

Ατομική Διπλωματική Εργασία

**ΥΛΟΠΟΙΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ ΠΟΥ ΑΦΟΡΑ
ΤΗΝ ΜΕΣΟΓΕΙΑΚΗ ΔΙΑΤΡΟΦΗ ΓΙΑ ΠΑΙΔΙΑ ΗΛΙΚΙΑΣ 3
ΜΕΧΡΙ 5 ΕΤΩΝ**

Σταύρη Χαριλάου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δεκέμβριος 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Υλοποίηση Εκπαιδευτικού Παιχνιδιού που αφορά την Μεσογειακή διατροφή για
παιδιά ηλικίας 3 μέχρι 5 ετών**

Σταύρη Χαριλάου

Επιβλέπων Καθηγητής

Καθηγητής Κωνσταντίνος Σ. Παττίχης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Δεκέμβριος 2022

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή Κωνσταντίνο Σ. Παττίχη για όλη την καθοδήγηση και την βοήθεια που μου παρείχε κατά την διάρκεια εκπόνησης της Ατομικής Διπλωματικής Εργασίας μου σαν επιβλέπων καθηγητής, αλλά και για όλες τις συμβουλές και στήριξη του τον τελευταίο χρόνο των σπουδών μου.

Επίσης, θα ήθελα να ευχαριστήσω την κυρία Ειρήνη Σχίζα, η οποία έχει συμβάλλει σημαντικά στην διεκπεραίωση της εργασίας με τις συμβουλές της, την ανατροφοδότηση και την βοήθεια που μου έχει προσφέρει.

Τέλος, ευχαριστώ την οικογένεια μου και τους φίλους μου για όλη την κατανόηση και υποστήριξη που μου παρείχαν σε όλα τα χρόνια των σπουδών μου.

Περίληψη

Το θέμα της Ατομικής Διπλωματικής Εργασίας είναι η ανάπτυξη ενός εκπαιδευτικού παιχνιδιού που αφορά την Μεσογειακή Διατροφή. Το παιχνίδι υλοποιήθηκε σε μηχανή Unity και ο σκοπός του είναι να μάθει με διασκεδαστικό τρόπο, σε παιδιά ηλικίας 3 μέχρι 5 ετών να δημιουργούν υγιεινά και ισορροπημένα γεύματα.

Πριν από την υλοποίηση του παιχνιδιού, ήταν απαραίτητο να γίνει εκτενής έρευνα έτσι ώστε να καθορίσουμε τις απαιτήσεις του συστήματος, να γνωρίζουμε πως πρέπει να υλοποιηθεί ένα παιχνίδι για τις συγκεκριμένες ηλικίες, τι αρέσει στα παιδιά και τι θεωρούν ελκυστικό, τι αρχές σχεδιασμού πρέπει να ακολουθηθούν καθώς επίσης και σε τι mobile operating system πρέπει να υλοποιηθεί το παιχνίδι.

Ακόμη, εξετάστηκε ποια παιχνίδια είναι δημοφιλείς στην Κύπρο με σκοπό να διαπιστώσουμε τι κατηγορίες παιχνιδιών είναι προτιμότερες, καθώς επίσης βρέθηκαν παρόμοια συστήματα που αφορούν την σωστή διατροφή, και έγινε ανάλυση για τα πλεονεκτήματα και τα μειονεκτήματα του κάθε παιχνιδιού.

Με την ολοκλήρωση του παιχνιδιού, το παιχνίδι δόθηκε σε διάφορους χρήστες για αξιολόγηση έτσι ώστε να προτείνουν εισηγήσεις με σκοπό να βελτιωθεί το παιχνίδι και στο τέλος να έχουμε ένα ολοκληρωμένο εκπαιδευτικό παιχνίδι που εξυπηρετεί όλες τις ανάγκες των χρηστών.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Γενική Εισαγωγή	1
	1.2 Εισαγωγή σε παιχνίδια	3
	1.3 Εισαγωγή σε παιχνίδια για παιδιά	3
	1.4 Στόχος Ατομικής Διπλωματικής Εργασίας	4
	1.5 Δομή Ατομικής Διπλωματικής Εργασίας	5
Κεφάλαιο 2	Ανασκόπηση.....	7
	2.1 Top Charts παιχνιδιών στην Κύπρο	8
	2.2 Δημοφιλέστερα παιχνίδια για παιδιά	9
	2.2.1 Thinkrolls 2	10
	2.2.2 Lola's Alphabet Train	11
	2.2.3 Kids Preschool Learning Games	12
	2.2.4 Toca Life World: Build Stories	13
	2.2.5 Bini Super ABC	14
	2.2.6 Pepi House	14
	2.2.7 Toca Kitchen 2	15
	2.2.8 Kids Balloon Pop Game	16
	2.2.9 Masha and the Bear	17
	2.2.10 Piano Kids	17
	2.2.11 Toca Lab : Elements	18
	2.2.12 Σύγκριση	19
	2.3 Δημοφιλέστερα Mobile Operating Systems	20
	2.4 Άλλα παρόμοια συστήματα υγιεινής διατροφής	22
	2.4.1 Pick Your Plate	22
	2.4.2 Food Decisions	23
	2.4.3 Sortify: Nutrition	24
	2.4.4 Healthier Me	24
	2.4.5 Cerino	25

2.5 Παρουσίαση	26
2.5.1 Πλεονεκτήματα και Μειονεκτήματα κάθε παιχνιδιού	26
2.5.2 Σύγκριση	28
2.6 Δωρεάν και Paid Παιχνίδια	30
2.7 Τι χαρακτηρές αρέσουν στα παιδιά	31
2.8 Είδος παιχνιδιών που αρέσουν στα παιδιά	32
2.9 Χρώματα που τραβούν την προσοχή των παιδιών	33
 Κεφάλαιο 3	
Απαιτούμενη γνώση και Τεχνολογίες, και Ανάλυση Απαιτήσεων και Προδιαγραφές.....	34
3.1 Unity	34
3.2 Visual Studio	34
3.3 Modelio	35
3.4 Ανάλυση Requirements για σχεδιασμό παιχνιδιών για παιδιά	35
3.5 Εξειδικευμένες αρχές που θα ακολουθήσει το παιχνίδι	39
3.6 Use case	40
3.6.1 Περιγραφή διαγράμματος	40
 Κεφάλαιο 4	
Σχεδιασμός Συστήματος και Υλοποίηση.....	42
4.1 Σχεδιασμός Συστήματος	42
4.2 Υλοποίηση	47
4.2.1 Κουμπιά – Πλοήγηση	47
4.2.2 Εκκλείδωμα χαρακτήρων και levels	49
4.2.3 Score	50
 Κεφάλαιο 5	
Αξιολόγηση Συστήματος - Αποτελέσματα - Συζήτηση.....	51
5.1 Αξιολόγηση Συστήματος	51
5.2 Αποτελέσματα	52
5.3 Συζήτηση	55
 Κεφάλαιο 6	
Συμπεράσματα και Μελλοντική Εργασία.....	56
6.1 Συμπεράσματα	56

Βιβλιογραφία	A-1
Παράρτημα Α – Οδηγίες Εγκατάστασης.....	A-4
Παράρτημα Β – Οδηγίες Χρήσης.....	A-7
Παράρτημα Γ – Κώδικες – Τεχνικός Οδηγός.....	A-13

Κατάλογος Εικόνων

Εικόνα 1.1: Πυραμίδα Μεσογειακής διατροφής

Εικόνα 1.3: Bronkie the Bronchiasaurus game

Εικόνα 3.4.1: Play Button

Εικόνα 3.4.2 : Το βιντεοπαιχνίδι Journey δημιουργεί στους παίκτες διαφορετικά συναισθήματα λόγω του περιβάλλοντος.

Εικόνα 4.1.1 Χαρακτήρας Μήλο

Εικόνα 4.1.2 Χαρακτήρας Πορτοκάλι

Εικόνα 4.1.3 Χαρακτήρας Αβοκάντο

Εικόνα 4.1.4 Χαρακτήρας Μπανάνα

Εικόνα 4.1.5 Χαρακτήρας Τυρί

Εικόνα 4.1.6 Χαρακτήρας Αυγό

Εικόνα 4.1.6 Χαρακτήρας Σταφύλι

Εικόνα 4.1.7 Χαρακτήρας Nutella

Εικόνα 4.1.8 Χαρακτήρες Παγωτό

Εικόνα 4.1.9 Χαρακτήρας Ελαιόλαδο

Εικόνα 4.1.10 Αρχικό μενού

Εικόνα 4.1.11 Level 1 Στάδιο 1 - Πρωινό

Εικόνα 4.1.12 Αύξηση score

Εικόνα 4.1.13 Μείωση score

Εικόνα 4.1.14 Χάρτης με τα levels

Εικόνα 4.2.1.1 Κλειδωμένος Χαρακτήρας

Εικόνα 4.2.1.2 Screenshot από το video του tutorial

Εικόνα 5.2.1 Απαντήσεις για την ηλικία

Εικόνα 5.2.2 Απαντήσεις για το φύλο

Εικόνα 5.2.3 Απαντήσεις για το τι άρεσε στους χρήστες

Εικόνα 5.2.4 Απαντήσεις για το τι δεν άρεσε στους χρήστες

Εικόνα 5.2.5 Εισηγήσεις

Κατάλογος Σχημάτων

Σχήμα 2.1.1 : Πίνακας με τα top charts παιχνιδιών για iphone

Σχήμα 2.2.1 : Πίνακας με τα top charts παιχνιδιών για ipad

Σχήμα 2.1.1 : Πίνακας με τα top charts παιχνιδιών για android

Σχήμα 2.2 : Πίνακας με τα top 11 παιχνίδια για παιδιά

Σχήμα 2.2.1.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Thinkrolls 2

Σχήμα 2.2.1.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Thinkrolls 2

Σχήμα 2.2.2.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Lola's Alphabet Train

Σχήμα 2.2.2.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Lola's Alphabet Train

Σχήμα 2.2.3.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Kids Preschool Learning Games

Σχήμα 2.2.3.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Kids Preschool Learning Games

Σχήμα 2.2.4.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Life World: Build Stories

Σχήμα 2.2.4.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Life World: Build Stories

Σχήμα 2.2.5.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Bini Super ABC

Σχήμα 2.2.5.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Bini Super ABC

Σχήμα 2.2.6.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Pepi House

Σχήμα 2.2.6.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Pepi House

Σχήμα 2.2.7.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Kitchen 2

Σχήμα 2.2.7.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Kitchen 2

Σχήμα 2.2.8.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Kids Balloon Pop Game

Σχήμα 2.2.8.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Kids Balloon Pop Game

Σχήμα 2.2.9.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Masha and the Bear

Σχήμα 2.2.9.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Masha and the Bear

Σχήμα 2.2.10.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Piano Kids

Σχήμα 2.2.10.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Piano Kids

Σχήμα 2.2.11.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Lab: Elements

Σχήμα 2.2.11.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Lab: Elements

Σχήμα 2.2.12.1 : Πίνακας σύγκρισης παιχνιδιών για παιδιά

Σχήμα 2.2.12.2 : Γραφική παράσταση για παιχνίδια για παιδιά

Σχήμα 2.3.1 : Πίνακας για το δημοφιλέστερο mobile operating system παγκόσμια

Σχήμα 2.3.2 : Γραφική παράσταση για το δημοφιλέστερο mobile operating system παγκόσμια

Σχήμα 2.3.3 : Πίνακας για το δημοφιλέστερο mobile operating system στην Κύπρο

Σχήμα 2.3.4 : Γραφική παράσταση για το δημοφιλέστερο mobile operating system στην Κύπρο.

