



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΟΒΑΡΟΥ
ΠΑΙΧΝΙΔΙΟΥ ΓΙΑ ΤΟΝ ΧΩΡΟ ΤΗΣ ΥΓΕΙΑΣ

ΠΑΠΑΙΩΑΝΝΟΥ ΠΑΝΑΓΙΩΤΗΣ (ΑΜ 2192)
ΣΠΥΡΗΣ ΧΡΗΣΤΟΣ (ΑΜ 2228)
(πρώην Τμήματος ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ, ΤΕΙ ΔΥΤ. ΕΛΛΑΔΑΣ)

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: ΚΟΥΤΡΑΣ ΑΘΑΝΑΣΙΟΣ

ΠΑΤΡΑ, 2022

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΜΗ ΛΟΓΟΚΛΟΠΗΣ

Βεβαιώνουμε, ότι είμαστε συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, γίνεται αναφορά στις όποιες πηγές εκ των οποίων έγινε χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Ακόμα δηλώνουμε ότι αυτή η γραπτή εργασία προετοιμάστηκε από εμάς προσωπικά και αποκλειστικά και ειδικά για την συγκεκριμένη πτυχιακή εργασία και ότι θα αναλάβουμε πλήρως τις συνέπειες εάν η εργασία αυτή αποδειχθεί ότι δεν μας ανήκει.

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ 1

ΑΜ

ΥΠΟΓΡΑΦΗ

ΠΑΠΑΙΩΑΝΝΟΥ ΠΑΝΑΓΙΩΤΗΣ

2192



ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ 2

ΑΜ

ΥΠΟΓΡΑΦΗ

ΣΠΥΡΗΣ ΧΡΗΣΤΟΣ

2228



ΠΕΡΙΛΗΨΗ

Το αντικείμενο στο οποίο επικεντρώνεται η παρούσα πτυχιακή εργασία, είναι ο σχεδιασμός και η ανάπτυξη ενός παιχνιδιού. Η κατηγορία που θα ασχοληθούμε είναι τα serious games. Ανήκουν σε μια πιο ειδική περίπτωση παιχνιδιών, τα οποία κυρίως εστιάζουν στο κομμάτι της υγείας. Αρχικά, παρουσιάζουμε μια ιστορική αναφορά σχετικά με τα serious games μαζί με παραδείγματα τέτοιου είδους εφαρμογών που δημιουργήθηκαν και συνεχίζουν να δημιουργούνται.

Το παιχνίδι που παρουσιάζουμε στα πλαίσια της εργασίας μας, υλοποιήθηκε με τη χρήση της μηχανής παιχνιδιού Unity και τη γλώσσα προγραμματισμού C#. Πρωτεύον σκοπός του παιχνιδιού είναι ο κάθε χρήστης - ασθενής να δοκιμαστεί / ασκηθεί σε μια σειρά παιχνιδιών που σκοπό έχουν την ενδυνάμωση των ικανοτήτων του χρήστη στον τομέα της μνήμης, καθώς και των αντανακλαστικών του.

Εκτός από την μηχανή παιχνιδιών Unity, χρησιμοποιήσαμε και τα παρακάτω προγράμματα για την υποβοήθηση της σχεδίασης και υλοποίησης των παιχνιδιών μας:

- Visual Studio(IDE)
- Plastic SCM
- Assets or extensions
- Adobe Photoshop
- Nvidia ShadowPlay

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ ΣΤΑ SERIOUS GAMES ΚΑΙ ΕΚΤΕΝΗΣ ΑΝΑΛΥΣΗ ΤΟΥΣ	12
1.1	ΕΙΣΑΓΩΓΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	12
1.2	ΚΑΤΗΓΟΡΙΕΣ SERIOUS GAMES	16
1.3	Εργαλεία υλοποίησης παιχνιδιών. Unity, visual studio και συνδεση μεταξύ τους.....	21
1.4	Βασική Ορολογία της μηχανής Unity.....	26
2	ΣΧΕΔΙΑΣΜΟΣ, ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ PROJECT “BRAIN EXERCISE”	27
2.1	ΧΡΗΣΙΜΟΤΗΤΑ ΕΝΤΑΞΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ.....	27
2.1.1	ΕΙΣΑΓΩΓΗ ΣΤΗΝ SQL	27
2.1.2	Πίνακες που χρησιμοποιήθηκαν κατά την εκπόνηση της εργασίας .	28
2.1.3	Ερωτήματα που χρησιμοποιούνται από την εφαρμογή μας	30
2.2	Ανάπτυξη παιχνιδιού ανά Σκηνή.....	38
2.2.1	DOCTOR-REGISTER	38
2.2.2	PATIENT-LOGIN	39
2.2.3	MAIN MENU	43
2.3	ΔΙΑΘΕΣΙΜΑ ΠΑΙΧΝΙΔΙΑ	52
2.3.1	Ταίριαξε τα όμοια.....	52
2.3.2	Αποφύγετε τα εμπόδια	55
2.3.3	Κλικ στο κουτί.....	57
2.3.4	Εκπαίδευση όρασης	60
3	ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ WPF ΜΕ ΣΚΟΠΟ ΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ DEVELOPER-CLIENT	61
3.1	ΕΙΣΑΓΩΓΗ ΣΤΟ WPF.....	61
3.2	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΕΦΑΡΜΟΓΗΣ	62

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1 America’s Army.....	13
Εικόνα 2 Λίστα serious games από το 1980-2009 και μέχρι και σήμερα	15
Εικόνα 3 Title Screen from “The Bradley Trainer” και εικόνες μέσα από το παιχνίδι	17
Εικόνα 4 Preview of the game “Versailles”	18
Εικόνα 5 Preview of the “The Oregon Trail”	19
Εικόνα 6 Game Cover and Screenshot of the game “Captain Novolin”	20
Εικόνα 7 Unity Logo	21
Εικόνα 8 Screenshot of the Unity Asset Store.....	22
Εικόνα 9 Screenshot from “Cuphead”	23
Εικόνα 10 Artwork from “Hollow Knight”	23
Εικόνα 11 Screenshot from “Phasmophobia”	23
Εικόνα 12 Game Cover of “Punishing Gray Raven”	24
Εικόνα 13 Logo of the Visual Studio	24
Εικόνα 14 Screenshot of how the Visual Studio(IDE)	25
Εικόνα 15 Screenshot of the Interface of Unity	26
Εικόνα 16 Explanatory graphic diagram of the SQL	27
Εικόνα 17 Screenshot from the table “DoctorPass”	28
Εικόνα 18 Screenshot from the table “Game”	28
Εικόνα 19 Screenshot from the table “Gameplay”	29
Εικόνα 20 Screenshot from the table “Player”	29
Εικόνα 21 Screenshot from the “Doctor-Register” window.....	38
Εικόνα 22 Screenshot from the “Patient-Login” window	39
Εικόνα 23 Screenshot from the “Δημιουργία Νέου Χρήστη” window	41
Εικόνα 24 Screenshot of the main menu of the game	43
Εικόνα 25 Screenshot from the Panel with the Categorized Games	43
Εικόνα 26 Screenshot from the button “Ταίριαξε τα όμοια”	44
Εικόνα 27 Screenshot from the buttons “Αποφύγετε τα εμπόδια”, “Κλικ στο κουτί”	44
Εικόνα 28 Screenshot from the button “Εκπαίδευση Όρασης”	45
Εικόνα 29 Screenshot from the Statistics Menu	46
Εικόνα 30 Screenshot for Example 1	48
Εικόνα 31 Screenshot for Example 2	49
Εικόνα 32 Screenshot of the Console Panel	50
Εικόνα 33 Screenshot with the 3 controls grids for the Console.....	50
Εικόνα 34 Screenshot of the Console’s Input Field	50
Εικόνα 35 Screenshot of some commands	51
Εικόνα 36 Screenshot of results from the table “Gameplay”	51
Εικόνα 37 Screenshot #1 from the Game “Ταιριάξτε τα όμοια”	53
Εικόνα 38 Screenshot #2 from the Game “Ταιριάξτε τα όμοια”	53
Εικόνα 39 Screenshot #3 from the Game “Ταιριάξτε τα όμοια”	53
Εικόνα 40 Screenshot #4 from the Game “Ταιριάξτε τα όμοια”	54

Εικόνα 41 Screenshot #5 from the Game “Ταιριάξτε τα όμοια”	54
Εικόνα 42 Screenshot #1 from the Game “Αποφύγετε τα εμπόδια”	55
Εικόνα 43 Screenshot #2 from the Game “Αποφύγετε τα εμπόδια”	57
Εικόνα 44 Screenshot #1 from the Game “Κλικ στο κουτί”	58
Εικόνα 45 Screenshot #2 from the Game “Κλικ στο κουτί”	60
Εικόνα 46 Screenshot της Αξιολογησης.....	62
Εικόνα 47 Example Screenshot	66
Εικόνα 48 Screenshot of the Gmail example	67
Εικόνα 49 Screenshot of the password prompt	68

ΕΥΡΕΤΗΡΙΟ ΑΠΟΣΠΑΣΜΑΤΩΝ ΚΩΔΙΚΑ

Code Snippet 1 Δημιουργία Πίνακα Player.....	30
Code Snippet 2 Δημιουργία Πίνακα Gameplay	30
Code Snippet 3 Δημιουργία Πίνακα Game	31
Code Snippet 4 Δημιουργία Πίνακα Doctor Pass.....	31
Code Snippet 5 Εισαγωγή εγγραφής στον πίνακα Player.....	32
Code Snippet 6 Εισαγωγή εγγραφής στον πίνακα Gameplay	32
Code Snippet 7 Εισαγωγή εγγραφής στον πίνακα DoctorPass	33
Code Snippet 8 Εισαγωγή εγγραφής στον πίνακα Gameplay (Reflex Game 1).....	33
Code Snippet 9 Εισαγωγή εγγραφής στον πίνακα (Memory Game).....	34
Code Snippet 10 Αρχικοποίηση αξιών στον πίνακα Game	34
Code Snippet 11 Επιλογή πρώτης γραμμής στον πίνακα Player με κριτήρια αναζήτησης το όνομα	35
Code Snippet 12 Επιλογή κωδικού στον πίνακα DoctorPass.....	35
Code Snippet 13 Επιλογή όλων από τον πίνακα Player με κριτήρια αναζήτησης το επιλεγμένο όνομα.....	36
Code Snippet 14 Επιλογή υψηλότερου σκορ από τον πίνακα Gameplay με κριτήρια αναζήτησης το ID του παιχνιδιού «Ταίριαξε τα όμοια»	36
Code Snippet 15 Επιλογή υψηλότερου σκορ από τον πίνακα Gameplay με κριτήρια αναζήτησης το ID του παιχνιδιού «Αποφύγετε τα εμπόδια».....	36
Code Snippet 16 Επιλογή υψηλότερου σκορ από τον πίνακα Gameplay με κριτήρια αναζήτησης το ID του παιχνιδιού «ΚΛΙΚ ΣΤΟ ΚΟΥΤΙ».....	37
Code Snippet 17 Επιλογή υψηλότερου σκορ, ημερομηνίας από τον πίνακα Player με πολλαπλά κριτήρια αναζήτησης	37
Code Snippet 18 Connection String της βάσης δεδομένων.....	37
Code Snippet 19 Μέθοδος για κρυπτογράφηση κειμένου από ASCII σε base64	39
Code Snippet 20 Έλεγχος ταυτοποίησης κωδικού γιατρού.....	40
Code Snippet 21 Μέθοδος για αποκρυπτογράφηση κειμένου από base64 σε ASCII.....	40
Code Snippet 22 Απενεργοποίηση αλληλεπίδρασης με το κουμπί	42
Code Snippet 23 Χρήση custom βιβλιοθήκης/API	46
Code Snippet 24 Μέθοδος δημιουργίας κύκλου με AnchoredPosition παραμετρικό	47
Code Snippet 25 Δημιουργία Ευθείας για την σύνδεση μεταξύ δύο σημείων δυναμικά με παραμετρικά δύο σημεία.....	47
Code Snippet 26 Εμφάνιση γράφου με παραμετρικά τα σημεία αξίας x,y	47
Code Snippet 27 Έλεγχος κενών στοιχείων	64
Code Snippet 28 Δημιουργία Smtplib για την εισαγωγή σε Gmail από την εφαρμογή.....	64
Code Snippet 29 Δημιουργία String Builder για το body του email.....	65
Code Snippet 30 Μονοπάτι της βάσης δεδομένων και προσθήκη στα συνημμένα του email.....	65

Code Snippet 31 Μονοπάτι της βάσης δεδομένων.....	68
Code Snippet 32 Μέθοδος για επαναφορά κωδικού γιατρού.....	68

1 ΕΙΣΑΓΩΓΗ ΣΤΑ SERIOUS GAMES ΚΑΙ ΕΚΤΕΝΗΣ ΑΝΑΛΥΣΗ ΤΟΥΣ

1.1 ΕΙΣΑΓΩΓΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Χάρη στις διάφορες, πιθανές, θετικές πτυχές των βιντεοπαιχνιδιών που παρουσιάζονται, ο ιατρικός και παραϊατρικός τομέας έχει λάβει θετικά αποτελέσματα και έχει αποκτήσει ενδιαφέρον για την παιχνιδοποίηση ορισμένων ιατρικών παρεμβάσεων (Ritterfeld et al. 2009).

Πριν από την εξήγηση των στόχων και των αρχών που έχουμε θέσει στα πλαίσια της πτυχιακής μας εργασίας, είναι σημαντικό να ορίσουμε τους όρους που χρησιμοποιούνται συνήθως στην έρευνα και τη βιβλιογραφία.

Με τον όρο σοβαρά παιχνίδια (serious games), ορίζουμε εκείνα τα παιχνίδια που είναι σχεδιασμένα με πρωταρχικό σκοπό την υγειονομική περίθαλψη, αποκατάσταση, εκπαίδευση, ή πρόληψη τραυματισμών. Παρόλο, που η ψυχαγωγία δεν παρέχεται πάντα από αυτά τα παιχνίδια, ως όρος χρησιμοποιείται πολύ συχνά. Για ορισμένους όμως, υπάρχει ένα έλλειμμα μεταξύ αυτών των δύο όρων. Σύμφωνα με αυτούς, εάν ένα παιχνίδι είναι πολύ σοβαρό, τότε αυτό δεν θεωρείται παιχνίδι πια.

Ο όρος serious game είναι αρκετά οξύμωρος θα μπορούσε να πει κανείς. Υπάρχουν πολλές αναφορές και παρόμοιες έννοιες με την σημασία του πραγματικού σκοπού ενός παιχνιδιού που αφορά κάτι το πιο «σοβαρό». Η πρώτη φορά που χρησιμοποιήθηκε η φράση αυτή, ήταν σε ένα βιβλίο με τίτλο «Serious Game» του Clark Abt (1970). Ο Abt ήταν ερευνητής που εργαζόταν σε ένα αμερικανικό ερευνητικό κέντρο κατά τη διάρκεια του ψυχρού πολέμου (Abt Associates, 2005). Ένας από τους στόχους του ήταν να χρησιμοποιεί παιχνίδια για την εκπαίδευση και την μόρφωση.

Στην πραγματικότητα σχεδίασε διάφορα παιχνίδια στον υπολογιστή, όπως το T.E.M.P.E.R. (Raytheon, 1961). Το παιχνίδι αυτό χρησιμοποιήθηκε από στρατιωτικούς αξιωματικούς για τη διερεύνηση της σύγκρουσης του ψυχρού πολέμου σε παγκόσμια κλίμακα. Ωστόσο, στο βιβλίο του ο Abt παρέχει επίσης παραδείγματα "μη-ψηφιακών" serious game, όπως παιχνίδια που σχετίζονται με τα μαθηματικά. Πιο συγκεκριμένα, τα παιχνίδια αυτά μπορούν να χρησιμοποιηθούν στα σχολεία. Ο Abt δίνει επίσης έναν σαφή ορισμό των "Σοβαρών Παιχνιδιών": Συγκεκριμένα, ορίζει ως "Σοβαρά παιχνίδια" τα παιχνίδια τα οποία μπορεί να παίζονται σοβαρά ή σε περιστασιακή βάση. Μας απασχολούν, με την έννοια ότι τα παιχνίδια αυτά έχουν σαφή και προσεκτικά μελετημένο εκπαιδευτικό σκοπό και δεν προορίζονται να παίζονται κυρίως για ψυχαγωγία. Συνεπώς, αυτό δεν σημαίνει όμως ότι τα σοβαρά παιχνίδια δεν είναι ή δεν πρέπει να είναι και διασκεδαστικά".

“Games may be played seriously or casually. We are concerned with serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. This does not mean that the serious games are not, or should not be, entertaining.” (Abt 1970).

Το παραπάνω, οριοθετεί το πεδίο εφαρμογής των serious games στα βιντεοπαιχνίδια, με παρόμοιο τρόπο με τους περισσότερους σύγχρονους ορισμούς που επικρατούν (Michael & Chen, 2005- Zyda, 2005). Στην πραγματικότητα, όλοι αυτοί οι ορισμοί φαίνεται να είναι επηρεασμένοι από το όραμα του Ben Sawyer και τη μελέτη του με τίτλο “Serious Games: Improving Public Policy through Game-based Learning and Simulation.” (Sawyer & Rejeski, 2002). Όπως υποδηλώνει και ο τίτλος, το έγγραφο αυτό αποτελεί ένα κάλεσμα για τη χρήση της τεχνολογίας και της γνώσης από τη βιομηχανία ψυχαγωγικών βιντεοπαιχνιδιών, για τη βελτίωση των προσομοιώσεων που βασίζονται σε παιχνίδια για δημόσιους οργανισμούς.

Για την ακρίβεια, ο Sawyer έγραψε για πρώτη φορά την εργασία του με τίτλο “Improving Public Policy through Game Based Learning and Simulations.”, όμως ο συνάδελφός του David Rejeski θεώρησε ότι από αυτόν τον τίτλο έλειπε κάτι. Ο Rejeski γνώριζε ένα βιβλίο με τίτλο "Serious Play" (Schrage, 1999), το οποίο περιγράφει λεπτομερώς, πώς ιδιωτικές εταιρείες χρησιμοποιούν προσομοιώσεις για την ενίσχυση της καινοτομίας. Αναφερόμενος σε αυτό το βιβλίο, ο Rejeski αποφάσισε να τροποποιήσει τον τίτλο της μελέτης του Sawyer ώστε να συμπεριληφθεί αυτός ο οξύμωρος τίτλος "Serious Games".

Τη μελέτη αυτή ακολούθησε γρήγορα η δημιουργία της "Serious Games Initiative", μιας ένωσης για την προώθηση της χρήσης των παιχνιδιών για σοβαρούς σκοπούς. Έτσι, το οξύμωρο "Σοβαρά παιχνίδια" αποκτούσε κάποια δυναμική στο μυαλό πολλών ανθρώπων (Sawyer, 2009).

Κατά σύμπτωση, το 2002 ήταν επίσης η ημερομηνία κυκλοφορίας του America's Army, ενός παιχνιδιού που ο Sawyer θεωρεί ως "το πρώτο επιτυχημένο και καλά εκτελεσμένο σοβαρό παιχνίδι που κέρδισε την απόλυτη αναγνωρισιμότητα του κοινού" (Gudmundsen, 2006).

