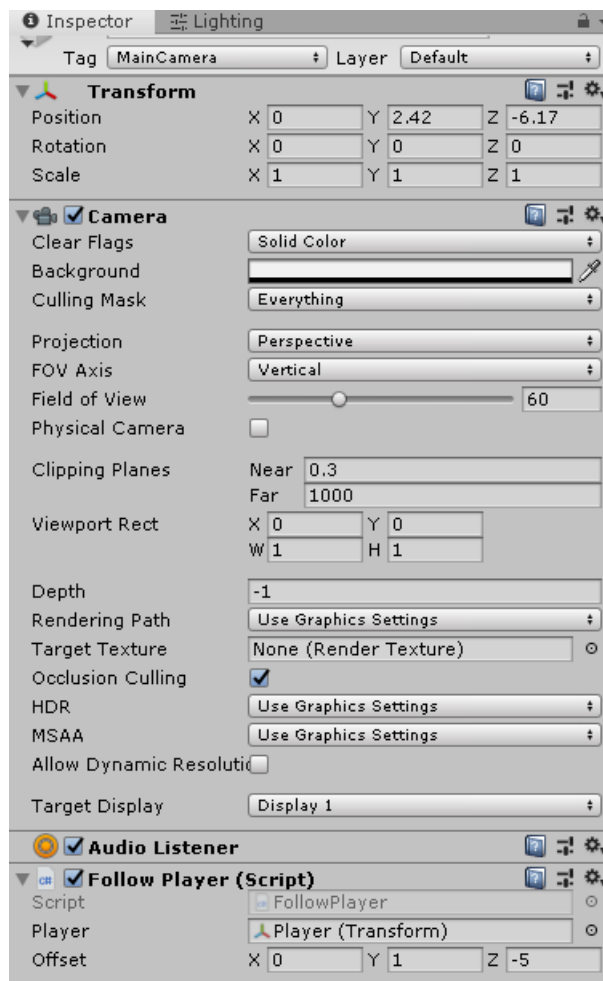
«τρέχει» όταν ο παίκτης έρχεται σε επαφή με κάποιο από τα εμπόδια. Έτσι θέτουμε την συνθήκη ότι αν ο παίκτης συγκρουστεί με ένα object που έχει το tag "Obstacle" να απενεργοποιήσει την κίνηση του παίκτη και στη συνέχεια να τελειώσει το παιχνίδι. Παρακάτω μπορείτε να δείτε το τελικό script.

```
1 using UnityEngine;
2
3 public class PlayerCollision : MonoBehaviour
4 {
5     public PlayerMovement movement; // Reference to PlayerMovement script
6
7     // Function runs when hit another object
8     // Get information about the collision and call it "collisionInfo"
9     void OnCollisionEnter(Collision collisionInfo)
10    {
11        // Check if object we collided with has a tag called "Obstacle"
12        if (collisionInfo.collider.tag == "Obstacle")
13        {
14            movement.enabled = false; //Disables player movement
15            FindObjectOfType<GameManager>().EndGame();
16        }
17    }
18 }
19
20
21
```

Μιας και αναφέραμε τα εμπόδια, ας δούμε και από τι αποτελούνται. Το object "Obstacle" αποτελείται από ένα 3D Cube Object του οποίου έχουμε αλλάξει τις διαστάσεις και το έχουμε κάνει Prefab, δηλαδή μπορούμε να το χρησιμοποιούμε αυτούσιο κάθε φορά στο σημείο που θέλουμε στο Unity κάνοντας το Duplicate πατώντας Ctrl+D. Δεν χρειάζεται δηλαδή για 2 εμπόδια να δημιουργήσουμε 2 διαφορετικά objects. Στο tag έχουμε βάλει τον τίτλο "Obstacle" έτσι ώστε να μπορεί να γίνεται αντιληπτό από το script που χρησιμοποιήσαμε παραπάνω και φυσικά του προσθέσαμε τα assets Box Collider για την σύγκρουση με άλλα αντικείμενα και Rigidbody για τα physics. Τέλος προστέθηκε ένα διαφορετικό material χρώματος σκούρο γκρι για να μπορεί να ξεχωρίζει από το λευκό δάπεδο.



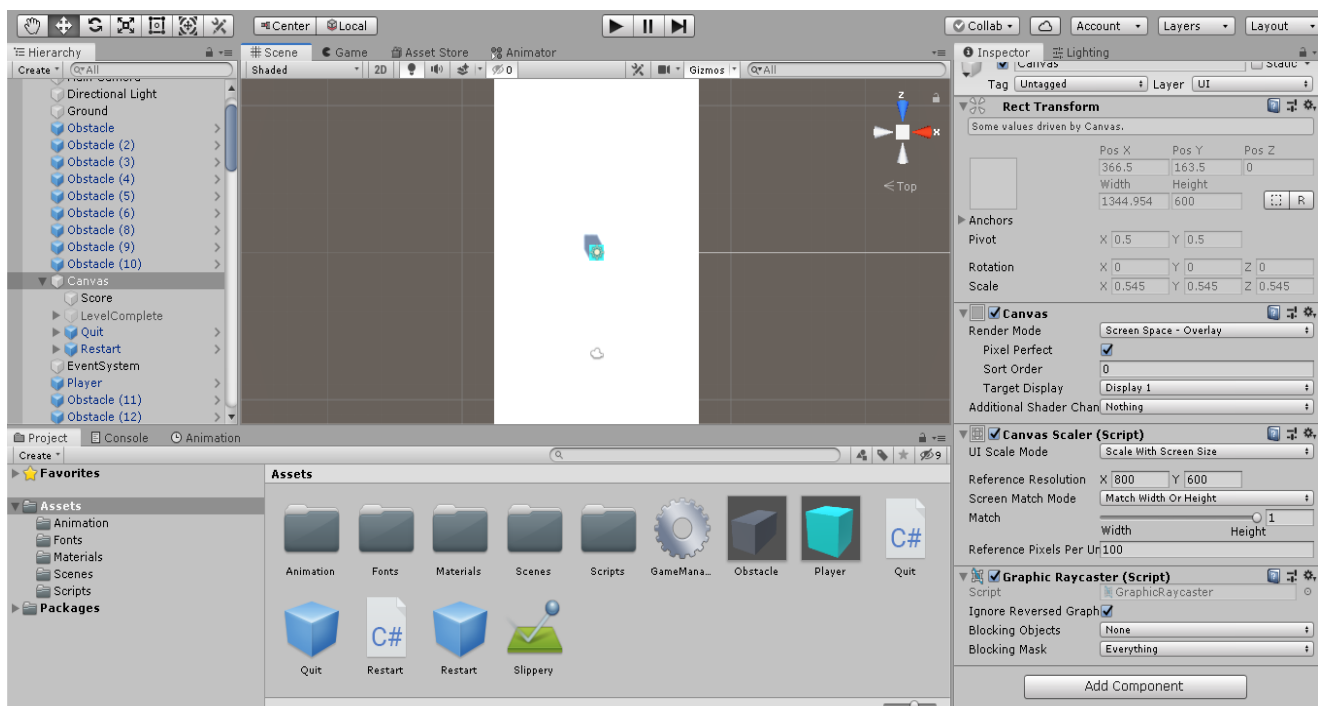
Ένα πολύ βασικό κομμάτι σε κάθε βιντεοπαιχνίδι είναι η κάμερα. Δηλαδή την οπτική που θα έχει ο χρήστης μέσα στον εκάστοτε εικονικό κόσμο. Για το συγκεκριμένο project επέλεξα η κάμερα να ακολουθεί τον παίκτη δίνοντάς του την οπτική του τρίτου προσώπου. Όπως μπορούμε να δούμε στα assets της κάμερας παρακάτω, έχει προστεθεί ένα script για να επιτευχθεί αυτό ονομαζόμενο “Follow Player” για ευνόητους λόγους.



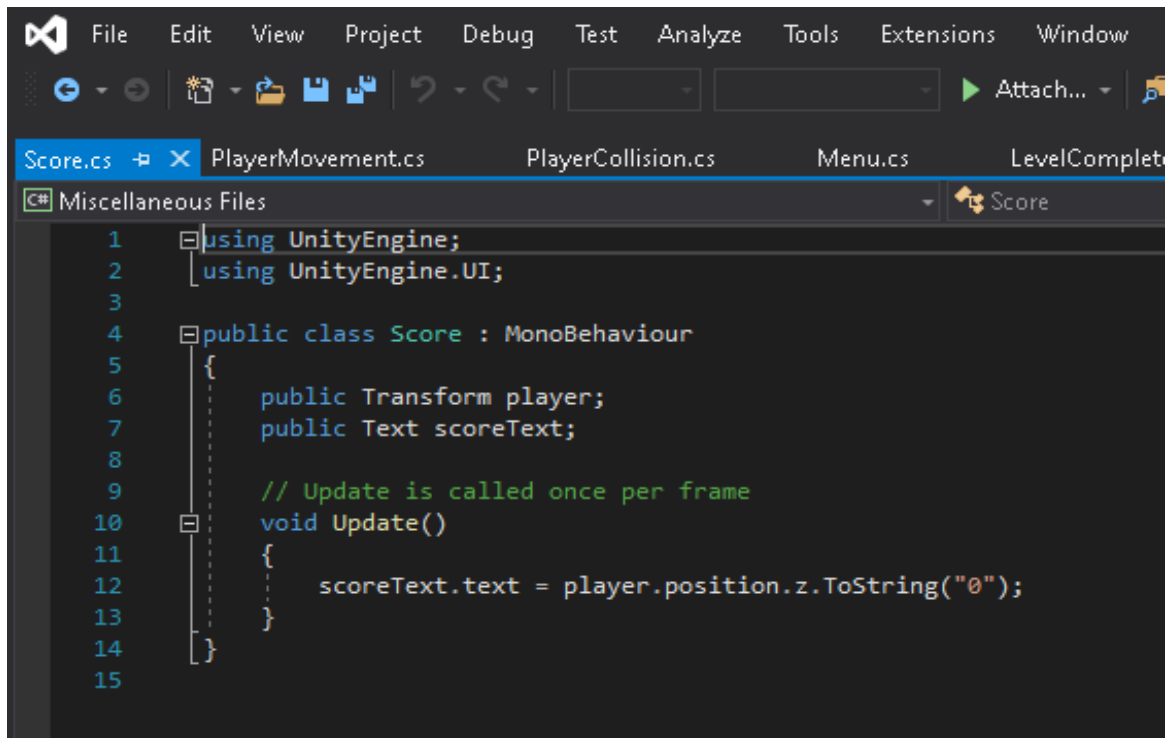
Όπως μπορούμε να δούμε στο script αυτό βάζουμε 2 αναφορές. Η πρώτη αφορά την θέση ποιου αντικειμένου θα ακολουθεί η κάμερα και η δεύτερη αφορά την τοποθέτηση της κάμερας. Προφανώς στην πρώτη αναφορά έχουμε βάλει το Player object που έχουμε το οποίο αναφέρεται στον παίκτη και στην δεύτερη περίπτωση έχουμε ρυθμίσει τις διαστάσεις στις οποίες τα βρίσκεται η κάμερα όπως μπορούμε να δούμε στην παραπάνω εικόνα στο Follow Player (Script). Με αυτή την επιλογή η κάμερα είναι 1 μέτρο πάνω από τον παίκτη και 5 μέτρα πίσω. Στον άξονα X είναι 0 καθώς θέλουμε η κάμερα να είναι ακριβώς πίσω από τον παίκτη. Τέλος με την εντολή transform.position προσθέτουμε την θέση που είναι ο παίκτης κάθε φορά με το offset που έχουμε ορίσει και έτσι η κάμερα ακολουθεί τον παίκτη συνεχώς. Παρακάτω μπορείτε να δείτε και το αντίστοιχο script.

```
Score.cs PlayerMovement.cs PlayerCollision.cs Menu.cs LevelComplete.cs FollowPlayer.cs EndTrigger.cs
Miscellaneous Files FollowPlayer
1 using UnityEngine;
2
3 public class FollowPlayer : MonoBehaviour
4 {
5     public Transform player;
6     public Vector3 offset;
7
8     // Update is called once per frame
9     void Update()
10    {
11        transform.position = player.position + offset;
12    }
13
14 }
15
```

Στη συνέχεια έχουμε το Canvas πάνω στο οποίο υπάρχουν όλα τα texts του παιχνιδιού, το Score, η ολοκλήρωση του κάθε επιπέδου καθώς και τα κουμπιά Quit και Restart.



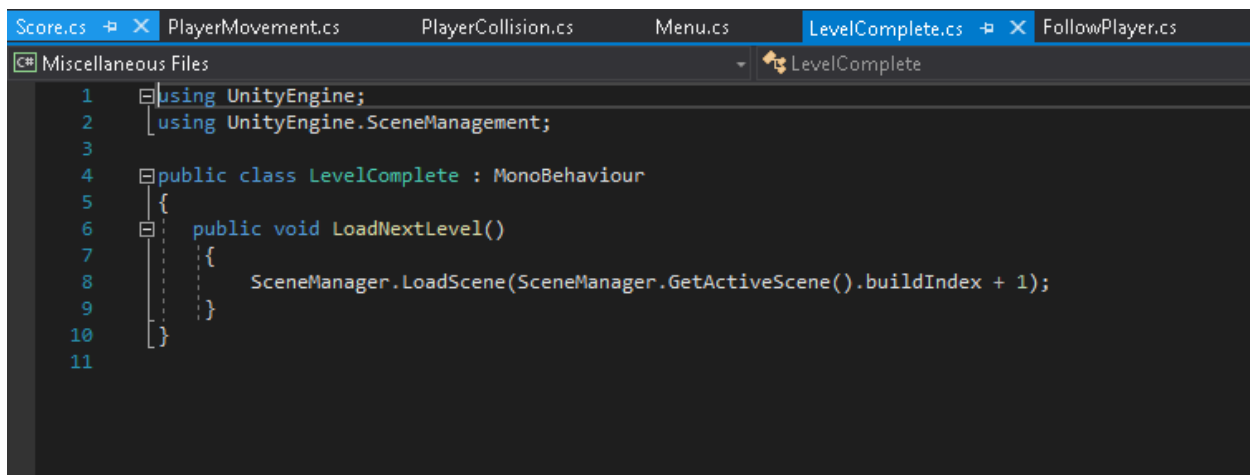
Το Score Asset είναι ουσιαστικά ένα text σε font Roboto το οποίο μετράει την απόσταση που έχει διανύσει ο παίκτης στο κάθε επίπεδο. Μέσα στο Score υπάρχει άλλο ένα script σε C# ονομαζόμενο Score. Αρχικά πρέπει να γίνουν δύο αναφορές. Μία για το αντικείμενο το οποίο θα διανύει την απόσταση και μία για την τιμή. Στη συνέχεια βάζουμε μία void η οποία καλείται σε κάθε frame και το text παίρνει την τιμή της θέσης του παίκτη. Έτσι το text δείχνει πάντα την θέση του παίκτη. Παρακάτω μπορείτε να δείτε το αντίστοιχο script.



The screenshot shows the Visual Studio IDE with the 'Score.cs' script open. The script is a C# MonoBehaviour class that uses UnityEngine and UnityEngine.UI. It contains a public Transform property 'player' and a public Text property 'scoreText'. The 'Update' method is implemented to update the score text with the player's position.z value formatted as a string with a leading zero.

```
1 using UnityEngine;
2 using UnityEngine.UI;
3
4 public class Score : MonoBehaviour
5 {
6     public Transform player;
7     public Text scoreText;
8
9     // Update is called once per frame
10    void Update()
11    {
12        scoreText.text = player.position.z.ToString("0");
13    }
14 }
15
```

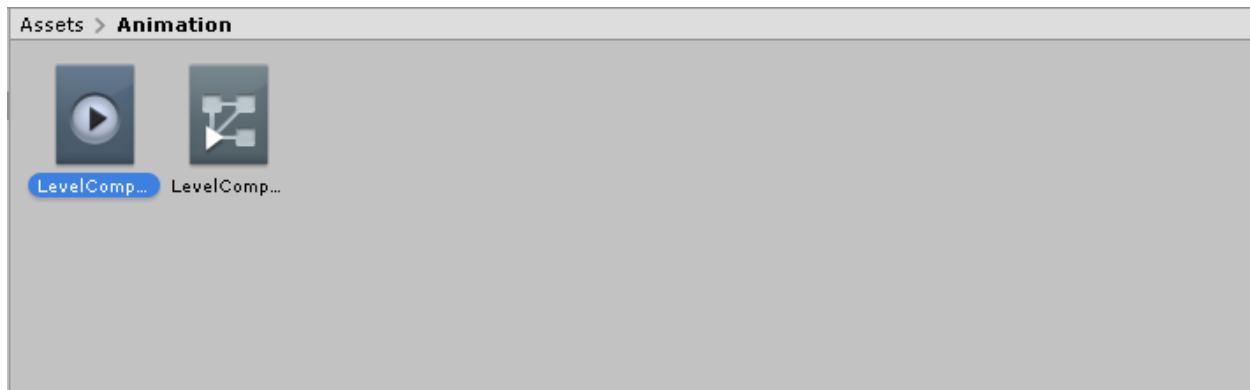
Το Level Complete asset αποτελείται από το text του, το animation και φυσικά το script σε C#. Το text του αποτελείται από τις λέξεις "Level Complete!" όταν βέβαια ο παίκτης φτάνει στον τερματισμό του κάθε επιπέδου. Όπως μπορούμε να δούμε στο παρακάτω script όταν συμβεί αυτό το SceneManager φορτώνει την επόμενη πίστα όπως βλέπουμε αναλυτικά στο παρακάτω script.



The screenshot shows the Visual Studio IDE with the 'LevelComplete.cs' script open. The script is a C# MonoBehaviour class that uses UnityEngine and UnityEngine.SceneManagement. It contains a public void method 'LoadNextLevel()' which calls SceneManager.LoadScene with the build index of the active scene plus one.

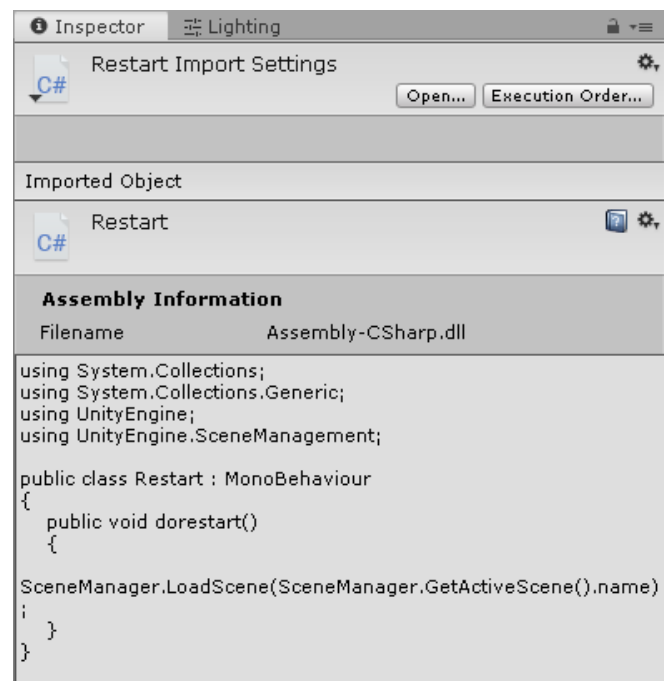
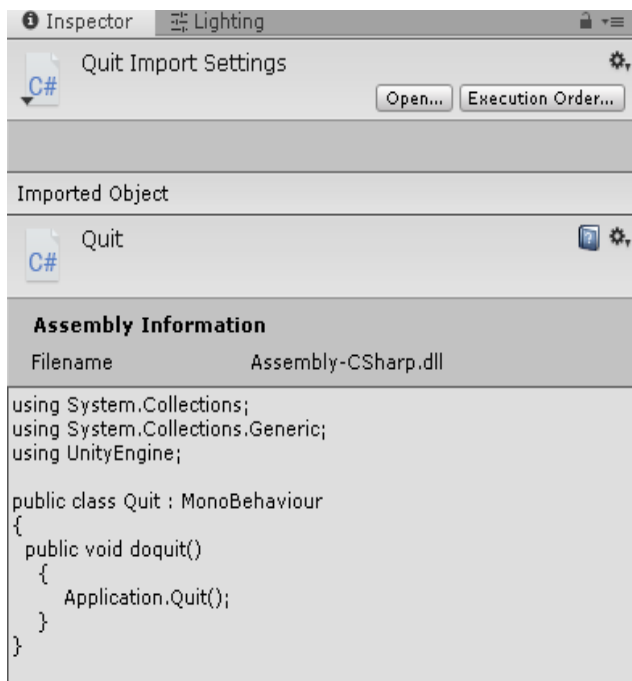
```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class LevelComplete : MonoBehaviour
5 {
6     public void LoadNextLevel()
7     {
8         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
9     }
10 }
11
```

Τέλος το animation του Level Complete κάνει άσπρη την οθόνη όταν ο παίκτης φτάσει στον τερματισμό και εμφανίζει φυσικά τις λέξεις “Level Complete!” και στη συνέχεια φορτώνει το επόμενο επίπεδο που έχουμε βάλει στην μηχανή.