Σχήμα 2.5.1 : Πίνακας με άλλα παρόμοια συστήματα

Σχήμα 2.5.2.1 : Πίνακας με πλεονεκτήματα και μειονεκτήματα άλλων παρόμοιων συστημάτων.

Σχήμα 2.5.2.2 : Γραφική παράσταση με πλεονεκτήματα και μειονεκτήματα άλλων παρόμοιων συστημάτων.

Σχήμα 2.5.2.3 : Πίνακας για το είδος παιχνιδιών άλλων παρόμοιων συστημάτων

Σχήμα 2.5.2.4 : Γραφική παράσταση για το είδος παιχνιδιών άλλων παρόμοιων συστημάτων

Σχήμα 2.7.1 : Γραφική παράσταση με το είδος χαρακτήρων που προτιμούν τα παιδιά

Σχήμα 2.7.1 : Γραφική παράσταση για το συναίσθημα που προτιμούν τα παιδιά

Σχήμα 2.8.1 : Γραφική παράσταση για τα rewards που προτιμούν τα παιδιά

Σχήμα 3.6.1 : Use case diagram

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή	1
1.2 Εισαγωγή σε παιχνίδια	3
1.3 Εισαγωγή σε παιχνίδια για παιδιά	3
1.4 Στόχος Ατομικής Διπλωματικής Εργασίας	4
1.5 Δομή Ατομικής Διπλωματικής Εργασίας	5

1.1 Γενική Εισαγωγή

Η σωστή διατροφή και η άσκηση παίζουν σημαντικό ρόλο στην υγεία του ανθρώπου. Ένας υγιεινός τρόπος διατροφής είναι η Μεσογειακή διατροφή. Η Μεσογειακή διατροφή προήλθε από τους λαούς που βρίσκονταν γύρω από την Μεσόγειο Θάλασσα και συγκεκριμένα από την Ελλάδα, την Ιταλία, την Ισπανία και τις παραλιακές χώρες της Μέσης Ανατολής. Η συγκεκριμένη διατροφή περιέχει τροφές από όλες τις ομάδες τροφίμων λαμβάνοντας υπόψη την ποσότητα και το πόσο συχνά καταναλώνεται η κάθε τροφή. Έμφαση δίνεται κυρίως σε τροφές όπως είναι τα φρούτα, τα λαχανικά, τα δημητριακά, τα όσπρια και το ελαιόλαδο. Η πυραμίδα της μεσογειακής διατροφής φαίνεται στην Εικόνα 1.1: Πυραμίδα Μεσογειακής Διατροφής. Η πυραμίδα αυτή αναφέρει τα εξής: Σε καθημερινή βάση πρέπει να καταναλώνονται αμυλούχες τροφές όπως είναι το ψωμί, τα δημητριακά και το ρύζι, τα γαλακτοκομικά προϊόντα, τα φρούτα, τα λαχανικά, τα όσπρια και οι ξηροί καρποί, και το ελαιόλαδο. Παράλληλα, πρέπει να γίνεται καθημερινή φυσική άσκηση. Σε εβδομαδιαία βάση πρέπει να καταναλώνονται τα πουλερικά, τα θαλασσινά, τα αυγά και τα γλυκά. Και σε μηνιαία βάση, μπορούμε να

καταναλώνουμε το κόκκινο κρέας. Σημαντική εξίσου είναι και η κατανάλωση νερού. Ποτά και ροφήματα με μεγάλη περιεκτικότητα σε ζάχαρη είναι καλό να αποφεύγονται.



Εικόνα 1.1: Πυραμίδα Μεσογειακής Διατροφής [19].

Οι διατροφικές συνήθειες του ανθρώπου εξαρτώνται από αρκετούς παράγοντες όπως είναι ο τρόπος ζωής, το βάρος, το ύψος και η ηλικία. Η ενημέρωση για την Μεσογειακή διατροφή και γενικότερα την ισορροπημένη διατροφή, πρέπει να γίνεται από μικρή ηλικία όπως είναι η προσχολική ηλικία. Η φυσική ανάπτυξη συνεχίζεται στην προσχολική ηλικία, η οποία αφορά ηλικίες 3 με 5 ετών, ενώ ταυτόχρονα τα παιδιά λόγω του ότι είναι ενεργητικά και παίζουν περισσότερο, έχουν αυξημένες ανάγκες σε ενέργεια και τροφή. Οι γονείς πρέπει να καλλιεργούν στα παιδιά, υγιεινές διατροφικές συνήθειες με σκοπό τα παιδιά να μην έχουν να αντιμετωπίσουν διατροφικές διαταραχές ή παχυσαρκία στο μέλλον. Υπάρχουν πολλοί τρόποι για να το επιτύχουν αυτό και ένας από αυτούς είναι μέσω της τεχνολογίας.

Τα τελευταία χρόνια λόγω της ανάπτυξης της τεχνολογίας, τα παιδιά από μικρή ηλικία μαθαίνουν να χρησιμοποιούν τα κινητά τηλέφωνα και άλλες διάφορες συσκευές για να ψυχαγωγηθούν. Αυτό έχει επιφέρει πολλά πλεονεκτήματα αλλά και μειονεκτήματα. Τα παιδιά επιλέγουν να μένουν σπίτι και να παίζουν με τις ώρες προκαλώντας έτσι προβλήματα υγείας όπως είναι η παχυσαρκία λόγω της καθιστικής ζωής και παράλληλα έχει αρνητικό αντίκτυπο στην γνωστική εξέλιξη και στην συγκέντρωση τους. Αν όμως η τεχνολογία χρησιμοποιηθεί με τον σωστό τρόπο, μπορεί να συμβάλει θετικά στην πληροφόρηση και την εκπαίδευση των παιδιών [19].

1.2 Εισαγωγή σε παιχνίδια

Υπάρχουν διάφορα είδη applications που μπορεί να κατεβάσει κάποιος στο κινητό του τηλέφωνο. Τα applications σχεδιάστηκαν για να εξυπηρετούν πολλούς σκοπούς και να διευκολύνουν τον άνθρωπο. Παράλληλα χρησιμοποιούνται και για την δουλειά αλλά και για την ψυχαγωγία. Για παράδειγμα υπάρχουν applications για ημερολόγια και emails, planners για να ακολουθούμε μια ισορροπημένη διατροφή αλλά και παιχνίδια.

Τα παιχνίδια διαχωρίζονται σε πολλές κατηγορίες, όπως είναι για παράδειγμα τα actions παιχνίδια, τα puzzle και τα online παιχνίδια. Ακόμη μπορεί να υπάρχει όριο ηλικίας ή μπορεί να είναι και κατάλληλο για όλες τις ηλικίες. Με την ανάπτυξη του διαδικτύου, υπάρχει δυνατότητα κάποιος να παίζει online με άλλους χρήστες ή και offline σε single player mode.

Σκοπός των παιχνιδιών, είναι ο κόσμος να ψυχαγωγείται επιλέγοντας αυτός τι επιθυμεί να παίζει ανάλογα με τις προτιμήσεις του και τα ενδιαφέροντα τους.

1.3 Εισαγωγή σε εκπαιδευτικά παιχνίδια για παιδιά

Πλέον, τα παιχνίδια παίζουν σημαντικό ρόλο στην εκπαίδευση και την πληροφόρηση των ανθρώπων. Συγκεκριμένα, μέσω των παιχνιδιών, τα παιδιά αναπτύσσουν τις αναλυτικές τους δεξιότητες, μαθαίνουν να διαχειρίζονται διάφορες καταστάσεις, όπως είναι για παράδειγμα ο πανικός, και γίνονται πιο έξυπνα αφού αποκτούν problem solving

skills, διαχειρίζονται πολλαπλά tasks και ενημερώνονται. Άρα τα παιχνίδια δεν είναι μόνο εργαλεία ψυχαγωγίας και χαλάρωσης, αλλά είναι και εργαλεία εκπαίδευσης [6].

Έχουν δημιουργηθεί πολλά παιχνίδια για παιδιά με σκοπό την εκμάθησή τους για θέματα υγείας. Για παράδειγμα, το 1995 έγινε release για το Super Nintendo Entertainment System, ένα εκπαιδευτικό παιχνίδι με το όνομα Bronkie the Bronchiasaurus. Βλέπε Εικόνα 1.3 : Bronkie the Bronchiasaurus game. Το συγκεκριμένο παιχνίδι ήθελε να διδάξει στα παιδιά για το άσθμα. Ουσιαστικά, το παιχνίδι αποτελείται από δεινόσαυρους οι οποίοι κατασκεύασαν μια μηχανή για να καθαρίσουν τον αέρα. Η μηχανή όμως κλάπηκε και τα κομμάτια της διασκορπίστηκαν στην πόλη. Οι 2 χαρακτήρες του παιχνιδιού πρέπει να βρουν τα εξαρτήματα που λείπουν. Για να το πετύχουν αυτό πρέπει να αποφύγουν κάποια εμπόδια όπως είναι ο καπνός του τσιγάρου. Το παιχνίδι, ακόμη, μαθαίνει στα παιδιά να χρησιμοποιούν σωστά τις συσκευές εισπνοής [7].



Εικόνα 1.3: Bronkie the Bronchiasaurus game [17].

1.4 Στόχος Ατομικής Διπλωματικής Εργασίας

Ο στόχος αυτής της Ατομικής Διπλωματικής Εργασίας είναι η υλοποίηση ενός εκπαιδευτικού παιχνιδιού που αφορά την Μεσογειακή Διατροφή. Το παιχνίδι θα είναι κατάλληλο για ηλικίες 3 μέχρι 5 ετών. Για να υλοποιηθεί τέτοιο παιχνίδι χρειάζεται να εξετάσουμε τι είναι δημοφιλές σε αυτές τις ηλικίες, τι παράγοντες πρέπει να λάβουμε

υπόψη μας, σε τι mobile operating system θα είναι διαθέσιμο το παιχνίδι και τι απαιτήσεις πρέπει να λάβουμε υπόψη μας έτσι ώστε το παιχνίδι να είναι εύκολο στην χρήση, να είναι ενδιαφέρον και παράλληλα να μαθαίνει στα παιδιά για την Μεσογειακή διατροφή με ένα διασκεδαστικό τρόπο.