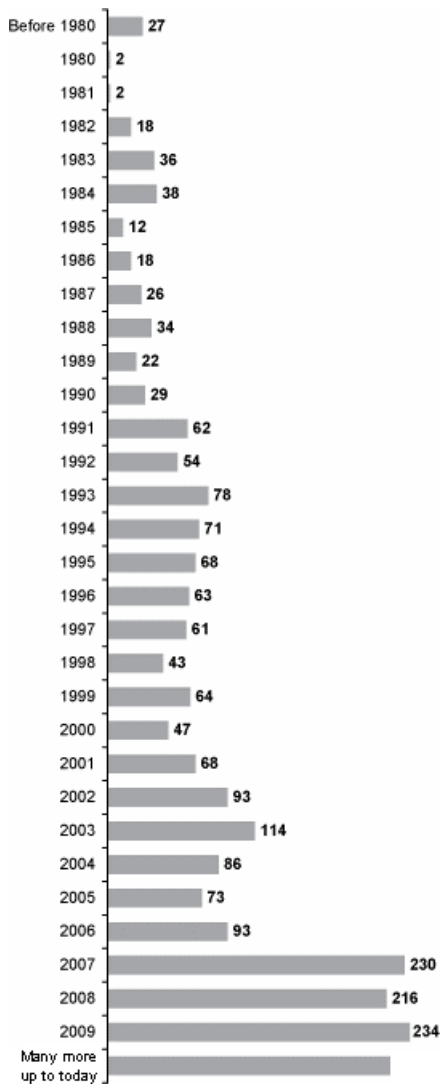


Εικόνα 1 America's Army

Ο συνδυασμός της δημοφιλούς επιτυχίας του America's Army και των προσπαθειών των Sawyer & Rejeski για την προώθηση τέτοιων παιχνιδιών, συντέλεσε στον προσδιορισμό του έτους 2002 ως η αφετηρία του «σημερινού κύκλου» των Serious Games. Αργότερα, ο Sawyer βελτίωσε τον ορισμό των serious games σε «οποιαδήποτε σημαντική χρήση των πόρων της βιομηχανίας ηλεκτρονικών παιχνιδιών / παιχνιδιών, των οποίων η κύρια αποστολή δεν είναι η ψυχαγωγία» (Sawyer, 2007).

Ο Michael Zyda, ο οποίος συμμετείχε στην ανάπτυξη του America's Army, πρότεινε έναν παρόμοιο ορισμό (Zyda, 2005) : «Ένας πνευματικός διαγωνισμός, που παίζεται με υπολογιστή σύμφωνα με συγκεκριμένους κανόνες, ο οποίος χρησιμοποιεί την ψυχαγωγία, για την προώθηση κυβερνητικών ή εταιρικών στόχων, εκπαίδευσης, υγείας, δημόσιας πολιτικής και στρατηγικής επικοινωνίας».

Σήμερα, τα περισσότερα serious games που κυκλοφορούν έχουν την τάση να ακολουθούν αυτή τη γραμμή, καθώς επιμένουν στη χρήση ψηφιακών παιχνιδιών, αντί να ακολουθούν τον ευρύτερο ορισμό των «σοβαρών παιχνιδιών» τόσο για τα ψηφιακά όσο και για τα μη ψηφιακά παιχνίδια που εισήχθησαν τη δεκαετία του '70.



Εικόνα 2 Λίστα serious games από το 1980-2009 και μέχρι και σήμερα

1.2 ΚΑΤΗΓΟΡΙΕΣ SERIOUS GAMES

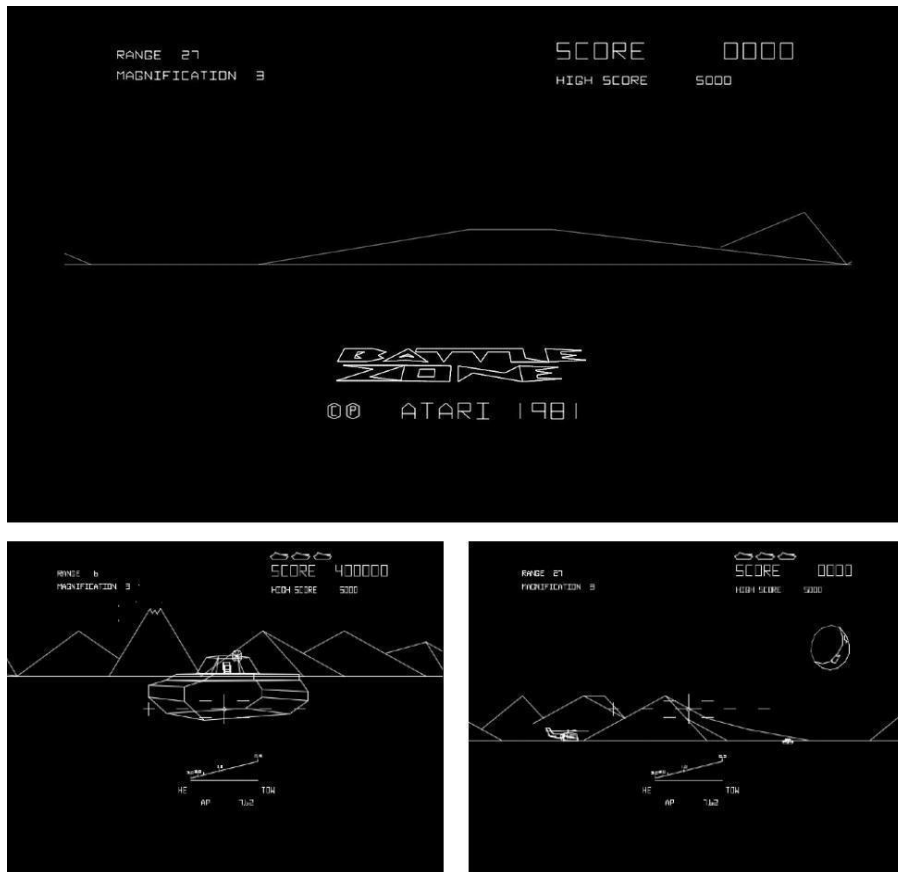
Μετά από αυτές τις πρωτοποριακές εμπειρίες, πολλά βιντεοπαιχνίδια κυκλοφόρησαν με σκοπό την υποστήριξη σοβαρών σκοπών πριν από το όρο των "Σοβαρών Παιχνιδιών". Ενδεικτικά αναφέρονται τα εξής τέσσερα παραδείγματα τέτοιων παιχνιδιών που αναφέρονται σε ένα ευρύ φάσμα τομέων.

Άμυνας

Εκτός από τα παιχνίδια που δημιουργήθηκαν για την Joint War Games Agency και πριν από την κυκλοφορία του America's Army, ο αμερικανικός στρατός έδειξε μεγάλο ενδιαφέρον για την ανάπτυξη ψυχαγωγικών βιντεοπαιχνιδιών για εκπαιδευτικούς σκοπούς. Ένα από τα πιο διάσημα παραδείγματα, είναι το "The Bradley Trainer" (Atari, 1981). Γνωστό και ως Military Battlezone ή Army Battlezone, το παιχνίδι αυτό αποτελεί μια προσαρμοσμένη έκδοση του Battlezone (Atari, 1980).

Το αρχικό παιχνίδι θέτει τον παίκτη στο ρόλο ενός άρματος μάχης σε έναν τρισδιάστατο κόσμο και του ζητά να ρίξει τα αντίπαλα οχήματα. Ο αμερικανικός στρατός προσέλαβε την Atari για να δημιουργήσει μια πιο ρεαλιστική έκδοση αυτού του παιχνιδιού, ώστε να μπορέσει να το χρησιμοποιεί ως εργαλείο άσκησης. Αντί για ένα φανταστικό танк, ο παίκτης ελέγχει τώρα το Bradley Fighting Vehicle, ένα στρατιωτικό χερσαίο όχημα οπλισμένο με ένα πυροβόλο με αλυσίδα και ένα κανόνι. Ο παίκτης πρέπει να καταρρίψει τα αντίπαλα ελικόπτερα και танкς πυροβολώντας με τα όπλα αυτού του πραγματικού οχήματος. Ο ρεαλισμός της προσομοίωσης πυροβολικού έχει βελτιωθεί ώστε να ανταποκρίνεται στον εκπαιδευτικό σκοπό αυτού του παιχνιδιού (James, 1997).

Παρόλο, που η Atari δέχτηκε να δημιουργήσει αυτή την προσαρμοσμένη έκδοση του παιχνιδιού της για τον αμερικανικό στρατό, αρκετοί από τους υπαλλήλους της ήταν κατηγορηματικά αντίθετοι, συμπεριλαμβανομένου του Ed Rotberg, του σχεδιαστή του Battlezone (Kent, 2001): "Δεν θέλαμε να έχουμε καμία επαφή με τον στρατό. Εγώ κατασκεύαζα παιχνίδια. Δεν επιθυμούσα να εκπαιδεύω ανθρώπους ώστε να σκοτώνουν". Έτσι, αν και ανεπίσημα, αυτή η αντίδραση, καταδεικνύει τις πολιτισμικές διαφορές μεταξύ του τομέα των ψυχαγωγικών βιντεοπαιχνιδιών και της σημερινής βιομηχανίας των "Σοβαρών Παιχνιδιών".



Εικόνα 3 Title Screen from “The Bradley Trainer” και εικόνες μέσα από το παιχνίδι

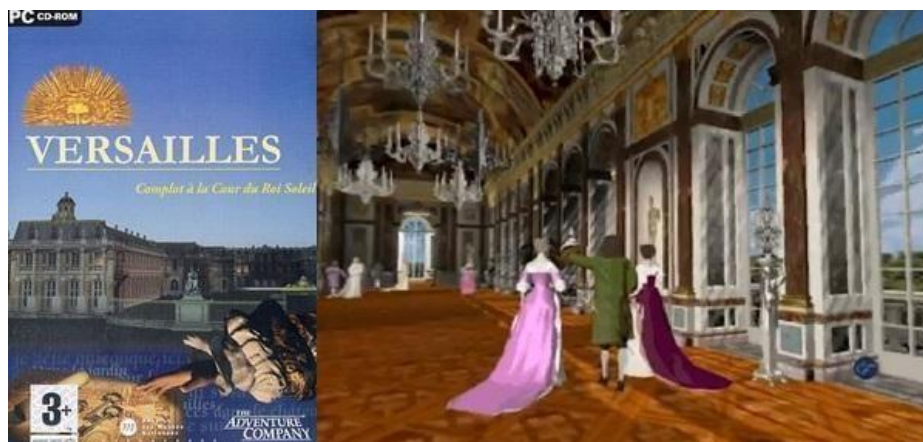
Τέχνη & Πολιτισμός

Το Versailles 1685 (Cryo, 1997) είναι η ναυαρχίδα των βιντεοπαιχνιδιών "πολιτιστικής ψυχαγωγίας". Τέτοιου είδους παιχνίδια συνδυάζουν την ψυχαγωγία και την πολιτιστική εκπαίδευση.

Το συγκεκριμένο παιχνίδι διαδραματίζεται στις Βερσαλλίες κατά τη διάρκεια της βασιλείας του Λουδοβίκου ΙΔ'. Σκοπός του παίκτη είναι να διερευνήσει και να εντοπίσει ποιος απειλεί να καταστρέψει τις Βερσαλλίες. Δίνεται η δυνατότητα στον παίκτη να κινηθεί ελεύθερα στο εσωτερικό του όμορφου αυτού χώρου, να συνομιλήσει με ιστορικούς χαρακτήρες (NPC) και να μάθει για τους πίνακες ζωγραφικής και τις τέχνες αυτής της εποχής.

Ο τίτλος αυτός έλαβε μια θερμή υποδοχή στην Ευρώπη με περισσότερα από 300.000 αντίτυπα να έχουν πωληθεί. Μάλιστα, άνοιξε το δρόμο για παρόμοιους τίτλους με διαφορετικές ιστορικές περιόδους, όπως το “Egypt 1156 BC Tomb of the pharaoh” (Cryo, 1997), “Byzantine: The Betrayal” (Discovery Channel Multimedia, 1997),

“China the Forbidden City” (Cryo, 1998), “Pilgrim Faith As A Weapon” (Axel Tribe, 1998), “Vikings” (Index+, 1998), “Rome” (Montparnasse Multimedia, 2000).



Εικόνα 4 Preview of the game “Versailles”

Εκπαίδευση

Το “The Oregon Trail” (MECC, 1971) ξεκίνησε ως ένα παιχνίδι text-only που δημιουργήθηκε από τρεις καθηγητές Ιστορίας, τον Don Rawitsch, τον Bill Heinemann και τον Paul Dillenberger. Ο παίκτης υποδύεται έναν Αμερικανό οδοιπόρο του 1848, ο οποίος έχει ως στόχο να φτάσει στο Όρεγκον για να εγκατασταθεί. Ο δρόμος προς το Όρεγκον είναι γεμάτος παγίδες, αλλά το παιχνίδι είναι εμπλουτισμένο με πληροφορίες που σχετίζονται με αυτή την περίοδο της αμερικανικής ιστορίας.

Το παιχνίδι αυτό "δημοσιεύθηκε" από τη Minnesota Educational Computing Consortium (MECC). Ο θεσμός αυτός βοήθησε τους εκπαιδευτικούς από τη Μινεσότα να χρησιμοποιούν τους υπολογιστές για διδασκαλία. Το “The Oregon Trail” ήταν τόσο δημοφιλές σε μαθητές αλλά και εκπαιδευτικούς που κυκλοφόρησαν πολλές αναβαθμίσεις. Το 1978 κυκλοφόρησε μια γραφιστική έκδοση του παιχνιδιού σε μορφή open-source κώδικα. Το παιχνίδι αυτό "δημοσιεύθηκε" από τη Minnesota Educational Computing Consortium (MECC). Ο θεσμός αυτός βοήθησε τους εκπαιδευτικούς από τη Μινεσότα να χρησιμοποιούν τους υπολογιστές για διδασκαλία. Το “The Oregon Trail” ήταν τόσο δημοφιλές σε μαθητές αλλά και εκπαιδευτικούς που κυκλοφόρησαν πολλές αναβαθμίσεις. Το 1978 κυκλοφόρησε μια γραφιστική έκδοση του παιχνιδιού σε μορφή open-source κώδικα, η οποία στη συνέχεια βελτιώθηκε και κυκλοφόρησε εμπορικά το 1985.

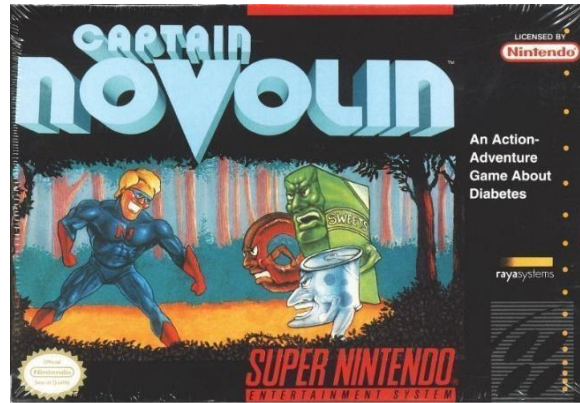


Εικόνα 5 Preview of the “The Oregon Trail”

Αυτό το παιχνίδι συνοδεύτηκε από διάφορες μεταγενέστερες αναβαθμισμένες εκδόσεις : “The Oregon Trail II (MECC, 1996)”, “The Oregon Trail: 3rd Edition (MECC, 1997)” - “spin-offs - The Amazon Trail (MECC, 1993)” & “The Africa Trail (MECC, 1997)”. Ωστόσο, το αρχικό παιχνίδι εξακολουθεί να είναι δημοφιλές σήμερα χάρη στις εκδόσεις που υπάρχουν για κινητά τηλέφωνα και σε εφαρμογή στο Facebook. Εν τέλει, αυτό το παιχνίδι δείχνει ξεκάθαρα ότι ένα "εκπαιδευτικό" ή "σοβαρό" παιχνίδι δεν είναι απαραίτητα το αντίθετο από ένα "δημοφιλές και εμπορικά επιτυχημένο" παιχνίδι.

Υγείας

Το “Captain Novolin” (Raya Systems, 1992) σχεδιάστηκε για να διδάσκει στα παιδιά πώς να διαχειρίζονται τον διαβήτη. Αυτό το παιχνίδι επιτρέπει στον παίκτη να παίξει τον ρόλο ενός υπερήρωα με διαβήτη, ο οποίος πρέπει να φροντίσει το επίπεδο γλυκόζης στο αίμα του, ενώ παράλληλα θα πρέπει να νικάει τους κακούς εξωγήινους του ανθυγιεινού φαγητού. Αυτό το παιχνίδι πλατφόρμας "υποκλέπτει" τον γνωστό μηχανισμό "collectable bonuses" για να μεταδώσει ένα μήνυμα. Τα μόνους που μπορεί να συλλέξει ο ήρωας είναι όλα αντικείμενα διατροφής. Έτσι, αν ο ήρωας συλλέξει πάρα πολλά από αυτά κινδυνεύει να αισθανθεί αδιαθεσία λόγω υψηλού επιπέδου γλυκόζης στο αίμα του. Ευτυχώς, πριν από κάθε επίπεδο, ένας διατροφολόγος ενημερώνει τους παίκτες σχετικά με το πόσα τρόφιμα επιτρέπεται να καταναλώσουν. Εξίσου σημαντικό θεωρείται ότι οι παίκτες θα πρέπει να διαχειρίζονται και την ινσουλίνη τους.



Εικόνα 6 Game Cover and Screenshot of the game “Captain Novolin”

Αυτό το παιχνίδι και τρεις άλλοι τίτλοι που σχετίζονται με το πεδίο της υγείας κυκλοφόρησαν για την κονσόλα Super Nintendo από την ίδια εταιρεία, την Raya Systems. Αν και δεν χαρακτηρίστηκαν ως "σοβαρά παιχνίδια", έχουν διεξαχθεί αρκετές ερευνητικές μελέτες για την ανάλυση των επιδράσεών τους στα παιδιά (Lieberman, 2001). Για παράδειγμα, το παιχνίδι Packy & Marlon (Raya Systems, 1994), παρόμοιο με το Captain Novolin με λειτουργία δύο παικτών, αναλύθηκε σε κλινική δοκιμή (Brown et al., 1997). Παρατηρήθηκε ότι η ομάδα των παιδιών στα οποία παρουσιάστηκε αυτό το παιχνίδι ήταν καλύτερη στη διαχείριση του διαβήτη τους. Ο αριθμός των περιπτώσεων, στις οποίες τα παιδιά αυτά χρειάστηκε να μεταβούν στο νοσοκομείο λόγω κρίσης γλυκόζης μειώθηκε κατά 77% συγκριτικά με την ομάδα που δεν δοκίμασε το παιχνίδι.

Συμπερασματικά, μέσω της μελέτης αποδεικνύεται ότι τα παιχνίδια βοήθησαν τα παιδιά να μάθουν πώς να διαχειρίζονται την ινσουλίνη και τις δόσεις της και να τρώνε υγιεινά γεύματα, ώστε να αποφεύγονται οι σχετικές κρίσεις με τη γλυκόζη.

1.3 Εργαλεία υλοποίησης παιχνιδιών. Unity, visual studio και συνδεση μεταξύ τους



Εικόνα 7 Unity Logo

Η Unity είναι μια μηχανή παιχνιδιών πολλαπλών πλατφόρμων που αναπτύχθηκε από την Unity Technologies, η οποία χρησιμοποιείται κυρίως για την ανάπτυξη βιντεοπαιχνιδιών και προσομοιώσεων για υπολογιστές, κονσόλες και κινητές συσκευές. Αρχικά ανακοινώθηκε μόνο για το OS X, στο Worldwide Developers Conference της Apple το 2005, αλλά έκτοτε επεκτάθηκε και στοχεύει σε 27 πλατφόρμες.

Η Unity είναι μια μηχανή παιχνιδιών, που υποστηρίζει 2D και 3D γραφικά, drag and drop αλλά και scripting μέσω της γλώσσας C#. Η Unity είναι ιδιαίτερα δημοφιλής για την ανάπτυξη παιχνιδιών για κινητά τηλέφωνα και μεγάλο μέρος της επικεντρώνεται σε αυτό. Το 2D κομμάτι της Unity είναι μια πρόσφατη προσθήκη στη μηχανή σε σχέση με το 3D κομμάτι της. Παρόλα αυτά, η Unity είναι μια επαρκής πλατφόρμα για την ανάπτυξη παιχνιδιών 2D, ακόμη και σε σύγκριση με άλλες αποκλειστικές μηχανές 2D, ιδίως αν πρόκειται να κυκλοφορήσει το παιχνίδι σε πολλές κινητές συσκευές.