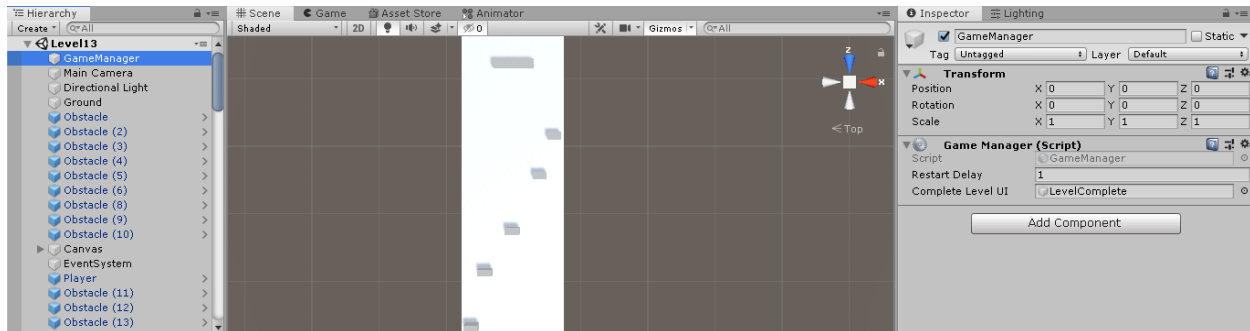
Τα assets Quit και Restart είναι δύο Buttons στο Canvas τα οποία αποτελούνται από τα αντίστοιχα texts και 2 πολύ απλά scripts. Όπως είναι λογικό, το Quit script τερματίζει την εφαρμογή και το restart script φορτώνει το ίδιο επίπεδο από την αρχή.

Στις παρακάτω εικόνες μπορείτε να δείτε αναλυτικά τους κώδικες.

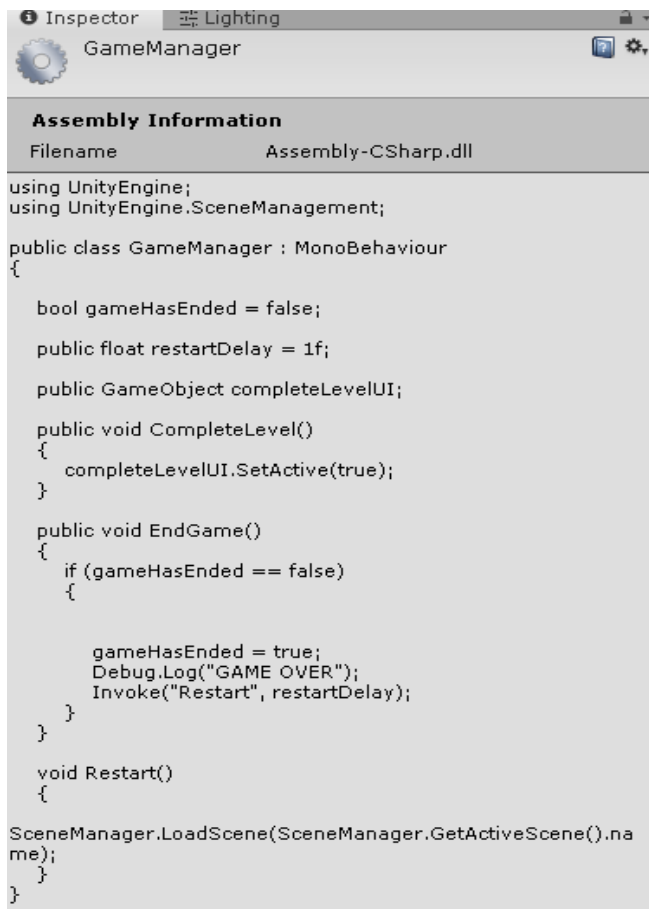


Μετέπειτα έχουμε το Asset Game Manager το οποίο παρόλο που δεν αποτελείται από κάποιο object ή text το οποίο μπορούμε να δούμε, χρησιμεύει ως διαμεσολαβητής

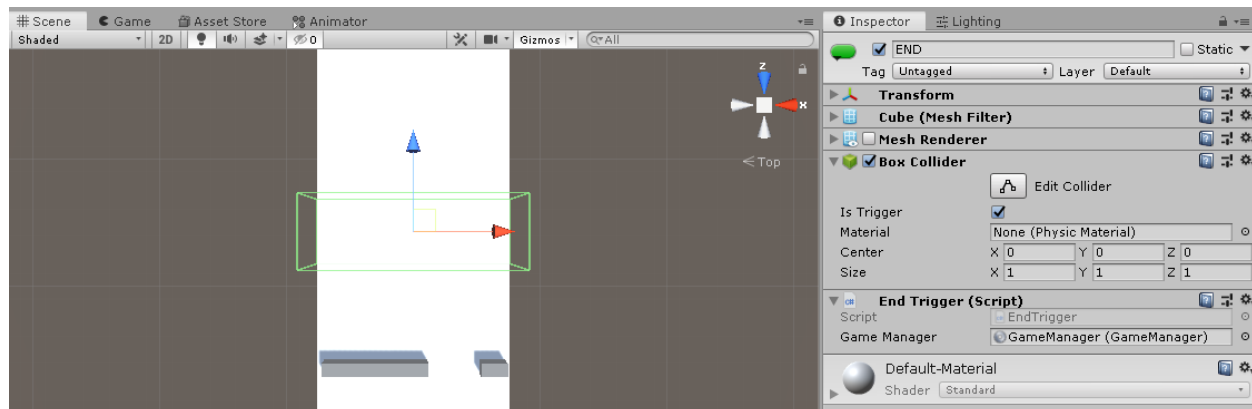
μεταξύ των επιπέδων. Ορίζει δηλαδή αν ο παίκτης έχασε και πρέπει να κάνει restart το επίπεδο ή αν ο παίκτης τερμάτισε και προχωράει μέσω του Level Complete που αναφέραμε παραπάνω.



Αποτελείται από ένα script το οποίο καθορίζει αν το παιχνίδι τελείωσε έτσι ώστε με μία καθυστέρηση η οποία στην περίπτωση μας είναι ένα δευτερόλεπτο να επανεκκινήσει το επίπεδο ή να προχωρήσει στο επόμενο μέσω του Level Complete Asset. Αυτό επιτυγχάνεται μέσω μίας true/false συνθήκης όπως μπορείτε να δείτε παρακάτω.



Φυσικά δεν θα μπορούσαμε να προχωρήσουμε σε κάποιο επίπεδο αν δεν υπήρχε και ένας τερματισμός. Το object END αποτελείται από έναν αόρατο κύβο χωρίς physics ο οποίος όταν έρθει σε επαφή με τον παίκτη τερματίζει το εκάστοτε επίπεδο και φορτώνει το επόμενο. Όπως μπορούμε να δούμε το Is Trigger στην κατηγορία Box Collider είναι ενεργοποιημένο για να επιτευχθεί αυτό και φυσικά δεν θα μπορούσε να λείπει ένα script το οποίο θα συνδέει το END object με τον GameManager.

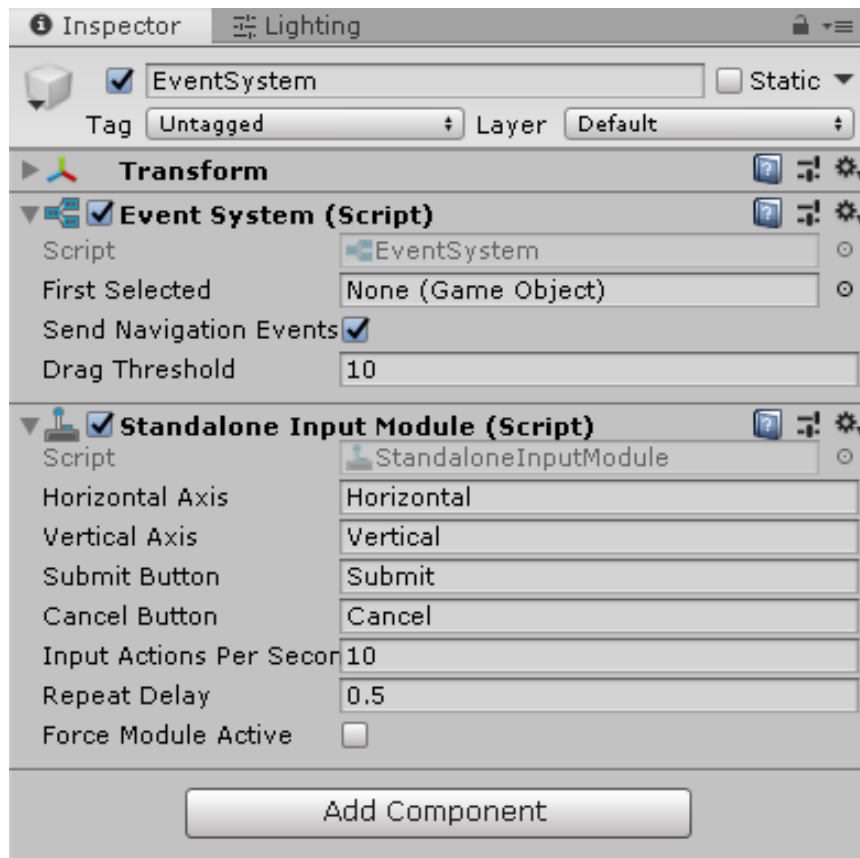


Το EndTrigger script ουσιαστικά βάζει την πολύ απλή συνθήκη πως αν ο παίκτης έρθει σε «σύγκρουση» με το END Object τότε καλεί τον GameManager και ολοκληρώνει το επίπεδο όπως φαίνεται στον παρακάτω κώδικα.

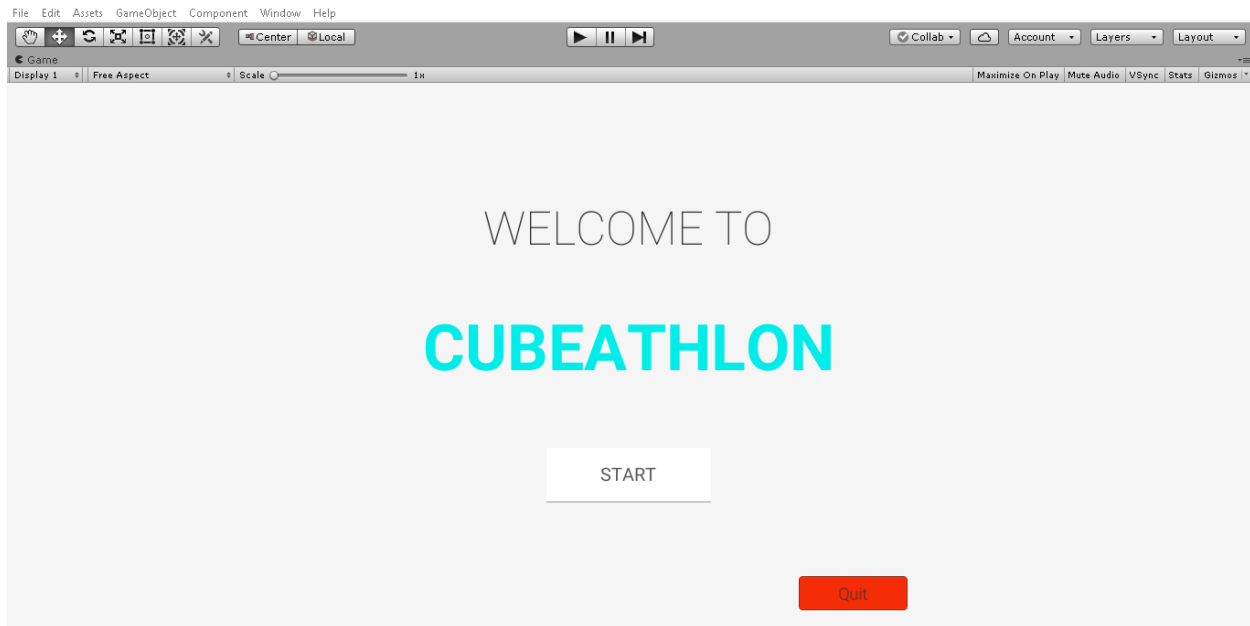
```
Score.cs PlayerMovement.cs PlayerCollision.cs Menu.cs LevelComplete.cs FollowPlayer.cs EndTrigger.cs
Miscellaneous Files EndTrigger
1 using UnityEngine;
2
3 public class EndTrigger : MonoBehaviour
4 {
5
6     public GameManager gameManager;
7     void OnTriggerEnter()
8     {
9         gameManager.CompleteLevel();
10    }
11 }
12
```

Από κει και έπειτα, δουλεύοντας σαν αλυσίδα, παίζει το animation και φορτώνει το επόμενο επίπεδο όπως αναφέραμε σε προηγούμενο στάδιο.

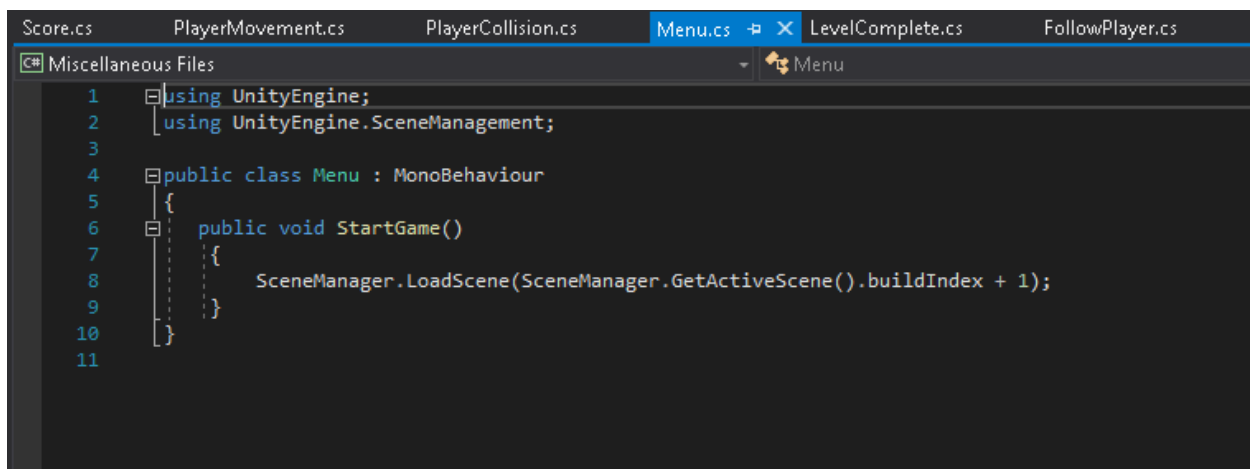
Τέλος υπάρχει ένα άδειο object, το EventSystem, το οποίο ορίζει την σειρά με την οποία γίνονται όλα τα προαναφερθέντα γεγονότα καθώς και τα γεγονότα αυτά καθαυτά.



Φυσικά δεν θα μπορούσε να είναι ένα ολοκληρωμένο βιντεοπαιχνίδι χωρίς το ένα αρχικό μενού και τίτλους τέλους. Το μενού αποτελείται από ένα στατικό Canvas και 2 Buttons, ένα για να ξεκινήσει ο χρήστης το παιχνίδι και ένα για να βγει από την εφαρμογή. Για άλλη μια φορά χρησιμοποιήθηκε η γραμματοσειρά Roboto.

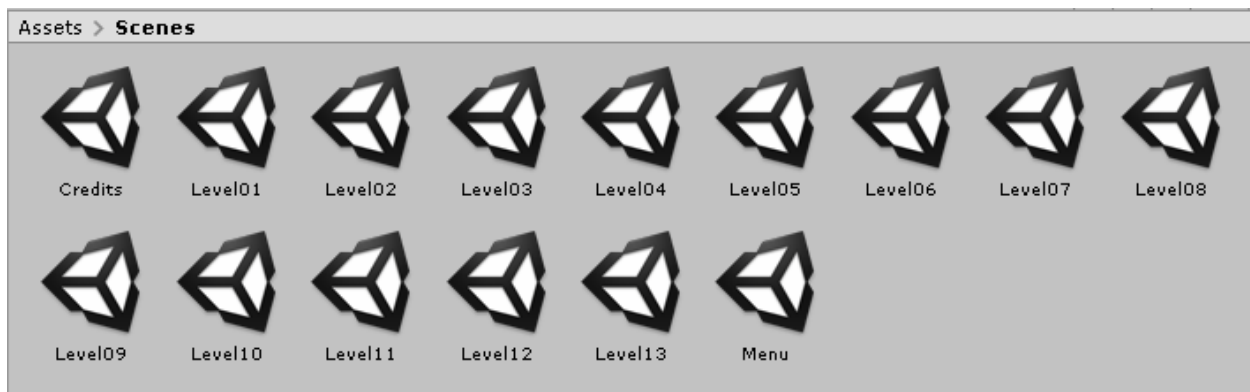


Παρακάτω θα βρείτε και το script που αφορά το Start Button καθώς το script για το Quit Button έχει αναφερθεί ήδη. Ουσιαστικά η public void που βάζουμε καλεί τον SceneManager και φορτώνει το επόμενο επίπεδο, το οποίο είναι και το πρώτο του παιχνιδιού.



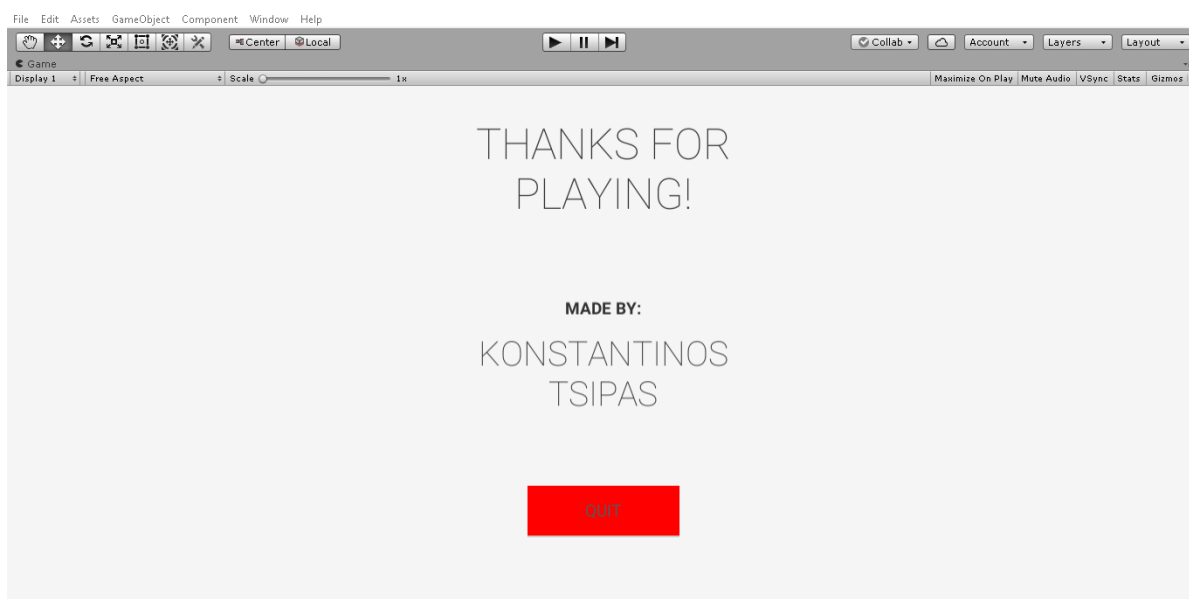
Αυτό φυσικά το ορίζουμε εμείς όταν έρθει η ώρα να χτίσουμε το βιντεοπαιχνίδι. Αν για παράδειγμα το Scene του Menu μπει μετά το Scene του πρώτου επιπέδου τότε αν ο παίκτης πατήσει το κουμπί Start θα μεταφερθεί αυτομάτως στο δεύτερο επίπεδο. Έχει να κάνει δηλαδή αποκλειστικά με την σειρά που θα επιλέξουμε εμείς στο build του

παιχνιδιού. Παρακάτω μπορείτε να δείτε όλα τα Scenes από τα οποία αποτελείται το παρόν βιντεοπαιχνίδι.