1.5 Δομή Ατομικής Διπλωματικής Εργασίας

Η δομή της Ατομικής Διπλωματικής Εργασίας αναλύεται στα Κεφάλαια 2 μέχρι 7 ως ακολούθως:

Κεφάλαιο 2:

Το κεφάλαιο 2 είναι η ανασκόπηση. Αρχικά, παρουσιάζονται τα top charts παιχνιδιών στην Κύπρο με σκοπό να δούμε ποια είναι τα δημοφιλέστερα παιχνίδια στην Κύπρο για τις κατηγορίες Top Free, Top Paid και Top Grossing. Έπειτα, μέσω στατιστικών, βλέπουμε ποια είναι τα δημοφιλέστερα mobile operating systems για να καταλήξουμε στο για πιο operating system θα υλοποιηθεί το παιχνίδι έτσι ώστε να έχει περισσότερες λήψεις. Ακόμη, παρουσιάζονται άλλα παρόμοια συστήματα υγιεινής διατροφής, τα πλεονεκτήματα και τα μειονεκτήματα τους και γίνεται σύγκριση. Στο τέλος του κεφαλαίου, γίνεται ανασκόπηση για τα δωρεάν και τα paid παιχνίδια με σκοπό να μάθουμε τι προτιμάει ο κόσμος.

Κεφάλαιο 3:

Στο κεφάλαιο 3 γίνεται αναφορά στο τι τεχνολογίες και τι γνώση χρειάζεται έτσι ώστε να υλοποιηθεί το παιχνίδι.

Κεφάλαιο 4:

Στο κεφάλαιο 4 γίνεται ανάλυση των απαιτήσεων και των προδιαγραφών που χρειάζονται για να σχεδιαστεί ένα παιχνίδι για παιδιά, καθώς και τι αρχές ακολουθεί το παιχνίδι που υλοποιήθηκε στα πλαίσια της Ατομικής Διπλωματικής Εργασίας.

Κεφάλαιο 5:

Στο κεφάλαιο 5 παρουσιάζεται ο τρόπος σχεδίασης του παιχνιδιού στο Unity και η υλοποίηση κώδικα για την λειτουργικότητα του παιχνιδιού.

Κεφάλαιο 6:

Στο κεφάλαιο 6 γίνεται η αξιολόγηση του παιχνιδιού και παρουσιάζονται όλες οι εισηγήσεις που έγιναν για την βελτίωση του.

Κεφάλαιο 7:

Στο κεφάλαιο 7 αναφέρεται η μελλοντική εργασία που μπορεί να γίνει έτσι ώστε τα παιχνίδια να γίνουν καλύτερα. Η μελλοντική εργασία περιέχει όλες τις εισηγήσεις που δεν έχουν υλοποιηθεί ακόμη.

Παράρτημα Α, Β και Γ:

Στα παραρτήματα υπάρχουν οδηγίες εγκατάστασης του παιχνιδιού σε συσκευή Android καθώς, οι οδηγίες χρήσης και οι κώδικες που δημιουργήθηκαν για την λειτουργικότητα του παιχνιδιού.

Κεφάλαιο 2

Ανασκόπηση

2.1 Top Charts παιχνιδιών στην Κύπρο	8
2.2 Δημοφιλέστερα παιχνίδια για παιδιά	9
2.2.1 Thinkrolls 2	10
2.2.2 Lola's Alphabet Train	11
2.2.3 Kids Preschool Learning Games	12
2.2.4 Toca Life World: Build Stories	13
2.2.5 Bini Super ABC	14
2.2.6 Pepi House	14
2.2.7 Toca Kitchen 2	15
2.2.8 Kids Balloon Pop Game	16
2.2.9 Masha and the Bear	17
2.2.10 Piano Kids	17
2.2.11 Toca Lab : Elements	18
2.2.12 Σύγκριση	19
2.3 Δημοφιλέστερα Mobile Operating Systems	20
2.4 Άλλα παρόμοια συστήματα υγιεινής διατροφής	22
2.4.1 Pick Your Plate	22
2.4.2 Food Decisions	23
2.4.3 Sortify: Nutrition	24
2.4.4 Healthier Me	24
2.4.5 Cerino	25
2.5 Παρουσίαση	26
2.5.1 Πλεονεκτήματα και Μειονεκτήματα κάθε παιχνιδιού	26
2.5.2 Σύγκριση	28
2.6 Δωρεάν και Paid Παιχνίδια	30
2.7 Τι χαρακτήρες αρέσουν στα παιδιά	31
2.8 Είδος παιχνιδιών που αρέσουν στα παιδιά	32
2.9 Χρώματα που τραβούν την προσοχή των παιδιών	33

2.1 Top Charts παιχνιδιών στην Κύπρο

Για να γίνει η υλοποίηση του παιχνιδιού, πρώτα πρέπει να εξετάσουμε τι παιχνίδια είναι δημοφιλή στην Κύπρο τον Ιούλιο του 2022. Οι κατηγορίες είναι τα δωρεάν παιχνίδια, τα paid παιχνίδια και τα top grossing. Τα top grossing παιχνίδια είναι αυτά που έχουν το μεγαλύτερο revenue είτε μέσω της τιμής του application, είτε μέσω των in-app purchases [10].

Από τα Σχήματα 2.1.1 : Πίνακας με τα top charts παιχνιδιών για iphone, 2.1.1: Πίνακας με τα top charts παιχνιδιών για ipad και 2.1.3: Πίνακας με τα top charts παιχνιδιών για android, παρατηρούμε πως το δημοφιλέστερο δωρεάν παιχνίδι είναι το Stumble Guys, ενώ το δημοφιλέστερο paid παιχνίδι είναι το Minecraft. Το παιχνίδι Stumble guys είναι ένα online παιχνίδι στο οποίο παίζεις με άλλους χρήστες και σκοπός είναι να φθάσεις στην γραμμή τερματισμού, ξεπερνώντας κάποια εμπόδια. Το παιχνίδι Minecraft είναι ένα open world παιχνίδι στο οποίο μπορείς να δημιουργήσεις το δικό σου σπίτι και να κάνεις mining. Και αυτό το παιχνίδι είναι online, επειδή μπορείς να μπεις σε διάφορους servers και να παίζεις με άλλους χρήστες. Καταλήγουμε στο συμπέρασμα πως οι χρήστες μεγαλύτερης ηλικίας, προτιμούν τα online παιχνίδια για να μπορούν να παίζουν σε πραγματικό χρόνο με άλλους χρήστες ανταγωνιστικά ή ομαδικά.

Iphone:			
Ranking:	Free Games:	Paid:	Top Grossing:
1	Stumble Guys	Minecraft	World Series of Poker-WSOP
2	Subway Surfers	Plague Inc.	Coin Master
3	Drive to Evolve	Papa's Freezeria To Go!	Rise of Empires: Fire and War
4	Western Sniper: Wild West FPS	ΠΑΛΕΡΜΟ	Candy Crush Saga
5	Candy Crush	Cluedo: Hasbro's Mystery Game	Age of Origins: Tower Defense

Σχήμα 2.1.1 : Πίνακας με τα top charts παιχνιδιών για iphone [10].

Ipad:			
Ranking:	Free Games:	Paid:	Top Grossing:
1	Stumble Guys	Plague Inc.	Roblox
2	Like a pizza	Minecraft	Candy Crush Saga
3	Bottle Jump 3D	Poppy Playtime Chapter 1	War Machines: Tanks Battle Game
4	Subway Surfers	Bloons TD 6	Dragon Mania Legends
5	Feeling Arrow	Take It Easy	Words With Friends 2 Word Game

Σχήμα 2.1.2 : Πίνακας με τα top charts παιχνιδιών για ipad [10].

Android:			
Ranking:	Free Games:	Paid:	Top Grossing:
1	Like a Pizza	Minecraft	Coin Master
2	Stumble Guys	Geometry Dash	Gunship Battle Total Warfare
3	Block Box Skyland Sword	Poppy Playtime Chapter 1	Roblox
4	Hair Tattoo: Barbershop Master	Football Manager 2022 Mobile	Candy Crush Saga
5	Calming Crosswords	Grand Theft Auto: San Andreas	Call of Duty Mobile Season 6

Σχήμα 2.1.3 : Πίνακας με τα top charts παιχνιδιών για android [10].

2.2 Δημοφιλέστερα παιχνίδια για παιδιά

Εκτός από τα δημοφιλέστερα παιχνίδια στην Κύπρο, είναι απαραίτητο να εξετάσουμε τι είδος παιχνιδιών αρέσουν στα παιδιά έτσι ώστε να καταλάβουμε τι κατηγορίες παιχνιδιών προτιμούν. Τα 11 πιο δημοφιλέστερα παιχνίδια για παιδιά, φαίνονται στο Σχήμα 2.2 : Πίνακας με τα top 11 παιχνίδια για παιδιά [16]. Για το κάθε παιχνίδι αναλύεται το πως παίζεται το παιχνίδι, τι είδος παιχνιδιού είναι, μερικά πλεονεκτήματα και μειονεκτήματα, αν το παιχνίδι είναι δωρεάν ή αν χρειάζεται να πληρώσουμε για να το εγκαταστήσουμε, καθώς και αν υπάρχουν in-app purchases, για τι ηλικίες είναι κατάλληλο αλλά αναφέρεται και ο αριθμός των downloads.

A/A	Game:	Paid/Free	Ages	Downloads
1	Thinkrolls 2	Paid (\$3,99)	3-9	10 thousands+
2	Lola's Alphabet Train	Free with in-app purchases	3-7	500 thousands+
3	Preschool learning games for kids	Free with in-app purchases	3-6	10 million+
4	Toca Life World	Free with in-app purchases	3-5	100 million+
5	Bini Super ABC	Free with in-app purchases	3	10 million+
6	Pepi House	Free with in-app purchases	3-7	10 million+
7	Toca Kitchen 2	Free	4+	100 million+
8	Kids Balloon Pop Game	Free with in-app purchases	2+	10 million+
9	Masha and the Bear	Free with in-app purchases	Up to 6	100 million+
10	Piano Kids	Free with in-app purchases	Up to 6	100 million+
11	Toca Lab: Elements	Paid (\$3,99)	6-8	500 thousands+

Σχήμα 2.2 : Πίνακας με τα top 11 παιχνίδια για παιδιά [16].

2.2.1 Thinkrolls 2

Σκοπός του παιχνιδιού είναι τα παιδιά να καθοδηγήσουν τους χαρακτήρες μέσα σε λαβύρινθους. Οι λαβύρινθοι έχουν αυξανόμενη δυσκολία. Τα παιδιά θα πρέπει να σκεφτούν πως μπορούν μετακινήσουν κάποια αντικείμενα για να μπορεί ο χαρακτήρας να προχωρήσει. Για παράδειγμα δεν μπορεί να περάσει αν δεν υπάρχει αντικείμενο στο νερό. Το παιχνίδι θέλει να βοηθήσει τα παιδιά να αναπτύξουν λογική σκέψη και να

μπορούν να επιλύουν προβλήματα. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.1.1: Πίνακας με πλεονεκτήματα του παιχνιδιού Thinkrolls 2 και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.1.2: Πίνακας με μειονεκτήματα του παιχνιδιού Thinkrolls 2.