Η Unity είναι επίσης μια καλή επιλογή για την ανάπτυξη παιχνιδιών που χρησιμοποιούν τεχνολογία Virtual Reality (VR), παρόλο που το VR είναι μια μικρή αλλά ραγδαία αναπτυσσόμενη αγορά αυτή τη στιγμή. Οι αγορές κινητών τηλεφώνων και PSVR (PlayStation VR) είναι οι μεγαλύτερες στο VR και η Unity είναι ήδη σε καλή θέση για τη μεταφορά παιχνιδιών σε πολλές από αυτές τις πλατφόρμες, όπως τον υπολογιστή, κονσόλες ή και σε πολλές διαφορετικές αγορές κινητών τηλεφώνων.

Η μηχανή στοχεύει στα ακόλουθα API γραφικών:

- Direct3D για Windows και Xbox One
- OpenGL για Linux
- macOS
- Windows
- OpenGL ES για Android
- iOS WebGL για web
- API για τις κονσόλες βιντεοπαιχνιδιών

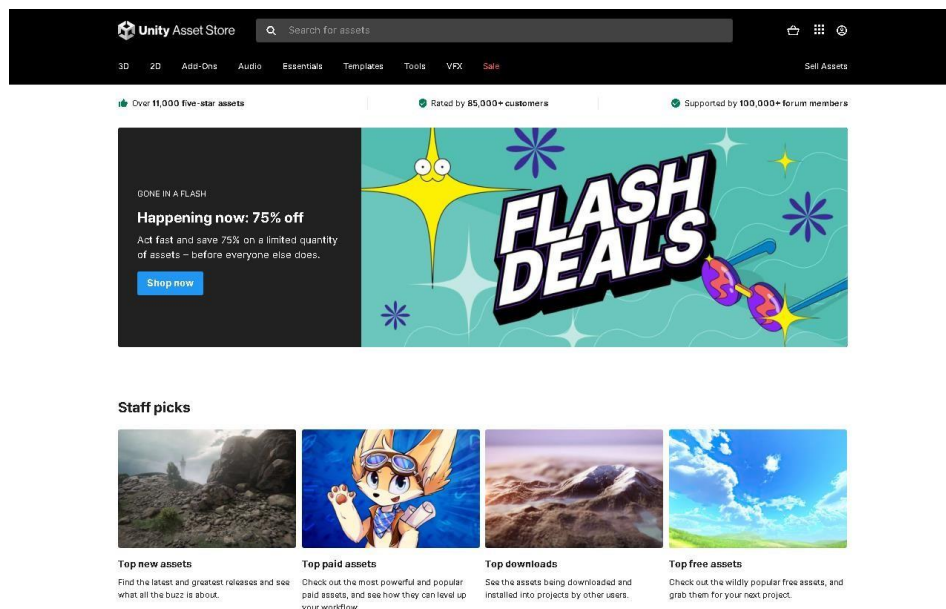
Για 3D παιχνίδια, η Unity επιτρέπει τον καθορισμό των ρυθμίσεων για την συμπίεση των υφών και την ανάλυση για κάθε πλατφόρμα που υποστηρίζει η μηχανή παιχνιδιών και παρέχει υποστήριξη για:

- Χαρτογράφηση (bump mapping)
- Χαρτογράφηση ανάκλασης (reflection mapping)
- Χαρτογράφηση παράλλαξης (parallax mapping)
- Screen space ambient occlusion (SSAO)
- Δυναμικές σκιές (dynamic shadows) με χρήση shadow maps
- Render-to-texture και εφέ μετα-επεξεργασίας πλήρους οθόνης

Η Unity προσφέρει επίσης υπηρεσίες στους προγραμματιστές, οι οποίες είναι οι εξής:

- Unity Ads
- Unity Analytics
- Unity Certification
- Unity Cloud Build
- Unity Everyplay
- Unity API
- Unity Multiplayer
- Unity Performance Reporting
- Unity Collaborate

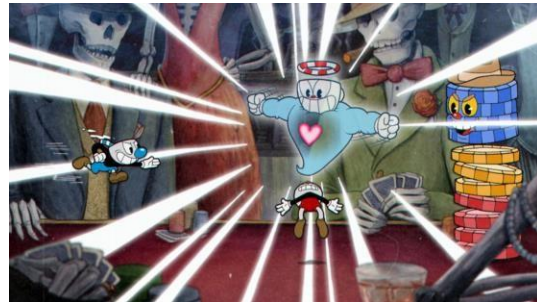
Εκτός από τα παραπάνω, η Unity διαθέτει ένα asset store όπου η κοινότητα των developers μπορεί να κατεβάζει και να ανεβάζει τόσο εμπορικούς όσο και δωρεάν πόρους από τρίτους, όπως υφές (**textures**), μοντέλα (**models**), **plugins**, επεκτάσεις επεξεργαστών (**editor extensions**), ακόμη και ολόκληρα παραδείγματα παιχνιδιών.



Εικόνα 8 Screenshot of the Unity Asset Store

Παρακάτω παραθέτουμε μερικές από τις πιο αξιοσημείωτες αναφορές παιχνιδιών τα οποία έχουν υλοποιηθεί στην πλατφόρμα Unity:

Cuphead



Εικόνα 9 Screenshot from “Cuphead”

Hollow Knight



Εικόνα 10 Artwork from “Hollow Knight”

Phasmophobia



Εικόνα 11 Screenshot from “Phasmophobia”

[Punishing: Gray Raven](#)



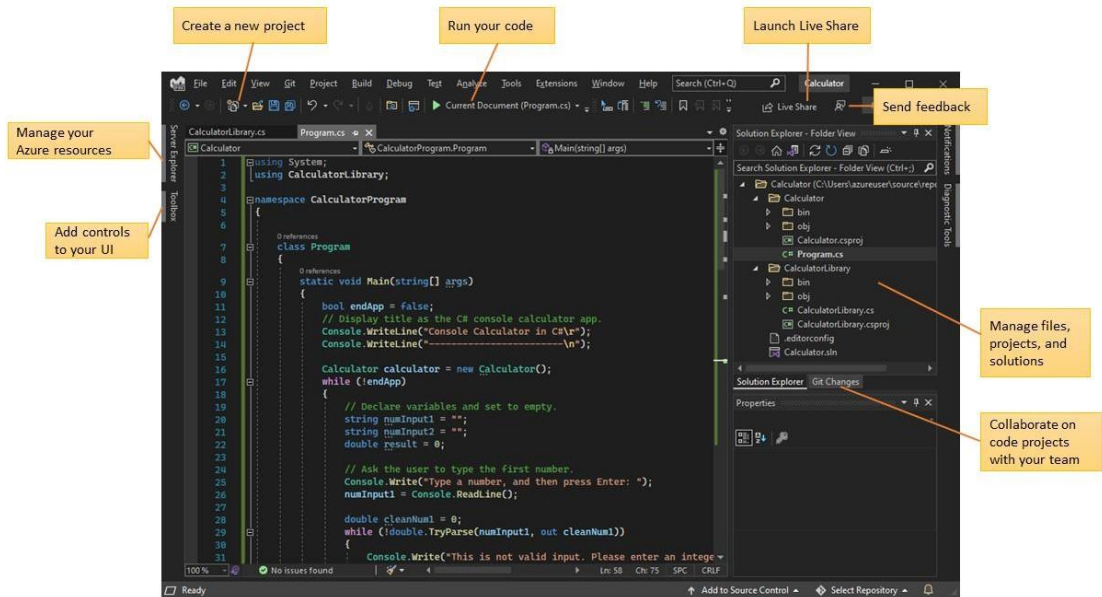
Εικόνα 12 Game Cover of “Punishing Gray Raven”



Εικόνα 13 Logo of the Visual Studio

Ένα integrated development environment (IDE) είναι ένα πλούσιο σε δυνατότητες πρόγραμμα που υποστηρίζει πολλές πτυχές της ανάπτυξης λογισμικού. Το [Visual Studio IDE](#) είναι ένα δημιουργικό βήμα εκκίνησης που μπορεί να χρησιμοποιηθεί για την επεξεργασία, αποσφαλμάτωση και δημιουργία κώδικα (build code) αλλά και στη συνέχεια τη δημοσίευση (**publish**) μιας εφαρμογής.

Πέρα από τον τυπικό editor και τον debugger που παρέχουν τα περισσότερα IDE, το Visual Studio περιλαμβάνει μεταγλωττιστές (**compilers**), εργαλεία συμπλήρωσης κώδικα, γραφικούς σχεδιαστές (**graphical designers**) και πολλά άλλα χαρακτηριστικά που βελτιώνουν τη διαδικασία της ανάπτυξης λογισμικού.

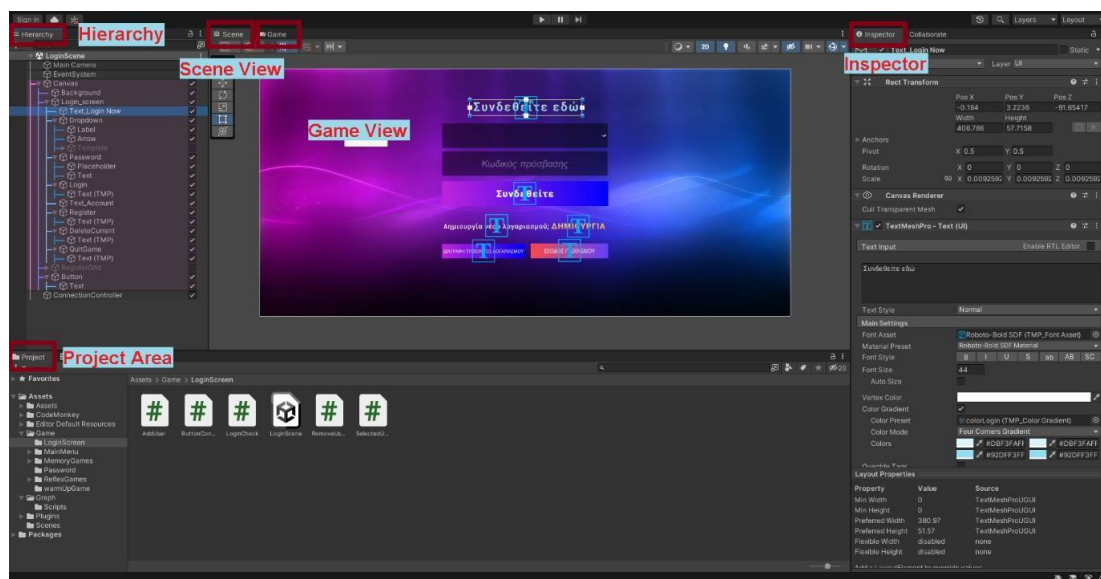


Εικόνα 14 Screenshot of how the Visual Studio(IDE)

Η παραπάνω εικόνα παρουσιάζει το Visual Studio με ένα ανοιχτό έργο (project) που δείχνει τα βασικά παράθυρα και τη λειτουργικότητά τους:

- Στο **Solution Explorer**, πάνω δεξιά, μπορούμε να προβάλουμε, να πλοηγηθούμε και να διαχειριστούμε τα αρχεία του κώδικά μας. Η Εξερεύνηση λύσεων μπορεί να βοηθήσει στην οργάνωση του κώδικά, ομαδοποιώντας τα αρχεία σε λύσεις και έργα.
- Το κεντρικό παράθυρο (**central editor**) του editor, εμφανίζει τα περιεχόμενα των αρχείων. Στο παράθυρο του επεξεργαστή, μπορούμε να επεξεργαστούμε κώδικα ή να σχεδιάσουμε μια διεπαφή χρήστη, όπως ένα παράθυρο με κουμπιά και πλαίσια κειμένου.
- Στο παράθυρο **Git Changes** κάτω δεξιά, μπορούμε να παρακολουθούμε στοιχεία εργασίας και να μοιραζόμαστε κώδικα με άλλους προγραμματιστές με τους οποίους συνεργαζόμαστε, χρησιμοποιώντας τεχνολογίες ελέγχου έκδοσης όπως το Git και το GitHub.

1.4 Βασική Ορολογία της μηχανής Unity



Εικόνα 15 Screenshot of the Interface of Unity

Interface

- **Scene View:** Εδώ θα δημιουργηθεί το επίπεδο για το παιχνίδι, τη σκηνή ή το τρισδιάστατο έργο του παιχνιδιού. Όλα τα αντικείμενα του παιχνιδιού θα τοποθετούνται και θα διαχειρίζονται εδώ.
- **Game View:** Εδώ θα μπορείτε να δείτε τα αποτελέσματα, την πρόοδο δημιουργίας του κάθε επιπέδου καθώς και τη σκηνή του παιχνιδιού. Θα πρέπει να υπάρχει τοποθετημένη μια κάμερα στη σκηνή για να βλέπουμε πώς θα φαίνεται η εφαρμογή μας στον χρήστη. Μερικές φορές η κάμερα αυτή ονομάζεται κάμερα προβολής (**Camera View**).
- **Hierarchy:** Αυτό το παράθυρο θα εμφανίζει όλα τα αντικείμενα παιχνιδιού (**Game Objects**) που έχουν τοποθετηθεί απευθείας στη σκηνή. Κατά βάση όλα όσα εμφανίζονται στην Προβολή παιχνιδιού, πρέπει να αναφέρονται εδώ. Αυτό θα περιλαμβάνει μη ορατά και ορατά αντικείμενα παιχνιδιού.
- **Project Area:** Αυτό είναι το κύριο παράθυρο του έργου. Στην πραγματικότητα εμφανίζει τι υπάρχει μέσα στο φάκελο **Assets**. Όλα τα αντικείμενα του παιχνιδιού (**Game Objects**), **Scripts**, **Textures**, **Folders**, **Models**, **Audio**, **Video** και κλπ. είναι προσβάσιμα από αυτό το πλαίσιο.
- **Inspector:** Σε αυτό το παράθυρο θα εμφανίζονται διάφορα χαρακτηριστικά (**attributes**) και ιδιότητες (**properties**) των επιλεγμένων αντικειμένων παιχνιδιού. Αναλόγως την επιλογή, θα εμφανίζονται και τα κατάλληλα χαρακτηριστικά και στοιχεία.

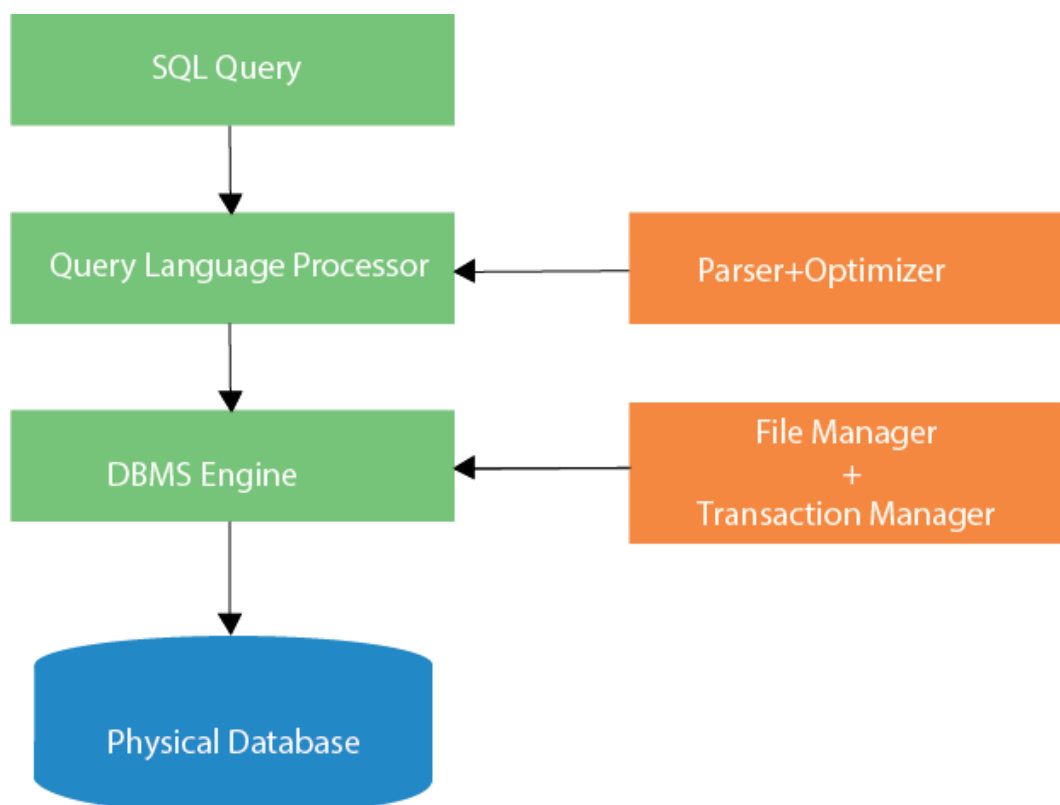
2 ΣΧΕΔΙΑΣΜΟΣ, ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ PROJECT “BRAIN EXERCISE”

2.1 ΧΡΗΣΙΜΟΤΗΤΑ ΕΝΤΑΞΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

2.1.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ SQL

SQL είναι συντομογραφία του **Structured Query Language**. Χρησιμοποιείται για την αποθήκευση και τη διαχείριση δεδομένων σε σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (**RDBMS**). Είναι μια τυπική γλώσσα για ένα σχεσιακό σύστημα βάσεων δεδομένων. Δίνει τη δυνατότητα στο χρήστη να δημιουργεί (**CREATE**), να διαβάζει (**READ**), να ενημερώνει (**UPDATE**) και να διαγράφει (**DELETE**) σχεσιακές βάσεις δεδομένων και πίνακες.

Όλα τα RDBMS όπως η MySQL, η Informix, η Oracle, η MS Access και ο SQL Server χρησιμοποιούν την SQL ως την καθιερωμένη γλώσσα βάσεων δεδομένων. Η SQL επιτρέπει στους χρήστες να υποβάλλουν ερωτήματα στη βάση δεδομένων με διάφορους τρόπους, χρησιμοποιώντας english-like εντολές.



Εικόνα 16 Explanatory graphic diagram of the SQL

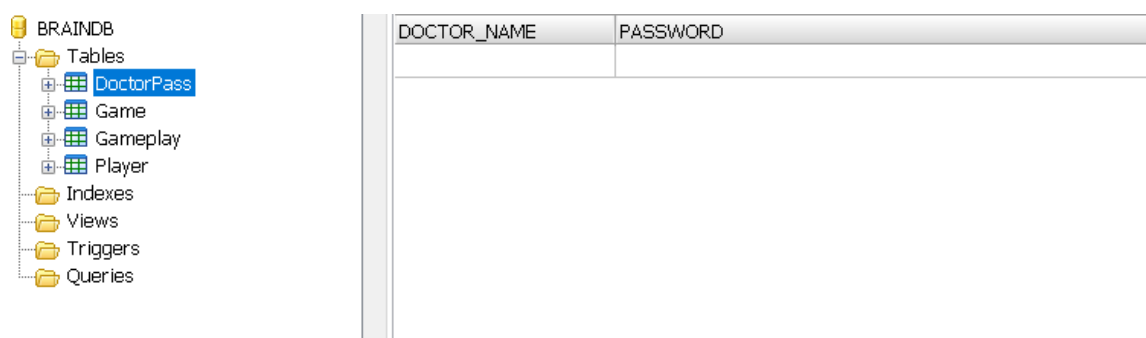
2.1.2 Πίνακες που χρησιμοποιήθηκαν κατά την εκπόνηση της εργασίας

Στα πλαίσια του έργου μας δημιουργήθηκαν οι παρακάτω SQL πίνακες:

DoctorPass:

Ο πίνακας αυτός περιλαμβάνει:

- το όνομα του γιατρού (*DOCTOR_NAME*) [**PRIMARY KEY**]
- τον κωδικό του (*PASSWORD*)



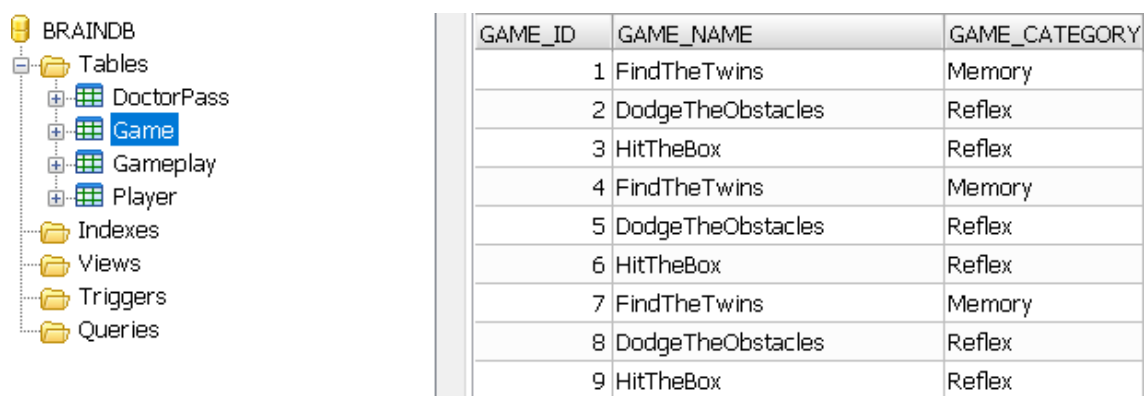
The screenshot shows the SQL Server Enterprise Manager interface. On the left, a tree view displays the database structure for 'BRAINDB', including folders for Tables, Indexes, Views, Triggers, and Queries. Under the 'Tables' folder, the 'DoctorPass' table is selected. On the right, the table structure is displayed in a grid format with two columns: 'DOCTOR_NAME' and 'PASSWORD'.