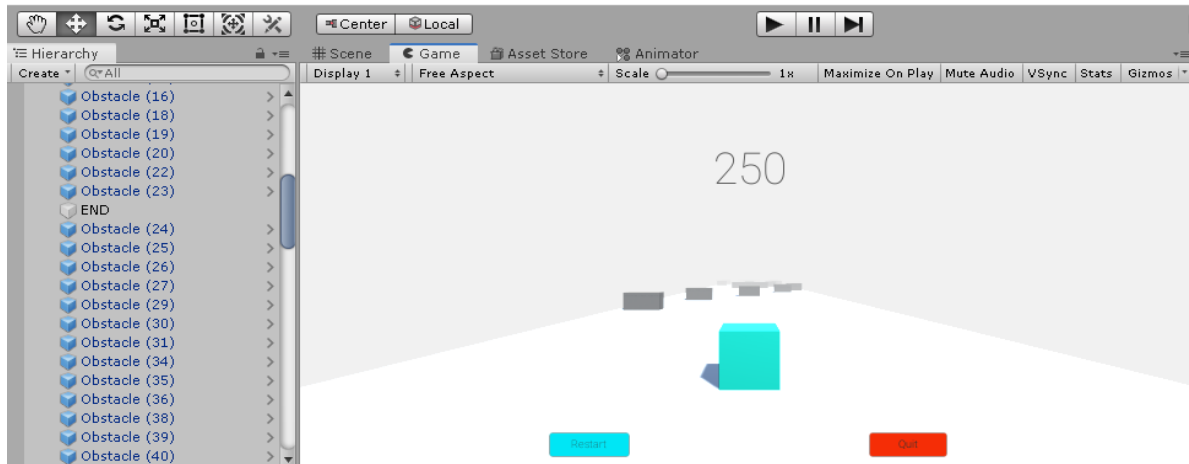
Φυσικά η κατασκευή του έγινε με την ακόλουθη σειρά. 1)Menu 2)Level01 3)Level02 4)Level03 5)Level04 6)Level05 7)Level06 8)Level07 9)Level08 10)Level09 11)Level10 12)Level11 13)Level12 14)Level13 15)Credits. Με αυτόν τον τρόπο υπάρχει μία συνεχής ροή στο παιχνίδι χωρίς απρόοπτα σφάλματα.

Τέλος έχουμε το Credits Scene, τους τίτλους τέλους δηλαδή του βιντεοπαιχνιδιού, στους οποίους εμφανίζεται ένα ευχαριστήριο μήνυμα στον παίκτη, το όνομα του δημιουργού και ένα κουμπί για να κλείσει την εφαρμογή.



Το Quit Button αποτελείται και αυτό από ένα Script που τερματίζει την εφαρμογή όπως αναφέραμε στην σελίδα 20.

Ως κλείσιμο για το πρώτο project μπορείτε να δείτε ένα preview όσον αφορά το πώς φαίνεται παρακάτω καθώς και θα υπάρχουν τα απαραίτητα links για να το κατεβάσετε.



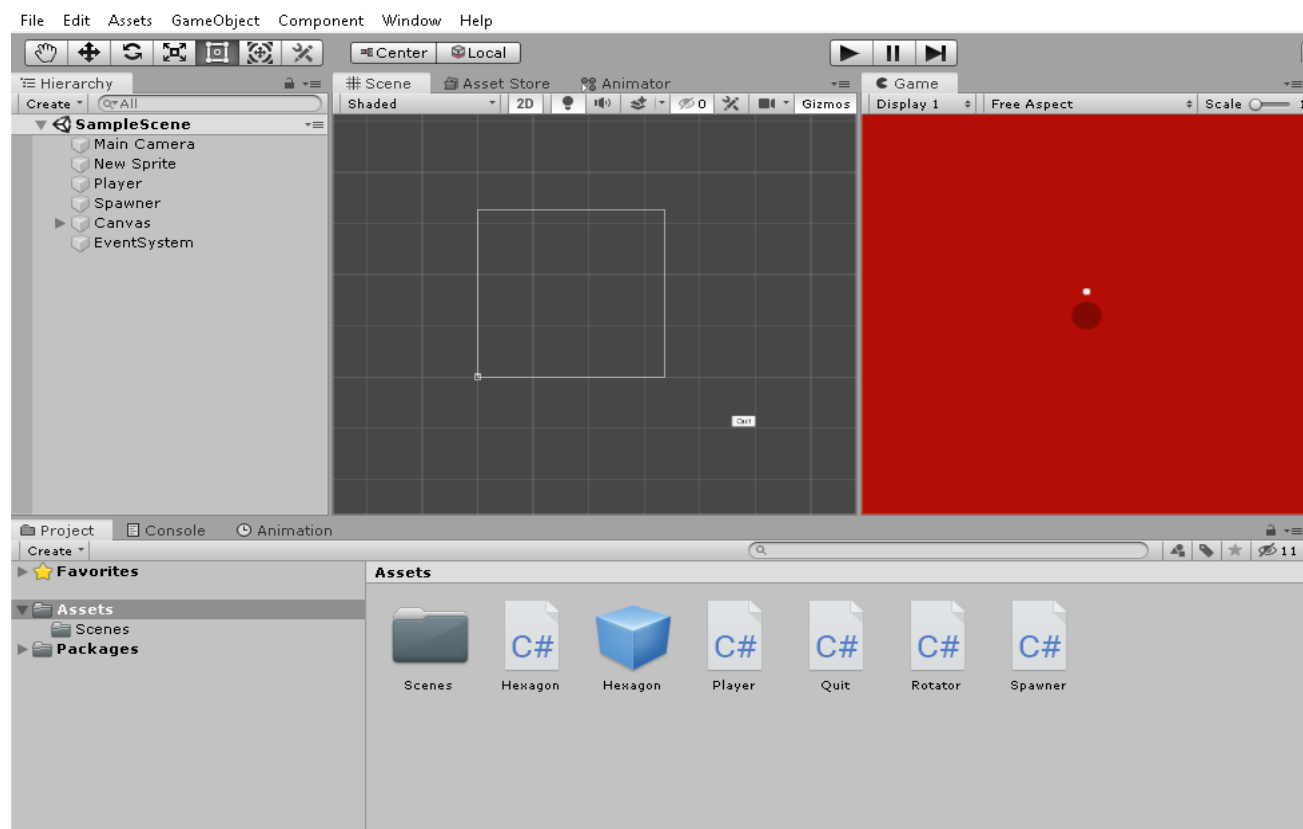
Dropbox Link:

<https://www.dropbox.com/sh/nck8fidgzc7hn9h/AACB9vXyVcWYmISzanMFSkloa?dl=0>

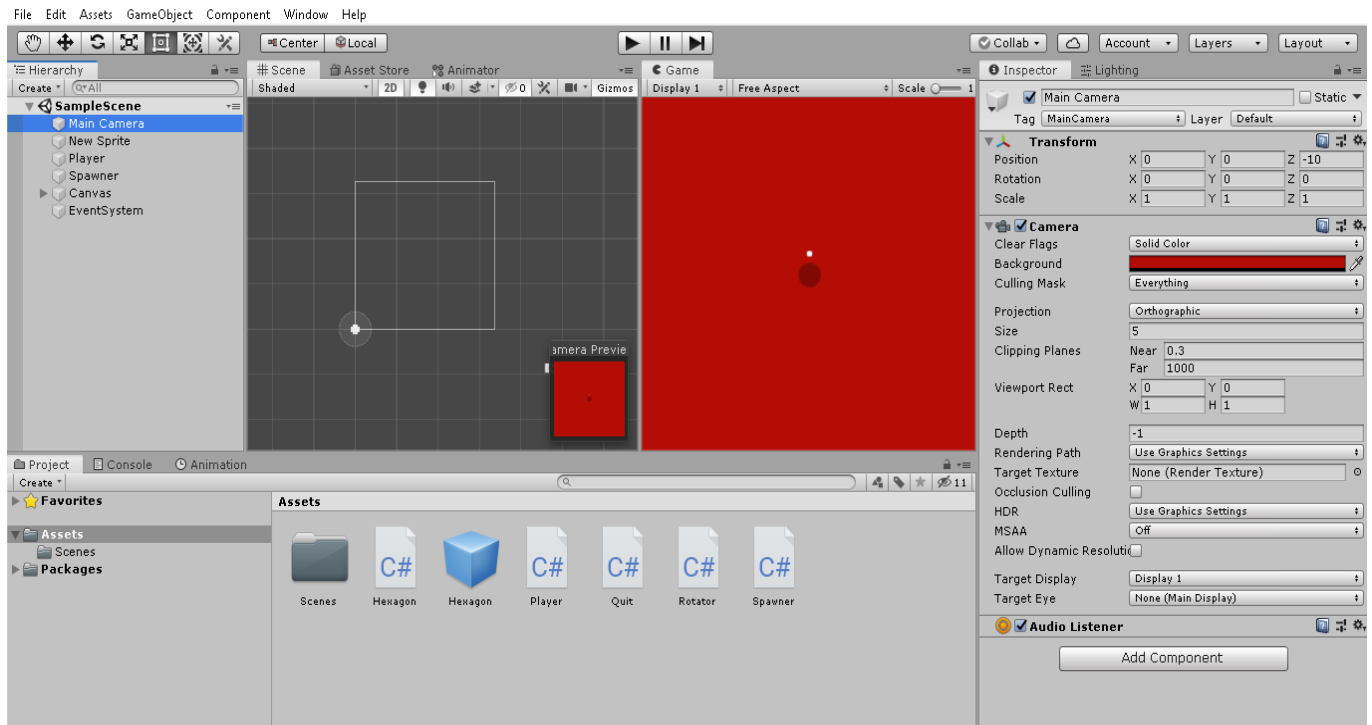
Project #2: Skillxagon

Το Skillxagon είναι ένα 2D παιχνίδι. Ο χειρισμός πραγματοποιείται με τα πλήκτρα “a” για να κινηθεί ο παίκτης αριστερόστροφα και “d” για δεξιόστροφα. Είναι μία μικροεφαρμογή για άτομα με έλλειψη συγκέντρωσης και συντονισμού και αφοσιώνεται στην βελτίωση του eye-to-hand coordination. Σκοπός του χρήστη είναι να περάσει την λευκή κουκίδα μέσα από τις κενές πλευρές των εξαγώνων που κλείνουν συνεχώς προς το κέντρο. Αποτελείται από ένα μόνο επίπεδο το οποίο δεν έχει τερματισμό και οποιαδήποτε στιγμή ο χρήστης μπορεί να τερματίσει την εφαρμογή με το κουμπί Quit που βρίσκεται στην οθόνη. Για άλλη μια φορά το βιντεοπαιχνίδι αυτό αναπτύχθηκε με την βοήθεια της Unity Engine και του Microsoft Visual Studio.

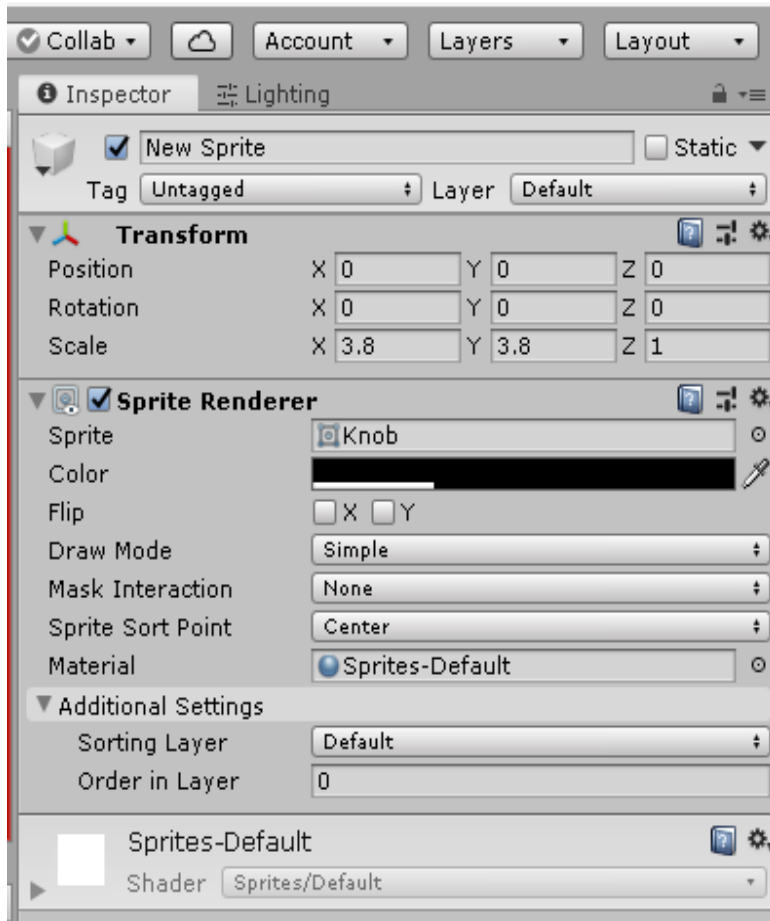
Αρχικά μπορούμε να δούμε στην παρακάτω εικόνα τα assets από τα οποία αποτελείται το βιντεοπαιχνίδι καθώς και ένα αρχικό στάδιο ανάπτυξης και ένα preview του περιβάλλοντος του παιχνιδιού.



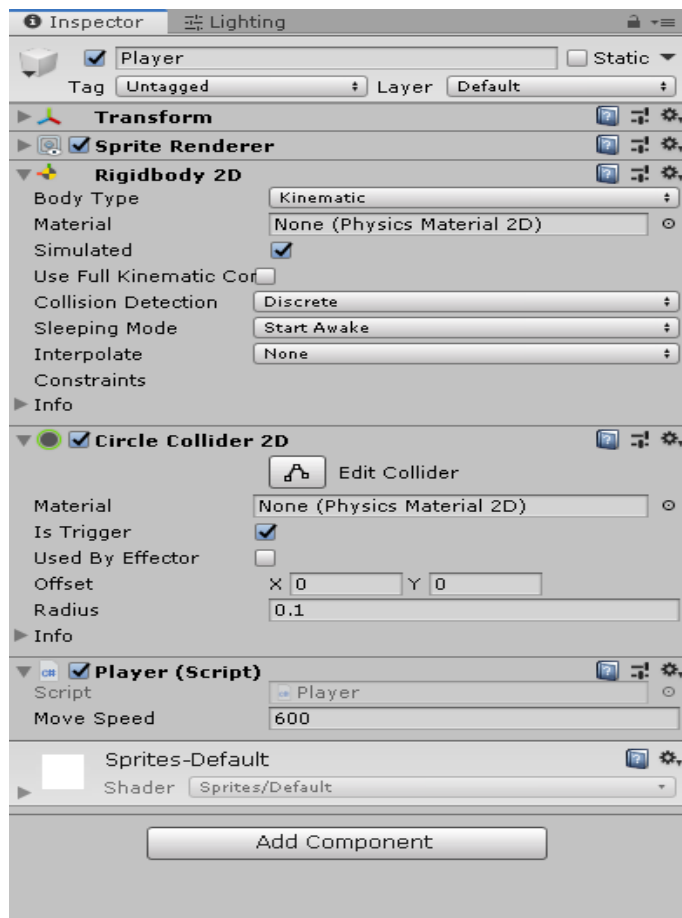
Το πρώτο βήμα ήταν να καθοριστεί η προοπτική του παιχνιδιού μέσω της κάμερας. Καθώς λοιπόν το βιντεοπαιχνίδι είναι τύπου 2D, αποφάσισα να του δώσω μία οπτική τύπου *soaring eagle*, δηλαδή ο χρήστης να έχει οπτική από πάνω (σαν αετός). Έτσι λοιπόν μπήκαν τα απαραίτητα assets για την κάμερα καθώς και επιλέχθηκε το χρώμα του background όπως μπορείτε να δείτε και παρακάτω στα components της κάμερας.



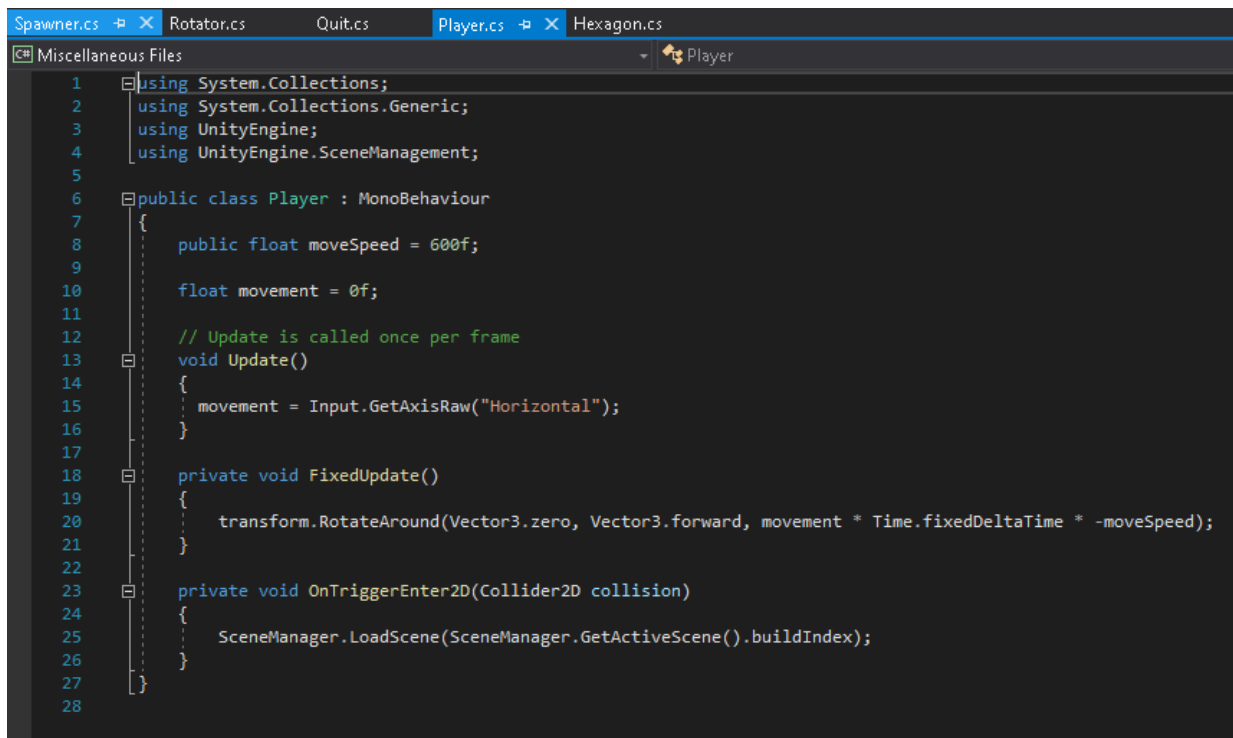
Μετάπειτα έπρεπε να υπάρχει ένα σημείο αναφοράς, γύρω από το οποίο θα περιστρέφεται ο παίκτης. Για τον λόγο αυτό πρόσθεσα ένα 2D Circle Object το οποίο ονόμασα *New Sprite*, του άλλαξα την κλίμακα για να αντιπροσωπεύει το αποτέλεσμα που επιθυμούσα και φυσικά του έδωσα material και χρώμα. Παρακάτω μπορούμε να δούμε τα components του αναλυτικά μέσα στο Unity.



Στη συνέχεια πραγματοποιήθηκε η ανάπτυξη του παίκτη. Αποτελείται από ένα 2D Circle Object και του έχει δοθεί η ετικέτα Player για τις αναφορές μέσα στα scripts. Του έχουν δοθεί επίσης τα Rigidbody 2D και το Circle Collider 2D για την συμπεριφορά του με το περιβάλλον του παιχνιδιού και της αλληλεπιδράσεις με άλλα σώματα και φυσικά τα physics. Δεν θα μπορούσαμε όμως να είχαμε παίκτη χωρίς να υπάρχει κίνηση, γι αυτό τον λόγο στα components υπάρχει ένα script ονομαζόμενο Player στο οποίο καθορίζεται η κίνηση του παίκτη. Στην παρακάτω εικόνα μπορείτε να δείτε αναλυτικά αυτά που αναφέρθηκαν.