Πλεονεκτήματα:

Ο τρόπος για να παίζεις το παιχνίδι δεν είναι περίπλοκος. Το παιδί απλά μετακινεί τον χαρακτήρα δεξιά και αριστερά ενώ με την βοήθεια της βαρύτητας, όταν υπάρχει κενό ο χαρακτήρας πέφτει. Το παιδί απλά μετακινεί τον χαρακτήρα και πατάει σε κάποια αντικείμενα που μπορούν να μετακινηθούν.
Το περιεχόμενο είναι εικονικό.
Υπάρχουν πολλοί λαβύρινθοι/levels.
Ανάπτυξη λογικής σκέψης και επίλυσης προβλημάτων.
Διαθέσιμο σε Google Play και App Store.

Σχήμα 2.2.1.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Thinkrolls 2.

Μειονεκτήματα:

Στο Google Play το παιχνίδι έχει τιμή €3.99 ενώ στο App Store έχει τιμή €4.99

Σχήμα 2.2.1.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Thinkrolls 2

2.2.2 Lola's Alphabet Train

Σκοπός είναι τα παιδιά να ολοκληρώσουν challenges τα οποία τους μαθαίνει το αλφάβητο. Κάποια challenges ζητούν να επιλέξει κάποιο συγκεκριμένο γράμμα ή τους εμφανίζονται κάποια αντικείμενα/ζώα/φαγητά και τους ζητάει να επιλέξουν αυτό που ξεκινά με κάποιο συγκεκριμένο γράμμα. Στο τέλος κάθε level, μπορείς να δεις το learning process του παιδιού. Το παιχνίδι γίνεται δύσκολο ανάλογα από το learning process. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.2.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Lola's Alphabet Train και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.2.2: Πίνακας με μειονεκτήματα του παιχνιδιού Lola's Alphabet Train.

Πλεονεκτήματα:

Το παιδί απλά πατά στην οθόνη για να επιλέξει αυτό που θέλει.
Το περιεχόμενο είναι εικονικό.
Στο τέλος κάθε level μπορείς να δεις το learning process.
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.
Διαθέσιμο σε Google Play και App Store.

Σχήμα 2.2.2.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Lola's Alphabet Train.

Μειονεκτήματα:

Περιέχει κείμενο στα tasks και το παιδί μπορεί να μην καταλαβαίνει τι πρέπει να κάνει.
Το παιχνίδι στο App Store έχει τιμή \$3.99.

Σχήμα 2.2.2.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Lola's Alphabet Train.

2.2.3 Kids Preschool Learning Games

Το παιχνίδι περιέχει 150+ mini games με σκοπό να μάθει στα παιδιά το αλφάβητο, τους αριθμούς, spelling, τα χρώματα κτλ. Είναι εκπαιδευτικό και παράλληλα διασκεδαστικό για τα παιδιά. Περιέχει audio και τραγούδια για να βοηθήσει τα παιδιά να μαθαίνουν πιο εύκολα και γρήγορα. Το παιδί μπορεί να πατά απλά στην οθόνη για να επιλέξει κάποιο αντικείμενο ή και για να ζωγραφίζει. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.3.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Kids Preschool Learning Games και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.3.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Kids Preschool Learning Games.

Πλεονεκτήματα:

Περιέχει πολύ εκπαιδευτικό περιεχόμενο.
Το περιεχόμενο είναι εικονικό.
Περιέχει πολλά mini games με αποτέλεσμα το παιδί να παίζει κάθε φορά κάτι καινούργιο.
Έχει audio που βοηθά στην καθοδήγηση του παιδιού και τραγούδια.
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.3.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Kids Preschool Learning Games.

Μειονεκτήματα:

Περιέχει adds με αποτέλεσμα το παιδί να πατήσει πάνω και να κατευθυνθεί αλλού.
Το παιχνίδι είναι διαθέσιμο μόνο στο Google play.

Σχήμα 2.2.3.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Kids Preschool Learning Games.

2.2.4 Toca Life World: Build Stories

Σκοπός του παιχνιδιού είναι να ενισχύσει την δημιουργικότητα των παιδιών. Είναι ένα πιο ελεύθερο παιχνίδι που δίνει την δυνατότητα στα παιδιά να δημιουργήσουν τους δικούς τους χαρακτήρες και να τους ντύσουν όπως αυτοί θέλουν. Επίσης, έχουν την δυνατότητα να δημιουργήσουν το δικό τους σπίτι αλλά και ολόκληρες πόλεις. Ακόμη, κάθε εβδομάδα δίνει rewards στους χρήστες του. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.4.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Life World: Build Stories και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.4.2: Πίνακας με μειονεκτήματα του παιχνιδιού Toca Life World: Build Stories.

Πλεονεκτήματα:

Το παιδί μπορεί να φτιάξει τον δικό του χαρακτήρα.
Το περιεχόμενο είναι εικονικό.
Το παιδί μπορεί να κάνει design σπίτια ή ακόμα και να φτιάξει τον δικό του κόσμο.
Κάθε εβδομάδα δίνεται reward
Είναι διαθέσιμο σε Google play και app store
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.4.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Life World: Build Stories.

Μειονεκτήματα:

Δεν είναι εκπαιδευτικό παιχνίδι.

Σχήμα 2.2.4.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Life World: Build Stories.

2.2.5 Bini Super ABC

Σκοπός του παιχνιδιού είναι να μάθει στα παιδιά το αλφάβητο. Περιέχει αρκετά mini games τα οποία είναι εικονικό περιεχόμενο και έχει audio. Για παράδειγμα, πρέπει να κάνει drag and drop το γράμμα Α στο σωστό κουτί και όταν το τοποθετήσει σωστά, ακούγεται το γράμμα Α για να μάθει να το λέει. Το παιχνίδι περιέχει πολλούς χαρακτήρες και ζωάκια με αποτέλεσμα να τραβά την προσοχή του παιδιού. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.5.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Bini Super ABC και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.5.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Bini Super ABC.

Πλεονεκτήματα:

Είναι εκπαιδευτικό.
Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση.
Περιέχει audio για την εύκολη εκμάθηση του παιδιού.
Είναι διαθέσιμο σε Google play και app store.
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.5.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Bini Super ABC.

Μειονεκτήματα:

Περιέχει διαφημίσεις.

Σχήμα 2.2.5.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Bini Super ABC.

2.2.6 Pepi House

Σκοπός του παιχνιδιού είναι να ενισχύσει την δημιουργικότητα των παιδιών. Το παιδί βλέπει ένα σπίτι με αρκετά δωμάτια και αλληλεπιδρά με τους χαρακτήρες. Για παράδειγμα, μπορεί να κάνει drag and drop φαγητό σε ένα χαρακτήρα για να το φάει. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.6.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Pepi House και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.6.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Pepi House.

Πλεονεκτήματα:

Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση
Είναι διαθέσιμο σε Google play και app store
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.6.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Pepi House.

Μειονεκτήματα:

Δεν είναι εκπαιδευτικό.
Δεν υπάρχει κάποιο reward.
Περιέχει διαφημίσεις.

Σχήμα 2.2.6.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Pepi House.

2.2.7 Toca Kitchen 2

Σκοπός του παιχνιδιού είναι να ενισχύσει την δημιουργικότητα των παιδιών. Το παιδί έχει τον ρόλο του σεφ. Μπορεί να φτιάξει οτιδήποτε θέλει δοκιμάζοντας μόνο του όποιο συνδυασμό φαγητών θέλει, δηλαδή να φτιάχνει τις δικές του συνταγές. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.7.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Kitchen 2 και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.7.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Kitchen 2.

Πλεονεκτήματα:

Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση (touch/drag and drop).
Είναι διαθέσιμο σε Google play και app store.
Το παιχνίδι είναι δωρεάν.

Σχήμα 2.2.7.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Kitchen 2.

Μειονεκτήματα:

Δεν είναι εκπαιδευτικό.
Δεν υπάρχει κάποιο reward.
Περιέχει διαφημίσεις.

Σχήμα 2.2.7.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Kitchen 2.

2.2.8 Kids Balloon Pop Game

Σκοπός του παιχνιδιού είναι να μάθει κυρίως αγγλικά στα παιδιά και να κάνει practice το vocabulary σε 10 διαφορετικές γλώσσες. Έχει 5 διαφορετικά παιχνίδια για κάθε γλώσσα. Μπορεί να σπάζει απλά μπαλόνια, μπαλόνια που περιέχουν γράμματα και να ακούν πως προφέρεται, μπαλόνια με αριθμούς, χρώματα και σχήματα. Το παιχνίδι περιέχει οπτικοακουστικό περιεχόμενο. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.8.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Kids Balloon Pop Game και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.8.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Kids Balloon Pop Game.

Πλεονεκτήματα:

Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση.
Περιέχει audio για τα γράμματα, χρώματα, σχήματα κτλ.
Είναι διαθέσιμο σε Google play και app store.
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.8.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Kids Balloon Pop Game.

Μειονεκτήματα:

Δεν υπάρχει κάποιο reward.
Περιέχει διαφημίσεις.

Σχήμα 2.2.8.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Kids Balloon Pop Game.

2.2.9 Masha and the Bear

Το παιχνίδι περιέχει 30 εκπαιδευτικά παιχνίδια και στοχεύει σε ηλικίες 6 χρονών. Το παιχνίδι δίνει την δυνατότητα στα παιδιά να ζωγραφίζουν, να μαθαίνουν λέξεις σε ξένες γλώσσες, να απομνημονεύουν αντικείμενα, να λύνουν puzzle, να παίζουν μουσικά όργανα και να μαθαίνουν αριθμούς από το 1 μέχρι το 10 και απλές μαθηματικές πράξεις. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.9.1: Πίνακας με πλεονεκτήματα του παιχνιδιού Masha and the Bear και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.9.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Masha and the Bear.

Πλεονεκτήματα:

Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση.
Είναι διαθέσιμο σε Google play και app store.
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.9.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Masha and the Bear.

Μειονεκτήματα:

Δεν υπάρχει κάποιο reward.
Περιέχει διαφημίσεις.

Σχήμα 2.2.9.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Masha and the Bear.

2.2.10 Piano Kids

Σκοπός του παιχνιδιού είναι να μάθει στα παιδιά τα μουσικά όργανα. Ουσιαστικά το παιδί μαθαίνει μουσική μέσω του παιχνιδιού για να βελτιώσει τις δεξιότητες του. Βοηθά στην ανάπτυξη της μνήμης, της συγκέντρωσης και της δημιουργικότητας. Όλη η οικογένεια μπορεί να συνθέτει τραγούδια μαζί. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.10.1: Πίνακας με πλεονεκτήματα του παιχνιδιού Piano Kids και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.10.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Piano Kids.

Πλεονεκτήματα:

Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση.
Περιέχει γνωστά τραγούδια τα οποία μπορούν να μάθουν.
Μαθαίνει τις νότες στα παιδιά.
Το παιχνίδι είναι δωρεάν και έχει in-app purchases στο Google Play.

Σχήμα 2.2.10.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Piano Kids.

Μειονεκτήματα:

Δεν υπάρχει κάποιο reward.
Δεν είναι διαθέσιμο σε app store.
Περιέχει διαφημίσεις.

Σχήμα 2.2.10.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Piano Kids.