Εικόνα 17 Screenshot from the table “DoctorPass”

Game:

Ο πίνακας αυτός περιλαμβάνει:

- τον κωδικό του παιχνιδιού (*GAME_ID*) [**PRIMARY KEY** | **AUTOINCREMENT NULL**]
- το όνομα του παιχνιδιού (*GAME_NAME*)
- κατηγορία παιχνιδιού (*GAME_GATEGORY*)



The screenshot shows the SQL Server Enterprise Manager interface. On the left, a tree view displays the database structure for 'BRAINDB', including folders for Tables, Indexes, Views, Triggers, and Queries. Under the 'Tables' folder, the 'Game' table is selected. On the right, the table structure and data are displayed in a grid format with three columns: 'GAME_ID', 'GAME_NAME', and 'GAME_CATEGORY'. The data rows are as follows:

GAME_ID	GAME_NAME	GAME_CATEGORY
1	FindTheTwins	Memory
2	DodgeTheObstacles	Reflex
3	HitTheBox	Reflex
4	FindTheTwins	Memory
5	DodgeTheObstacles	Reflex
6	HitTheBox	Reflex
7	FindTheTwins	Memory
8	DodgeTheObstacles	Reflex
9	HitTheBox	Reflex

Εικόνα 18 Screenshot from the table “Game”

Gameplay:

Αυτός ο πίνακας περιέχει κυρίως τα περισσότερα χαρακτηριστικά για κάθε παιχνίδι και ασθενή. Αποτελείται από:

- το *GAMEPLAY_ID* [**PRIMARY KEY** | **AUTOINCREMENT NULL**]
- το *PLAYER_ID*, το *GAME_ID*

- την ημερομηνία που έπαιξε (*PLAYED_DATE*)
- την ώρα έναρξης (*STARTING_TIME*)
- την ώρα λήξης (*FINISH_TIME*)
- την διάρκεια (*PLAYED_DURATION*)
- την δυσκολία που επέλεξε (*GAME_DIFFICULTY*)
- συνολικό σκορ που συγκέντρωσε (*SCORE*)

GAMEPLAY_ID	PLAYER_ID	GAME_ID	Played_Date	Starting_time	finish_time	Played_duration	Game_Difficulty	Score
1	1	2	10/01/2022	21:11:10	21:12:11	01:01	EASY	606
2	2	2	10/01/2022	21:11:10	21:12:11	01:01	HARD	989
3	2	3	10/01/2022	21:11:10	21:12:11	01:01	MEDIUM	376
4	1	3	10/01/2022	21:11:10	21:12:11	01:01	HARD	759
5	3	2	10/01/2022	21:11:10	21:12:11	01:01	HARD	577
6	1	1	10/01/2022	21:11:10	21:12:11	01:01	EASY	160
7	3	1	10/01/2022	21:11:10	21:12:11	01:01	HARD	39
8	3	3	10/01/2022	21:11:10	21:12:11	01:01	EASY	806
9	3	3	10/01/2022	21:11:10	21:12:11	01:01	MEDIUM	538
10	2	1	10/01/2022	21:11:10	21:12:11	01:01	HARD	150
11	2	1	10/01/2022	21:11:10	21:12:11	01:01	HARD	215
12	1	3	10/01/2022	21:11:10	21:12:11	01:01	HARD	125
13	2	1	10/01/2022	21:11:10	21:12:11	01:01	EASY	896
14	1	3	10/01/2022	21:11:10	21:12:11	01:01	HARD	466
15	1	1	10/01/2022	21:11:10	21:12:11	01:01	MEDIUM	499
16	2	1	11/01/2022	21:11:10	21:12:11	01:01	EASY	310
17	1	1	11/01/2022	21:11:10	21:12:11	01:01	HARD	297
18	1	1	11/01/2022	21:11:10	21:12:11	01:01	EASY	825
19	2	1	11/01/2022	21:11:10	21:12:11	01:01	EASY	387
20	3	1	11/01/2022	21:11:10	21:12:11	01:01	MEDIUM	328
21	2	3	11/01/2022	21:11:10	21:12:11	01:01	HARD	674

Εικόνα 19 Screenshot from the table “Gameplay”

Player:

Ο πίνακας αυτός περιέχεται από:

- το κωδικό του ασθενή (*PLAYER_ID*) [PRIMARY KEY | AUTOINCREMENT NULL]
- τον όνομα (*PLAYER_NAME*)
- την ηλικία (*PLAYER_AGE*)
- το φύλο (*PLAYER_GENDER*)

PLAYER_ID	PLAYER_NAME	PLAYER_AGE	PLAYER_GENDER

Εικόνα 20 Screenshot from the table “Player”

2.1.3 Ερωτήματα που χρησιμοποιούνται από την εφαρμογή μας

CREATE

Create tables #1 (PLAYER)

```
1 Query = "create table if not exists Player(  
2     PLAYER_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
3     PLAYER_NAME VARCHAR(30) NOT NULL,  
4     PLAYER_AGE INT NOT NULL,  
5     PLAYER_GENDER VARCHAR(10));"
```

Code Snippet 1 Δημιουργία Πίνακα Player

Create tables #2 (GAMEPLAY)

```
1 Query = create table if not exists Gameplay(  
2     GAMEPLAY_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
3     PLAYER_ID INTEGER,  
4     GAME_ID INTEGER,  
5     Played_Date VARCHAR(10),  
6     Starting_time VARCHAR(10),  
7     finish_time VARCHAR(10),  
8     Played_duration VARCHAR(10),  
9     Game_Difficulty VARCHAR(10),  
10    Score INTEGER,  
11    FOREIGN key (GAME_ID) references Game(GAME_ID),  
12    FOREIGN key (PLAYER_ID) references Player(PLAYER_ID));
```

Code Snippet 2 Δημιουργία Πίνακα Gameplay

Create table #3 (GAME)

```
1 Query = create table if not exists Game (  
2     GAME_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
3     GAME_NAME VARCHAR(25) NOT NULL,  
4     GAME_CATEGORY VARCHAR(10) NOT NULL);
```

Code Snippet 3 Δημιουργία Πίνακα Game

Create table #4 (DOCTORPASS)

```
1 Query = create table if not exists DoctorPass (  
2     DOCTOR_NAME VARCHAR(20) NOT NULL,  
3     PASSWORD VARCHAR(50) NOT NULL);
```

Code Snippet 4 Δημιουργία Πίνακα Doctor Pass

INSERT

Add new user

```
1 Query = $insert into Player (  
2     PLAYER_NAME,  
3     PLAYER_AGE,  
4     PLAYER_GENDER)  
5     values (  
6     '{inputName.text}',  
7     '{inputAge.text}',  
8     '{gender.options[gender.value].text}');
```

Code Snippet 5 Εισαγωγή εγγραφής στον πίνακα Player

Memory game

```
1 Query = $INSERT INTO GAMEPLAY (  
2     PLAYER_ID, GAME_ID, PLAYED_DATE,  
3     STARTING_TIME, FINISH_TIME, PLAYED_DURATION,  
4     GAME_DIFFICULTY, SCORE)  
5     VALUES  
6     ('{Rplayer_id}',  
7     '{Rgame_id}',  
8     '{Rplayed_date}',  
9     '{Rstart_time}',  
10    '{Rfinish_time}',  
11    '{Rplayed_duration}',  
12    '{Rgame_difficulty}',  
13    '{Rscore}');
```

Code Snippet 6 Εισαγωγή εγγραφής στον πίνακα Gameplay

Εισαγωγή στον πίνακα Doctor Pass

```
1 Query = $INSERT INTO DoctorPass(  
2     DOCTOR_NAME,  
3     PASSWORD)  
4     VALUES  
5     ('{docName.text}',  
6     '{encodedBytes}');
```

Code Snippet 7 Εισαγωγή εγγραφής στον πίνακα DoctorPass

Reflex Game ()

```
1 Query = $INSERT INTO GAMEPLAY(  
2     PLAYER_ID,  
3     GAME_ID,  
4     PLAYED_DATE,  
5     STARTING_TIME,  
6     FINISH_TIME,  
7     PLAYED_DURATION,  
8     GAME_DIFFICULTY,  
9     SCORE)  
10    VALUES(  
11    '{PlayerPrefs.GetInt("PLAYER_ID")}'  
12    '{PlayerPrefs.GetInt("GameID")}'  
13    '{currentDay}'  
14    '{starting_time}'  
15    '{finish_time}'  
16    '{correctedDuration}'  
17    '{PlayerPrefs.GetString("Game2DifficultyName")}'  
18    '{newScore}');
```

Code Snippet 8 Εισαγωγή εγγραφής στον πίνακα GamePlay (Reflex Game 1)

Memory Game #1 ()

```
1 Query = $INSERT INTO GAMEPLAY(  
2     PLAYER_ID,  
3     GAME_ID,  
4     PLAYED_DATE,  
5     STARTING_TIME,  
6     FINISH_TIME,  
7     PLAYED_DURATION,  
8     GAME_DIFFICULTY,  
9     SCORE)  
10 VALUES(  
11     '{PlayerPrefs.GetInt("PLAYER_ID")}'  
12     '{PlayerPrefs.GetInt("GameID")}'  
13     '{currentDay}'  
14     '{starting_time}'  
15     '{finish_time}'  
16     '{correctedDuration}'  
17     '{PlayerPrefs.GetString("Game3DifficultyName")}'  
18     '{newScore}');
```

Code Snippet 9 Εισαγωγή εγγραφής στον πίνακα (Memory Game)

Αρχικά values για το πίνακα GAME

```
1 Query = $INSERT INTO GAME(  
2     GAME_NAME,  
3     GAME_CATEGORY)  
4 VALUES  
5     ('FindTheTwins', 'Memory'),  
6     ('DodgeTheObstacles', 'Reflex'),  
7     ('HitTheBox', 'Reflex');
```

Code Snippet 10 Αρχικοποίηση αξιών στον πίνακα Game

SELECT

Select first row player

```
1 Query = $SELECT EXISTS(  
2     SELECT 1 FROM Player  
3     WHERE PLAYER_NAME='{inputName.text}' LIMIT 1);
```

Code Snippet 11 Επιλογή πρώτης γραμμής στον πίνακα Player με κριτήρια αναζήτησης το όνομα

Select password

```
1 Query = SELECT PASSWORD FROM DOCTORPASS;
```

Code Snippet 12 Επιλογή κωδικού στον πίνακα DoctorPass

Dropdown

Στοιχεία επιλεγμένου ασθενή

```
1 Query = $SELECT * FROM Player
2     WHERE PLAYER_NAME = '{selectedAccountName}';
```

Code Snippet 13 Επιλογή όλων από τον πίνακα Player με κριτήρια αναζήτησης το επιλεγμένο όνομα

Φόρτωση high score από το παιχνίδι "Ταίριαξε τα όμοια"

```
1 Query = $SELECT MAX(Score) FROM Gameplay
2     WHERE PLAYER_ID =
3     '{PlayerPrefs.GetInt("PLAYER_ID")}' AND GAME_ID = '1';
```

Code Snippet 14 Επιλογή υψηλότερου σκορ από τον πίνακα Gameplay με κριτήρια αναζήτησης το ID του παιχνιδιού «Ταίριαξε τα όμοια»

Φόρτωση highscore από το παιχνίδι "Αποφύγετε τα εμπόδια"

```
1 Query = $SELECT MAX(Score) FROM Gameplay
2     WHERE PLAYER_ID =
3     '{PlayerPrefs.GetInt("PLAYER_ID")}' AND GAME_ID = '2';
```

Code Snippet 15 Επιλογή υψηλότερου σκορ από τον πίνακα Gameplay με κριτήρια αναζήτησης το ID του παιχνιδιού «Αποφύγετε τα εμπόδια»

Φόρτωση highscore από το παιχνίδι "Κλικ στο κουτί"

```
1 Query = $SELECT MAX(Score) FROM Gameplay
2 WHERE PLAYER_ID =
3 '{PlayerPrefs.GetInt("PLAYER_ID")}' AND GAME_ID = '3';
```

Code Snippet 16 Επιλογή υψηλότερου σκορ από τον πίνακα Gameplay με κριτήρια αναζήτησης το ID του παιχνιδιού «Κλικ στο κουτί»

Στοιχεία για την δημιουργία x (high score), y (εκτελέσιμες προσπάθειες) αξόνων του κάθε παιχνιδιού

```
1 Query = $SELECT MAX(Score),played_date FROM Gameplay
2 WHERE PLAYER_ID =
3 '{PlayerPrefs.GetInt("PLAYER_ID")}'
4 AND GAME_ID = '{PlayerPrefs.GetInt("selectedGame")}'
5 AND game_difficulty =
6 '{PlayerPrefs.GetString("selectedDifficulty")}';
```

Code Snippet 17 Επιλογή υψηλότερου σκορ, ημερομηνίας από τον πίνακα Player με πολλαπλά κριτήρια αναζήτησης

Connection String

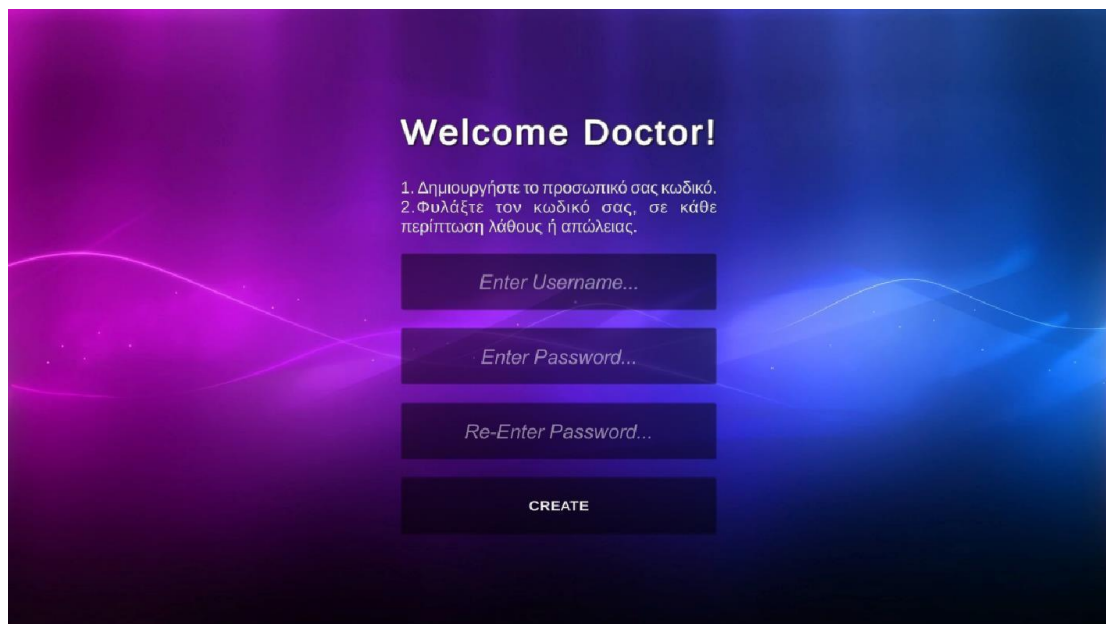
```
1 string connection = "URI=file:" + Application.persistentDataPath + "/BrainDB.s3db";
2
```

Code Snippet 18 Connection String της βάσης δεδομένων

2.2 Ανάπτυξη παιχνιδιού ανά Σκηνή

2.2.1 DOCTOR-REGISTER

Η πρώτη σκηνή του παιχνιδιού προορίζεται για τον γιατρό – βοηθό των συγκεκριμένων ασθενών όπου καλείται να βοηθήσει. Όπως βλέπουμε στην παρακάτω εικόνα πρόκειται για ένα Register System. Το συγκεκριμένο View θα εμφανιστεί καθόλη την διάρκεια ζωής του μόνο μία φορά διότι γίνεται έλεγχος στην βάση δεδομένων εάν έχει δημιουργηθεί ο λογαριασμός του γιατρού.



Εικόνα 21 Screenshot from the “Doctor-Register” window

Ο χρήστης – γιατρός, εφόσον καταχωρήσει τα απαιτούμενα προσωπικά στοιχεία (Username , Password, Confirm Password) και πατήσει το Button (Create) γίνεται έλεγχος «IsNullOrEmpty» σε κάθε Input Field έτσι ώστε να αποφευχθεί η εισαγωγή κενών στοιχείων στην βάση ενώ στην συνέχεια γίνεται έλεγχος ταύτισης μεταξύ των δυο κωδικών που χρειάζεται να εισάγει ο χρήστης προκειμένου να κάνει register, όπως απεικονίζεται στην εικόνα 1. Μετά τον έλεγχο και των κωδικών δύναται να συμβούν τα παρακάτω :

- 1) Μη ταύτιση των κωδικών και εμφάνιση του μηνύματος “**Password does not Match**”.
- 2) Ταύτιση των κωδικών και εμφάνιση του μηνύματος “**Success**” όπου επιτυγχάνεται η εγγραφή του χρήστη-γιατρού στην βάση δεδομένων και μας οδηγεί στο επόμενο Scene του “**Patient Login**”.

Κατά την επιτυχία δημιουργίας account για τον γιατρό, ο κωδικός πριν αποθηκευτεί στην βάση γίνεται encrypted σε Base64 για μεγαλύτερη ασφάλεια.