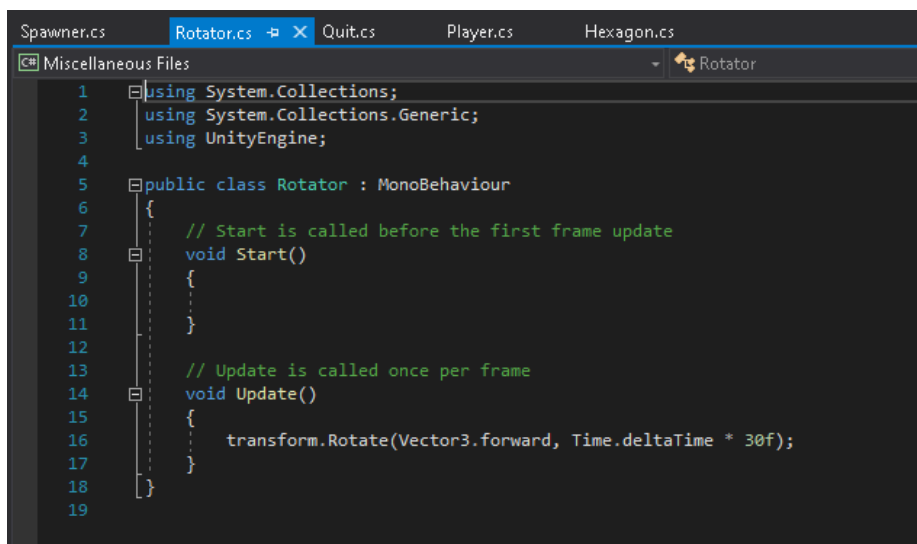


Στο C# script ονομαζόμενο Player αρχικά βάλαμε μία public float που θα καθορίζει την ταχύτητα με την οποία θα κινείται ο παίκτης. Στη συνέχεια θέτουμε τον άξονα ο οποίος είναι οριζόντιος και αμέσως μετά τον τρόπο κίνησης ο οποίος είναι περιστροφικός γύρω από το New Sprite που αναφέρθηκε παραπάνω. Γύρω δηλαδή από το σημείο (0,0). Τέλος έθεσα μία συνθήκη κατά την οποία αν ο παίκτης έρθει σε επαφή με κάποιο άλλο αντικείμενο, στην προκειμένη περίπτωση της κλειστές πλευρές του εξαγώνου, να γίνεται η επανεκκίνηση του επιπέδου. Μπορείτε να βρείτε τον κώδικα στην παρακάτω εικόνα.



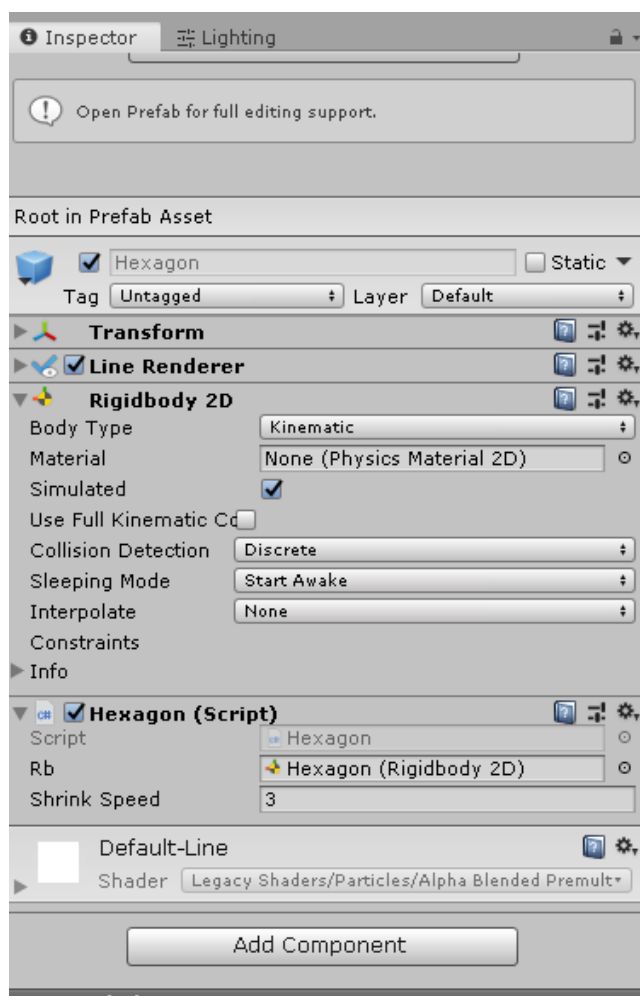
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class Player : MonoBehaviour
7 {
8     public float moveSpeed = 600f;
9
10    float movement = 0f;
11
12    // Update is called once per frame
13    void Update()
14    {
15        movement = Input.GetAxisRaw("Horizontal");
16    }
17
18    private void FixedUpdate()
19    {
20        transform.RotateAround(Vector3.zero, Vector3.forward, movement * Time.fixedDeltaTime * -moveSpeed);
21    }
22
23    private void OnTriggerEnter2D(Collider2D collision)
24    {
25        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
26    }
27 }
28
```

Παρ' όλα αυτά όμως δεν αρκεί αυτό το script για την περιστροφή του παίκτη. Στα assets λοιπόν δημιουργούμε άλλο ένα C# Script ονομαζόμενο Rotator. Με την βοήθεια αυτού του script η private void FixedUpdate() στην παραπάνω εικόνα πλέον είναι συντακτικά σωστή και μπορεί να δουλέψει και πλέον ο παίκτης μπορεί να περιστραφεί γύρω από το σημείο (0,0).



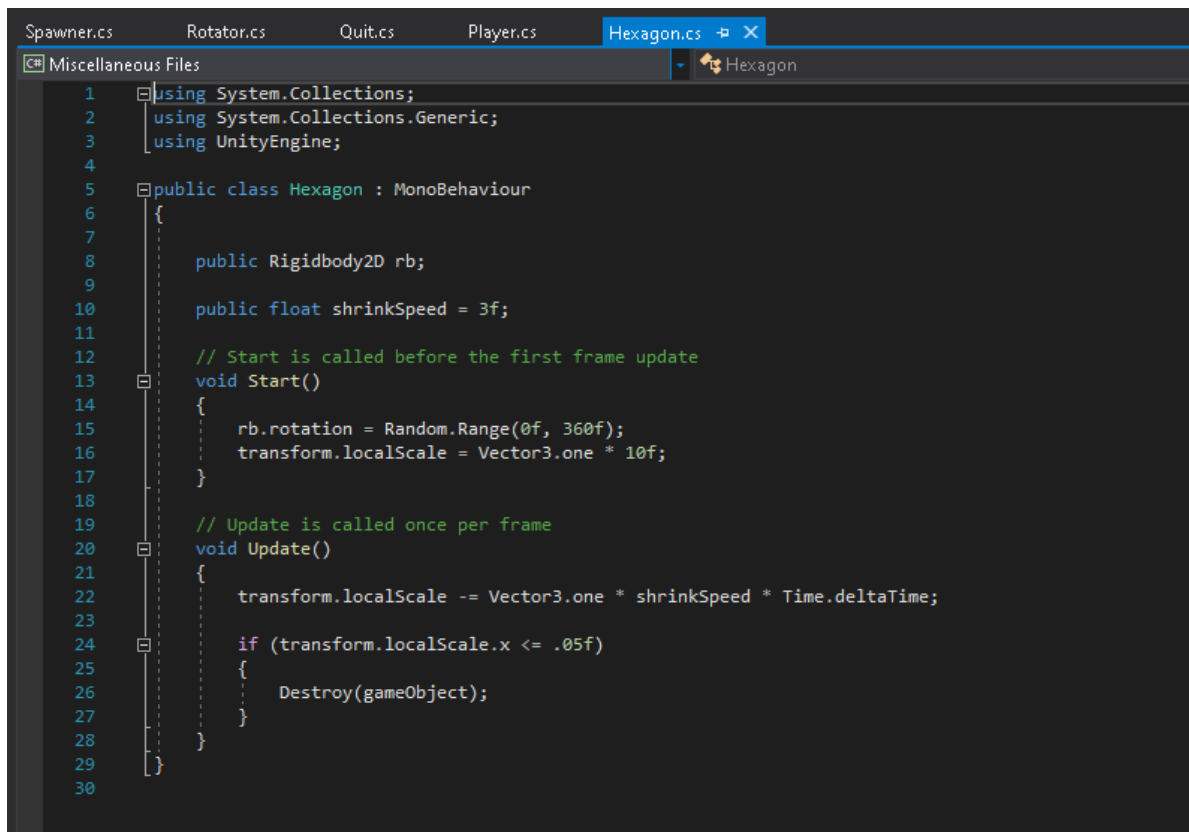
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rotator : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10    }
11
12    // Update is called once per frame
13    void Update()
14    {
15        transform.Rotate(Vector3.forward, Time.deltaTime * 30f);
16    }
17 }
18
19
```

Ας δούμε όμως τα «εμπόδια» του παιχνιδιού, τα εξάγωνα όπως προαναφέραμε. Το Hexagon Object αποτελείται από ένα 2D Line Object το οποίο αλλάζοντας τις κλίμακες του κατέληξε στην τελική του μορφή, ένα εξάγωνο δηλαδή με μία ανοιχτή πλευρά. Του δόθηκε λευκό χρώμα και προφανώς το component Rigidbody 2D για να μπορεί να αλληλεπιδρά με άλλα objects. Στη συνέχεια έγινε η σύνταξη ενός C# script ονομαζόμενο Hexagon για το παρόν αντικείμενο και τέλος έγινε prefab για να μπορεί να χρησιμοποιηθεί αυτούσιο σε κάθε περίπτωση χωρίς να χρειάζεται να δημιουργήσουμε άλλα Objects. Παρακάτω μπορούμε να δούμε μια προεπισκόπηση των components απο τα οποία αποτελείται το Hexagon Object.



Στο Hexagon script ξεκίνησα με μία public για το Rigidbody 2D component έτσι ώστε το Hexagon να αναγνωρίζει όταν έρχεται σε επαφή με κάποιο άλλο αντικείμενο με το ίδιο component. Στη συνέχεια με την βοήθεια μιας public float έθεσα το ρυθμό συρρίκνωσης

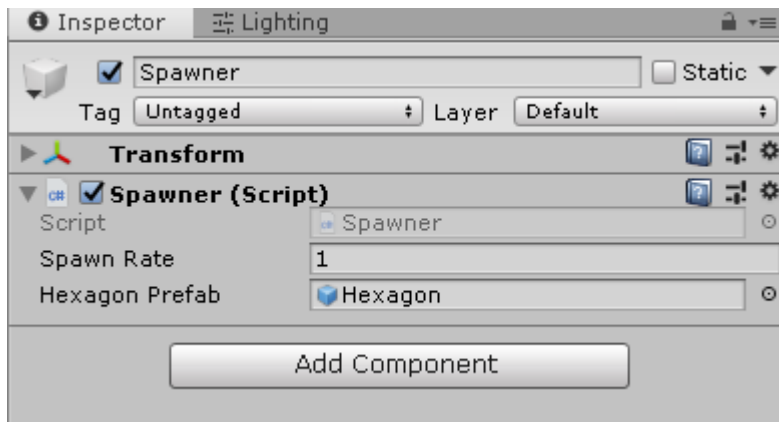
του εξαγώνου καθώς αυτό θα κλείνει προς τον παίκτη, δηλαδή γύρω από το σημείο (0,0). Έπειτα προγραμματίσα το εξάγωνο πριν ξεκινήσει να κλείνει προς τον παίκτη, να περιστρέφεται τυχαία μεταξύ 0 και 360 μοίρες, έτσι κάθε φορά η ανοιχτή πλευρά του εξαγώνου θα είναι σε διαφορετικό σημείο. Του δόθηκε στη συνέχεια κλίμακα και σε συνδυασμό με τον ρυθμό συρρίκνωσης έθεσα την συνθήκη πως όταν το εξάγωνο συρρικνωθεί τόσο ώστε να φτάσει στο κεντρικό σημείο τότε το συγκεκριμένο object να καταστραφεί καθώς θα μας είναι πλέον άχρηστο. Παρακάτω μπορείτε να δείτε αναλυτικά τον κώδικα στο Microsoft Visual Studio.



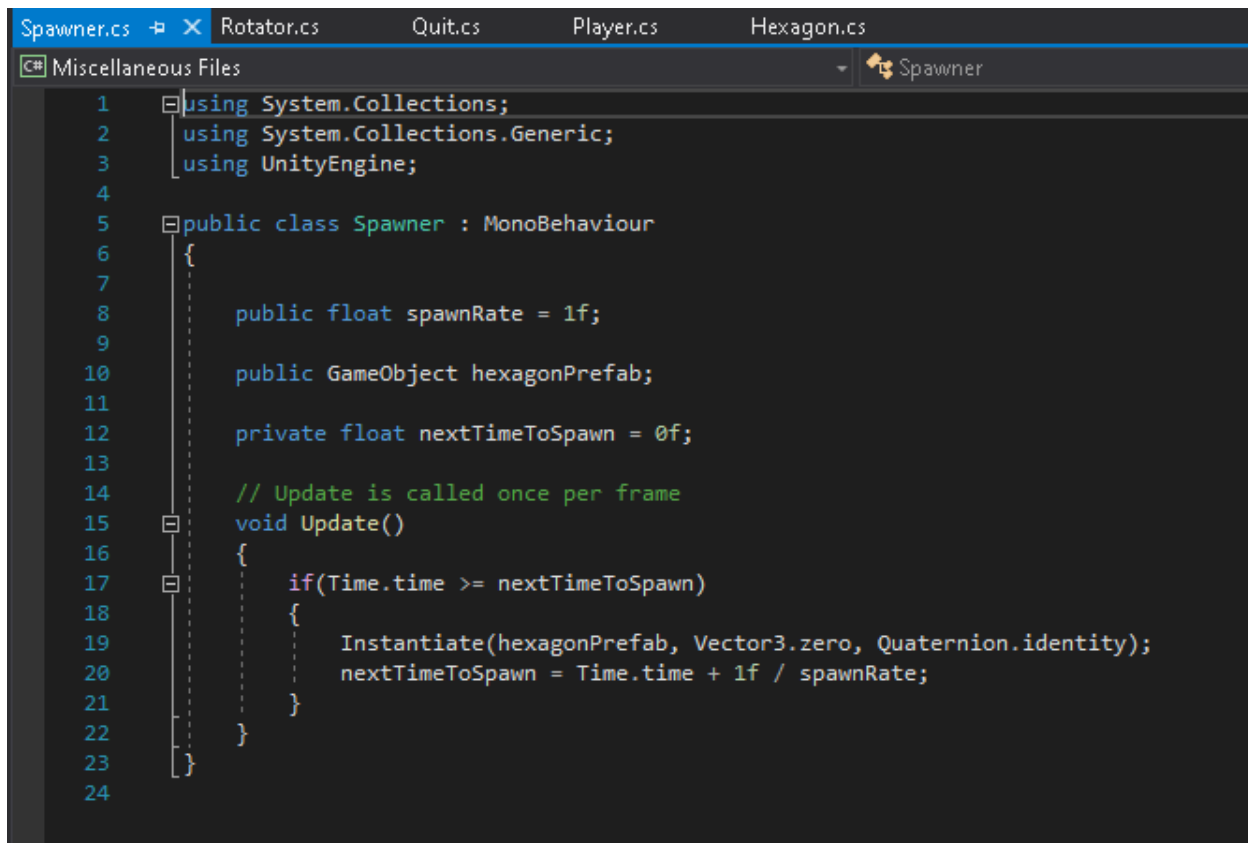
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Hexagon : MonoBehaviour
6  {
7
8      public Rigidbody2D rb;
9
10     public float shrinkSpeed = 3f;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         rb.rotation = Random.Range(0f, 360f);
16         transform.localScale = Vector3.one * 10f;
17     }
18
19     // Update is called once per frame
20     void Update()
21     {
22         transform.localScale -= Vector3.one * shrinkSpeed * Time.deltaTime;
23
24         if (transform.localScale.x <= .05f)
25         {
26             Destroy(gameObject);
27         }
28     }
29 }
30
```

Ας μην ξεχνάμε παρ' όλα αυτά ότι όλα τα παραπάνω που αφορούν το Hexagon Object αφορούν ένα και μόνο ένα Object. Για αυτόν λοιπόν τον λόγο αρχικά το κάναμε prefab, για να μπορεί να χρησιμοποιηθεί πολλές φορές αυτούσιο με όλα τα components του και στην συνέχεια δημιουργήσαμε ένα Empty Object, αντικείμενο το οποίο υπάρχει αλλά δεν φαίνεται, ονομαζόμενο Spawner για να μπορούμε μέσω αυτού να δημιουργούμε

εξάγωνο συνεχώς. Το Spawner Object όπως μπορούμε να δούμε παρακάτω αποτελείται μόνο από ένα C# script ονομαζόμενο Spawner.

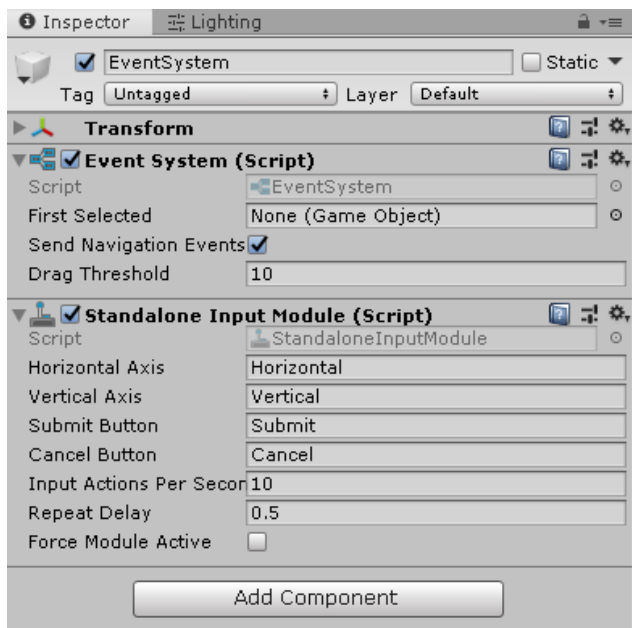


Ας δούμε όμως αναλυτικά από τι αποτελείται το εν λόγω script. Αρχικά θετουμε τον ρυθμό με τον οποίο θα δημιουργούνται καινούργια αντικείμενα και στη συνέχεια ποια αντικείμενα θα είναι αυτά. Προφανώς στην περίπτωση μας είναι το Hexagon Object το οποίο με τον κώδικα που συντάξαμε προηγουμένως θα κλείνει προς τον παίκτη. Έπειτα με την βοήθεια μίας private float, η οποία είναι κρυφή και δεν μπορούμε να την δούμε στο Unity όπως τις άλλες 2, θέτουμε τον τελευταίο όρο ο οποίος είναι το πότε θα δημιουργηθεί το επόμενο εξάγωνο και μέσω μιας συνθήκης θέτουμε ουσιαστικά τον χρόνο να είναι ένα δευτερόλεπτο όπως μπορείτε να δείτε τον κώδικα στην παρακάτω εικόνα.

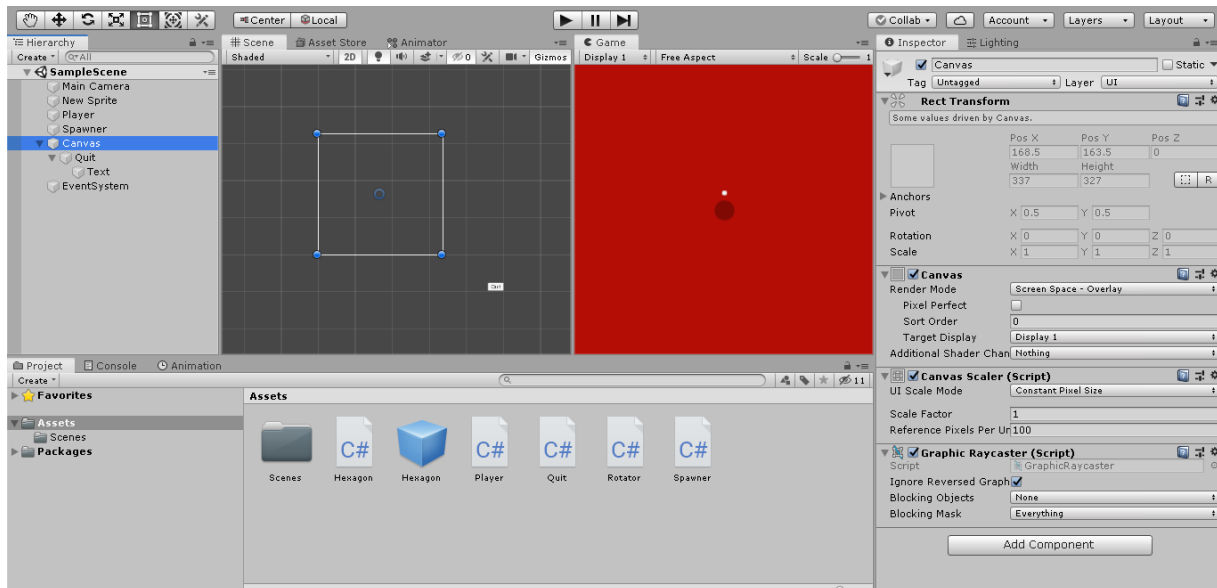


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Spawner : MonoBehaviour
6 {
7
8     public float spawnRate = 1f;
9
10    public GameObject hexagonPrefab;
11
12    private float nextTimeToSpawn = 0f;
13
14    // Update is called once per frame
15    void Update()
16    {
17        if(Time.time >= nextTimeToSpawn)
18        {
19            Instantiate(hexagonPrefab, Vector3.zero, Quaternion.identity);
20            nextTimeToSpawn = Time.time + 1f / spawnRate;
21        }
22    }
23 }
24
```

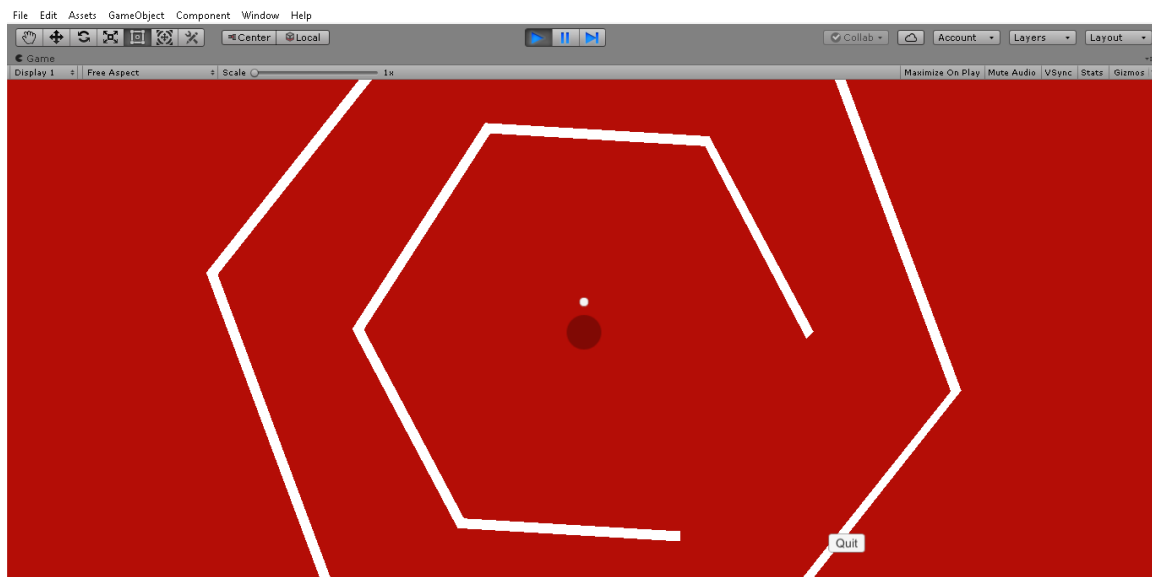
Εννοείται επίσης πως για να λειτουργούν όλα σωστά με την κάθε σκηνή και τις συναρτήσεις, έχουμε βάλει για άλλη μία φορά ένα Event System για να μας βοηθήσει σε αυτό.