2.2.11 Toca Lab: Elements

Σκοπός του παιχνιδιού είναι να μάθει στα παιδιά τον περιοδικό πίνακα. Τους δίνει τη δυνατότητα να κάνουν πειράματα κάνοντας drag and drop τα elements. Μπορούν να δουν την αλληλεπίδραση μεταξύ των στοιχείων. Μερικά στοιχεία είναι σε μορφή χαρακτήρων και μπορούν να κινούνται στον χώρο με σκοπό η εκμάθηση να είναι διασκεδαστική. Τα πλεονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.11.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Lab: Elements και τα μειονεκτήματα του παιχνιδιού φαίνονται στο Σχήμα 2.2.11.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Lab: Elements.

Πλεονεκτήματα:

Το περιεχόμενο είναι εικονικό.
Απλό στην χρήση
Υπάρχει αλληλεπίδραση των στοιχείων για να καταλαβαίνει το παιδί το αποτέλεσμα
Μαθαίνει στα παιδιά τον περιοδικό πίνακα
Είναι διαθέσιμο σε google play και app store

Σχήμα 2.2.11.1 : Πίνακας με πλεονεκτήματα του παιχνιδιού Toca Lab: Elements.

Μειονεκτήματα:

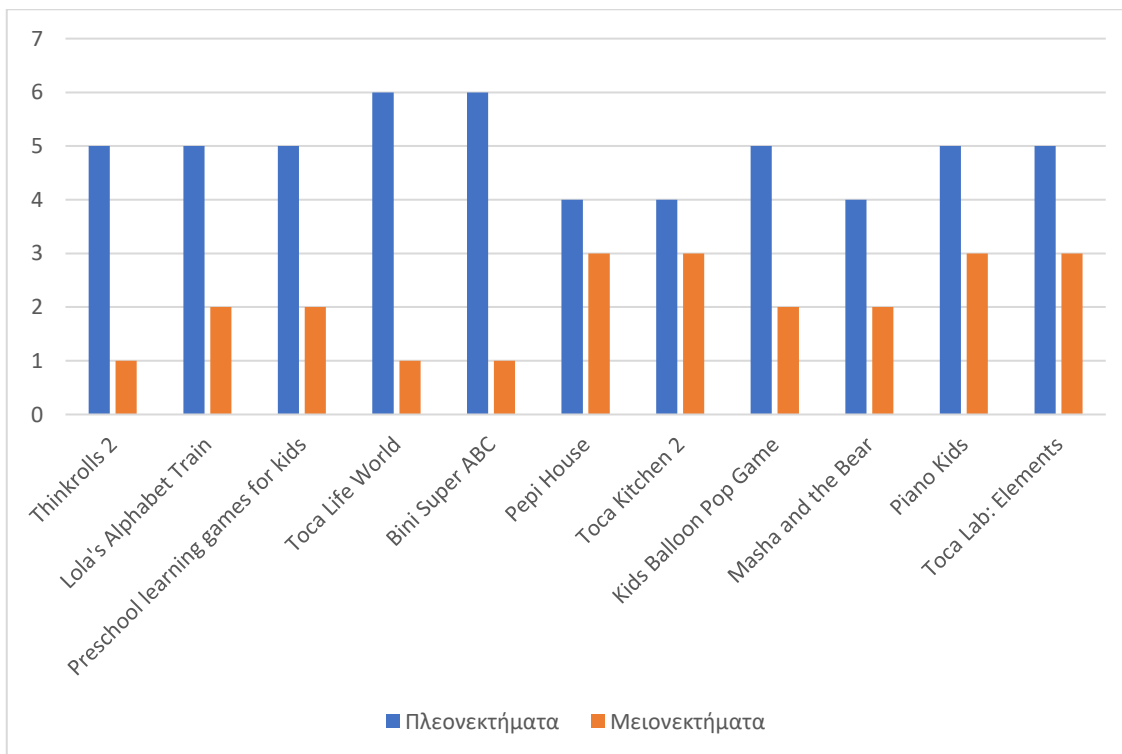
Δεν υπάρχει κάποιο reward.
Είναι paid app.
Περιέχει διαφημίσεις.

Σχήμα 2.2.11.2 : Πίνακας με μειονεκτήματα του παιχνιδιού Toca Lab: Elements.

2.2.12 Σύγκριση

Όνομα παιχνιδιού	Πλεονεκτήματα	Μειονεκτήματα
Thinkrolls 2	5	1
Lola's Alphabet Train	5	2
Preschool learning games for kids	5	2
Toca Life World	6	1
Bini Super ABC	6	1
Pepi House	4	3
Toca Kitchen 2	4	3
Kids Balloon Pop Game	5	2
Masha and the Bear	4	2
Piano Kids	5	3
Toca Lab: Elements	5	3

Σχήμα 2.2.12.1 : Πίνακας σύγκρισης παιχνιδιών για παιδιά.



Σχήμα 2.2.12.2 : Γραφική παράσταση για παιχνίδια για παιδιά

Από το Σχήμα 2.2.12.1 : Πίνακας σύγκρισης παιχνιδιών για παιδιά και το Σχήμα 2.2.12.2: Γραφική παράσταση για παιχνίδια για παιδιά, παρατηρούμε πως τα περισσότερα πλεονεκτήματα τα έχει το εκπαιδευτικό παιχνίδι Bini Super ABC αλλά και το παιχνίδι Toca Life World που αν και δεν είναι εκπαιδευτικό ενισχύει την δημιουργικότητα των παιδιών, ενώ τα λιγότερα μειονεκτήματα τα έχει το παιχνίδι Thinkrolls 2, το Toca Life World και το Bini Super ABC.

Τα 4 παιχνίδια με τα περισσότερα μειονεκτήματα είναι τα εξής: Pepi House, Toca Kitchen 2, Piano Kids, Toca Lab: Elements. Οι κύριοι λόγοι που αυτά τα παιχνίδια έχουν τα περισσότερα μειονεκτήματα είναι λόγω του ότι περιέχουν διαφημίσεις και επειδή δεν υπάρχει κάποιο reward.

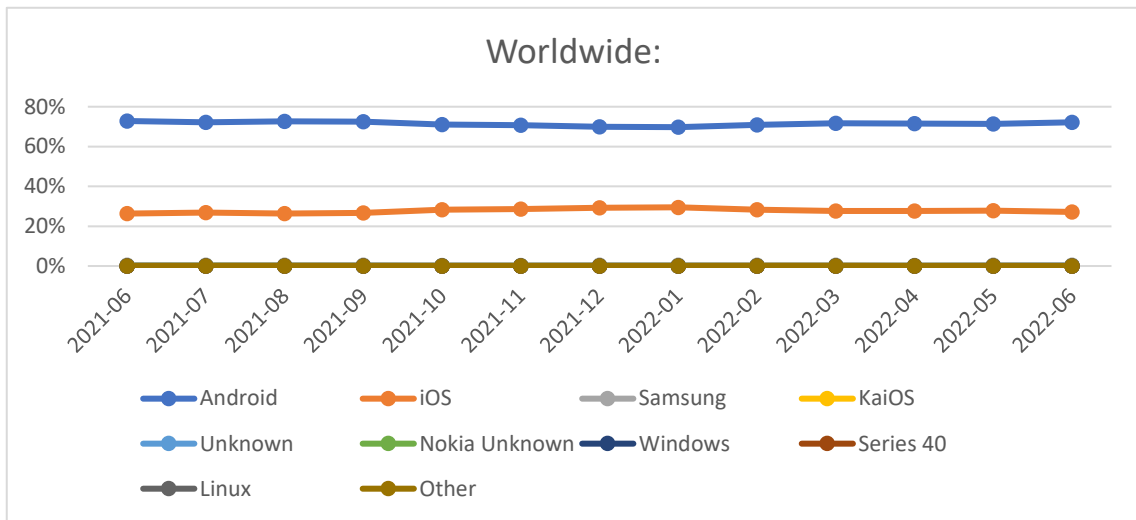
2.3 Δημοφιλέστερο Mobile Operating System

Worldwide										
Date	Android	iOS	Samsung	KaiOS	Unknown	Nokia Unknown	Windows	Series 40	Linux	Other
2021-06	72.83%	26.35%	0.41%	0.18%	0.14%	0.03%	0.02%	0.02%	0.02%	0.02%
2021-07	72.21%	26.92%	0.43%	0.19%	0.14%	0.03%	0.02%	0.01%	0.02%	0.02%
2021-08	72.74%	26.42%	0.42%	0.19%	0.14%	0.02%	0.02%	0.01%	0.02%	0.02%
2021-09	72.44%	26.75%	0.41%	0.17%	0.14%	0.02%	0.02%	0.01%	0.01%	0.02%
2021-10	71.09%	28.21%	0.38%	0.11%	0.13%	0.02%	0.01%	0.01%	0.01%	0.02%
2021-11	70.74%	28.54%	0.38%	0.14%	0.13%	0.02%	0.01%	0.01%	0.01%	0.02%
2021-12	70.01%	29.24%	0.43%	0.13%	0.12%	0.02%	0.01%	0.01%	0.01%	0.02%
2022-01	69.74%	29.49%	0.45%	0.14%	0.11%	0.02%	0.01%	0.01%	0.01%	0.02%
2022-02	70.94%	28.29%	0.43%	0.17%	0.10%	0.01%	0.01%	0.01%	0.01%	0.02%
2022-03	71.70%	27.57%	0.42%	0.14%	0.10%	0.01%	0.01%	0.01%	0.01%	0.02%
2022-04	71.59%	27.68%	0.39%	0.12%	0.15%	0.02%	0.01%	0.01%	0.01%	0.01%
2022-05	71.45%	27.83%	0.41%	0.12%	0.12%	0.01%	0.01%	0.01%	0.01%	0.01%
2022-06	72.12%	27.22%	0.42%	0.08%	0.11%	0.01%	0.01%	0.01%	0.01%	0.01%

Σχήμα 2.3.1 : Πίνακας για το δημοφιλέστερο mobile operating system παγκόσμια [13].

Από το Σχήμα 2.3.1 : Πίνακας για το δημοφιλέστερο mobile operating system παγκόσμια, παρατηρούμε πως τα δύο δημοφιλέστερα mobile operating systems είναι το Android και το iOS της Apple. Από τον Ιούνιο του 2021 μέχρι τον Ιούνιο του 2022, παρατηρήθηκε πως το mobile operating system που επιλέγει ο περισσότερος κόσμος παγκοσμίως είναι το Android με ποσοστό μεγαλύτερο από 70%.

Από το Σχήμα 2.3.2 : Γραφική παράσταση για το δημοφιλέστερο mobile operating system παγκόσμια, μπορούμε να δούμε πως υπάρχει μεγάλη διαφορά μεταξύ των operating systems, αφού το iOS βρίσκεται στην δεύτερη θέση με ποσοστό κοντά στο 20% [13].