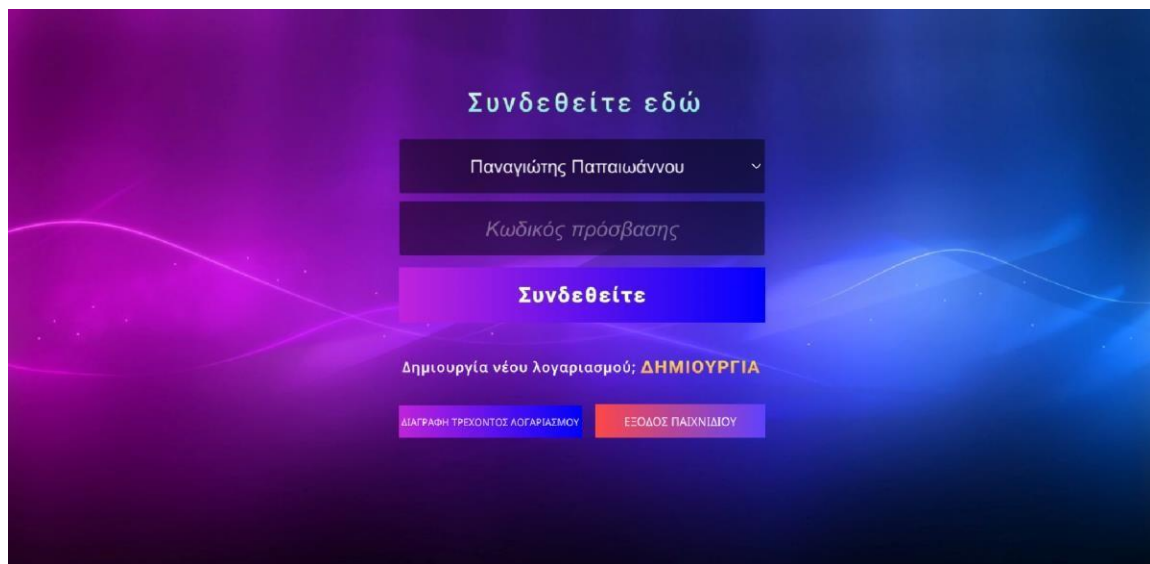
```
1 private string Base64Encode(string passwordForDatabase)
2     {
3         var passwordBytes = Encoding.UTF8.GetBytes(passwordForDatabase);
4
5         return Convert.ToBase64String(passwordBytes);
6     }
```

Code Snippet 19 Μέθοδος για κρυπτογράφηση κειμένου από ASCII σε base64

2.2.2 PATIENT-LOGIN

Μετά την πραγματοποίηση του “Doctor Login Scene”, το συγκεκριμένο Scene γίνεται αυτομάτως το προκαθορισμένο scene που θα βλέπει ο χρήστης και προορίζεται για τους ασθενείς του γιατρού. Ο γιατρός έχει τη δυνατότητα να προβεί στις παρακάτω ενέργειες κάνοντας click στο αντίστοιχο button:

- “Δημιουργία”
- “Διαγραφή τρέχοντος λογαριασμού”
- “Συνδεθείτε “
- “Έξοδος παιχνιδιού”



Εικόνα 22 Screenshot from the “Patient-Login” window

Όπως διακρίνουμε στην εικόνα 2, υπάρχει το button του “κωδικός πρόσβασης” ο οποίος είναι ο κωδικός του **ΓΙΑΤΡΟΥ** που δημιουργήσαμε στο προηγούμενο View (Doctor Register). Κάθε κίνηση (δημιουργία χρήστη, διαγραφή χρήστη, σύνδεση με συγκεκριμένο χρήστη) είναι αδύνατον να πραγματοποιηθεί χωρίς να προστεθεί ο κωδικός του γιατρού έτσι ώστε να

εξασφαλίσουμε να αποφύγουμε μία απρόβλεπτη κίνηση κάποιου ασθενή και να προστατεύσουμε την βάση δεδομένων. Όπως βλέπουμε και στο παρακάτω snippet γίνεται πάντα έλεγχος εάν ο κωδικός στο Input Field είναι ίδιος με τον κωδικό της βάσης.

```
1  if (ifpassword.text.ToString() == plainPassword)
2      {
3          //DoSomething
4      }
5  else
6      {
7          Debug.Log("Wrong Password Doctor!");
8  }
9
```

Code Snippet 20 Έλεγχος ταυτοποίησης κωδικού γιατρού

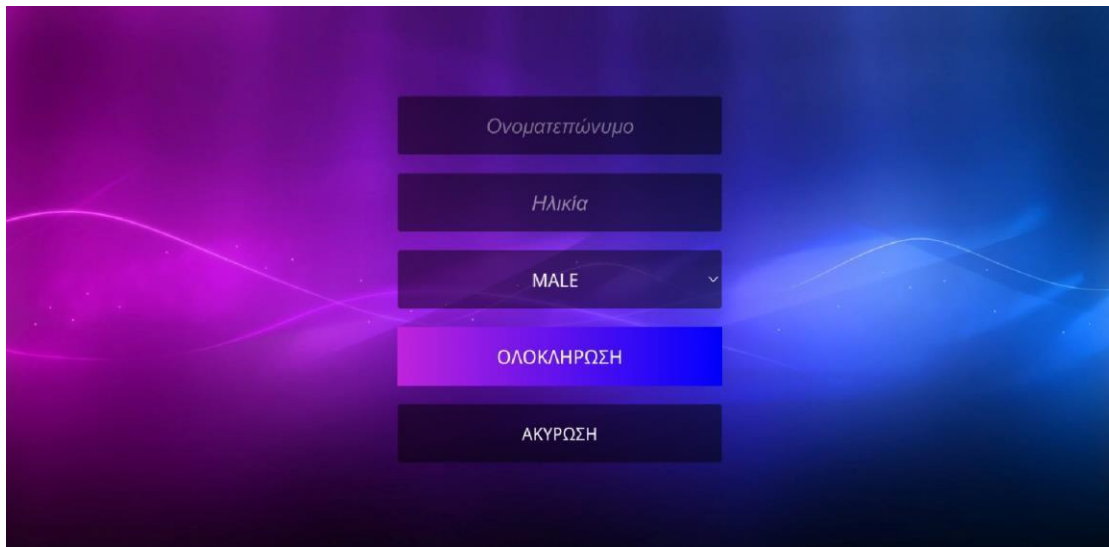
Απαραίτητο κρίνεται, πριν γίνει ο έλεγχος να ολοκληρωθεί το decode του κωδικού που λαμβάνεται από την βάση δεδομένων από base64 σε ASCII. Η διαδικασία του decode περιγράφεται στο παρακάτω snippet.

```
1  private string Base64Decode(string base64password)
2      {
3          var encodedBytes = System.Convert.FromBase64String(base64password);
4          return System.Text.Encoding.UTF8.GetString(encodedBytes);
5      }
```

Code Snippet 21 Μέθοδος για αποκρυπτογράφηση κειμένου από base64 σε ASCII

Δημιουργία Νέου Χρήστη

Για την δημιουργία νέου ασθενή πατώντας το button (Δημιουργία) το πρόγραμμα μας οδηγεί σε μία νέα φόρμα όπου ο ασθενής καλείται να εισάγει τα στοιχεία του (Όνοματεπώνυμο, Ηλικία και Γένος) .



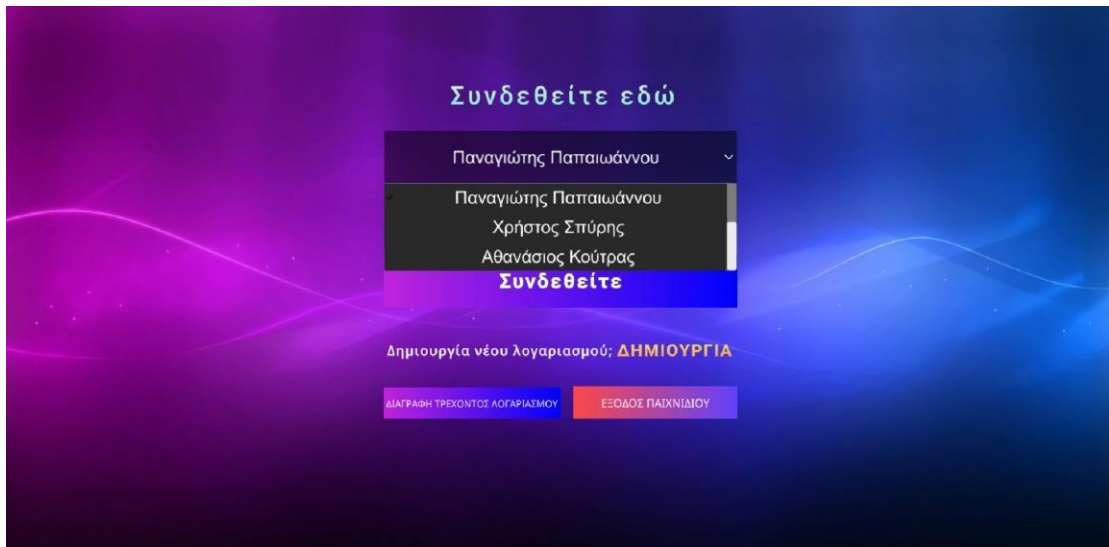
Εικόνα 23 Screenshot from the “Δημιουργία Νέου Χρήστη” window

Εφόσον πατήσει το button (Ολοκλήρωση) γίνεται ‘**IsNullOrEmpty**’ έλεγχος σε κάθε Input Field και ενώ παράλληλα ελέγχεται στην βάση εάν ο χρήστης υπάρχει ήδη με το συγκεκριμένο όνομα που μας οδηγεί σε δύο περιπτώσεις:

- Υπάρχει στην βάση και μας εμφανίζει μήνυμα ‘**Please Insert Another Name**’
- Δεν υπάρχει στην βάση, οπότε επιτυγχάνεται η δημιουργία του ασθενή

Κατά την επίτευξη δημιουργίας του νέου χρήστη, διακρίνουμε στο αρχικό μενού ότι ο χρήστης εμφανίζεται στη Dropdown list όπως μπορούμε να δούμε στην παρακάτω εικόνα. Αυτό γίνεται καθώς στη Dropdown list εμφανίζονται όλοι οι ασθενείς χρήστες που έχουμε στην βάση δεδομένων.

Σε περίπτωση που ο χρήστης δεν επιθυμεί να συνεχίσει την εγγραφή του έχει τη δυνατότητα να επιλέξει το button (Ακύρωση) που μας οδηγεί στο αρχικό μενού του Patient Logic.



Εικόνα 25 Screenshot of the Dropdown list

Διαγραφή Υπάρχοντος Χρήστη

Για την διαγραφή ενός υπάρχοντος χρήστη χρειάζεται να διαλέξουμε τον επιθυμητό χρήστη από το Dropdown και να πατήσουμε το button (Διαγραφή τρέχοντος Λογαριασμού).

Η μη ύπαρξη χρηστών στην Dropdown list συντελεί στη μη αναγκαιότητα χρήσης του button (Διαγραφή τρέχοντος Λογαριασμού) και κατά συνέπεια τη μη εφικτή η λειτουργία του (non-interactable button)



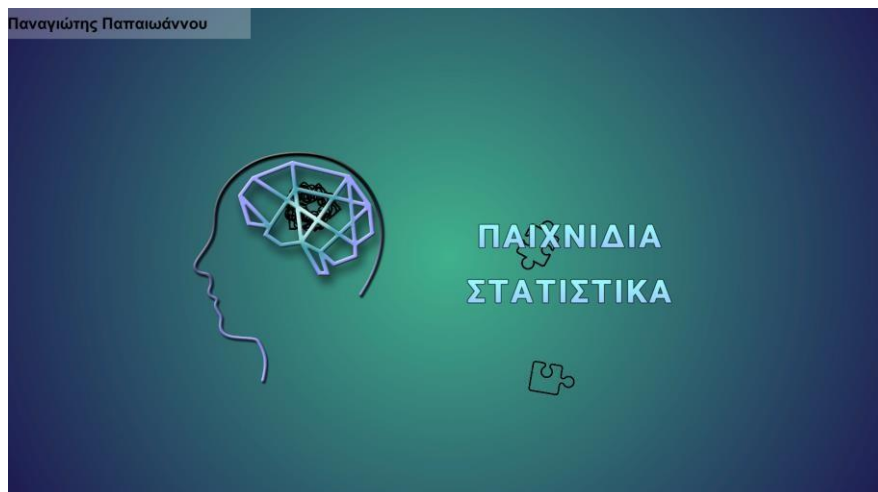
Code Snippet 22 Απενεργοποίηση αλληλεπίδρασης με το κουμπί

Σύνδεση Υπάρχοντος Επιλεγμένου Χρήστη

Για να συνδεθούμε στο Main Menu χρειάζεται να διαλέξουμε τον επιθυμητό χρήστη από το Dropdown και να πατήσουμε το button (Συνδεθείτε). Καθώς πατήσουμε το button (Συνδεθείτε) φορτώνουμε όλα τα στοιχεία του ασθενή που υπάρχουν αποθηκευμένα στην βάση καθώς και το Player Progress του από τον πίνακα Gameplay της βάσης και τα αποθηκεύουμε σε μία built-in κλάση που ονομάζεται "[PlayerPrefs](#)" για να τα διαχειριστούμε όπως θέλουμε. Στην πρώτη σύνδεση κάθε νέου χρήστη γίνονται αρχικοποιήσεις σε όλα τα στατιστικά του Player Progress.

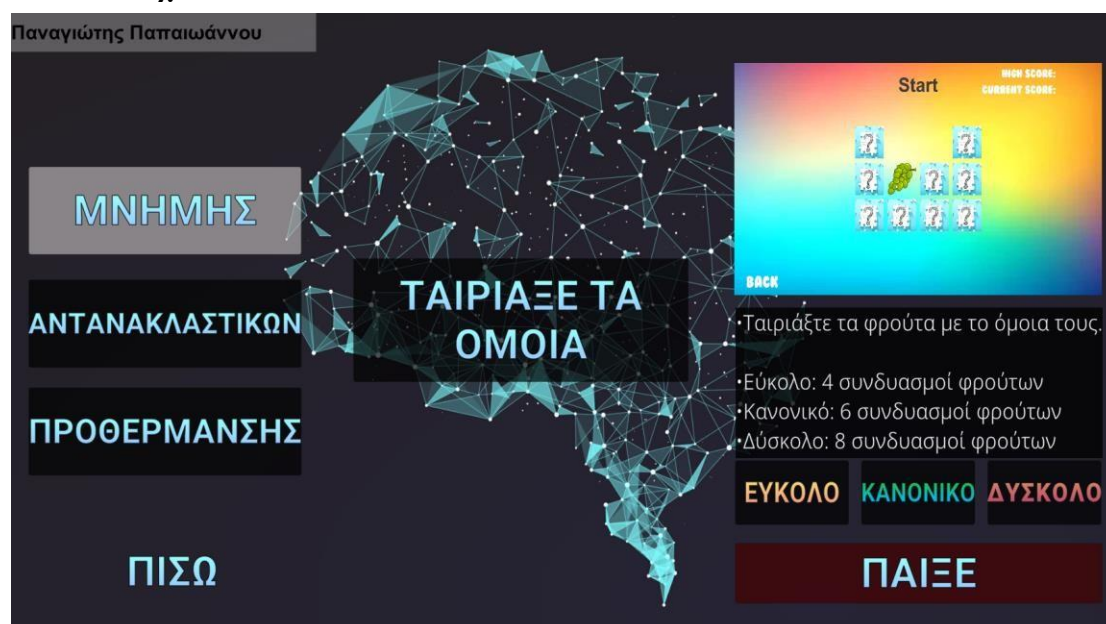
2.2.3 MAIN MENU

Το Main Menu Scene είναι το βασικό scene όπου μπορούμε να πλοηγηθούμε στην κατηγορία των παιχνιδιών, στα στατιστικά του συνδεδεμένου χρήστη, να ενεργοποιήσουμε την κονσόλα προγραμματιστών ή να κάνουμε αποσύνδεση. Με βάση την παρακάτω εικόνα βλέπουμε ότι μέσω του PlayerPrefs όπου αναφέραμε στο Patient Login μας εμφανίζει στην επάνω αριστερά γωνία το ονοματεπώνυμο του συνδεδεμένου χρήστη. Από το σημείο αυτό ότι αλλαγές γίνουν στην βάση δεδομένων στον πίνακα GamePlay αναφέρονται αποκλειστικά σε αυτόν τον χρήστη μέχρι να γίνει αποσύνδεση του.



Εικόνα 24 Screenshot of the main menu of the game

Πάνελ Παιχνιδιών



Εικόνα 25 Screenshot from the Panel with the Categorized Games

Στη παραπάνω εικόνα διακρίνουμε στα αριστερά τις κατηγορίες παιχνιδιών (Μνήμης, Αντανακλαστικών και Προθέρμανσης). Αναλόγως ποιά κατηγορία θα διαλέξει ο χρήστης εμφανίζονται τα διαθέσιμα παιχνίδια της συγκεκριμένης κατηγορίας που απεικονίζονται στο κέντρο της εικόνας.

Κατηγορία Μνήμης:

Στην κατηγορία μνήμης υπάρχει το παιχνίδι «Ταίριαξε τα όμοια».

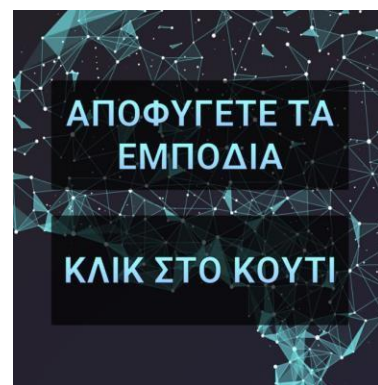


Εικόνα 26 Screenshot from the button “Ταίριαξε τα όμοια”

Κατηγορία Αντανακλαστικών

Στην κατηγορία αντανακλαστικών υπάρχουν τα εξής παιχνίδια:

- Αποφύγεται τα εμπόδια
- Κλικ στο κουτί



Εικόνα 27 Screenshot from the buttons “Αποφύγεται τα εμπόδια”, “Κλικ στο κουτί”

Κατηγορία Προθέρμανσης

Στην κατηγορία προθέρμανσης υπάρχει το παιχνίδι «Εκπαίδευση όρασης» το οποίο δεν συλλέγει δεδομένα στην βάση δεδομένων



Εικόνα 28 Screenshot from the button “Εκπαίδευση Όρασης”

Στην συνέχεια, αναλόγως το παιχνίδι που θα επιλέξει ο χρήστης, εμφανίζεται διαφορετικό γκρουπ στο δεξί μέρος της εικόνας το οποίο αποτελείται από ένα Video Preview του παιχνιδιού, περιγραφή του παιχνιδιού καθώς και τα επίπεδα δυσκολίας (Εξαιρείται το παιχνίδι προθέρμανσης).

Κάνοντας επιλογή του παιχνιδιού και δυσκολίας που θέλουμε μπορούμε να πατήσουμε το κουμπί (Παίξε) για να μας μεταφέρει στο View του παιχνιδιού.

Πάνελ Στατιστικών

Το πάνελ στατιστικών είναι το πιο βασικό πάνελ της εφαρμογής καθώς ο γιατρός μπορεί να δείξει στον ασθενή της πορεία που έχει κάνει από την πρώτη του επαφή με την εφαρμογή.



Εικόνα 29 Screenshot from the Statistics Menu

Στην παραπάνω εικόνα υπάρχουν 2 dropdown (Ονομασίες παιχνιδιών, Δυσκολίες) όπου καθορίζουν τον γράφο, την υψηλότερη βαθμολογία, τις εκτελέσιμες προσπάθειες και την διάρκεια του συγκεκριμένου παιχνιδιού στην συγκεκριμένη δυσκολία.

Όπως αναφέρθηκε παραπάνω έχουμε φορτώσει ήδη όλα τα δεδομένα του ασθενή καθώς και τις προσπάθειες του σε κάθε παιχνίδι οπότε μπορούμε πολύ εύκολα διαλέξουμε τι πληροφορία θέλουμε να εμφανιστεί στον γράφο μας.

Η δημιουργία του γράφου χτίστηκε επάνω σε ένα custom API (Application Programming Interface) από τον developer [«Code Monkey»](#).



Code Snippet 23 Χρήση custom βιβλιοθήκης/API

Με ορισμένα Queries καταγράψαμε τις τιμές του σκορ ταξινομημένες ανά ημερομηνία σε μία λίστα καθώς και τις συγκεκριμένες ημερομηνίες που ο ασθενής έπαιξε. Έτσι μπορέσαμε να προσθέσουμε στο X axis τις ημερομηνίες και στον Y axis τα σκορ με ελάχιστο σκορ το 0 και μέγιστο την υψηλότερη βαθμολογία του συγκεκριμένου παιχνιδιού χωρισμένα σε 10άδες ώστε ότι σκορ και να πραγματοποιήσει να μην επηρεάσει το ύψος του γράφου και να μας οδηγήσει σε σφάλματα. Εφόσον οι άξονες έχουν τοποθετηθεί δημιουργήσαμε ένα gameObject τύπου "Image" όπου θα παίρνει σαν παραμέτρους τους δύο άξονες και θα προστίθεται στον γράφο.