Και τέλος δεν θα μπορούσε να λείπει ένα Canvas στο οποίο υπάρχει ένα Button Object ονομαζόμενο Quit το οποίο θα επιτρέπει στον χρήστη να κλείνει την εφαρμογή όποτε επιθυμήσει.



Παρακάτω μπορείτε να βρείτε μία προεπισκόπηση του βιντεοπαιχνιδιού αυτού καθώς και το ανάλογο link για να το κατεβάσετε.



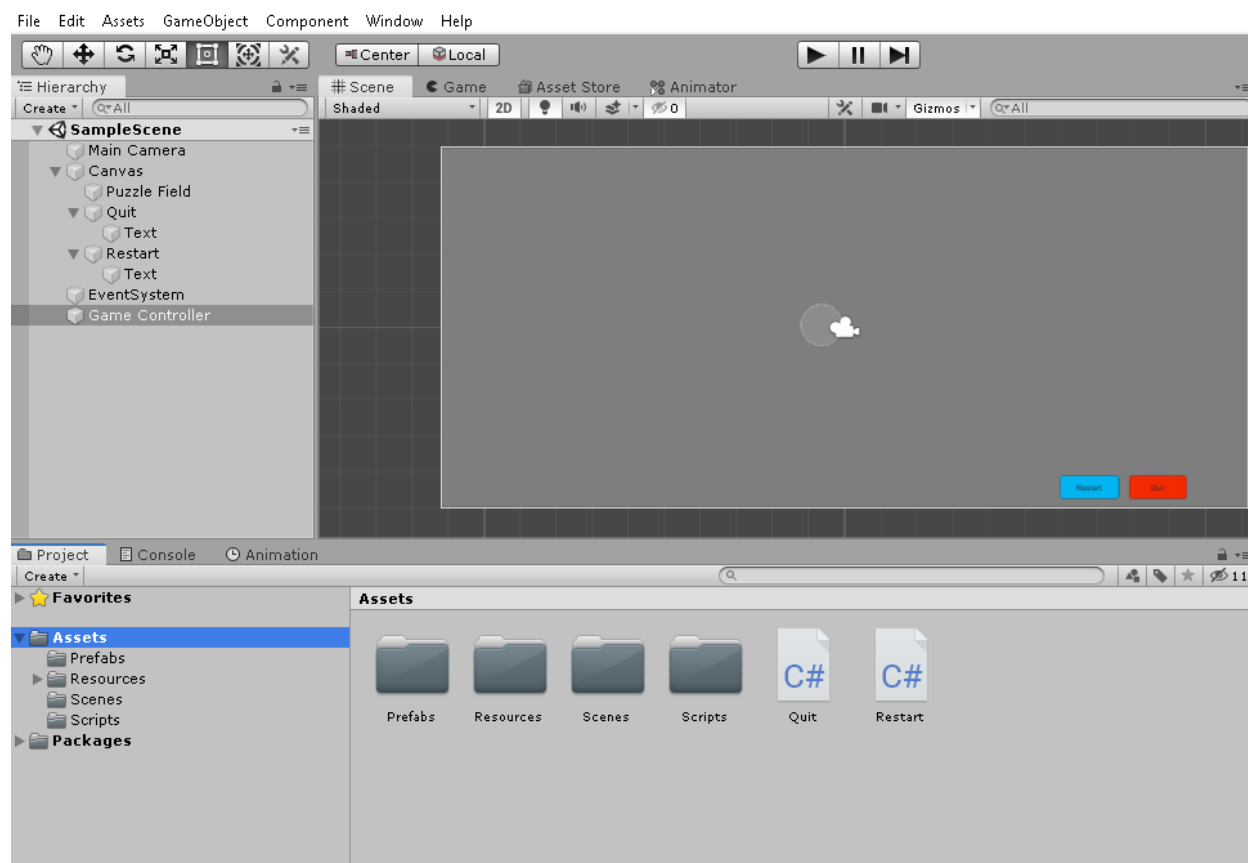
Dropbox link:

<https://www.dropbox.com/sh/aerh1kse6rima47/AABKZm6YoCdboQHFgKjd8kxRa?dl=0>

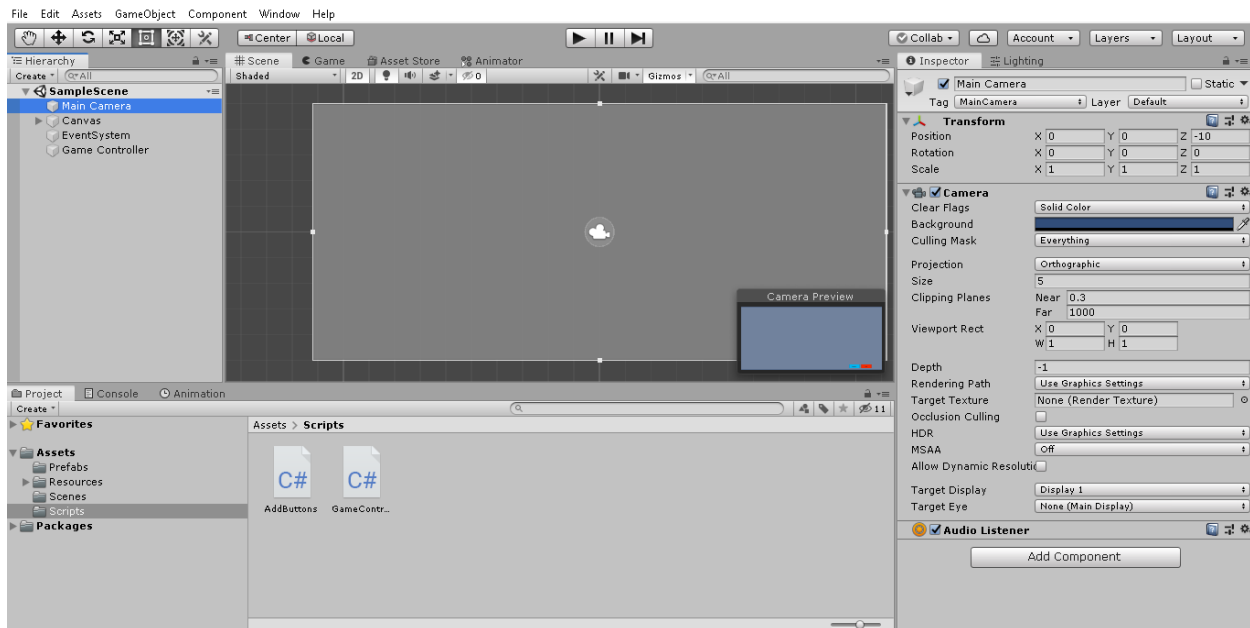
Project #3: Memory Puzzle Game

Το Memory Puzzle Game είναι ένα 2D παιχνίδι μνήμης. Ο χειρισμός πραγματοποιείται με το ποντίκι και το αριστερό κλικ του ποντικιού. Ο παίκτης προσπαθεί να βρει τα κοινά ζευγάρια καρτών οι οποίες είναι κρυμμένες. Όταν ο παίκτης βρει ένα ζευγάρι, αυτό εξαφανίζεται ενώ αν δεν βρει το κοινό ζευγάρι οι κάρτες ξανακλείνουν. Όταν βρει όλα τα ζευγάρια κερδίζει. Είναι μια εφαρμογή για άτομα με νοητικές παθήσεις που αφορούν την έλλειψη μακροχρόνιας και βραχυχρόνιας μνήμης. Το βιντεοπαιχνίδι αυτό πραγματοποιήθηκε με την βοήθεια του Unity Engine και του Microsoft Visual Studio.

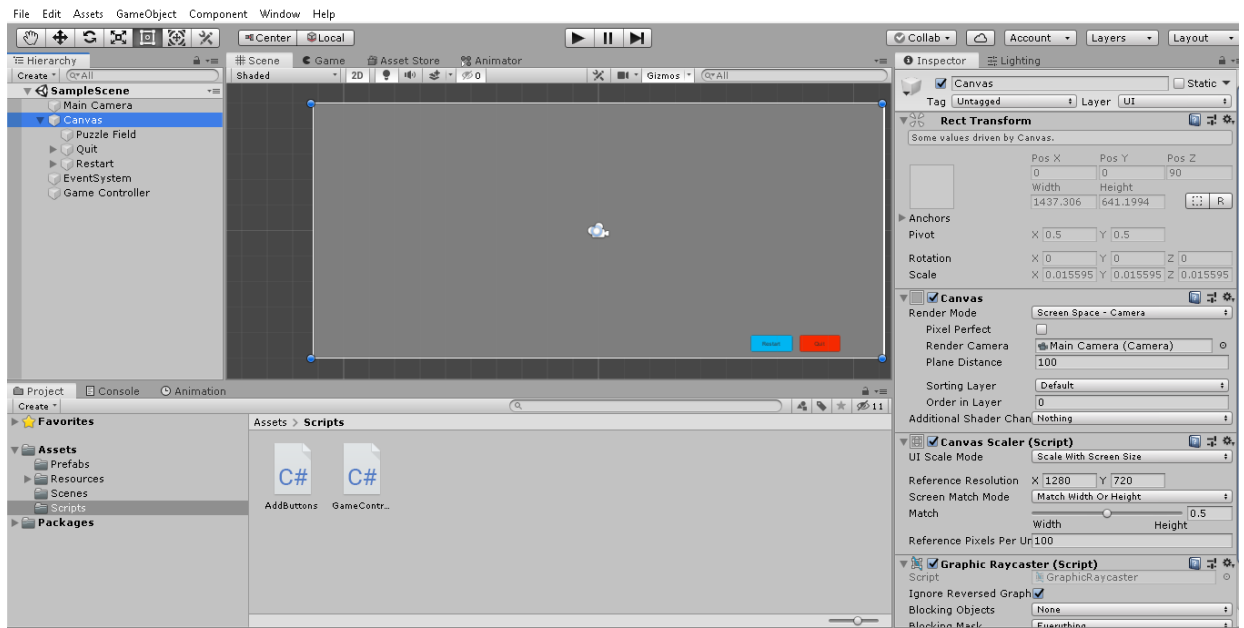
Αρχικά μπορούμε στην παρακάτω εικόνα να δούμε τα assets από τα οποία αποτελείται το βιντεοπαιχνίδι. Αυτά είναι: Η κύρια κάμερα, ο καμβάς στον οποίο υπάρχει και το puzzle field δηλαδή το περιβάλλον μας και τα κουμπιά της εξόδου και της επανεκκίνησης quit και restart αντίστοιχα, το σύστημα γεγονότων και φυσικά η λειτουργία του παιχνιδιού.



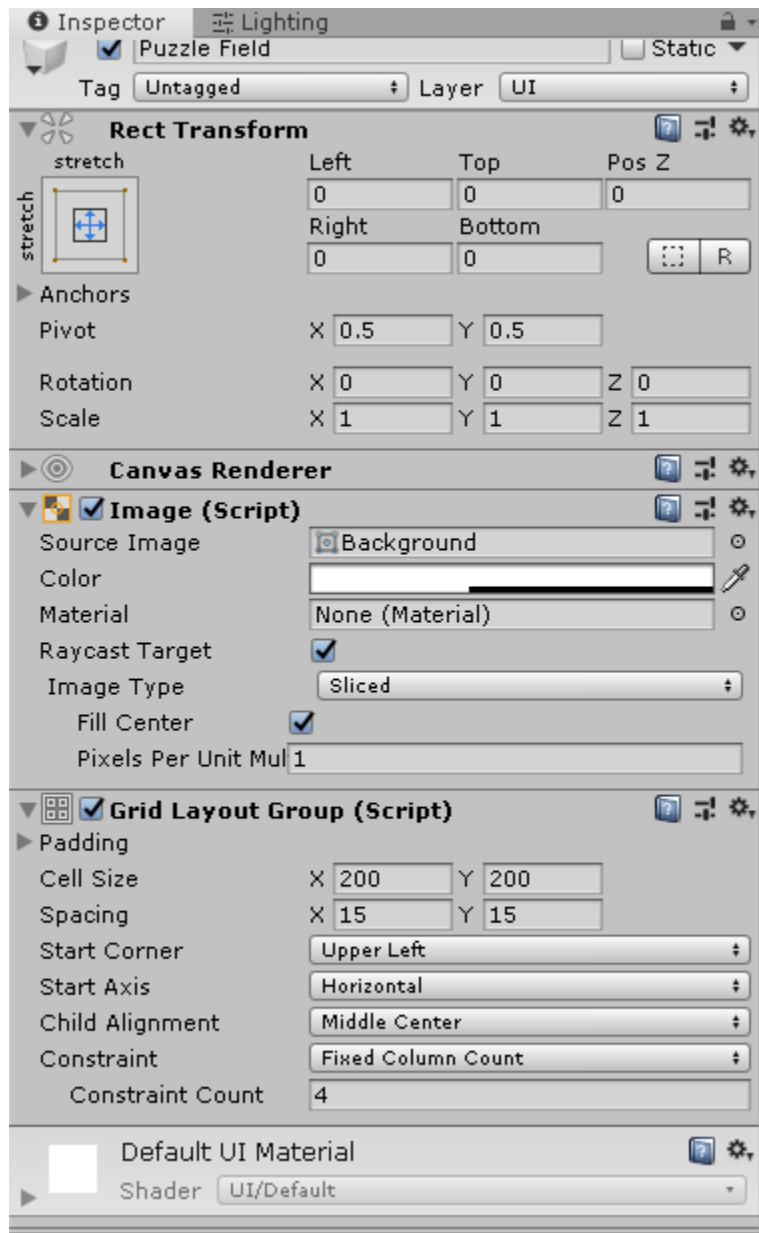
Για άλλη μία φορά η οπτική της κάμερας είναι soaring eagle τύπου έτσι ώστε ο παίκτης να μπορεί να βλέπει όλο το ταμπλό και στην προκειμένη περίπτωση η οπτική δεν θα μπορούσε να είναι τίποτε άλλο παρά αυτό. Με αυτήν την λογική επιλέχθηκαν οι απαραίτητες ρυθμίσεις καθώς και το χρώμα του background όπως μπορείτε να δείτε παρακάτω.



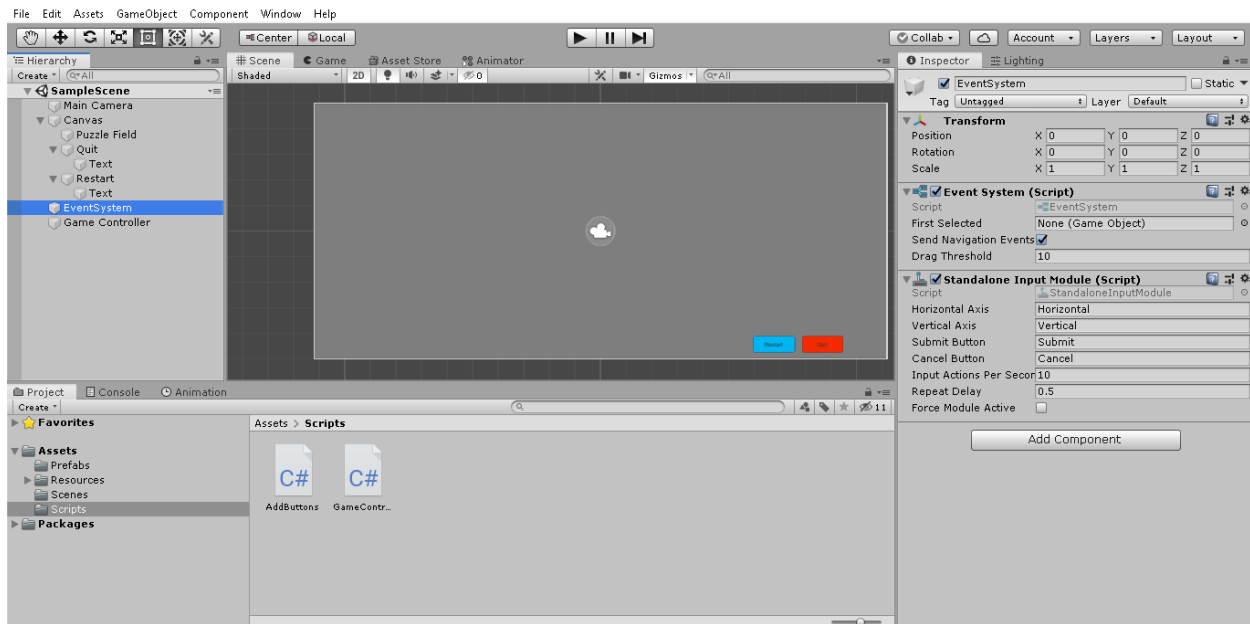
Στη συνέχεια μπορούμε να δούμε μια προεπισκόπηση του Canvas object μέσα στο οποίο υπάρχει το Puzzle Field object το οποίο οφείλεται για την διαρρύθμιση και την κατανομή των καρτών στο ταμπλό καθώς και τα κουμπιά Quit και Restart.