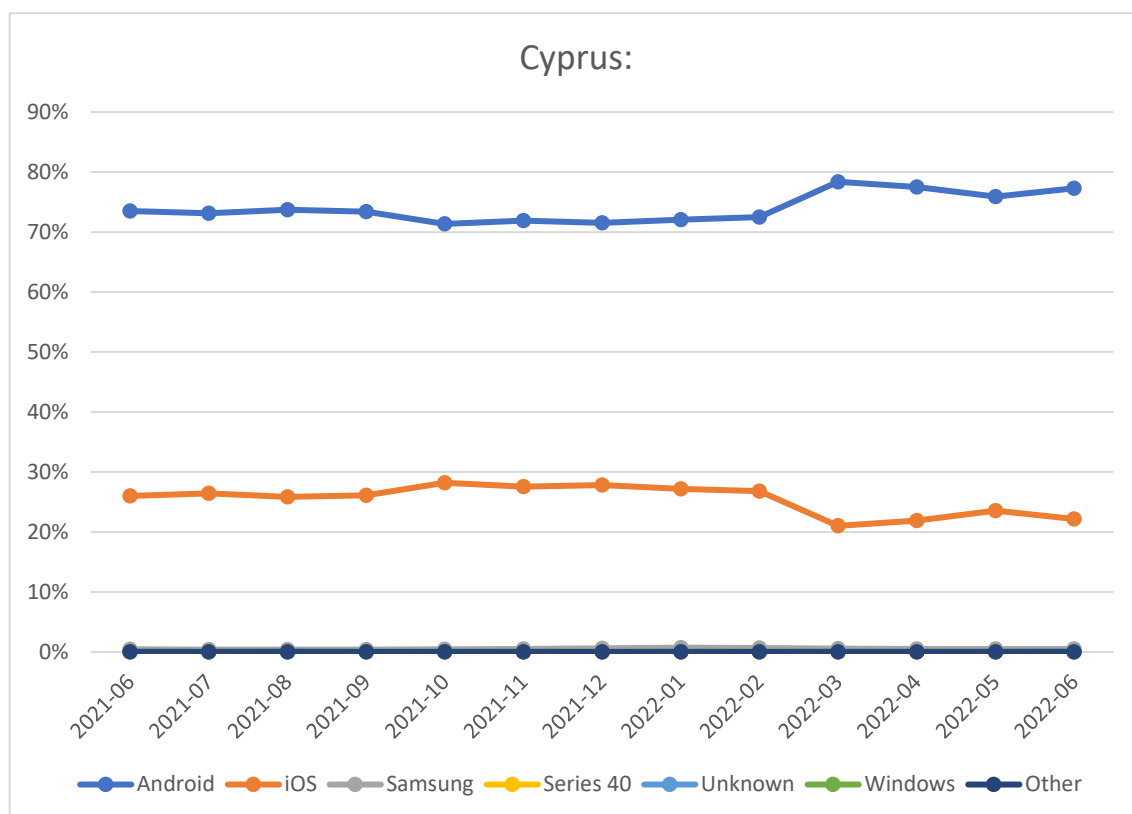


Σχήμα 2.3.2 : Γραφική παράσταση για το δημοφιλέστερο mobile operating system παγκόσμια [13].

Όσον αφορά την Κύπρο, το Android συνεχίζει να έχει την πρώτη θέση και στην δεύτερη βρίσκεται το iOS. Τα αποτελέσματα βρίσκονται στο Σχήμα 2.3.3 : Πίνακας για το δημοφιλέστερο mobile operating system στην Κύπρο.

Cyprus:							
Date	Android	iOS	Samsung	Series 40	Unknown	Windows	Other
2021-06	73.50%	25.98%	0.44%	0.02%	0.02%	0.01%	0.02%
2021-07	73.12%	26.45%	0.38%	0.02%	0.02%	0.01%	0.01%
2021-08	73.72%	25.82%	0.40%	0.02%	0.01%	0.01%	0.03%
2021-09	73.42%	26.12%	0.41%	0.02%	0.01%	0.01%	0.01%
2021-10	71.35%	28.20%	0.42%	0.01%	0.01%	0.01%	0.01%
2021-11	71.90%	27.56%	0.49%	0.01%	0.01%	0.02%	0.01%
2021-12	71.51%	27.82%	0.59%	0.03%	0.02%	0.01%	0.01%
2022-01	72.05%	27.16%	0.71%	0.02%	0.02%	0.01%	0.02%
2022-02	72.47%	26.80%	0.66%	0.03%	0.02%	0.01%	0.01%
2022-03	78.38%	21.02%	0.54%	0.02%	0.03%	0.01%	0.01%
2022-04	77.50%	21.92%	0.51%	0.03%	0.03%	0.01%	0.01%
2022-05	75.92%	23.53%	0.47%	0.02%	0.03%	0.01%	0.01%
2022-06	77.29%	22.16%	0.47%	0.02%	0.02%	0.01%	0.03%

Σχήμα 2.3.3 : Πίνακας για το δημοφιλέστερο mobile operating system στην Κύπρο [14].



Σχήμα 2.3.4 : Γραφική παράσταση για το δημοφιλέστερο mobile operating system στην Κύπρο [14].

Από το Σχήμα 2.3.4 : Γραφική παράσταση για το δημοφιλέστερο mobile operating system στην Κύπρο, μπορούμε να συμπεράνουμε πως το κατάλληλο mobile operating system για να κάνουμε release κάποιο application είναι το Android, αφού παγκόσμια και Παγκύπρια έχει το μεγαλύτερο ποσοστό αγοράς [14].

2.4 Άλλα παρόμοια συστήματα υγιεινής διατροφής

Για να μπορέσουμε να υλοποιήσουμε ένα παιχνίδι, πρώτα πρέπει να λάβουμε υπόψη μας παρόμοια συστήματα και να τα αξιολογήσουμε.

2.4.1 Pick Your Plate

Το συγκεκριμένο παιχνίδι δίνει την δυνατότητα στον χρήστη να ταξιδέψει σε όλο το κόσμο για μάθει πως να δημιουργεί υγιεινά γεύματα. Αρχικά ο χρήστης επιλέγει μια από τις διαθέσιμες χώρες που υπάρχουν στον χάρτη. Αφού επιλέξει μια χώρα, θα εμφανιστούν

διάφορα τρόφιμα από τα οποία ο χρήστης μπορεί να επιλέξει ότι θέλει χωρίς όμως να ξεπεράσει το budget του. Για κάθε επιλογή, βλέπουμε στο σχήμα των guidelines της συγκεκριμένης χώρας, να ανεβαίνει το score. Η κάθε χώρα έχει διαφορετικό φαγητό. Πριν ξεκινήσει ο χρήστης να επιλέγει τρόφιμα για να δημιουργήσει πρωινό, μεσημεριανό και βραδινό, το παιχνίδι εμφανίζει μερικές πληροφορίες όσον αφορά τα guidelines της χώρας που έχει επιλεγεί. Στο τέλος, ανάλογα με το πόσο ακολουθήθηκαν τα guidelines δίνεται και το ανάλογο μετάλλιο.

Το παιχνίδι είναι διαθέσιμο σε Google Play και σε App Store. Και στα δύο stores είναι δωρεάν, δεν διαθέτει In-App Purchases και είναι κατάλληλο για ηλικίες 9 μέχρι 11 ετών.

2.4.2 Food Decisions

Μέσω του παιχνιδιού, ο χρήστης μπορεί να επιλέξει υγιεινά τρόφιμα από εστιατόρια, supermarket, την κουζίνα του και από εργαστήριο. Στο πάνω αριστερό μέρος της οθόνης, εμφανίζονται τα tasks που πρέπει να κάνει ο player. Όταν επιλέγει κάποιο τρόφιμο, του εμφανίζονται μερικές πληροφορίες όπως για παράδειγμα πόσα calories είναι. Το παιχνίδι είναι Online έτσι ώστε να μπορεί ο χρήστης να παίζει με τους φίλους του τα διάφορα Mini Games που προσφέρει το παιχνίδι. Ένα από τα Mini Games ονομάζεται Food Olympics. Το συγκεκριμένο mini game, εμφανίζει στην αρχή τι πρέπει να βρει ο παίκτης. Για παράδειγμα, αν η κατηγορία που εμφανίζεται είναι τα γαλακτοκομικά, ο παίκτης πρέπει να τρέξει σε κάποιο τρόφιμο που να ανήκει στην κατηγορία αυτή.

Είναι ένα Open World παιχνίδι που επιτρέπει στους χρήστες να επιλέγουν ότι φαγητό θέλουν και να κινούνται στον χάρτη ελεύθερα. Στο τέλος κάθε ημέρας, το παιχνίδι εμφανίζει ένα health feedback ανάλογα με τις επιλογές του χρήστη.

Το παιχνίδι είναι διαθέσιμο στο Steam. Πρώτα πρέπει να γίνει εγκατάσταση του Steam στον υπολογιστή και έπειτα η εγκατάσταση του παιχνιδιού. Είναι διαθέσιμο και στο Google Play του Android όμως δεν είναι multiplayer, είναι single player και είναι μόνο για practice. Ακόμη, είναι δωρεάν χωρίς in-app purchases.

2.4.3 Sortify: Nutrition

Σκοπός αυτού του παιχνιδιού είναι ο παίκτης να ταξινομήσει τα διάφορα τρόφιμα που έχει στην διάθεση του. Επιλέγει αυτός τις ετικέτες και τις τοποθετεί στα κουτιά. Ανάλογα με την ετικέτα, πρέπει να τοποθετηθούν στο κουτί και τα ανάλογα τρόφιμα. Οι πόντοι προσθέτονται και αφαιρούνται με βάση τις επιλογές του χρήστη. Στο τέλος εμφανίζεται ένα review με τις επιλογές του χρήστη και πόσους πόντους έχασε ή κέρδισε. Σκοπός είναι να μαζέψει όσο το δυνατόν περισσότερους πόντους.

Το παιχνίδι είναι απλό στην χρήση λόγω του ότι είναι drag and drop με την χρήση του mouse. Επίσης, είναι online, δηλαδή χρειάζεται σύνδεση με το internet, και είναι κατάλληλο για όλες τις ηλικίες.

2.4.4 Healthier Me

Το συγκεκριμένο παιχνίδι θέλει να μάθει στα παιδιά να παίρνουν σωστές αποφάσεις όσον αφορά την σωστή διατροφή και την φυσική κατάσταση. Αποτελείται από δύο lessons. Στο πρώτο lesson ο μαθητής πρέπει να βοηθήσει τον Chester επιλέγοντας 3 ισορροπημένα και υγιεινά γεύματα. Με αυτό τον τρόπο ο μαθητής μαθαίνει τις διάφορες ομάδες τροφίμων, όπως είναι για παράδειγμα τα φρούτα και τα λαχανικά, φτιάχνοντας γεύματα τα οποία ακολουθούν τα food guidelines των ΗΠΑ. Ανάλογα με το τι επέλεξε από τα διαθέσιμα τρόφιμα, το score αυξάνεται ή μειώνεται σε κάθε κατηγορία.

Στο δεύτερο lesson, ο μαθητής πρέπει να βοηθήσει τον Chester να προετοιμαστεί για ένα παιχνίδι που θα παίξει με τους φίλους του το οποίο ονομάζεται “beastball”. Ουσιαστικά θα τον βοηθήσει να δυναμώσει, να αποκτήσει αντοχή και ευελιξία για να νικήσει. Στο κάτω μέρος του παιχνιδιού, βλέπουμε πως υπάρχουν 3 μπάρες που δείχνουν το score της δύναμης, της αντοχής και της ευελιξίας.