Code Snippet 24 Μέθοδος δημιουργίας κύκλου με AnchoredPosition παραμετρικό

Στην συνέχεια χρειαζόμαστε ένα gameObject όπου θα είναι τύπου “Line” για να συνδέει τους κύκλους από τον πρώτο μέχρι και τον τελευταίο ώστε να δίνει μία αισθητική στον γράφο.

```
1 private void CreateDotConnection(Vector2 dotPositionA, Vector2 dotPositionB) {
2     GameObject gameObject = new GameObject("dotConnection", typeof(Image));
3     gameObject.transform.SetParent(graphContainer, false);
4     gameObject.GetComponent<Image>().color = new Color(1,1,1, .5f);
5     RectTransform rectTransform = gameObject.GetComponent<RectTransform>();
6     Vector2 dir = (dotPositionB - dotPositionA).normalized;
7     float distance = Vector2.Distance(dotPositionA, dotPositionB);
8     rectTransform.anchorMin = new Vector2(0, 0);
9     rectTransform.anchorMax = new Vector2(0, 0);
10    rectTransform.sizeDelta = new Vector2(distance, 3f);
11    rectTransform.anchoredPosition = dotPositionA + dir * distance * .5f;
12    rectTransform.localEulerAngles = new Vector3(0, 0, UtilsClass.GetAngleFromVectorFloat(dir));
13 }
```

Code Snippet 25 Δημιουργία Ευθείας για την σύνδεση μεταξύ δύο σημείων δυναμικά με παραμετρικά δύο σημεία

Και τέλος εφόσον έχουν φτιαχτεί όλα πρέπει να εμφανίσουμε όλα τα values στον γράφο.

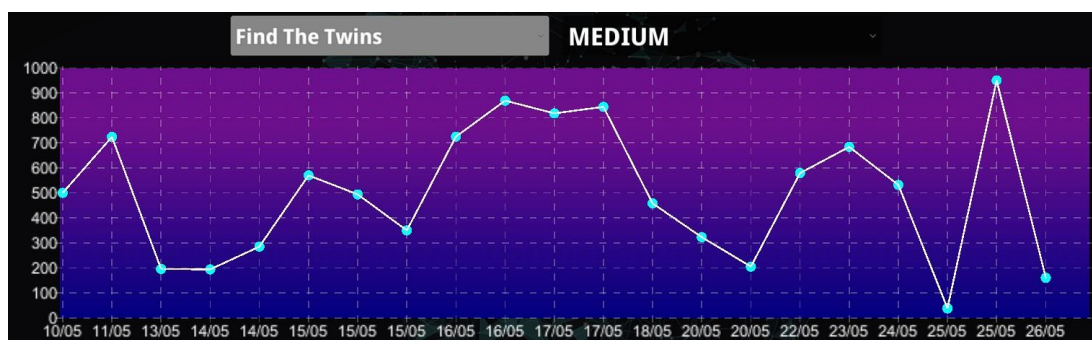
```
1 private void ShowGraph(List<int> valueList) {
2     float graphHeight = graphContainer.sizeDelta.y;
3     float yMaximum = 100f;
4     float xSize = 50f;
5
6     GameObject lastCircleGameObject = null;
7     for (int i = 0; i < valueList.Count; i++) {
8         float xPosition = xSize + i * xSize;
9         float yPosition = (valueList[i] / yMaximum) * graphHeight;
10        GameObject circleGameObject = CreateCircle(new Vector2(xPosition, yPosition));
11        if (lastCircleGameObject != null) {
12            CreateDotConnection(lastCircleGameObject.GetComponent<RectTransform>().anchoredPosition,
13                                circleGameObject.GetComponent<RectTransform>().anchoredPosition);
14        }
15        lastCircleGameObject = circleGameObject;
16    }
17 }
```

Code Snippet 26 Εμφάνιση γράφου με παραμετρικά τα σημεία αξίας x,y

Παραδείγματα ορισμένων δοκιμών για να δούμε εάν λειτουργεί ο γράφος όπως πρέπει

Παράδειγμα 1^ο

Επιλογή παιχνιδιού	Ταίριαξε τα όμοια
Επιλογή δυσκολίας	Κανονική
Μέγιστο σκορ	950
Πρώτη ημερομηνία δοκιμής	10/05
Τελευταία ημερομηνία δοκιμής	26/05

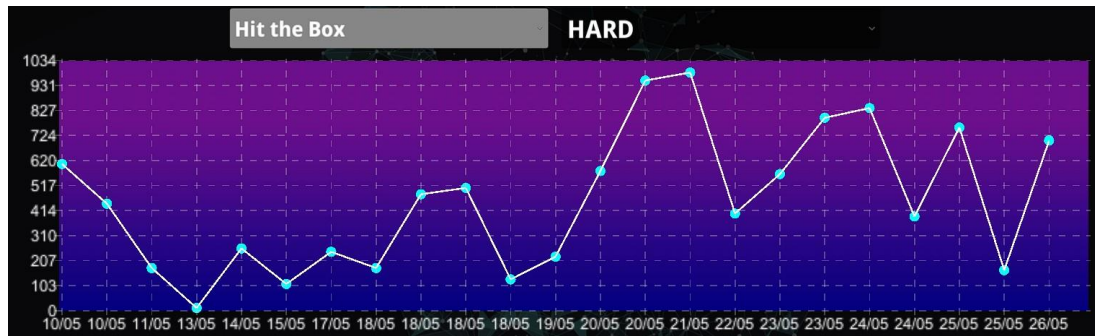


Εικόνα 30 Screenshot for Example 1

Όπως διακρίνουμε στην παραπάνω εικόνα όλα τα στατιστικά είναι σωστά ακόμα και εάν ο χρήστης δεν έχει παίξει κάποιο χρονικό διάστημα (μεταξύ 18/05 – 20/05) και στον άξονα Υ έχουμε προσθέσει ένα μικρό περιθώριο +50 για να μην βγαίνει το gameObject εκτός γράφου.

Παράδειγμα 2^ο

Επιλογή παιχνιδιού	Κλικ στο κουτί
Επιλογή δυσκολίας	Δύσκολη
Μέγιστο σκορ	984
Πρώτη ημερομηνία δοκιμής	10/05
Τελευταία ημερομηνία δοκιμής	26/05



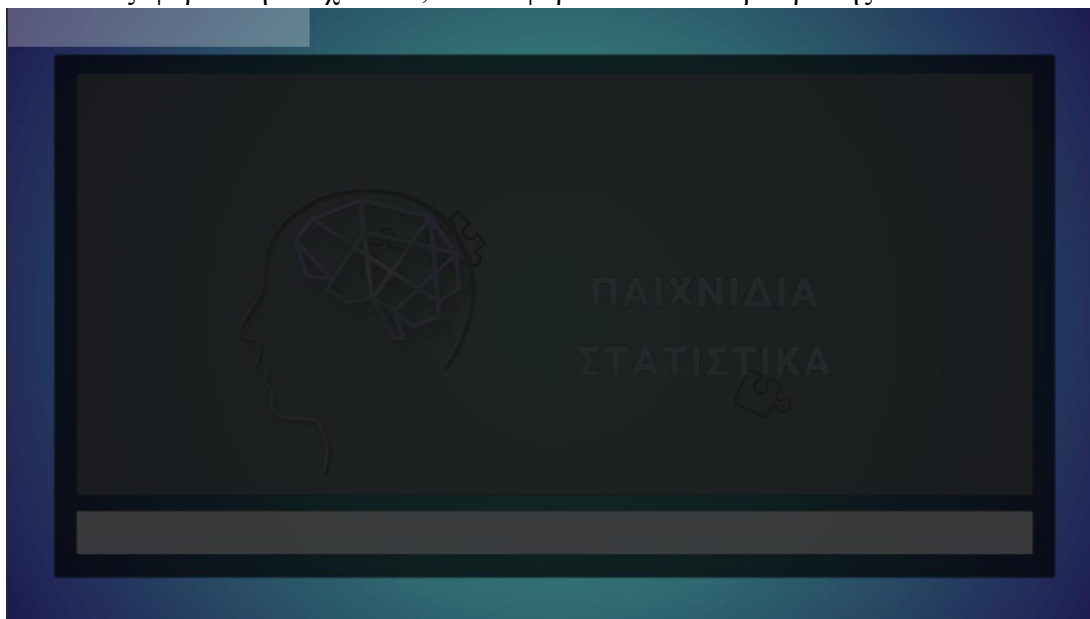
Εικόνα 31 Screenshot for Example 2

Με το δεύτερο παράδειγμα μπορούμε να επιβεβαιώσουμε ότι ο γράφος λειτουργεί άψογα καθώς όλα τα values είναι στην θέση που χρειάζονται.

Οι δοκιμές των στατιστικών έγιναν δημιουργώντας ένα script όπου εισάγει στον πίνακα Gameplay της βάσης 150 εγγραφές με τυχαία στατιστικά σε 4 χρήστες (600 εγγραφές σύνολο).

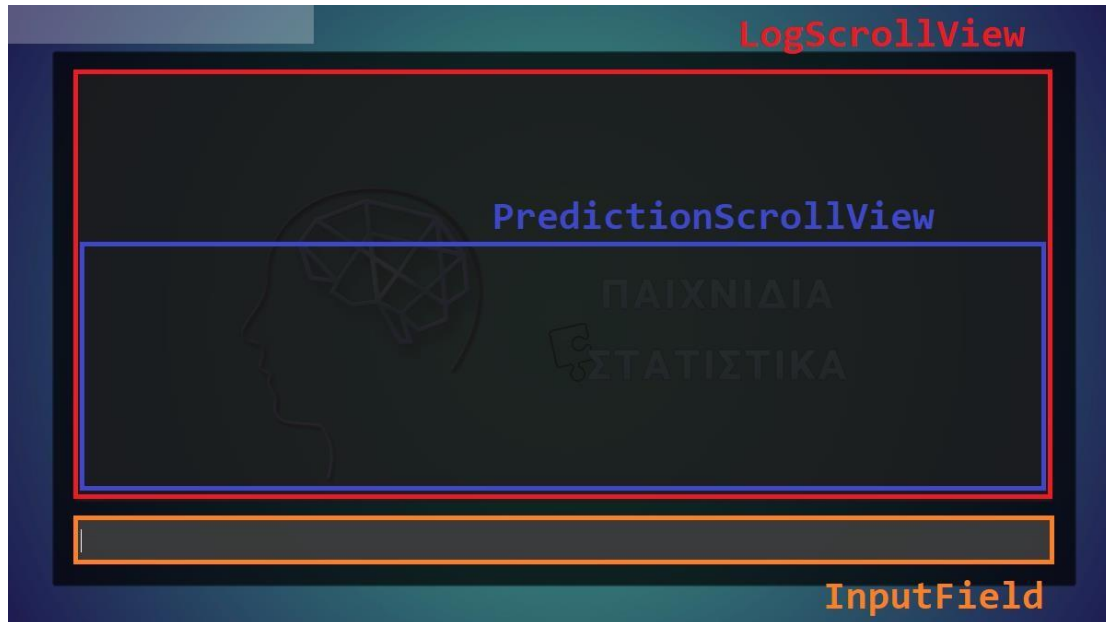
Κονσόλα Προγραμματιστών

Η κονσόλα προγραμματιστών είναι ένα από τα πιο ενδιαφέροντα χαρακτηριστικά που προσθέσαμε στην εφαρμογή μας το οποίο δεν αφορά τον ασθενή ή τον γιατρό αλλά μόνο τους developers. Μας διευκολύνει στο να τρέξουμε ορισμένα commands για να πραγματοποιήσουμε ενέργειες που δεν θα μπορούσαμε να τις κάνουμε διαφορετικά όπως debugging, άμεση επαφή με την βάση για να δούμε τα δεδομένα που θέλουμε, απευθείας φόρτωση παιχνιδιών, επαναφορά κωδικού πρόσβασης και πολλά άλλα.



Εικόνα 32 Screenshot of the Console Panel

Πατώντας το κουμπι «~» ενεργοποιείται το console και ταυτόχρονα κάνουμε σύνδεση με την βάση δεδομένων η οποία παραμένει ανοιχτή μέχρι να το απενεργοποιήσουμε κάνοντας χρήση του ίδιου κουμπιού.



Εικόνα 33 Screenshot with the 3 controls grids for the Console

Στην παραπάνω εικόνα παρουσιάζονται 3 controls μέσα σε grid για να είναι ολοκληρωμένο το console :

- 1) InputField : Πεδίο εισαγωγής εντολής
- 2) PredictionScrollView : Εμφανίζει τις προβλεπόμενες εντολές με βάση αυτά που έχει γράψει ο developer στο InputField.
- 3) LogScrollView : Εμφανίζει τα περιεχόμενα/αποτελέσματα της καταχωρημένης εντολής σε ένα πλαίσιο με δυνατότητα κύλισης.

Στο πεδίο εισαγωγής εντολής έχει γίνει η υλοποίηση αυτόματης συμπλήρωσης κειμένου με το πάτημα του κουμπιού “tab” γρηγορότερη και πιο εύκολη χρήση. Ο προγραμματιστής καθώς συνεχίζει και γράφει παρατηρούμε ότι μας εμφανίζει όλες τις διαθέσιμες εντολές που έχουμε βάλει να εμφανίζονται στο PredictionScrollView.



Εικόνα 34 Screenshot of the Console's Input Field

Για την υλοποίηση αυτού έχουμε φτιάξει μία λίστα από εντολές που ο προγραμματιστής μπορεί να βάλει. Για κάθε αλφαριθμητικό που προσθέτει γίνεται

συνεχής έλεγχος σε όλη την λίστα και μας εμφανίζει την πρώτη επιλογή στο πεδίο εισαγωγής εντολής και όλες τις υπόλοιπες στο PredictionScrollView ενώ του δίνεται η επιλογή να διαλέξει κάνοντας κλικ στην επιθυμητή εντολή. Κάθε εντολή ξεκινάει με slash «/» και στην συνέχεια αναλόγως τι θέλει να πραγματοποιήσει προσθέτει αντίστοιχο γράμμα ή λέξη.

Query	/q
Φόρτωση Παιχνιδιού	/load
Get (Sensitive data)	/get
Καθαρισμός του LogScrollView	/clear
Έξοδος εφαρμογής	/quit

Ο αριθμός εντολών είναι συγκεκριμένος στην εφαρμογής μας, μιας και αποτελεί βάση, σε enterprise applications όμως οι εντολές αυτές που μπορούν να προστεθούν μπορούν να είναι πολύ περισσότερες.

```

/quit
/clear
/load game1
/load game2
/load game3
/load game4
/get database
/get decoded_pass
/q select * from gameplay

```

Εικόνα 35 Screenshot of some commands

Μετά την επιλογή κάποιας εντολής πατώντας το κουμπί «Enter» μας εμφανίζει το αποτέλεσμα στο LogScrollView όπου μπορούμε να δούμε τα στατιστικά.

```

> /clear

> /q select * from gameplay where score > '500'
7 | 2 | 3 | 10/05/2022 | 12:52:24 | 12:53:45 | 01:21 | EASY | 914
9 | 3 | 3 | 10/05/2022 | 10:00:51 | 10:02:12 | 01:21 | EASY | 566
11 | 1 | 3 | 10/05/2022 | 03:16:47 | 03:18:41 | 01:54 | EASY | 844
12 | 3 | 1 | 10/05/2022 | 12:07:44 | 12:09:24 | 01:40 | MEDIUM | 817
13 | 2 | 3 | 10/05/2022 | 01:02:29 | 01:04:12 | 01:43 | EASY | 669
14 | 2 | 3 | 10/05/2022 | 17:20:01 | 17:21:54 | 01:53 | HARD | 606
16 | 1 | 3 | 10/05/2022 | 05:01:30 | 05:01:39 | 00:09 | MEDIUM | 762
27 | 3 | 1 | 10/05/2022 | 03:20:45 | 03:21:11 | 00:26 | EASY | 597
28 | 3 | 3 | 10/05/2022 | 08:38:21 | 08:40:07 | 01:46 | HARD | 541
30 | 3 | 3 | 10/05/2022 | 11:06:32 | 11:07:28 | 00:56 | HARD | 583
31 | 2 | 3 | 11/05/2022 | 14:22:52 | 14:23:28 | 00:36 | EASY | 649
34 | 1 | 1 | 11/05/2022 | 02:33:44 | 02:34:17 | 00:33 | EASY | 585
38 | 2 | 1 | 11/05/2022 | 17:10:01 | 17:11:14 | 01:13 | HARD | 738
39 | 1 | 2 | 11/05/2022 | 16:05:02 | 16:05:55 | 00:53 | MEDIUM | 569
41 | 2 | 3 | 11/05/2022 | 22:16:12 | 22:17:48 | 01:36 | MEDIUM | 710
43 | 1 | 3 | 11/05/2022 | 08:42:51 | 08:44:23 | 01:32 | HARD | 955
44 | 2 | 3 | 11/05/2022 | 18:23:43 | 18:24:40 | 00:57 | EASY | 736
45 | 3 | 3 | 11/05/2022 | 11:58:33 | 11:59:02 | 00:29 | EASY | 587
51 | 3 | 3 | 11/05/2022 | 19:37:41 | 19:39:05 | 01:24 | MEDIUM | 834
53 | 1 | 1 | 11/05/2022 | 16:08:42 | 16:09:11 | 00:29 | MEDIUM | 724
54 | 1 | 3 | 11/05/2022 | 21:33:33 | 21:34:33 | 01:00 | EASY | 542
55 | 1 | 1 | 11/05/2022 | 04:47:45 | 04:49:04 | 01:19 | EASY | 593
56 | 2 | 3 | 11/05/2022 | 18:16:02 | 18:16:24 | 00:22 | MEDIUM | 826

```

Εικόνα 36 Screenshot of results from the table “Gameplay”

2.3 ΔΙΑΘΕΣΙΜΑ ΠΑΙΧΝΙΔΙΑ

2.3.1 Ταίριαξε τα όμοια

Ένα παιχνίδι μνήμης που στοχεύει στο να θέσει σε δοκιμασία τη μνήμη του παίχτη και ταυτόχρονα να βελτιώσει τις αποδόσεις της.

Περιγραφή παιχνιδιού

Το «Ταίριαξε τα όμοια» ανήκει στην κατηγορία παιχνιδιών μνήμης. Είναι ένα πολύ κοινό παιχνίδι που ο χρήστης διακρίνει στην οθόνη του «ανοιχτές» κάρτες με ορισμένα φρούτα. Στα διαθέσιμα δευτερόλεπτα που έχει πρέπει να απομνημονεύσει όσες περισσότερες κάρτες μπορεί. Όταν περάσουν αυτά τα δευτερόλεπτα οι κάρτες θα «κλείσουν» και ο χρήστης θα πρέπει να βρει όλες τις όμοιες κάρτες ανά ζευγάρια (Π.χ. κεράσι - κεράσι) ώστε να εξαφανιστούν από την οθόνη. Στην περίπτωση που ο χρήστης βρει λάθος ζευγάρι (π.χ. κεράσι - μπανάνα) οι επιλεγμένες κάρτες θα «κλείσουν» όμως ο χρήστης θα έχει επίγνωση για το τι φρούτο υπάρχει σε αυτές τις λανθασμένες μαντεπιές. Σκοπός του παιχνιδιού είναι να βρει τα σωστά ζευγάρια όσο πιο γρήγορα γίνεται και με τις ελάχιστες λανθασμένες μαντεπιές ζευγαριών.