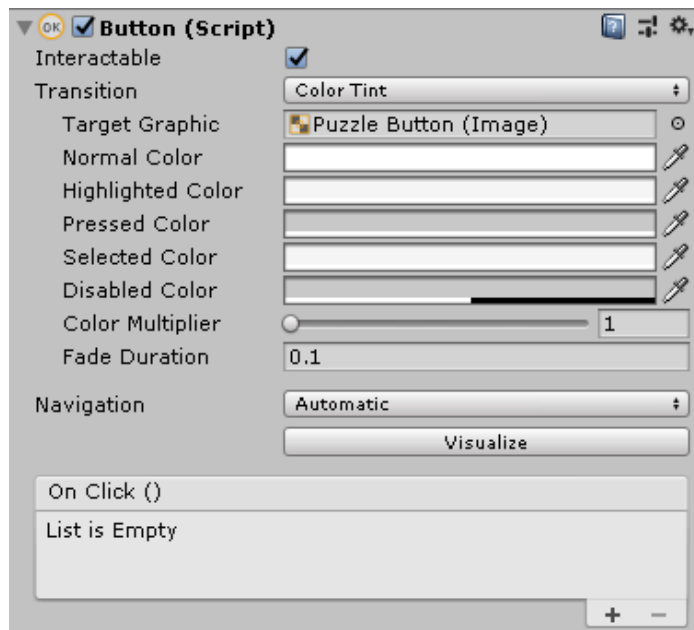
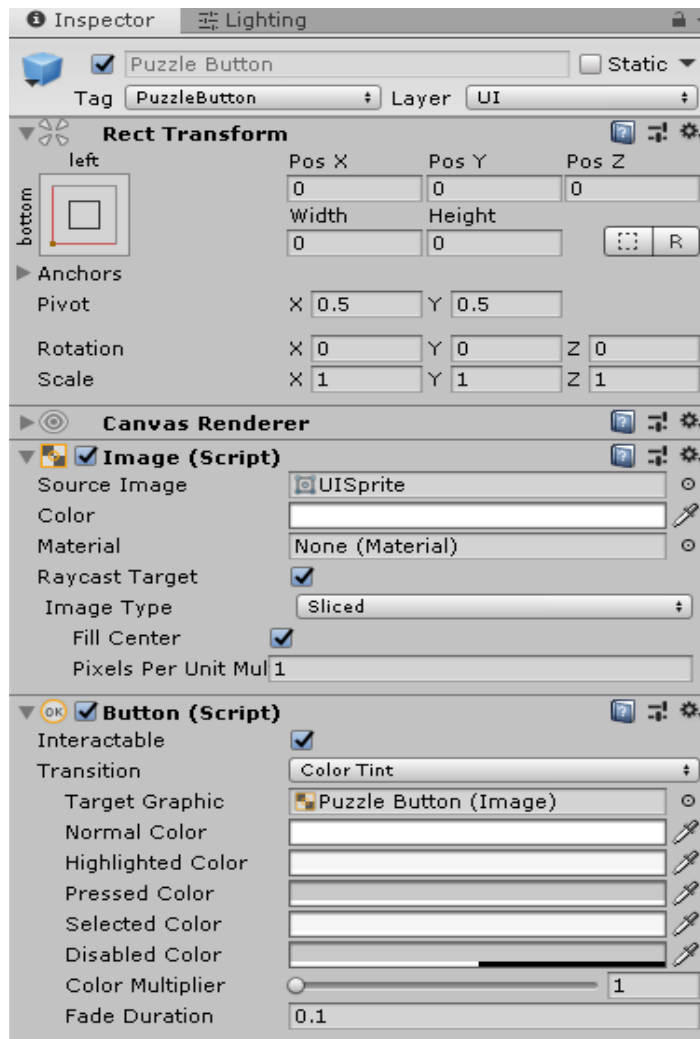
Το Puzzle Field object όπως αναφέρθηκε, φροντίζει για το layout των καρτών στο περιβάλλον. Για τον λόγο αυτό χρησιμοποιήθηκε το asset Grid Layout Group που μπορούμε να δούμε στην παρακάτω εικόνα το οποίο κάνει ακριβώς αυτό και με τις απαραίτητες ρυθμίσεις το δικό μας layout είναι 4x3. Αποτελείται δηλαδή από έναν πίνακα από 12 κάρτες οι οποίες είναι 4 οριζόντια και 3 κάθετα. Τα κουμπιά Quit και Restart στη συνέχεια τοποθετήθηκαν χειροκίνητα στο κάτω αριστερό μέρος της εφαρμογής.



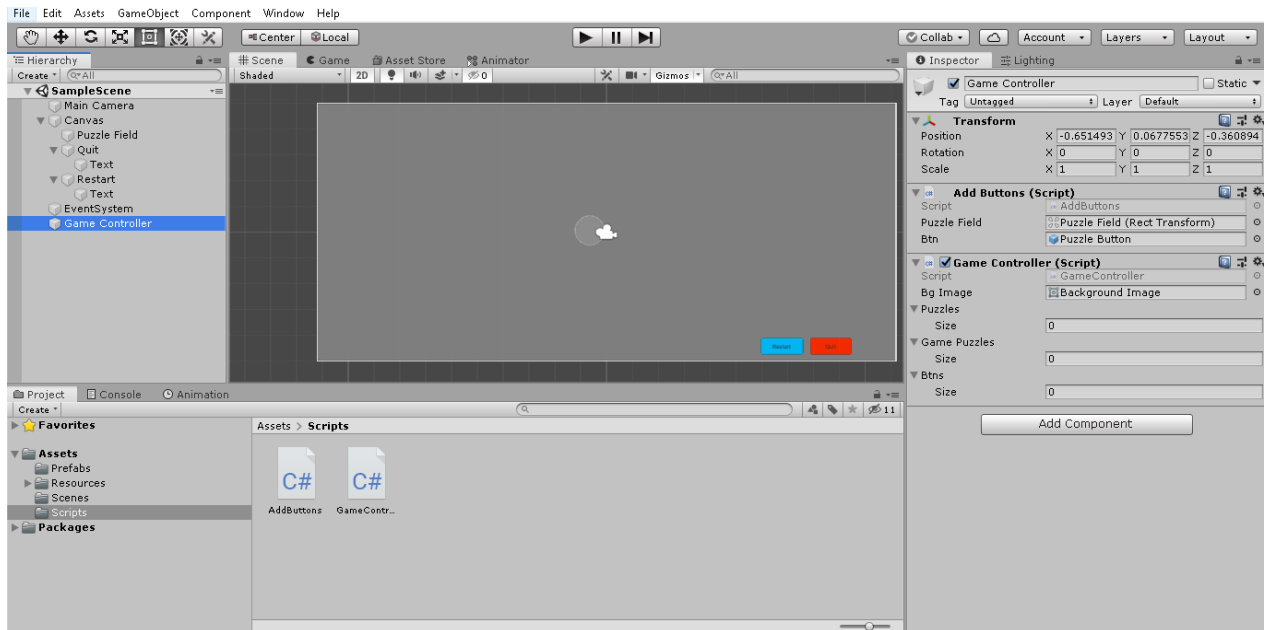
Στη συνέχεια μπορούμε να δούμε μία προεπισκόπηση του Event System object το οποίο ρυθμίζει την σειρά γεγονότων μέσα στα scripts σε C# τα οποία δημιουργήθηκαν για την παρούσα εφαρμογή και θα δούμε στην συνέχεια.



Ακόμη ένα πολύ σημαντικό object για την εφαρμογή αυτή είναι το Puzzle Button object το οποίο δεν μπορούμε να το δούμε στην προεπισκόπηση γιατί το μετέτρεψα σε prefab για μπορεί να χρησιμοποιηθεί πολλές φορές χωρίς να χρειάζεται να γεμίσουμε το περιβάλλον με άλλα object κάνοντας έτσι και τα script πολύ πιο εύκολα στην σύνταξη. Παρακάτω μπορούμε να δούμε τα assets από τα οποία αποτελείται το Puzzle Button object prefab. Ουσιαστικά, αυτό το αντικείμενο θα είναι η κλειστή κάρτα που θα βρίσκεται στο ταμπλό και με τα scripts τα οποία θα αναλύσουμε στην συνέχεια θα μπορέσουμε να δούμε πως θα αλλάζει η εικόνα στο αριστερό κλικ του ποντικιού καθώς και θα οριστούν τα ζεύγη καρτών.

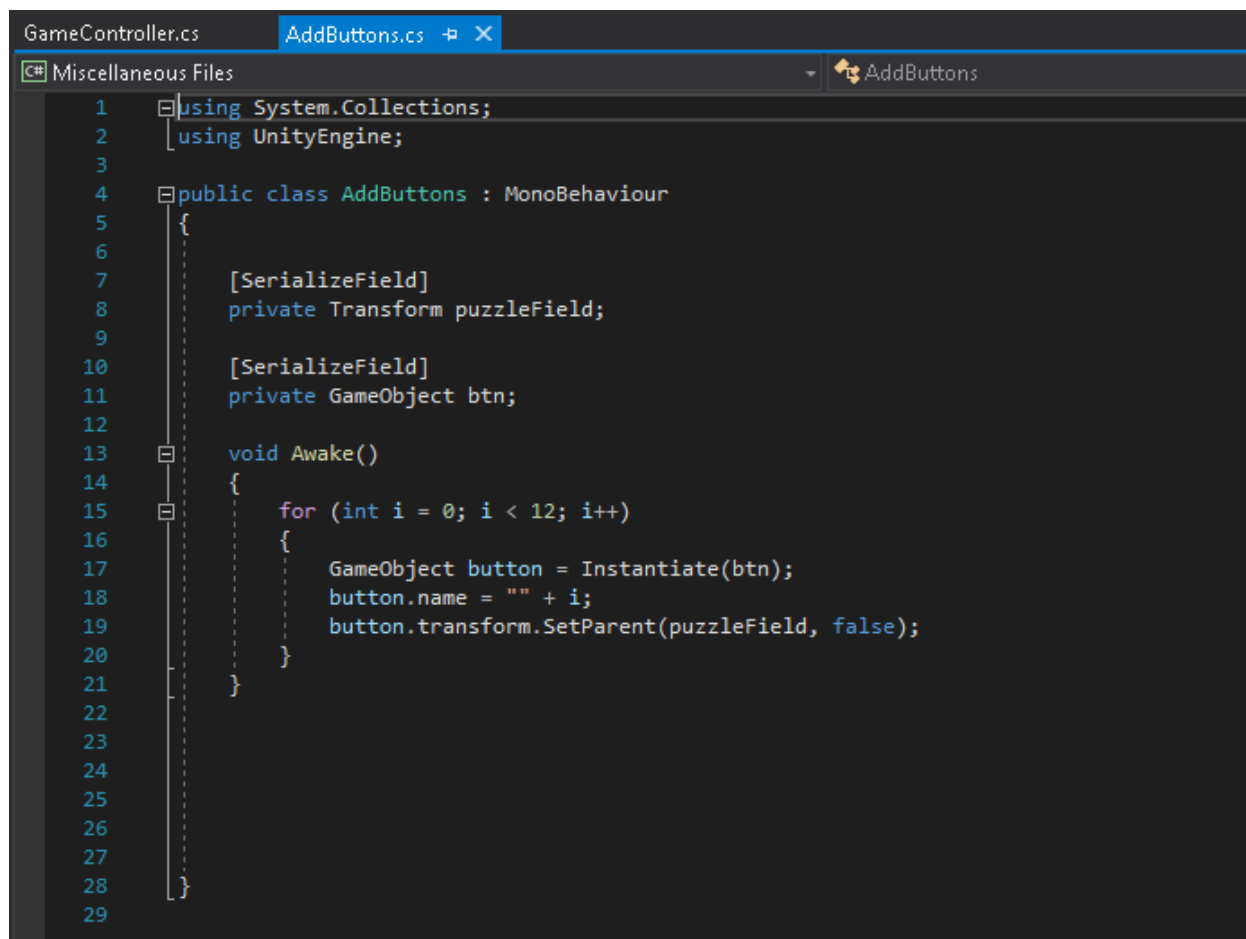


Φυσικά αυτό το βιντεοπαιχνίδι δεν θα μπορούσε να γίνει πραγματικότητα αν δεν υπήρχε το Game Controller object το οποίο στην ουσία είναι ένα κενό αντικείμενο στο περιβάλλον του Unity, χωρίς υπόσταση αλλά, όπως μπορούμε να δούμε στην παρακάτω εικόνα, αποτελείται από 2 scripts σε C#. Το Add Buttons script και το Game Controller script τα οποία θα δούμε πιο αναλυτικά στην συνέχεια καθώς είναι η αρχή και το τέλος του παρόν βιντεοπαιχνιδιού.



Τέλος στα assets έχουμε τα sprites, τις εικόνες δηλαδή που θα βρίσκονται στο πάνω μέρος και στο κάτω μέρος των καρτών. Όπως μπορείτε να δείτε στις παρακάτω εικόνες για το πάνω μέρος των καρτών επέλεξα το λευκό δέντρο με το μαύρο φόντο και για το κάτω μέρος των καρτών και μέσω της βοήθειας ενός script που θα δούμε στη συνέχεια, επιλέχθηκε ένα sheet από πολλά και διαφορετικά sprites τα οποία επιλέγονται τυχαία σε κάθε παιχνίδι και εφόσον έχουμε 12 κάρτες θα χρειαστούμε 6 sprites συνολικά εφόσον θα έχουν ένα κοινό ζευγάρι. Με την επιλογή split που δίνει το Unity για .png εικόνες, χωρίσαμε το sprite sheet σε πολλά μικρά διαφορετικά sprites τα οποία μπορούν να χρησιμοποιηθούν το καθένα ξεχωριστά.

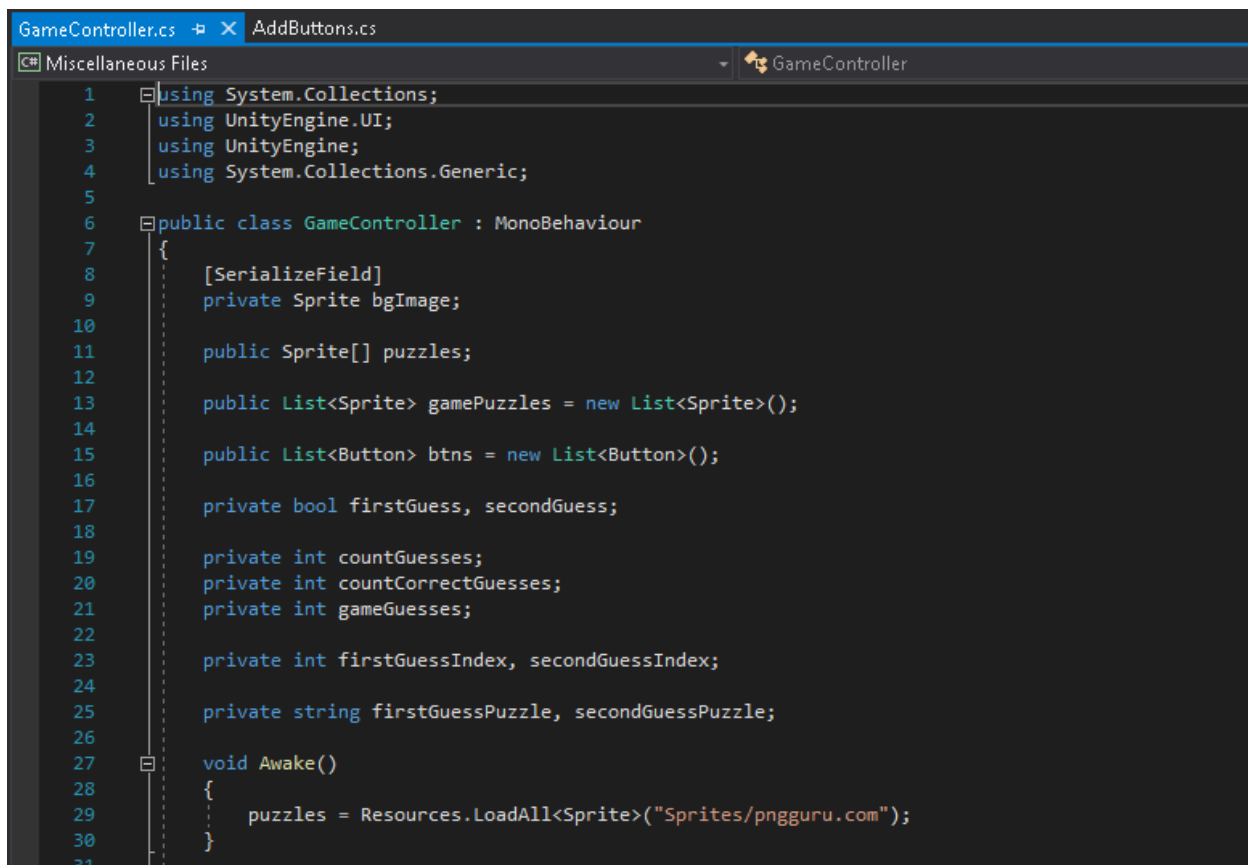
Το πρώτο script ονομαζόμενο AddButtons υλοποιεί το Puzzle Field με διαστάσεις 4x3. Στη συνέχεια κάνει τα prefabs Puzzle Button να μπορούν να αλληλεπιδρούν και τους δίνει ταυτότητα. Παρακάτω μπορείτε να δείτε τον κώδικα αυτούσιο.



```
1 using System.Collections;
2 using UnityEngine;
3
4 public class AddButtons : MonoBehaviour
5 {
6
7     [SerializeField]
8     private Transform puzzleField;
9
10    [SerializeField]
11    private GameObject btn;
12
13    void Awake()
14    {
15        for (int i = 0; i < 12; i++)
16        {
17            GameObject button = Instantiate(btn);
18            button.name = "" + i;
19            button.transform.SetParent(puzzleField, false);
20        }
21    }
22
23
24
25
26
27
28
29 }
```

Στη συνέχεια έχουμε το μεγαλύτερο script που συντάχθηκε και από τα 3 projects, το script GameController. Ας δούμε αρχικά τι κάνει οπτικά αυτό το script. Πρώτα βάζει την εικόνα που θέλουμε στο πάνω μέρος της κάρτας για όλες τις κάρτες. Στη συνέχεια από το sprite sheet που είδαμε παραπάνω παίρνει 6 sprites από 2 φορές το καθένα και τα τοποθετεί τυχαία κάτω από τις κάρτες. Μετέπειτα πρέπει να θέσουμε τις συνθήκες. Όταν δηλαδή γίνεται αριστερό κλικ σε ένα Puzzle Button να μπορούμε να δούμε την εικόνα που βρίσκεται από κάτω και ότι αυτή η εικόνα θα παραμείνει ανοιχτή μέχρι να επιλέξουμε άλλη μία. Αν οι εικόνες είναι ίδιες τότε περιμένει ένα μικρό χρονικό διάστημα και διαγράφονται τα Puzzle Buttons. Αν όχι τότε πάλι περιμένει ένα μικρό χρονικό διάστημα και η κάρτα «κλείνει» και έχουμε την εικόνα που θέσαμε για το πάνω μέρος

της κάρτας. Ο χρόνος που θέτουμε για να περιμένει πριν εξαφανιστούν οι κάρτες ή πριν ξανακλείσουν έχει σκοπό ο παίκτης να έχει αρκετό χρόνο να απομνημονεύσει την κρυμμένη εικόνα. Όταν ο παίκτης βρει όλα τα ζευγάρια έχουμε βάλει συνθήκη να τελειώνει το παιχνίδι και στη συνέχεια μπορεί να παίξει ξανά πατώντας το κουμπί Restart ή να κλείσει την εφαρμογή πατώντας το κουμπί Quit. Οι εικόνες στο κάτω μέρος των καρτών τοποθετούνται τυχαία. Αυτό σημαίνει ότι σχεδόν ποτέ το ένα Puzzle θα είναι ίδιο με κάποιο άλλο. Σε αυτό το σημείο να αναφερθούμε ότι δεν υπάρχει πάνω και κάτω μέρος καρτών, απλά το αναφέρουμε έτσι για την διευκόλυνση στην επεξήγηση. Αυτό που συμβαίνει στην πραγματικότητα είναι ότι η εικόνα που έχουμε στο «πάνω μέρος της κάρτας» αλλάζει στο sprite το οποίο τοποθετήθηκε τυχαία με την αρχή του παιχνιδιού. Όλα τα παραπάνω γίνονται πραγματικότητα με τις void Awake(), void Start(), void GetButtons(), void AddPuzzles(), void AddListeners(), public void PickAPuzzle(), IEnumerator CheckIfThePuzzlesMatch(), void CheckIfTheGameIsFinished() και void Shuffle() που μπορείτε να δείτε παρακάτω αναλυτικά.



```
1 using System.Collections;
2 using UnityEngine.UI;
3 using UnityEngine;
4 using System.Collections.Generic;
5
6 public class GameController : MonoBehaviour
7 {
8     [SerializeField]
9     private Sprite bgImage;
10
11     public Sprite[] puzzles;
12
13     public List<Sprite> gamePuzzles = new List<Sprite>();
14
15     public List<Button> btns = new List<Button>();
16
17     private bool firstGuess, secondGuess;
18
19     private int countGuesses;
20     private int countCorrectGuesses;
21     private int gameGuesses;
22
23     private int firstGuessIndex, secondGuessIndex;
24
25     private string firstGuessPuzzle, secondGuessPuzzle;
26
27     void Awake()
28     {
29         puzzles = Resources.LoadAll<Sprite>("Sprites/pngguru.com");
30     }
31 }
```

```
GameController.cs  AddButtons.cs
[C#] Miscellaneous Files  GameController

30  }
31
32  void Start()
33  {
34      GetButtons();
35      AddListeners();
36      AddGamePuzzles();
37      Shuffle(gamePuzzles);
38      gameGuesses = gamePuzzles.Count / 2;
39  }
40
41  void GetButtons()
42  {
43      GameObject[] objects = GameObject.FindGameObjectsWithTag("PuzzleButton");
44
45      for (int i = 0; i < objects.Length; i++)
46      {
47          btns.Add(objects[i].GetComponent<Button>());
48          btns[i].image.sprite = bgImage;
49      }
50  }
51
```

```
GameController.cs  AddButtons.cs
[C#] Miscellaneous Files  GameController

51
52  void AddGamePuzzles()
53  {
54      int loopers = btns.Count;
55      int index = 0;
56
57      for (int i = 0; i < loopers; i++)
58      {
59          if(index == loopers / 2)
60          {
61              index = 0;
62          }
63
64          gamePuzzles.Add(puzzles[index]);
65
66          index++;
67      }
68  }
69
70  void AddListeners()
71  {
72      foreach (Button btn in btns)
73      {
74          btn.onClick.AddListener(() => PickAPuzzle());
75      }
76  }
77
```