Το παιχνίδι είναι διαθέσιμο σε PC και είναι δωρεάν μόνο σε K-12 teachers/students. Οι ηλικίες που στοχεύει είναι από 6 ετών και άνω, δηλαδή για μαθητές δημοτικού.

2.4.5 Cerino – Put it in the basket

Το παιχνίδι αυτό δεν είναι ακόμη διαθέσιμο, όμως είναι καλό να αναφερθεί γιατί αφορά την Μεσογειακή διατροφή. Αρχικά, το παιχνίδι περιέχει αρκετά mini-games έτσι ώστε τα παιδιά να έχουν εύρος παιχνιδιών για να υπάρχει περισσότερο ενδιαφέρον.

Το πρώτο mini-game είναι το put it in the basket. Ουσιαστικά, το παιδί πρέπει να κινεί τον χαρακτήρα δεξιά και αριστερά έτσι ώστε ο χαρακτήρας να βάζει στο καλάθι τα σωστά τρόφιμα. Υπάρχουν διαφορετικά levels για κάθε κατηγορία της πυραμίδας. Για παράδειγμα, στο πρώτο level, μαζεύονται τρόφιμα καθημερινής κατανάλωσης όπως είναι τα φρούτα και τα λαχανικά.

Λόγω του ότι το παιχνίδι απευθύνεται σε παιδιά προσχολικής ηλικίας, στα πρώτα levels τα τρόφιμα που πρέπει να μπουν στο καλάθι φωσφορίζουν με σκοπό να βοηθήσουν τα παιδιά. Έπειτα, όταν το παιδί ξαναπαίζει, το τρόφιμα δεν θα φωσφορίζουν έτσι ώστε το παιδί να καταλάβει από μόνο του ποια τρόφιμα πρέπει να μαζέψει. Υπάρχει βοηθητικός πίνακας που δείχνει εικονικά ποια τρόφιμα πρέπει να μαζευτούν. Σε περίπτωση που προστεθεί κάτι άλλο στο καλάθι, χάνονται points από το score.

2.5 Παρουσίαση

A/A	Παιχνίδι:	Paid/Free:	Ηλικία:	Downloads:	PC/Android/iOS:
1	Pick Your Plate	Free	9 - 11	1 χιλ+ (Google Play)	Android/iOS
2	Food Decisions	Free	-	-	PC/Android
3	Sortify: Nutrition	Free	Για όλες τις ηλικίες	-	Online
4	Healthier Me	Free only for K-12 Teachers/Students	6+	-	PC
5	Cerino	Free with in-app purchases	3 - 5	Ακόμη δεν είναι διαθέσιμο	Android

Σχήμα 2.5.1 : Πίνακας με άλλα παρόμοια συστήματα.

2.5.1 Πλεονεκτήματα και Μειονεκτήματα κάθε παιχνιδιού

Pick Your Plate:

Πλεονεκτήματα:

1. Μπορεί να είναι κατάλληλο για ηλικίες 3-5 χρονών γιατί απλά θα πατάει στην οθόνη και θα βλέπει ότι γεμίζει, για παράδειγμα η πυραμίδα - Απλό στην χρήση.
2. Υπάρχει reward με μετάλλια για την κάθε χώρα.
3. Το παιχνίδι χρησιμοποιεί εικονικό περιεχόμενο. Ένα παιδί ηλικίας 4-5 χρονών δεν ξέρει να διαβάζει.
4. Τα παιδιά μπορούν να μάθουν τα food guidelines διάφορων χωρών.
5. Διαθέσιμο σε Android και σε iOS.

Μειονεκτήματα:

1. Αν και για κάθε χώρα υπάρχουν πληροφορίες, το παιδί δεν θα μπορεί να τις διαβάσει.

2. Για κάθε γεύμα, υπάρχει ένα budget, όμως το παιδί αν και μπορεί να ξέρει λίγους αριθμούς, δεν ξέρει αριθμητική (πρόσθεση/αφαίρεση).

Food Decisions:

Πλεονεκτήματα:

1. Επειδή είναι 3D είναι πιο ρεαλιστικό.
2. Υπάρχουν tasks που πρέπει να κάνει το παιδί με σκοπό να μάθει πιο εύκολα για την υγιεινή διατροφή.
3. Διαθέσιμο σε Android και σε PC.

Μειονεκτήματα:

1. Το παιδί δεν ξέρει να διαβάζει άρα δεν θα καταλαβαίνει ούτε τι γράφουν τα tasks, ούτε τι αναγράφεται για κάθε τρόφιμο (για παράδειγμα τα calories).
2. Επειδή είναι 3D ίσως να είναι περίπλοκη η χρήση του πληκτρολογίου, για παράδειγμα θα πρέπει να χρησιμοποιεί τα πλήκτρα WASD για να κινείται και ταυτόχρονα να χρησιμοποιεί το mouse για να βλέπει γύρω του.

Sortify Nutrition:

Πλεονεκτήματα:

1. Εύκολο στην χρήση.
2. Το περιεχόμενο είναι εικονικό και χρησιμοποιεί ελάχιστο κείμενο.

Μειονεκτήματα:

1. Στο τέλος του παιχνιδιού δείχνει μόνο ποια τρόφιμα έβαλε στα σωστά στα κουτιά και ποια τρόφιμα έβαλε στα λάθος κουτιά, δεν δείχνει στο παιδί σε ποια κουτιά έπρεπε να τοποθετήσει τα τρόφιμα που έκανε λάθος.
2. Απλά μαθαίνει στα παιδιά σε ποια κατηγορία ανήκει το κάθε τρόφιμο.
3. Το παιχνίδι είναι διαθέσιμο μόνο online.

Healthier Me:

Πλεονεκτήματα:

1. Υπάρχει score. Τα παιδιά μπορούν να ξαναπαίξουν για να ξεπεράσουν το προσωπικό τους score.
2. Περιέχει κυρίως εικονικό περιεχόμενο αντί κείμενο.
3. Το παιχνίδι αφορά όχι μόνο την διατροφή αλλά και την σωματική άσκηση.

Μειονεκτήματα:

1. Access μόνο σε k12 teachers και students.
2. Διαθέσιμο μόνο σε υπολογιστή.

Cerino:

Πλεονεκτήματα:

1. Είναι εκπαιδευτικό παιχνίδι.
2. Στα πρώτα levels, τα τρόφιμα που πρέπει να μπουν στο καλάθι, φωτίζονται για να βοηθήσουν το παιδί.
3. Περιέχει οπτικοακουστικό περιεχόμενο.
4. Υπάρχει score.

Μειονεκτήματα:

1. Στο main menu υπάρχουν 2 κουμπιά, το play και το exit, τα οποία έχουν την ίδια μορφή. Τα παιδιά όμως δεν ξέρουν να διαβάζουν άρα υπάρχει περίπτωση να θέλουν να παίξουν και να πατήσουν σε λανθασμένο κουμπί.
2. Το ακουστικό περιεχόμενο είναι αρκετό με αποτέλεσμα τα παιδιά να αποφεύγουν να το ακούσουν και να πατούν skip.

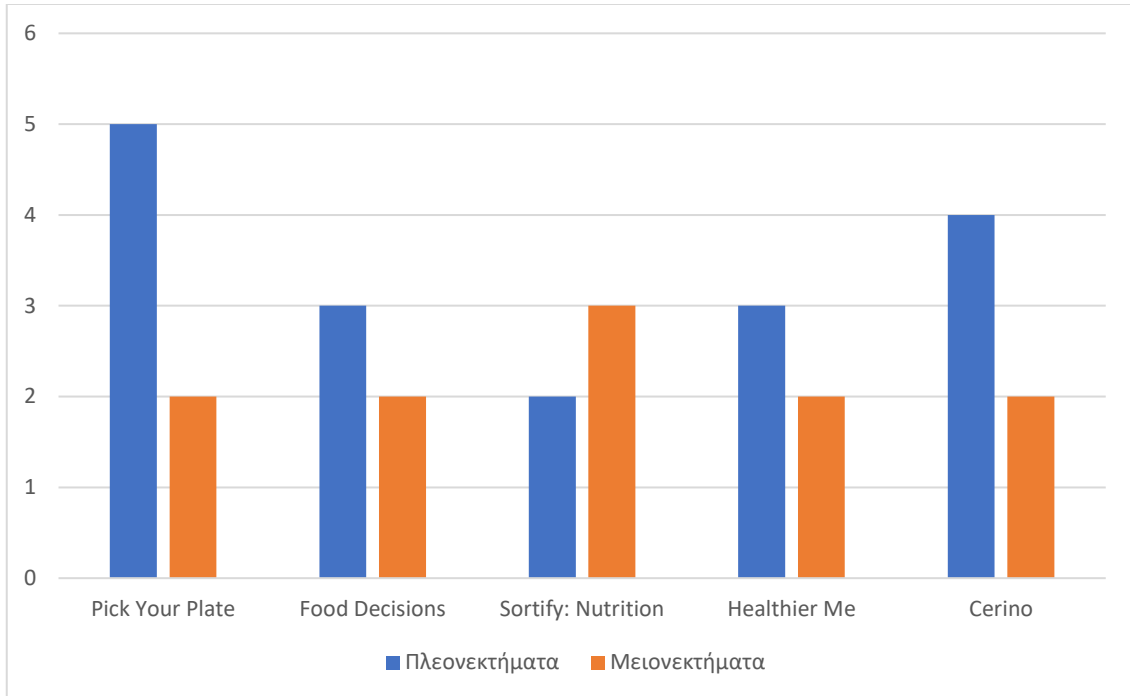
2.5.2 Σύγκριση

Στο Σχήμα 2.5.2.1 : Πίνακας με πλεονεκτήματα και μειονεκτήματα άλλων παρόμοιων συστημάτων, βλέπουμε πόσα πλεονεκτήματα και πόσα μειονεκτήματα έχουν αναφερθεί στο Υποκεφάλαιο 2.5.1 Πλεονεκτήματα και Μειονεκτήματα κάθε παιχνιδιού.

Όνομα Παιχνιδιού	Πλεονεκτήματα	Μειονεκτήματα
Pick Your Plate	5	2
Food Decisions	3	2
Sortify: Nutrition	2	3
Healthier Me	3	2
Cerino	4	2

Σχήμα 2.5.2.1 : Πίνακας με πλεονεκτήματα και μειονεκτήματα άλλων παρόμοιων συστημάτων.