Δυσκολίες παιχνιδιού

	Ζευγάρια καρτών	Διαθέσιμα Δευτερόλεπτα	Πολλαπλασιαστής
Εύκολο	4	5	X1
Κανονικό	6	10	X2
Δύσκολο	8	15	X3

Controls

Left Mouse click (αριστερό κουμπί ποντικιού)

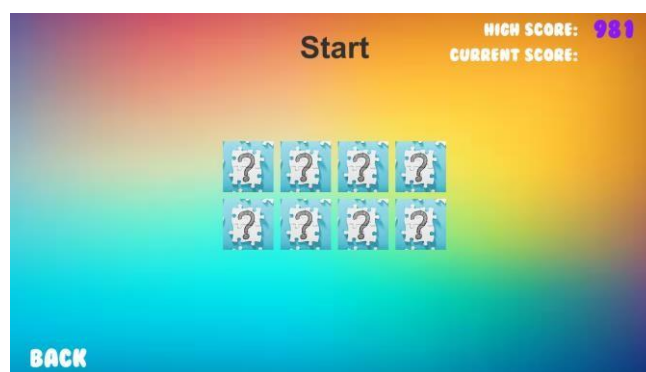
Υπολογισμός Σκορ

Score = (1500 – TotalGuesses * 50) * multiplier

Αναπαράσταση παιχνιδιού σε σειρά



Εικόνα 37 Screenshot #1 from the Game “Ταιριάξτε τα όμοια”



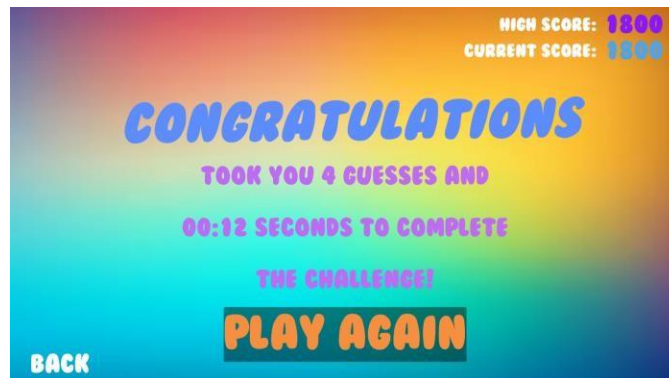
Εικόνα 38 Screenshot #2 from the Game “Ταιριάξτε τα όμοια”



Εικόνα 39 Screenshot #3 from the Game “Ταιριάξτε τα όμοια”



Εικόνα 40 Screenshot #4 from the Game “Ταιριάξτε τα όμοια”



Εικόνα 41 Screenshot #5 from the Game “Ταιριάξτε τα όμοια”

Υλοποίηση παιχνιδιού

Κατά την έναρξη της σκηνής του παιχνιδιού στην Awake function φορτώνουμε όλα τα δεδομένα που χρειαζόμαστε από την βάση δεδομένων καθώς και τα απαραίτητα resources.

Στην Start function καλούμε με σειρά τις μεθόδους που θα χρειαστούμε ώστε να γίνει η τοποθέτηση των κουμπιών μέσα στο transparent grid , να προστεθεί Listener κάθε κουμπί του grid ώστε να έχουν όλα την ίδια ενέργεια, να προστεθούν όλα τα resources και να γίνει «τυχαία ανάμιξη καρτών».

Μόλις γίνει και το «τυχαία ανάμιξη καρτών» ξεκινάει η αντίστροφη μέτρηση όπου μόλις φτάσει στο μηδέν αλλάζουμε το background των κουμπιών και ο χρήστης μπορεί να κάνει πλέον interact.

Στην συνέχεια, κάθε φορά που ο χρήστης διαλέγει μία κάρτα αλλάζει το background στο αντίστοιχο φρούτο που έχουμε θέσει εξ 'αρχής και κρατάμε το index της κάρτας σε μία μεταβλητή. Το ίδιο γίνεται και για την επιλογή της δεύτερης κάρτας

αλλά μετά γίνεται έλεγχος των 2 καρτών εάν είναι ίδιες (βασισμένες των index των καρτών) και μας οδηγεί σε 2 περιπτώσεις:

Στην περίπτωση διαφορετικών indexes, η συνθήκη μας είναι false οπότε και «κλείνουμε» τις συγκεκριμένες κάρτες

Στην περίπτωση που τα indexes είναι ίδια και η συνθήκη μας είναι true, εξαφανίζουμε από το grid τα συγκεκριμένα κουμπιά και ερχόμαστε ξανά στο σημείο όπου ο χρήστης πρέπει να διαλέξει άλλο ζευγάρι καρτών. Όταν τα CorrectGuesses δηλαδή οι σωστές μαντεπιές είναι ίσες με τον αριθμό ζευγαριών τότε γίνεται ο υπολογισμός του score, έλεγχος για HighScore, προστίθεται η συγκεκριμένη προσπάθεια στον πίνακα Gameplay της βάσης δεδομένων και μας εμφανίζεται ένα πάνελ όπου τελείωσε το παιχνίδι όπως στην εικόνα ταδε.

2.3.2 Αποφύγετε τα εμπόδια

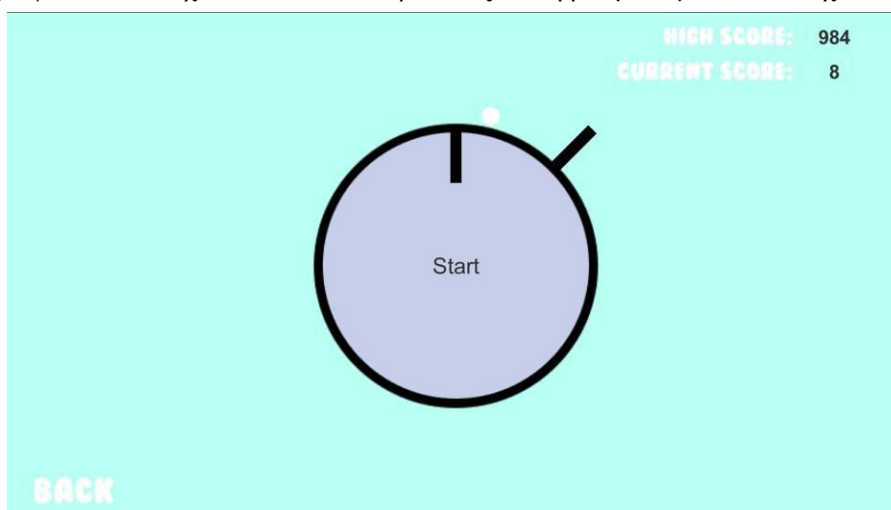
Ένα παιχνίδι ανακλαστικών που στοχεύει στο ότι ο παίχτης πρέπει να έχει την συγκέντρωση να αποφύγει τα εμπόδια που συναντάει, καθώς και να είναι σε ετοιμότητα και για τα επόμενα που ακολουθούν.

Περιγραφή παιχνιδιού

Το «Αποφύγετε τα εμπόδια» ανήκει στην κατηγορία παιχνιδιών ανακλαστικών. Εφόσον, ο παίχτης αποφασίσει την δυσκολία που θέλει να δοκιμάσει, στη συνέχεια εφόσον πατήσει το κουμπί “Παίξτε” το παιχνίδι ξεκινάει.

Στην εκκίνηση του παιχνιδιού γίνεται αντίστροφη μέτρηση 5 δευτερολέπτων και το παιχνίδι ξεκινάει. Αυτό που παρουσιάζεται στην οθόνη και αντικρίζει ο χρήστης - παίκτης είναι ένας μεγάλος κύκλος και ένας ακόμη μικρότερος, ο οποίος είναι και ο παίκτης μας. Στην συνέχεια ο χρήστης συναντά εμπόδια στη πορεία του, τα οποία πρέπει να αποφύγει όσο περισσότερα από αυτά μπορεί.

Το συγκεκριμένο παιχνίδι τερματίζει οπότε ο χρήστης βρεθεί σε επαφή με κάποιο εμπόδιο. Τελικώς, εφόσον το παιχνίδι τελειώσει παρουσιάζεται η βαθμολογία που πέτυχε.



Εικόνα 42 Screenshot #1 from the Game “Αποφύγετε τα εμπόδια”

Δυσκολίες παιχνιδιού

	Εμπόδια	Διαθέσιμα Δευτερόλεπτα	Πολλαπλασιαστής
Εύκολο	4	5	X1
Κανονικό	6	5	X2
Δύσκολο	8	5	X3

Χειριστήρια

- Mouse Left Click (Αριστερό κουμπί ποντικιού)
- Space (Κενό πληκτρολογίου)

Ο χρήστης μπορεί να διαλέξει ένα από τα παραπάνω χειριστήρια για την αποφυγή των εμποδίων.

Υπολογισμός Σκορ

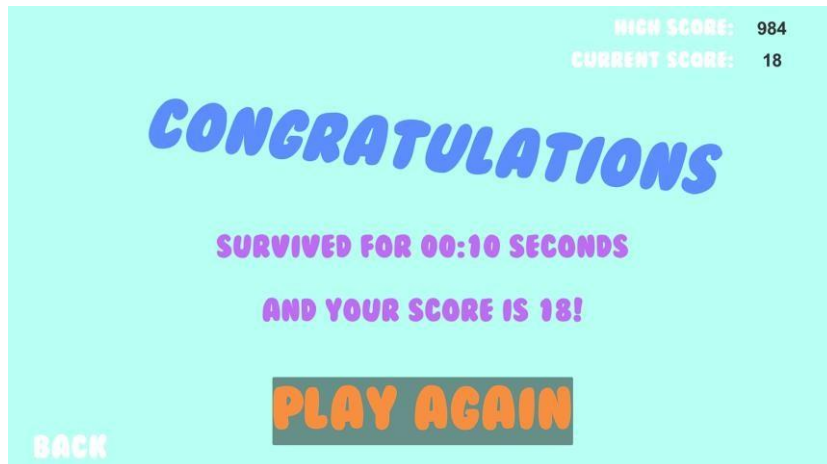
$$\text{Score} = \text{obstacles} * \text{multiplier}$$

Υλοποίηση παιχνιδιού

Κατά την έναρξη της σκηνής του παιχνιδιού στην Awake function φορτώνουμε τα δεδομένα που χρειάζονται καθώς και δημιουργούμε προγραμματιστικά τα εμπόδια στο εσωτερικό αλλά και στο εξωτερικό φλοιό του κύκλου που θα ακολουθάει ο Player.

Για να μπορέσουν να εφαρμόσουν σωστά τα εμπόδια επάνω στον κύκλο πρέπει να τους δώσουμε σωστο rotation στο Zaxis. Για να υλοποιηθεί αυτό πρέπει να βασιστούμε στο θεώρημα του 'Όιλερ' (Euler) και να καλέσουμε την συνάρτηση Quaternion.Euler(x,y,z). Όταν ο χρήστης ξεκινήσει το παιχνίδι η επιλογή εμποδίου στο εσωτερικό ή εξωτερικό φλοιό του κύκλου είναι τυχαία με πιθανότητα 50/50. Κάθε φορά που ο χρήστης περνάει ένα εμπόδιο αυτομάτως εμφανίζεται το επόμενο εμπόδιο, εξαφανίζεται το προηγούμενο και αυτό καθορίζεται αναλόγως του index του συγκεκριμένου εμποδίου όπου βρίσκεται εκείνη την στιγμή και αναβαθμίζεται το score του.

Στην μοναδική περίπτωση που ο χρήστης δεν καταφέρει να αποφύγει ένα εμπόδιο, τότε έχουμε μια σειρά από events για τον υπολογισμό του σκορ, σύγκριση υψηλότερου σκορ, εισαγωγή νέας σειράς στον πίνακα Gameplay της βάσης δεδομένων με τα συγκεκριμένα στοιχεία και φόρτωση του καμβά με τον τίτλο επιτυχίας.



Εικόνα 43 Screenshot #2 from the Game “Αποφύγετε τα εμπόδια”

2.3.3 Κλικ στο κουτί

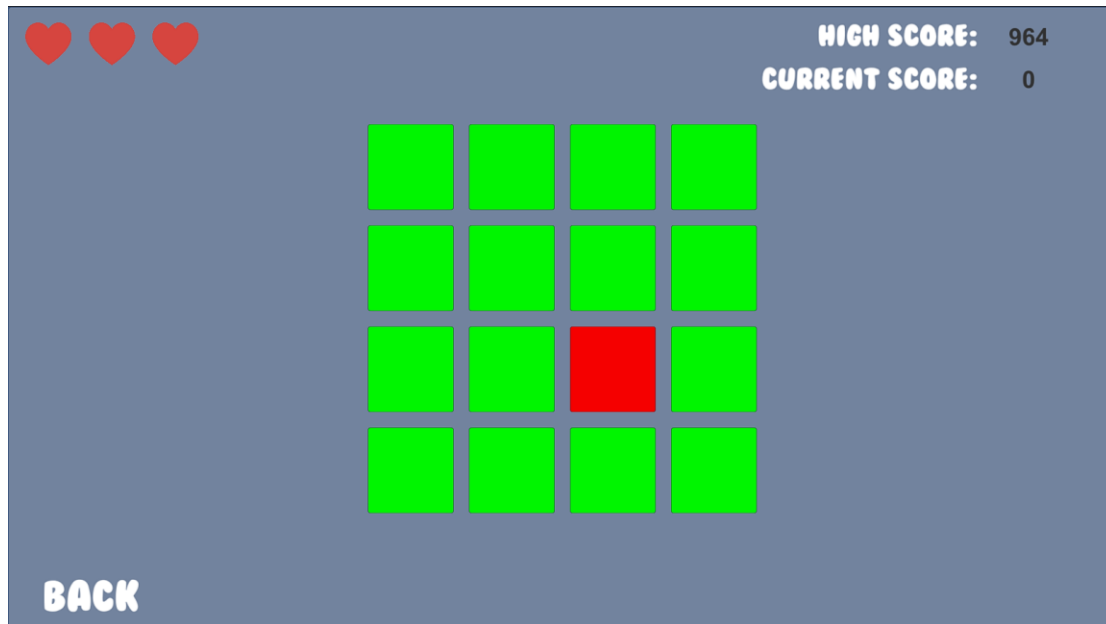
Ένα παιχνίδι αντανακλαστικών που στοχεύει στο ότι ο παίχτης πρέπει να κάνει κλικ στο επιθυμητό κουτί, όσο πιο γρήγορα μπορεί.

Περιγραφή παιχνιδιού

Το παιχνίδι “Κλικ στο κουτί” είναι μια ακόμη επιλογή στη κατηγορία των παιχνιδιών αντανακλαστικών. Εφόσον ο παίχτης αποφασίσει την δυσκολία που θέλει να δοκιμάσει, στη συνέχεια εφόσον πατήσει το κουμπί “Παίξτε” το παιχνίδι ξεκινάει.

Στη οθόνη του παίχτη εμφανίζεται ένα πλέγμα από κουτιά - τετράγωνα. Τα πράσινα είναι τα μη-έγκυρα και το κόκκινο είναι ο στόχος. Επίσης, δίνονται στον χρήστη τρεις ζωές όπου φαίνονται στην οθόνη του. Στόχος του παίκτη, είναι να κάνει κλικ στο κόκκινο κουτί – τετράγωνο από το οποίο και θα συλλέξει πόντους που θα προσμετρήσουν στην συνολική του βαθμολόγηση. Εάν ο παίχτης δεν καταφέρει να κάνει κλικ στο στόχο του ή καθυστερήσει τότε χάνει μια ζωή.

Το παιχνίδι τερματίζει όταν ο παίχτης χάσει όλες τις διαθέσιμες ζωές του ενώ κατά τον τερματισμό του παιχνιδιού θα εμφανιστεί στην οθόνη το τελικό σκορ που κατάφερε κατά την προσπάθειά του.



Εικόνα 44 Screenshot #1 from the Game “Κλικ στο κουτί”

Δυσκολίες παιχνιδιού

	Κουτιά	Διαθέσιμα Δευτερόλεπτα	Πολλαπλασιαστής
Εύκολο	9	5	X1
Κανονικό	16	5	X2
Δύσκολο	25	5	X3

Χειριστήρια

- Mouse Left Click (Αριστερό κουμπί ποντικιού)

Υπολογισμός Σκορ

$$\text{Score} = \text{CorrectButtons} * \text{multiplier}$$

Υλοποίηση παιχνιδιού

Κατά την έναρξη της σκηνής του παιχνιδιού στην Awake function φορτώνουμε όλα τα δεδομένα που χρειαζόμαστε από την βάση δεδομένων καθώς και τα απαραίτητα

resources και ορίζουμε 3 ζωές. Στην συνέχεια στην Start function προσθέτουμε μία μέθοδο «StartRound()» όπου είναι η βασική μας μέθοδος για την υλοποίηση του παιχνιδιού. Καθώς καλείται η StartRound δημιουργούμε ένα πάνελ όπου μέσα γίνονται Instantiate όλα τα κουμπιά αναλόγως την δυσκολία του παιχνιδιού και δίνουμε σε ένα Random κουμπί διαφορετικό tag και χρώμα ώστε να αναγνωρίζουμε ποιο κουμπί πρέπει να πατήσουμε. Εφόσον όλα είναι κουμπιά μπορούμε να έχουμε πρόσβαση στο OnClick event και να καλέσουμε εμφωλευμένο event “AddListener(UnityAction call)” για να δημιουργήσουμε μία custom μέθοδο όπου θα μπορέσουμε να υλοποιήσουμε τι θα κάνει το κουμπί κάθε φορά που το πατάμε. Επίσης, η AddListener χρειάζεται ένα UnityAction. Σαν παράμετρο χρησιμοποιούμε «Lambda Expression» την custom μέθοδο και περνάμε το κουμπί στην μέθοδο. Κάθε φορά που ο χρήστης πατάει ένα κουμπί υπάρχουν γίνεται έλεγχος εάν πατήθηκε το σωστό κουμπί ή όχι.

1) Επιλογή σωστού button ή μη επιλογή button (Idle State)

Όταν ο χρήστης πατήσει το σωστό button (Δηλαδή το button με το κόκκινο χρώμα) αναβαθμίζεται το σκορ του και προσθέτει μία ζωή.

2) Επιλογή λάθους button ή μη επιλογή button (Idle State)

Όταν ο χρήστης επιλέξει λάθος button ή δεν πατήσει τίποτα τότε χάνει μία ζωή

Όπως αναφέραμε παραπάνω, η StartRound() μέθοδος βρίσκεται στην Start() και για να μπορέσουμε να επαναλαμβάνουμε την ίδια μέθοδο συνέχεια καλούμε ένα behavior της Unity που ονομάζεται «InvokeRepeating(string methodName, float time, float repeatRate)». Ως παραμέτρους περνάμε την “StartRound”, σε πόσα δευτερόλεπτα θέλουμε να ξεκινήσει (στην περίπτωση μας 5.0f όπου είναι 5 δευτερόλεπτα) και πόσο γρήγορα θέλουμε να καλείται η Start, κάτι που ορίζει ανάλογα και την δυσκολία του παιχνιδιού.

Όταν ο χρήστης φτάσει στο σημείο να έχει 0 ζωές τότε καλούμε την CancelInvoke() για να μην ξανακαλέσουμε την Start, γίνεται υπολογισμός του σκορ, έλεγχος του υψηλότερου σκόρ και εμφάνιση του τίτλου επιτυχίας.



Εικόνα 45 Screenshot #2 from the Game “Κλικ στο κουτί”

2.3.4 Εκπαίδευση όρασης

Περιγραφή παιχνιδιού

Το παιχνίδι αυτό εξυπηρετεί κυρίως τον σκοπό της εξάσκησης του χρήστη - παίκτη ώστε σε καθημερινή βάση να βελτιώνει προοδευτικά τη συγκέντρωση του. Για να μπορέσει να επιτευχθεί αυτό, πρέπει να είναι σε ετοιμότητα τα ανακλαστικά του καθώς και η μια από τις αισθήσεις του που χρησιμοποιεί, όπως είναι η όραση.

Αρχικά, όταν επιλέγεται το παιχνίδι, δίνεται η επιλογή του στόχου και στην συνέχεια πατώντας το κουμπί “Παίξτε” το παιχνίδι ξεκινάει.

Το κύριο σημείο εδώ, είναι να εκπαιδευτεί ο χρήστης στον συντονισμό της εστίασης των ματιών του. Αυτό επιτυγχάνεται όταν ο παίκτης - ασθενής εστιάζει τα μάτια του στο στόχο/σημάδι, μέχρι να παρατηρήσει ότι δημιουργείται μια “οπτική ψευδαίσθηση” και οι πράσινες κουκκίδες σταδιακά αλλάζουν χρώμα σε μωβ και τελικώς αρχίζουν να εξαφανίζονται. Το παραπάνω μπορεί να διαρκέσει όσο χρόνο επιθυμεί να αφιερώσει ο κάθε παίκτης.