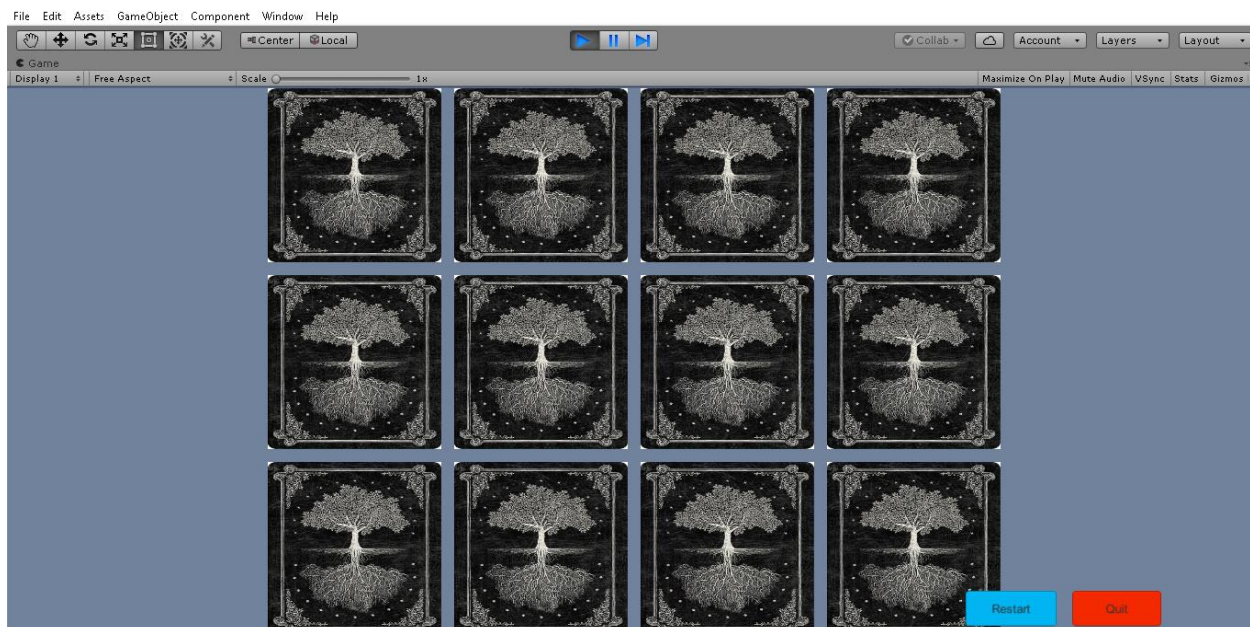
```
GameController.cs ➤ AddButtons.cs
Miscellaneous Files GameController
78 public void PickAPuzzle()
79 {
80     string name = UnityEngine.EventSystems.EventSystem.current.currentSelectedGameObject.name;
81
82     if (!firstGuess)
83     {
84         firstGuess = true;
85
86         firstGuessIndex = int.Parse(UnityEngine.EventSystems.EventSystem.current.currentSelectedGameObject.name);
87
88         firstGuessPuzzle = gamePuzzles[firstGuessIndex].name;
89
90         btns[firstGuessIndex].image.sprite = gamePuzzles[firstGuessIndex];
91     }
92     else if (!secondGuess)
93     {
94         secondGuess = true;
95
96         secondGuessIndex = int.Parse(UnityEngine.EventSystems.EventSystem.current.currentSelectedGameObject.name);
97
98         secondGuessPuzzle = gamePuzzles[secondGuessIndex].name;
99
100        btns[secondGuessIndex].image.sprite = gamePuzzles[secondGuessIndex];
101
102        countGuesses++;
103
104        StartCoroutine(CheckIfThePuzzlesMatch());
105    }
106 }
107
```

```
GameController.cs ➤ AddButtons.cs
Miscellaneous Files GameController
105     }
106 }
107
108 IEnumerator CheckIfThePuzzlesMatch()
109 {
110     yield return new WaitForSeconds(1f);
111
112     if(firstGuessPuzzle == secondGuessPuzzle)
113     {
114         yield return new WaitForSeconds(.5f);
115
116         btns[firstGuessIndex].interactable = false;
117         btns[secondGuessIndex].interactable = false;
118
119         btns[firstGuessIndex].image.color = new Color(0, 0, 0, 0);
120         btns[secondGuessIndex].image.color = new Color(0, 0, 0, 0);
121
122         CheckIfTheGameIsFinished();
123     }
124     else
125     {
126         yield return new WaitForSeconds(.5f);
127
128         btns[firstGuessIndex].image.sprite = bgImage;
129         btns[secondGuessIndex].image.sprite = bgImage;
130     }
131
132     yield return new WaitForSeconds(.5f);
133
134     firstGuess = secondGuess = false;
135 }
136
```

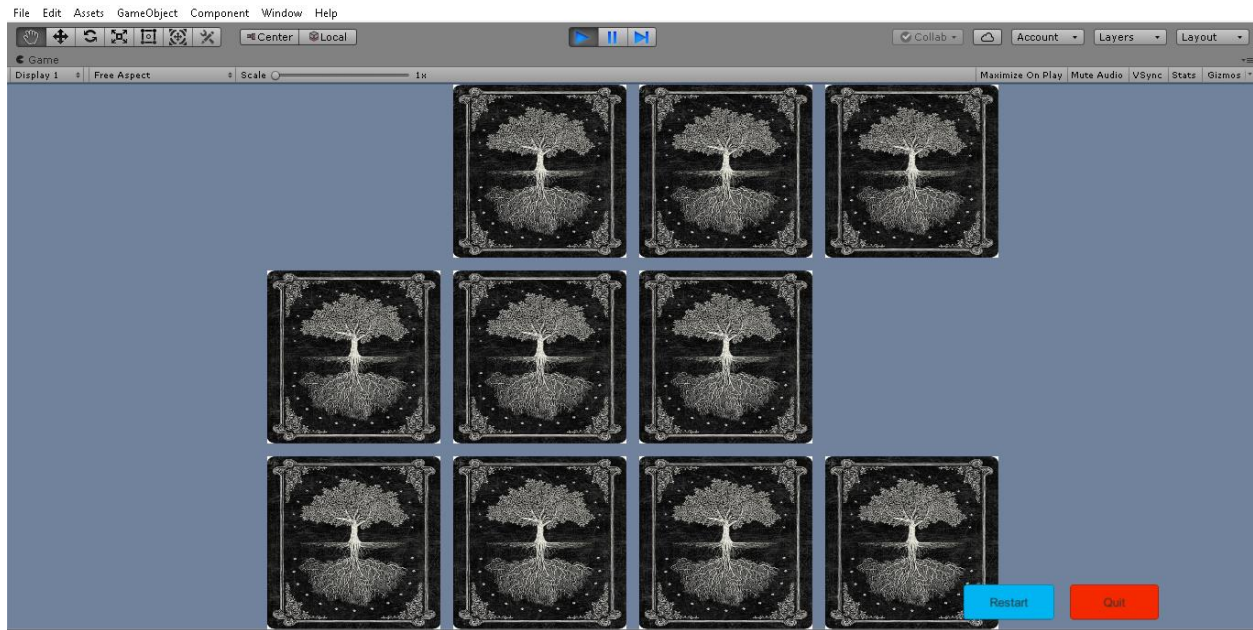
```
GameController.cs  AddButtons.cs
Miscellaneous Files  GameController

135     }
136
137     void CheckIfTheGameIsFinished()
138     {
139         countCorrectGuesses++;
140
141         if(countCorrectGuesses == gameGuesses)
142         {
143             Debug.Log("game finished");
144             Debug.Log("it took you" + countGuesses + " guess(es) to finish the game");
145         }
146     }
147
148     void Shuffle(List<Sprite> list)
149     {
150         for (int i = 0; i < list.Count; i++)
151         {
152             Sprite temp = list[i];
153             int randomIndex = Random.Range(i, list.Count);
154             list[i] = list[randomIndex];
155             list[randomIndex] = temp;
156         }
157     }
158
159 }
```

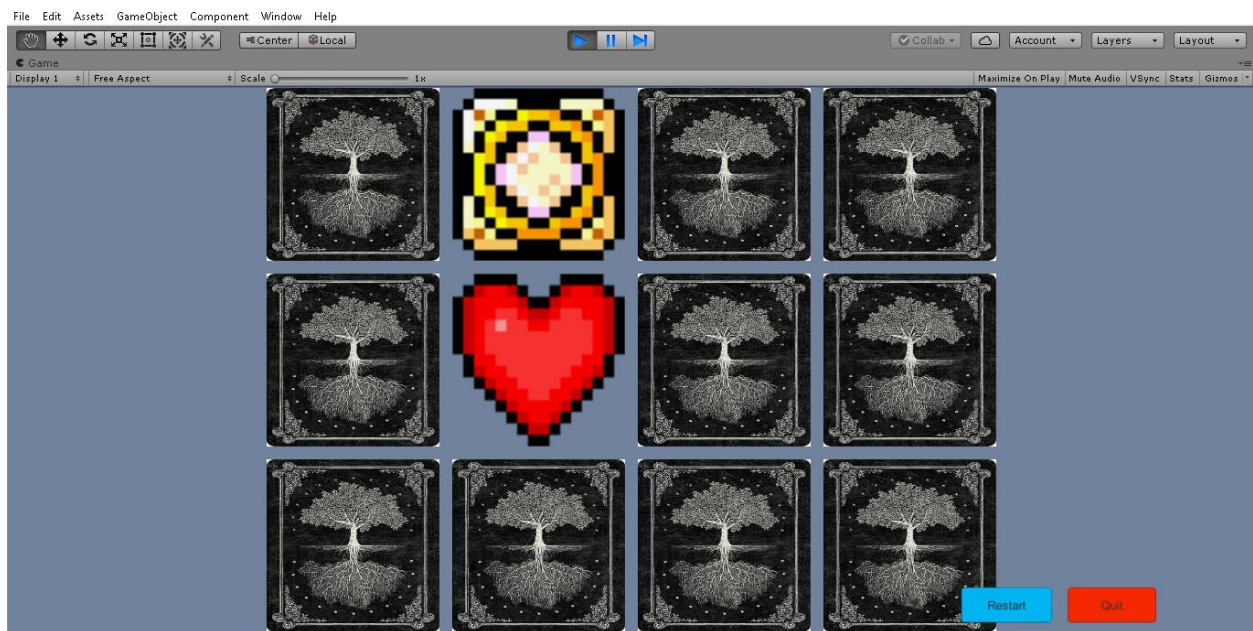
Έχοντας πραγματοποιήσει όλα τα παραπάνω βήματα η αρχική εικόνα του παιχνιδιού είναι αυτή που απεικονίζεται παρακάτω.



Παρακάτω μπορούμε να δούμε ένα παράδειγμα μετά από μία σωστή μαντεψιά.



Και στην τελευταία εικόνα μπορούμε να δούμε πώς απεικονίζονται τα sprites όταν ο παίκτης πατήσει πάνω στο Puzzle Button με αριστερό κλικ.



Dropbox Link:

<https://www.dropbox.com/sh/ichuvpgctjyh7cj/AADdY3awvJYipWQoY5E4Fmlya?dl=0>

Πρώτες Εκδόσεις

Οι παραπάνω εφαρμογές έφτασαν στην τελική τους μορφή μετά από μία σειρά αλλαγών και ενημερώσεων. Καθοριστικοί παράγοντες για τις αλλαγές τους ήταν:

- Οι αντιδράσεις του focus group
- Bugs και glitches των εφαρμογών
- Η δυσκολία των πρώτων εκδόσεων
- Η έλλειψη σημαντικών χαρακτηριστικών

Η εφαρμογή Cubeathlon στην πρώτη της μορφή δεν είχε τις επιλογές για επανεκκίνηση και κλείσιμο μέσω των κουμπιών Restart και Quit αντίστοιχα. Οι αποστάσεις μεταξύ των εμποδίων ήταν πολύ μικρές μεταξύ τους καθιστώντας το αδύνατο για χρήση από το focus group γι αυτό και έδρασα επί του θέματος κάνοντας το πιο εύκολο. Ο χειρισμός του παίκτη έγινε πιο ομαλός και όχι στιβαρός όπως ήταν στην αρχή καθώς και αφαιρέθηκε η επιτάχυνση του παίκτη με το πέρασμα των επιπέδων. Τέλος τα 10 επίπεδα έγιναν 13 δίνοντας την δυνατότητα για περαιτέρω ενασχόληση με την εφαρμογή.

Η εφαρμογή Skillxagon υπέστη τις λιγότερες αλλαγές. Δεν πραγματοποιήθηκε καμία αλλαγή πέρα από την προσθήκη των επιλογών της επανεκκίνησης και τερματισμού της εφαρμογής με την εισαγωγή των κουμπιών Restart και Quit αντίστοιχα.

Η εφαρμογή Memory Puzzle Game υπέστει σοβαρές αλλαγές καθώς υπήρχαν πολλά λάθη στον κώδικα καθιστώντας το αδύνατο για χρήση. Στη συνέχεια υπήρξε πρόβλημα με τις εικόνες των καρτών και έπρεπε να δουλευτεί απ' την αρχή με καινούργιες εικόνες και sprite sheets. Φυσικά για άλλη μια φορά προστέθηκαν οι επιλογές της επανεκκίνησης της εφαρμογής και το κλείσιμο της με την προσθήκη των κουμπιών Restar και Quit αντίστοιχα. Τα ζεύγη καρτών από 3 έγιναν 6, δηλαδή από 6 κάρτες στο ταμπλό πλέον υπάρχουν 12, αυξάνοντας λίγο την δυσκολία αλλά δίνοντας την δυνατότητα στον χρήστη να ασχοληθεί περισσότερο με την εφαρμογή. Τέλος ιδιαίτερη προσοχή δόθηκε στην τυχαιότητα των καρτών με την έναρξη του παιχνιδιού. Σαν developer ήθελα κανένα παιχνίδι να μην είναι ίδιο με το προηγούμενο και πράγματι οι

πιθανότητες όλες οι κάρτες να βρίσκονται στο ίδιο σημείο με το προηγούμενο παιχνίδι είναι απειροελάχιστες.

Μελλοντικές Πιθανές Αναβαθμίσεις

Βεβαίως η ανάπτυξη ενός βιντεοπαιχνιδιού ή μίας εφαρμογής πάντα έχει περιθώρια βελτίωσης. Σε αυτήν την υποενότητα λοιπόν θα δούμε μερικά πιθανά σενάρια ανάπτυξης των project αυτής της εργασίας για το καθένα ξεχωριστά.

Ξεκινώντας με την εφαρμογή Cubathlon. Ένα πιθανό σενάριο ανάπτυξης θα μπορούσε να είναι η συλλογή αντικειμένων (collectables) από συγκεκριμένες δίοδους της πίστας δίνοντας στον παίκτη το αίσθημα ολοκλήρωσης καθώς και προσθέτοντας μία επιλεκτική δυσκολία, προτρέποντάς τον να συνεχίσει να ασχολείται με την εφαρμογή. Μία προσθήκη θα μπορούσε επίσης να είναι ένας μηχανισμός δίνοντας την ιδιότητα στον χαρακτήρα που χειρίζεται ο παίκτης να αναπηδά για να προσπεράσει μερικά εμπόδια τα οποία για παράδειγμα θα μπορούσαν να καλύπτουν όλο το πλάτος της πίστας δίνοντας του καμία άλλη επιλογή. Στο ίδιο πλαίσιο, θα μπορούσε να υπάρχει αυξημένη δυσκολία στα επίπεδα κάνοντας τον χαρακτήρα να κινείται όλο και γρηγορότερα σε κάθε επίπεδο αναγκάζοντας τον παίκτη να δρα πιο γρήγορα. Τέλος θα μπορούσαν να προστεθούν περισσότερες πίστες προσφέροντας ώρες απασχόλησης καθώς και η επιλογή να σώσει ο παίκτης την πρόοδο του και να συνεχίσει από το τρέχων επίπεδο στο οποίο έφτασε.

Όσον αφορά την εφαρμογή Skillxagon. Θα μπορούσαν να προστεθούν πολλά επίπεδα αυξανόμενης δυσκολίας. Ο τρόπος προόδου των επιπέδων θα μπορούσε κάλλιστα να είναι τα πόσα εξάγωνα ο παίκτης θα μπορέσει να προσπελάσει για να μπορέσει να προχωρήσει. Η αύξηση της δυσκολίας θα μπορέσει να επιτευχθεί με την αύξηση της ταχύτητας του ρυθμού δημιουργίας των εξαγώνων. Σε αρχικά επίπεδα για παράδειγμα τα εξάγωνα θα μπορούσαν να έχουν 2 πλευρές ανοιχτές και σε μετέπειτα επίπεδα μόνο μια. Ένας τελευταίος τρόπος αύξησης δυσκολίας θα μπορούσε να είναι η περιστροφή

της κάμερας. Με αυτόν τον τρόπο αύξησης δυσκολίας όμως το βιντεοπαιχνίδι δεν θα ήταν φιλικό για άτομα με επιληψία.

Τέλος για την εφαρμογή Memory Puzzle Game θα μπορούσαν να προστεθούν για άλλη μία φορά περισσότερα επίπεδα με αυξανόμενη δυσκολία. Θα μπορούσε για παράδειγμα ο παίκτης να αρχίζει με 3 ζευγάρια κάρτες στο πρώτο επίπεδο και να αυξάνεται ο αριθμός των ζευγαριών κατά 1 με το κάθε πέρασμα επιπέδου. Ένας ακόμη ενδιαφέρον τρόπος αύξησης δυσκολίας στα τελευταία επίπεδα του παιχνιδιού θα μπορούσε να είναι στις 3 αποτυχημένες προσπάθειες εύρεσης ζεύγους, οι κάρτες να ανακατεύονται τυχαία στο ταμπλό.

Σε όλες τις εφαρμογές επίσης θα μπορούσε να πραγματοποιηθεί μια συνεργασία με έναν animator έτσι ώστε να είναι πιο όμορφες. Στην εφαρμογή Cubeathlon για παράδειγμα θα μπορούσε ο κύβος που είναι ο παίκτης να γίνει ένα αυτοκίνητο, οι κύβοι που είναι τα εμπόδια να γίνουν κώνοι ή οδικά φράγματα και προφανώς το τερματικό μιας πίστας να γίνει μια γραμμή αγώνων ταχύτητας.

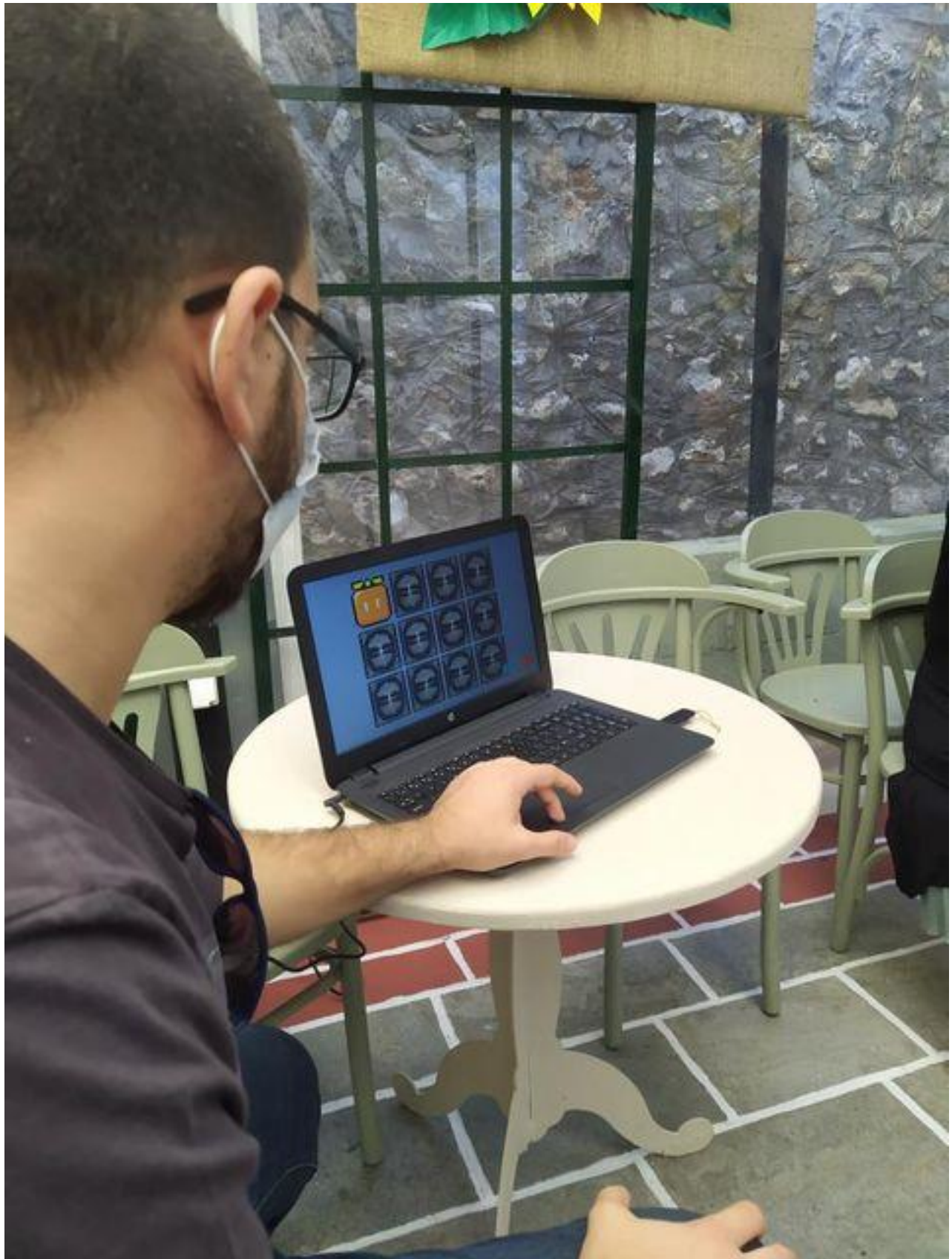
Κεφάλαιο 5: Παρουσίαση των εφαρμογών σε άτομα με ήπιες γνωστικές διαταραχές

Μετά την ολοκλήρωση των εφαρμογών στις τελικές τους μορφές πραγματοποιήθηκε η παρουσίασή τους σε 2 άτομα αρχικά έτσι ώστε να μπορέσουν να διαπιστωθούν αποτελέσματα για τις εν λόγω εφαρμογές. Το ένα άτομο πάσχει από διαγνωσμένη έλλειψη συγκέντρωσης και το άλλο από άνοια. Αρχικά τους δόθηκε λεπτομερής περιγραφή με τους κανόνες και τον χειρισμό των εφαρμογών. Στη συνέχεια τους δόθηκε μία περίοδος προσαρμογής στο περιβάλλον της κάθε εφαρμογής καθώς μπόρεσαν να τα δουν στην πράξη. Και τα 2 άτομα ήταν ιδιαίτερα ενθουσιώδη με την διάδραση των βιντεοπαιχνιδιών. Παρατηρήθηκαν βελτιώσεις στον τρόπο παιχνιδιού και στα 2 άτομα από την δεύτερη κιόλας προσπάθεια τους καθώς κάθε φορά έφταναν σε μεγαλύτερο επίπεδο χωρίς να χάσουν. Τα άτομα στη συνέχεια έδειξαν την διάθεση να συνεχίσουν να ασχολούνται με αυτά τα βιντεοπαιχνίδια και δεν δυσανασχέτησαν καθ' όλη την διαδικασία. Μετά το τέλος της ενασχόλησης τους με τις εφαρμογές έδειξαν ιδιαίτερη ζωντάνια και όχι αδρανοποίηση.

Στο πλαίσιο της πτυχιακής μου εργασίας με κάλεσαν από την Ένωση Κυριών Δράμας – Σπίτι Ανοιχτής Φιλοξενίας να πάρω μέρος σε μία ημερίδα, με οκτώ άτομα από την τρίτη ηλικία με αρχές Alzheimer's, με αφορμή την παγκόσμια ημέρα κατά της νόσου του Alzheimer's η οποία πραγματοποιήθηκε στις 25/9/2020 στο εργαστήριο δημιουργικής γραφής της Ένωσης Κυριών Δράμας – Σπίτι Ανοιχτής Φιλοξενίας υπό την επίβλεψη του στελέχους, Όλγας Βλαχοπούλου.

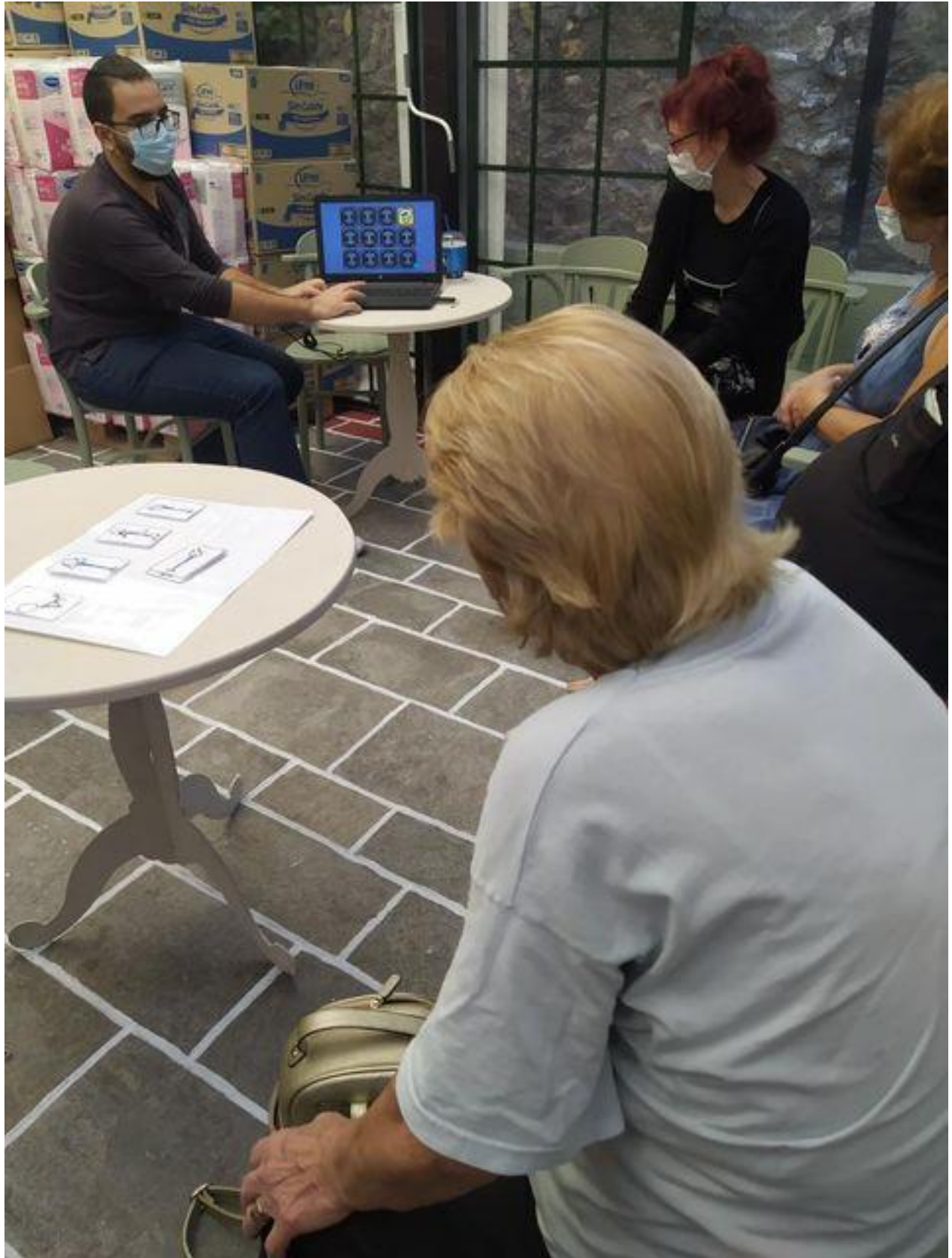
Σκοπός μου ήταν να τους παρουσιάσω τις μελέτες από το Πανεπιστήμιο του Μόντρεαλ αλλά και από το Τμήμα Βιοϊατρικής του Διεθνές Ινστιτούτου Επιστημών Ζωής της Ινδονησίας. Τους εξήγησα πώς η ενασχόληση με τέτοιου είδους εφαρμογές μπορεί να δείξει σημαντικά αποτελέσματα όπως αναφέρεται και στις δύο έρευνες και ότι συμβάλλει στην δημιουργία φαιάς ουσίας στον εγκέφαλο. Στη συνέχεια, όπως και με τα προηγούμενα δύο άτομα, τους παρουσίασα τα βιντεοπαιχνίδια και τους εξήγησα τον χειρισμό, τον σκοπό και τους κανόνες. Τα άτομα της τρίτης ηλικίας ήταν ιδιαίτερα

ενθουσιώδη με την όλη κατάσταση όταν συνειδητοποίησαν ότι θα μπορέσουν να τα δουν στην πράξη. Ένας ένας λοιπόν και τηρώντας τα μέτρα ασφαλείας λόγω της πανδημίας του COVID-19, έπαιρναν το λάπτοπ και με την πληρη καθοδήγησή μου σε κάθε βήμα έπαιζαν τα βιντεοπαιχνίδια. Για ακόμη μία φορά τα άτομα έδειξαν βελτίωση από την δεύτερη και τρίτη κιόλας προσπάθεια φτάνοντας σε σημείο πιο μακριά από το προηγούμενο καθώς και μια ζωντάνια και θέληση να φτάσουν σε επόμενο επίπεδο. Τόσο που στο τέλος με ρώτησαν προσωπικά αν τα εγγόνια τους μπορούν να τις βρουν για να μπορέσουν να ασχοληθούν και στο σπίτι. Η ημερίδα στέφθηκε με απόλυτη επιτυχία και τα άτομα της τρίτης ηλικίας καθώς και τα στελέχη της Ένωσης Κυριών Δράμας – Σπίτι Ανοιχτής Φιλοξενίας με ευχαρίστησαν που μπόρεσα να διαθέσω τον χρόνο μου για αυτόν τον σκοπό. Παρακάτω μπορείτε να δείτε φωτογραφικό υλικό από την ημερίδα.











Τα αποτελέσματα ήταν πολύ θετικά και από τα δύο test audiences καθιστώντας αυτή την μελέτη και την δημιουργία των εφαρμογών επιτυχής καθώς τα άτομα επιθυμούσαν να ασχοληθούν επιπλέον με αυτές τις εφαρμογές αλλά και αποκτώντας μία ζωντάνια με την ενασχόληση τους με αυτές της εφαρμογές. Ακόμα και αυτοί που ήταν επιφυλακτικοί στην αρχή στη συνέχεια μπήκαν στο πνεύμα καθώς τους κέντρισαν το ενδιαφέρον οι εφαρμογές και ο τρόπος παιχνιδιού αλλά και γνωρίζοντας ότι αυτές οι εφαρμογές συμβάλλουν στην αντιμετώπιση των νόσων που αντιμετωπίζουν καθημερινά.

Κεφάλαιο 6: Τελικά Συμπεράσματα

Κλείνοντας την πτυχιακή εργασία πρέπει να αναφερθεί για άλλη μία φορά η σημαντικότητα των δύο κλάδων μαζί αλλά και ξεχωριστά και το πόσο ραγδαία αποτελέσματα μπορούν να επιφέρουν όταν συνεργάζονται. Αναφέρομαι φυσικά στην ανάπτυξη εφαρμογών και στην βιοπληροφορική, δύο τομείς που τα τελευταία χρόνια ειδικά έχουν γνωρίσει πολύ σημαντική ανάπτυξη. Μέσα από αυτή την πτυχιακή εργασία μπόρεσα κι εγώ να αναγνωρίσω ακόμη περισσότερο τα αποτελέσματα που μπορούν να επιφέρουν η κάθε μία επιστήμη από μόνη της αλλά και οι δύο μαζί. Η ενασχόληση με εφαρμογές συγκεκριμένων προδιαγραφών, όπως οι παρούσες εφαρμογές αυτής της εργασίας, για άτομα με ήπια γνωστικά προβλήματα μαζί με μία σωστή και στοχευμένη φαρμακευτική αγωγή πιστεύω ότι θα μπορούσαν να κάνουν τεράστια βήματα στην καταπολέμηση τέτοιων νόσων βοηθώντας τα άτομα τα οποία υποφέρουν. Τα αποτελέσματα των ερευνών διάφορων πανεπιστημίων ανά τον κόσμο αλλά και αυτής της εργασίας είναι δείκτης πως πρέπει να ασχοληθούμε περισσότερο με αυτές τις επιστήμες και τεχνολογίες.

Τέλος θα αναφερθούν μερικές πιθανές καλές πρακτικές από την αναμενόμενη χρήση αυτών ή παρόμοιων εφαρμογών καθώς και την καινοτομία τους σε σχέση με παλαιότερες πρακτικές ή πρακτικές που εφαρμόζονται αυτή την στιγμή. Μία πολύ καλή εφαρμογή των projects θα ήταν σε οίκους ευγηρίας για άτομα που πάσχουν από ήπιες γνωστικές διαταραχές. Κάποια στιγμή μες την μέρα θα μπορούσαν να ασχολούνται με τις εφαρμογές αυτές οξύνοντας τις αισθήσεις τους και παράγοντας την φαιά ουσία που έχουν ανάγκη. Τα αποτελέσματα μπορεί να φανούν σύντομα αλλά σε πολλές περιπτώσεις θα φανούν αργότερα στην πορεία. Το κλειδί είναι η μόνιμη και εντατική ενασχόληση έτσι ώστε να υπάρχει εγρήγορση και η υπεκφυγή κατάσταση λήθης του εγκεφάλου. Πολλοί οίκοι ευγηρίας μέχρι και τώρα χρησιμοποιούν σταυρόλεξα και επιτραπέζια, τα οποία παρ' όλο που δείχνουν κάποια αποτελέσματα δεν προσφέρουν την διάδραση και την αίσθηση του κατορθώματος και ολοκλήρωσης. Το αρνητικό στα επιτραπέζια είναι ότι τις περισσότερες φορές απαιτούνται 2 ή περισσότερα άτομα για να είναι δυνατή η ενασχόληση με αυτά. Στην περίπτωση των εφαρμογών δεν υφίστανται

τέτοια προβλήματα καθώς το κάθε άτομο μπορεί να ασχοληθεί μόνο του καθώς και να δημιουργηθούν πίνακες με την καλύτερη επίδοση έτσι ώστε να τους ωθήσουν να προσπαθήσουν περισσότερο βοηθώντας τους εαυτούς τους με αυτόν τον τρόπο. Μία ακόμη αναμενόμενη πρακτική αυτών ή παρόμοιων εφαρμογών είναι η χρήση τους σε κλινικές που φροντίζουν άτομα με άνοια ή Alzheimer's καθώς είναι δομημένες έτσι ώστε να έχουν καλύτερη δράση σε άτομα που πάσχουν από αυτές τις νόσους και σε συνδυασμό με μία σωστή φαρμακευτική αγωγή θα μπορούσαν να δείξουν ραγδαία θετικά αποτελέσματα. Επιπρόσθετα, δεν θα ήταν απίθανη η χρήση των εφαρμογών σε σχολεία για παιδιά που πάσχουν από έλλειψη συγκέντρωσης στο πλαίσιο των διδακτικών ωρών της πληροφορικής ή του ολοήμερου.

Όπως πάντα μιλάμε για εφαρμογές, εφαρμογές οι οποίες μπορούν μονίμως να αναπτύσσονται και να προσαρμόζονται στις ανάγκες μίας ομάδας ανθρώπων ή του κάθε ατόμου ξεχωριστά. Οι ίδιες εφαρμογές της παρούσας πτυχιακής εργασίας μπορούν να εκδοθούν και σε μορφή που να υποστηρίζονται από λογισμικό Android και IOS για κινητά τηλέφωνα και όχι απαραίτητα για PC. Άρα δύο πολύ θετικά στοιχεία είναι η συνεχής ενημέρωση και ανάπτυξη των εφαρμογών αλλά και η προσβασιμότητα σε άλλες συσκευές (multiplatform). Επίσης ένας ακόμα κλάδος που τώρα κάνει τα καινούρια του βήματα είναι η επταυξημένη πραγματικότητα. Αν και επικίνδυνη για άτομα που υποφέρουν από επιληπτικά επεισόδια, η επταυξημένη πραγματικότητα είναι μία τεχνολογία που στην περίπτωση μας θα μπορούσε να συνδυαστεί με την 3D εφαρμογή Cubeathlon και να προκύψει μία 3D εφαρμογή επταυξημένης πραγματικότητας με την χρήση του Oculus ή του VR, τα οποία είναι εξαρτήματα/συσκευές που κάνουν δυνατή την χρήση της επταυξημένης πραγματικότητας μέσω του PC ή Sony Playstation αντίστοιχα. Το αρνητικό είναι ότι αυτές οι συσκευές είναι αρκετά ακριβές και δεν είναι προσβάσιμες στο ευρύ κοινό ακόμα, το οποίο είναι και αρνητικό στις υπόλοιπες πρακτικές που αναφέραμε παραπάνω, το κόστος. Πολλοί οίκοι ευγηρίας ή κλινικές μπορεί να μην έχουν την οικονομική δυνατότητα να επενδύσουν σε ηλεκτρονικούς υπολογιστές ή tablets για την χρήση των εφαρμογών. Γι αυτόν τον λόγο και οι εφαρμογές είναι όσο καλύτερα βελτιστοποιημένες γίνεται έτσι ώστε να μην απαιτούνται ακριβά μηχανήματα με μεγάλη μνήμη RAM, κάρτα γραφικών κλπ. Με ένα πολύ απλό λάπτοπ ο καθένας μπορεί να εκτελέσει επιτυχώς αυτές τις εφαρμογές χωρίς κανένα

πρόβλημα. Όπως μπορούμε να δούμε δεν υπάρχει τερματικό στάδιο σε αυτούς τους τομείς, το μόνο που χρειάζεται είναι ευρηματικότητα, ώρες εργασίας και πάθος για την ενασχόληση από τους συμμετέχοντες.

Πηγές

- Does Video Gaming Have Impacts on the Brain: Evidence from a Systematic Review <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6826942/>
- Health Benefits of Digital Videogames for Older Adults: A Systematic Review of the Literature <https://www.liebertpub.com/doi/full/10.1089/g4h.2012.0046>
- [Videogames and Health Improvement: A Literature Review of Randomized Controlled Trials](#)
- Playing Video Games Could Help Prevent Alzheimer's <https://www.esquire.com/lifestyle/health/a14379049/video-games-prevent-alzheimers/>
- Microsoft Visual Studio https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- Unity (Game Engine) [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- Early History of Video games https://en.wikipedia.org/wiki/Early_history_of_video_games
- C Sharp https://el.wikipedia.org/wiki/C_Sharp
- Microsoft Visual Studio <https://visualstudio.microsoft.com/>

- Microsoft Visual Studio Tutorial
<https://visualstudio.microsoft.com/vs/getting-started/>
- Microsoft Visual Studio: “Everything you need to know”
<https://www.edureka.co/blog/visual-studio-tutorial/>
- Unity Engine
<https://unity.com/>
- Unity Engine Tutorials
<https://learn.unity.com/>
- Brackeys – Become a Developer
<https://brackeys.com/>
- C Sharp Tutorial
<https://www.w3schools.com/cs/>
- Unity Asset Store
<https://store.unity.com/>
- Bioinformatics
<https://en.wikipedia.org/wiki/Bioinformatics>
- Bioinformatics the application of computational tools to organize, analyze, understand, visualize and store information associated with biological macromolecules
<https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/bioinformatics>
- How to Write your Thesis
https://www.ldeo.columbia.edu/~martins/sen_sem/thesis_org.html
- Creating and Managing a Computer Game Project
<https://www.instructables.com/Creating-and-managing-a-computer-game-project/>

- Learning Science Through Computer Games and Simulations
<https://www.ics.uci.edu/~wscacchi/GameLab/Recommended%20Readings/Learning-Science-Games-2011.pdf>