3 ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ WPF ΜΕ ΣΚΟΠΟ ΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ DEVELOPER-CLIENT

Στην διάρκεια της υλοποίησης της εφαρμογής μας Brain Exercises ως developers χρειαζόμασταν ένα external application για την επικοινωνία μεταξύ των developers και γιατρού. Βασιζόμενοι σε αυτό δημιουργήσαμε μία εφαρμογή, υλοποιώντας την σε WPF (Windows Presentation Foundation) και σε .net Core 3.1 framework. Ο σκοπός της εφαρμογής αυτής εκτός της επικοινωνίας είναι και η αυτόματη μεταφορά της τοπικής βάσης δεδομένων που δημιουργείται κατά την πρώτη εκκίνηση της εφαρμογής Brain Exercises ώστε να μπορούμε να λαμβάνουμε τα δεδομένα των ασθενών του κάθε ιατρού καθώς και η γρήγορη αποκατάσταση του κωδικού πρόσβασης του ιατρού σε περίπτωση που αυτός χαθεί ή δεν μπορεί να τον θυμηθεί.

3.1 ΕΙΣΑΓΩΓΗ ΣΤΟ WPF

- Τι είναι το Windows Presentation Foundation;

Το Windows Presentation Foundation είναι ένα δωρεάν εργαλείο και ένα σύνολο από υπηρεσίες ανοιχτού κώδικα για προγραμματιστές και προγραμματιστές της Visual C# (VC#) με πολυετή εμπειρία στον προγραμματισμό C#, την ανάπτυξη διεπαφής χρήστη και τη σύνταξη κώδικα C#. Από την έναρξή του, το έργο αναβαθμίζεται συνεχώς για να παρέχει στους προγραμματιστές περισσότερο έλεγχο, μεγαλύτερη ευκολία στη χρήση, καλύτερη ασφάλεια και αξιοπιστία.

- Πώς χρησιμοποιείται?

Η υλοποίηση του Windows Presentation Foundation στηρίζεται πάνω σε βασικές τεχνολογίες Windows και βιβλιοθήκες λογισμικού, συμπεριλαμβανομένων των C#, Visual Basic, Go και GoT, καθώς και πολλών άλλων βιβλιοθηκών, όπως OpenCL, C. Ο στόχος αυτού του έργου είναι να παρέχει την ευελιξία να έχουμε ένα έργο για κάθε τύπο ανάπτυξης.

- Πώς χρησιμοποιείται από τους προγραμματιστές;

Αυτό είναι κυρίως ένα εργαλείο για προγραμματιστές, όπως ακριβώς και κάθε άλλο παρόμοιο εργαλείο. Μπορεί επίσης να χρησιμοποιηθεί στην ανάπτυξη και τη δημιουργία εφαρμογών για iOS και Android. Το πρόγραμμα μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη και τη δημιουργία περιεχομένου ιστού για τις εφαρμογές ιστού, ακόμη και για παιχνίδια.

3.2 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΕΦΑΡΜΟΓΗΣ

Κατά την διάρκεια υλοποίηση της εφαρμογής BrainExercise σαν Developers είδαμε ότι χρειαζόμασταν μία external εφαρμογή που θα χρησιμοποιεί ως utility από τον γιατρό για τις διαφορετικές ανάγκες που του παρέχει. Η συγκεκριμένη εφαρμογή προσφέρει 3 απαραίτητα features τα οποία είναι :

1. Να μπορεί να μας στέλνει ο γιατρός την βάση δεδομένων ώστε σαν developers να μπορέσουμε να κάνουμε καταγραφή.
2. Την άμεση ανάκτηση μοναδικού κωδικού πρόσβασης που χρειάζεται ο γιατρός ώστε να κάνει οποιαδήποτε κίνηση στην εφαρμογή BrainExercises.
3. Το feedback που μπορεί ο γιατρός να μας παρέχει ώστε να μπορέσουμε να αναβαθμίσουμε την εφαρμογή και να γίνει ακόμα καλύτερη και user friendly.

BrainExerciseExternalApp

Αξιολόγηση Σχετικά με εμάς

- Παρακαλώ συμπληρώστε την παρακατω φορμα αξιολογησης καθως και στην περιγραφη αναφερατε αν σας αρεσε ή δεν σας αρεσε.
- Σε περιπτωση που ξεχασατε τον κωδικο πατηστε το κουμπι Forgot Password και θα σας εμφανιστει στην οθονη σας

Όνομα*

Επώνυμο*

Email*

Περιγραφή

Εκτίμηση

Bad Poor Fair Good Excellent

Send Feedback Forgot Password

Εικόνα 46 Screenshot της Αξιολογησης

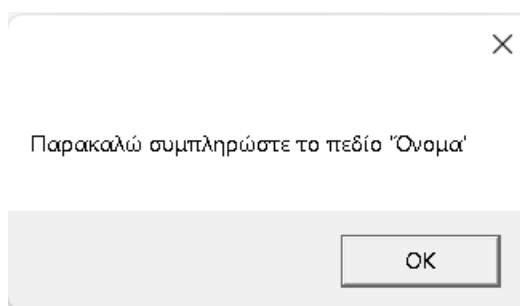
Όπως παρατηρούμε από την παραπάνω εικόνα κατά την εκκίνηση της εφαρμογής υπάρχει ένα Tab Control που αποτελείται από την «αξιολόγηση» όπως επίσης και το «Σχετικά με εμάς».

Ακόμη κατά την φάση της αξιολόγησης υπάρχουν 2 labels τα οποία εξηγούν στον γιατρό πως να χρησιμοποιήσει την εφαρμογή, στην συνέχεια υπάρχει η φόρμα αξιολόγησης και τέλος δύο κουμπιά για την εφαρμογή της επιλογής του.

1) Send Feedback button

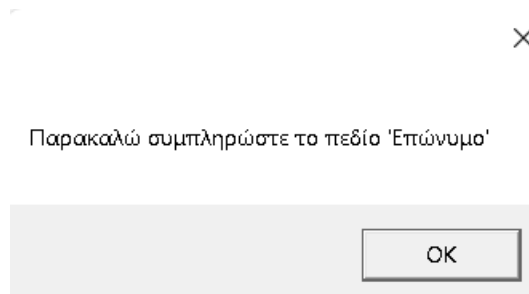
Στην περίπτωση που ο γιατρός θέλει να μας στείλει την βάση δεδομένων και/ή να μας αξιολογήσει θα πρέπει να πατήσει το κουμπι Send Feedback. Βέβαια, απαραίτητη θεωρείται η υποχρεωτική συμπλήρωση των πεδίων με το σύμβολο «*» διότι γίνεται έλεγχος «isNullOrEmpty» για την αποφυγή ελειπών στοιχείων πριν την αποστολή της βάσης δεδομένων.

Όταν το πεδίο «Όνομα» είναι κενό



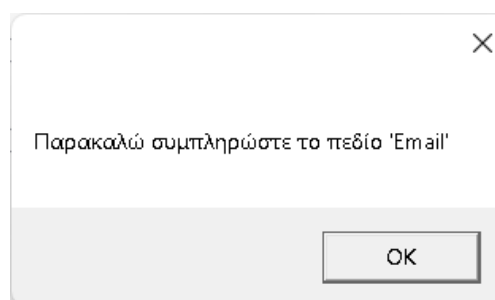
A validation dialog box with a close button (X) in the top right corner. The text inside reads: "Παρακαλώ συμπληρώστε το πεδίο 'Όνομα'". At the bottom right, there is an "OK" button.

Όταν το πεδίο «Επώνυμο» είναι κενό



A validation dialog box with a close button (X) in the top right corner. The text inside reads: "Παρακαλώ συμπληρώστε το πεδίο 'Επώνυμο'". At the bottom right, there is an "OK" button.

Όταν το πεδίο «Email» είναι κενό



A validation dialog box with a close button (X) in the top right corner. The text inside reads: "Παρακαλώ συμπληρώστε το πεδίο 'Email'". At the bottom right, there is an "OK" button.


```
1 if (onoma.Text == null || onoma.Text == String.Empty)
2 {
3     MessageBox.Show("Παρακαλώ συμπληρώστε το πεδίο 'Όνομα'");
4     return;
5 }
6 else if (eponimo.Text == null || eponimo.Text == String.Empty)
7 {
8     MessageBox.Show("Παρακαλώ συμπληρώστε το πεδίο 'Επώνυμο'");
9     return;
10 }
11 else if (email.Text == null || email.Text == String.Empty)
12 {
13     MessageBox.Show("Παρακαλώ συμπληρώστε το πεδίο 'Email'");
14     return;
15 }
```

Code Snippet 27 Έλεγχος κενών στοιχείων

Εφόσον, ο γιατρός συμπληρώσει τα υποχρεωτικά στοιχεία έχει την επιλογή να βάλει μία περιγραφή και να αξιολογήσει την εφαρμογή που διαθέτει 5 επίπεδα αξιολόγησης (bad, poor, fair, good, excellent) .

Την στιγμή που πατήσει το κουμπί Send Feedback ένα script φτιάχνει ένα email με το κατάλληλο configuration ώστε να μπορέσει να σταλθεί μέσω Gmail Client.

```
1 SmtplibClient client = new SmtplibClient("smtp.gmail.com", 587);
2     client.EnableSsl = true;
3     client.DeliveryMethod = SmtplibDeliveryMethod.Network;
4     client.UseDefaultCredentials = false;
5     client.Credentials = new NetworkCredential("brainexercisesclicien@gmail.com", "*****");
6     MailMessage mail = new MailMessage();
```

Code Snippet 28 Δημιουργία SmtplibClient για την εισαγωγή σε Gmail από την εφαρμογή

Κατόπιν, για την ολοκληρωμένη δημιουργία ενός email χρειαζόμαστε το subject, body, attachment, sender και receiver.

Για το συγκεκριμένο παράδειγμα δημιουργήσαμε εμείς 2 Gmail accounts :

- Sender: brainexercisescient@gmail.com
- Receiver: brainexercisegames@gmail.com

Ο συγκεκριμένος τρόπος είναι αρκετά αποτελεσματικός διότι για μικρές εφαρμογές όπως η δική μας, η Google για να αποτρέψει το Spam θα πρέπει να έχει απενεργοποιηθεί από τις ρυθμίσεις το Gmail το “Less Secure App” κάτι που είναι επικίνδυνο.

Στην συνέχεια, εφόσον αναφέραμε τον μηχανισμό που θα χρησιμοποιήσουμε για την επικοινωνία έχουμε βάλει στατικό subject (“Database”), ένα StringBuilder για την εύκολη δημιουργία του body όπως βλέπουμε και στο παρακάτω Snippet:

```

1  sb.AppendLine("Στοιχεία Γιατρού ");
2      sb.AppendLine();
3      sb.AppendLine("Όνομα : " + onoma.Text);
4      sb.AppendLine("Επώνυμο : " + eponimo.Text);
5      sb.AppendLine("Email : " + email.Text);
6      sb.AppendLine("Περιγραφή : " + perigrifi.Text);
7      sb.AppendLine("Εκτίμηση : " + selectedRadioButton);
8      sb.AppendLine();
9      sb.AppendLine("Το παρόν email στάθληκε στις : " +
10     DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss"));

```

Code Snippet 29 Δημιουργία String Builder για το body του email

Για να επιτευχθεί η δημιουργία των παραπάνω, καθοριστικό ρόλο παίζει η επισύναψη του attachment στο οποίο θα περιλαμβάνεται η βάση δεδομένων μας. Για να γίνει αυτό πρέπει να δούμε πως μπορούμε να έχουμε access στο AppData από ένα c# WPF app και να προσθέσουμε αυτό το μονοπάτι.

```

1  string path = @"C:\Users\" + Environment.UserName + @"\AppData\LocalLow\DefaultCompany\Brain\BrainDB.s3db";
2
3  mail.Attachments.Add(new Attachment(path));

```

Code Snippet 30 Μονοπάτι της βάσης δεδομένων και προσθήκη στα συνημμένα του email

Όπως παρατηρούμε αυτό είναι ακριβώς το ίδιο μονοπάτι που έχουμε και στο BrainExercise με την μόνη διαφορά ότι στην Unity θα πρέπει να βάλουμε Application.PersistantDataPath ενώ εδώ χρειαζόμαστε Enviroment.UserName.

The screenshot shows a web browser window titled "BrainExerciseExternalApp" with a close button (X) in the top right corner. The page has a header with two tabs: "Αξιολόγηση" (selected) and "Σχετικά με εμάς". Below the header, there are two bullet points: "• Παρακαλώ συμπληρώστε την παρακατω φορμα αξιολογησης καθώς και στην περιγραφη αναφερετε αν σας αρεσε ή δεν σας αρεσε." and "• Σε περιπτωση που ξεχασατε τον κωδικο πατηστε το κουμπι Forgot Password και θα σας εμφανιστει στην οθονη σας". The form contains several input fields: "Όνομα*" with the value "Χρήστος", "Επώνυμο*" with "Σπύρης", and "Email*" with "chrisspyris@gmail.com". There is a larger text area for "Περιγραφή" containing two lines: "1)Πολύ καλή εφαρμογή" and "2)Χρειάζεται διόρθωση στο σημείο "X"". Below the text area is a "Εκτίμηση" section with five colored circles: red (Bad), orange (Poor), yellow (Fair), light green (Good), and dark green (Excellent). The "Excellent" circle is selected. At the bottom, there are two buttons: "Send Feedback" and "Forgot Password".

Εικόνα 47 Example Screenshot

Database Inbox x



brainexercisesclien@gmail.com

to me ▾

🇬🇷 Greek ▾ > English ▾ [Translate message](#)

Στοιχεία Γιατρού

Όνομα : Χρήστος

Επώνυμο : Σπύρης

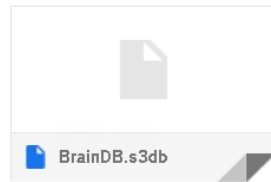
Email : chrisspyris@gmail.com

Περιγραφή : 1)Πολύ καλή εφαρμογή

2)Χρειάζεται διόρθωση στο σημείο "X"

Εκτίμηση : Excellent

Το παρόν email στάθληκε στις : 16/05/2022 16:51:13



Εικόνα 48 Screenshot of the Gmail example

Ακολουθώντας τα παραπάνω βήματα, ολοκληρώνεται η δημιουργία του email το οποίο θα μας παρέχει ήδη 2 από τα 3 features που έχει η εφαρμογή μας (να λάβουμε την βάση δεδομένων και να πάρουμε feedback από τον γιατρό).

2) Forgot Password

Παραπάνω αναφέραμε ότι μπορούμε να έχουμε access από την εφαρμογή μας στην βάση δεδομένων. Όμως με το path που βάλουμε παραπάνω έχουμε μόνο δικαιώματα read και όχι write. Για να ανακτήσουμε τον κωδικό πρόσβασης πρέπει να έχουμε όλα τα δικαιώματα καθώς επίσης να μπορούμε να αποκωδικοποιήσουμε τον κωδικό πρόσβασης του γιατρού από τον πίνακα Doctor ώστε να μπορεί να εμφανιστεί στην εφαρμογή μας σαν ASCII και όχι σαν base64 format.

Για να μπορέσουμε να πάρουμε τα δικαιώματα πρέπει να βάλουμε το DataSource μέσα στην αρχή του Uri όπως βλέπουμε στο Snippet παρακάτω:

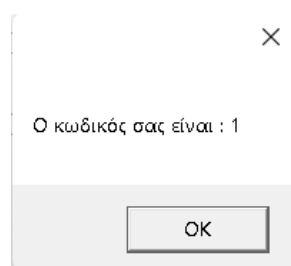
```
1 string path1 = @"Data Source=C:\Users\" + Environment.UserName
2               + @"\AppData\LocalLow\DefaultCompany\Brain\BrainDB.s3db";
```

Code Snippet 31 Μονοπάτι της βάσης δεδομένων

Από την στιγμή αυτή έχουμε πλήρη έλεγχο για να χρησιμοποιήσουμε String Queries ή LINQ API και ως αποτέλεσμα έχουμε το εξής:

```
1 private void ForgotPassword_Click(object sender, RoutedEventArgs e)
2     {
3         string path1 = @"Data Source=C:\Users\" + Environment.UserName +
4                       @" \AppData\LocalLow\DefaultCompany\Brain\BrainDB.s3db";
5
6         SQLiteConnection sqlite_conn = new SQLiteConnection(path1);
7         sqlite_conn.Open();
8
9         string stringQuery = "SELECT PASSWORD FROM DOCTORPASS";
10        var SqliteCmd = new SQLiteCommand();
11        SqliteCmd = sqlite_conn.CreateCommand();
12        SqliteCmd.CommandText = stringQuery;
13
14        var reader = SqliteCmd.ExecuteReader();
15        while (reader.Read())
16        {
17            base64password = reader.GetString(0);
18        }
19
20        var plainPassword = Base64Decode(base64password);
21        MessageBox.Show("Ο κωδικός σας είναι : " + plainPassword);
22    }
```

Code Snippet 32 Μέθοδος για επαναφορά κωδικού γιατρού



Εικόνα 49 Screenshot of the password prompt

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η διαδικασία της υλοποίησης της πτυχιακής αυτής πραγματοποιήθηκε με αρκετή μελέτη πάνω στη Unity, καθώς και με τη χρήση της γλώσσας προγραμματισμού C#. Σημαντικό εδώ είναι πως παρόλο τη παραπάνω μελέτη που πραγματοποιήθηκε μέσω της βιβλιογραφίας και την συνολική δημιουργία του serious game, αποκτήθηκαν νέες και επιπρόσθετες γνώσεις.

Επιπλέον, υπήρχαν πολλά ερεθίσματα τα οποία σημειώθηκαν για μελλοντικές ενέργειες. Τέτοιες μπορεί να είναι όπως το να προστεθούν επιπλέον γλώσσες στο παιχνίδι και περισσότερες αναλύσεις για να είναι συμβατό με μεγαλύτερη πληθώρα υπολογιστών στο μέλλον. Επιπροσθέτως, υπάρχουν πολλές ιδέες και σκέψεις για επιπλέον κατηγορίες και παραπάνω παιχνίδια αντίστοιχα.

Σημαντική πληροφορία που πρέπει να αναφερθεί εδώ είναι ότι η βάση είναι δημιουργημένη σε μια πλατφόρμα, που μέσω αυτής είναι εύκολο να γίνει migrate σε μια άλλη. Αυτό σημαίνει παραπάνω controllers και haptics, θα μπορούσαν να δουλέψουν ακόμη και για περισσότερα σημεία του σώματος. Ένα παράδειγμα εδώ είναι ότι μπορεί να χρησιμοποιηθεί “joystick” για να βοηθήσει κάποιος που αντιμετωπίζει πρόβλημα στον καρπό. Επίσης, το VR είναι μια ακόμη επιλογή που θα μπορούσε να δουλέψει για “full body” δοκιμασίες.

ΑΝΑΦΟΡΕΣ

- Alvarez, J., Alvarez, V., Djaouti, D., & Michaud, L. (2010). *Serious Games: Training & Teaching - Healthcare - Defence & security - Information & Communication*. IDATE.
- Brown, S. J., Lieberman, D. A., Gemeny, B. A., Fan, Y. C., Wilson, D. M., & Pasta, D. J. (1997). Educational video game for juvenile diabetes: results of a controlled trial. *Informatics for Health and Social Care*, 22(1), 77- 89.
- Caspian Learning. (2008). *Serious Games in Defence Education*. Royaume-Uni: Caspian Learning
- Herz, J. C. (1997). *Joystick Nation: How Videogames Ate Our Quarters, Won Our Hearts, and Rewired Our Minds* (1er ed.). Little, Brown and Company.
- Ioannides, A. (2017). *The art of Lorem Ipsum*. Ανάκτηση 5 1, 2017, από <http://ipsum.com>
- Kohler, C. (2004). *Power-Up: How Japanese Video Games Gave the World an Extra Life*. BRADY GAMES.
- Kostopoulos, G. (2010). This the title of the journal. *IEEE Transactions on Brain Imaging*, 12(2), 233-250.
- Koutras, A. (2016). *This is the book's title* (2nd εκδ.). Patras: Elsevier.
- Michael, D., & Chen, S. (2005). *Serious Games: Games That Educate, Train, and Inform* (1er ed.). Course Technology PTR.
- Smillie, K. (2010). *Some Topics in Computing*. Keith Smillie. Retrieved from <http://webdocs.cs.ualberta.ca/~smillie/Topics/Topics.html>
- Smith, R. D. (2009). *Military Simulation & Serious Games: Where We Came From and Where We Are Going*. Modelbenders LLC.