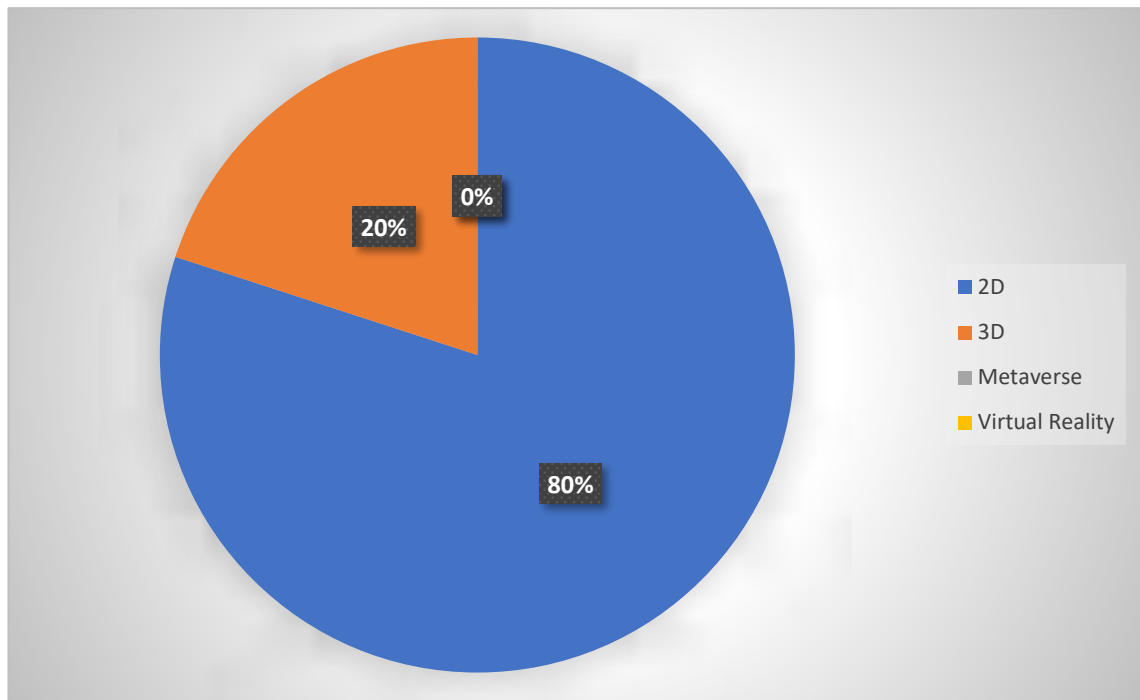
Από το Σχήμα 2.5.2.3 : Πίνακας για το είδος παιχνιδιών άλλων παρόμοιων συστημάτων και το Σχήμα 2.5.2.4 : Γραφική παράσταση για το είδος παιχνιδιών άλλων παρόμοιων συστημάτων, παρατηρούμε πως τα περισσότερα παιχνίδια είναι 2D και μόνο το παιχνίδι Food Decisions είναι 3D.



Σχήμα 2.5.2.2 : Γραφική παράσταση με πλεονεκτήματα και μειονεκτήματα άλλων παρόμοιων συστημάτων.

2D	3D	Metaverse	Virtual Reality
Pick Your Plate	Food Decisions		
Sortify: Nutrition			
Healthier Me			
Cerino			
4	1	0	0

Σχήμα 2.5.2.3 : Πίνακας για το είδος παιχνιδιών άλλων παρόμοιων συστημάτων.



Σχήμα 2.5.2.4 : Γραφική παράσταση για το είδος παιχνιδιών άλλων παρόμοιων συστημάτων.

2.6 Δωρεάν και Paid Παιχνίδια

Τα περισσότερα παιχνίδια που είναι διαθέσιμα στις διάφορες πλατφόρμες, όπως το Google Play του Android, είναι δωρεάν. Τα δωρεάν παιχνίδια, είναι δωρεάν για να παίξεις άλλα υπάρχει η δυνατότητα οι χρήστες να κάνουν in-app purchases είτε για να αγοράσουν χαρακτήρες, να ξεκλειδώσουν levels και άλλα πολλά. Υπάρχουν 2 κατηγορίες χρηστών, αυτοί που κάνουν in-app purchases, οι οποίοι ονομάζονται premium players, και οι χρήστες που δεν ξοδεύουν καθόλου χρήματα και ονομάζονται non-spending players. Το θετικό ενός δωρεάν παιχνιδιού, είναι πως δίνει την δυνατότητα στον χρήστη πρώτα να παίξει το παιχνίδι και μετέπειτα εάν θέλει μπορεί να κάνει in-app purchases. Σε περίπτωση που δεν του αρέσει το παιχνίδι, μπορεί να το διαγράψει χωρίς να σκέφτεται πως το έχει πληρώσει.

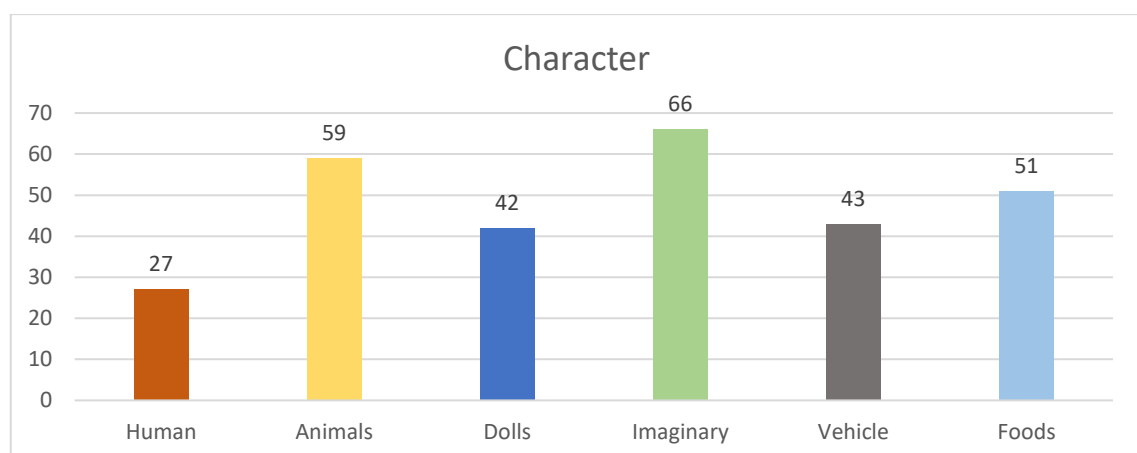
Μερικοί developers βομβαρδίζουν το παιχνίδι με in-app purchases για να εξαναγκάσουν τον χρήστη να αγοράσει. Σκοπός τους είναι να κερδίσουν χρήματα και αρκετές φορές οι χρήστες αγοράζουν επειδή δεν μπορούν να προχωρήσουν στο παιχνίδι και άλλοι απλά διαγράφουν την εφαρμογή.

Οι περισσότεροι δεν είναι διατεθειμένοι να πληρώσουν για παιχνίδια, για αυτό και το μεγαλύτερο ποσοστό των downloads είναι των δωρεάν παιχνιδιών και εφαρμογών. Ο κύριος λόγος που δεν θέλουν να πληρώσουν για να αγοράσουν ένα παιχνίδι είναι επειδή δεν ξέρουν αν αξίζει να το αγοράσουν αφού φοβούνται μήπως δεν τους αρέσει το παιχνίδι [11].

2.7 Τι χαρακτήρες αρέσουν στα παιδιά

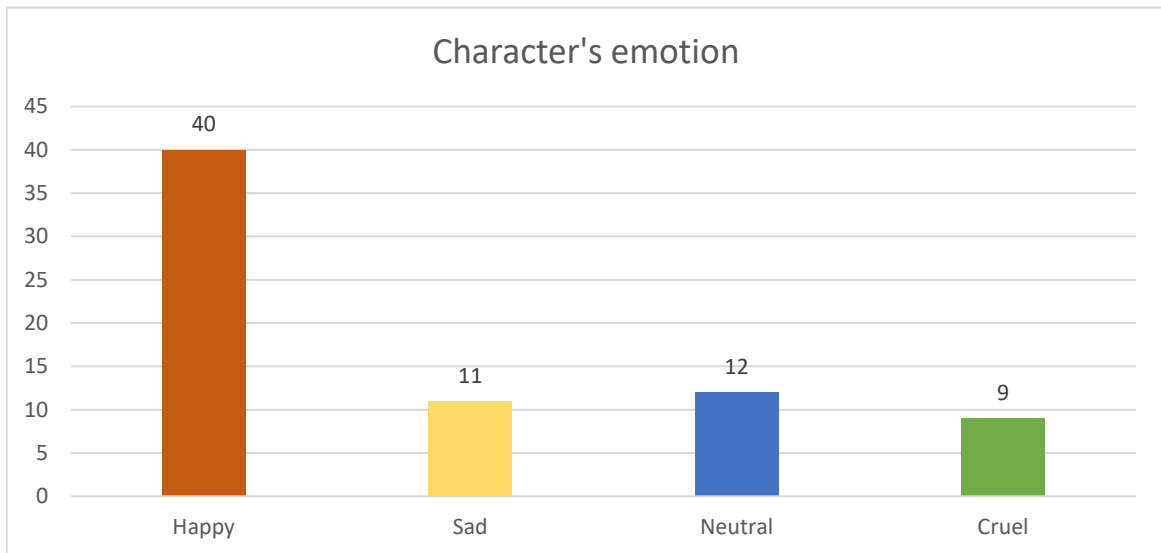
Ένας καλός τρόπος για να τραβήξουμε την προσοχή των παιδιών είναι να χρησιμοποιήσουμε κάποιο χαρακτήρα στο παιχνίδι. Είναι σημαντικό ο χαρακτήρας να είναι ενδιαφέρον έτσι ώστε το παιχνίδι να είναι ελκυστικό για τα παιδιά. Ο χαρακτήρας μπορεί να είναι ένα creature ή ένα κινούμενο αντικείμενο.

Τα παιδιά προσχολικής ηλικίας θεωρούν ότι ένας χαρακτήρας είναι ελκυστικός όταν περιέχει φανταστικά στοιχεία. Τα φανταστικά στοιχεία που προτιμούν είναι ο χαρακτήρας να έχει κάποια δύναμη ή να έχει μορφή cartoon. Ακόμη, παρατηρούν την εξωτερική εμφάνιση αλλά και τα ρούχα και τα accessories του χαρακτήρα. Επίσης, παρατηρήθηκε από έρευνα, πως τα παιδιά προτιμούν φανταστικούς χαρακτήρες και ζώα τα οποία έχουν ανθρώπινη μορφή [9,15]. Στο Σχήμα 2.7.1 : Γραφική παράσταση με το είδος χαρακτήρων που προτιμούν τα παιδιά, παρουσιάζονται τα αποτελέσματα για το είδος των χαρακτήρων.



Σχήμα 2.7.1 : Γραφική παράσταση με το είδος χαρακτήρων που προτιμούν τα παιδιά [15].

Σημαντικό στοιχείο του χαρακτήρα είναι και το συναίσθημα. Τα παιδιά προτιμούν ο χαρακτήρας να είναι χαρούμενος και διασκεδαστικός. Το συναίσθημα που δεν άρεσε στα περισσότερα παιδιά ήταν όταν ο χαρακτήρας ήταν cruel [15]. Στο Σχήμα 2.7.1 : Γραφική παράσταση για το συναίσθημα που προτιμούν τα παιδιά, παρουσιάζονται τα αποτελέσματα για το συναίσθημα.

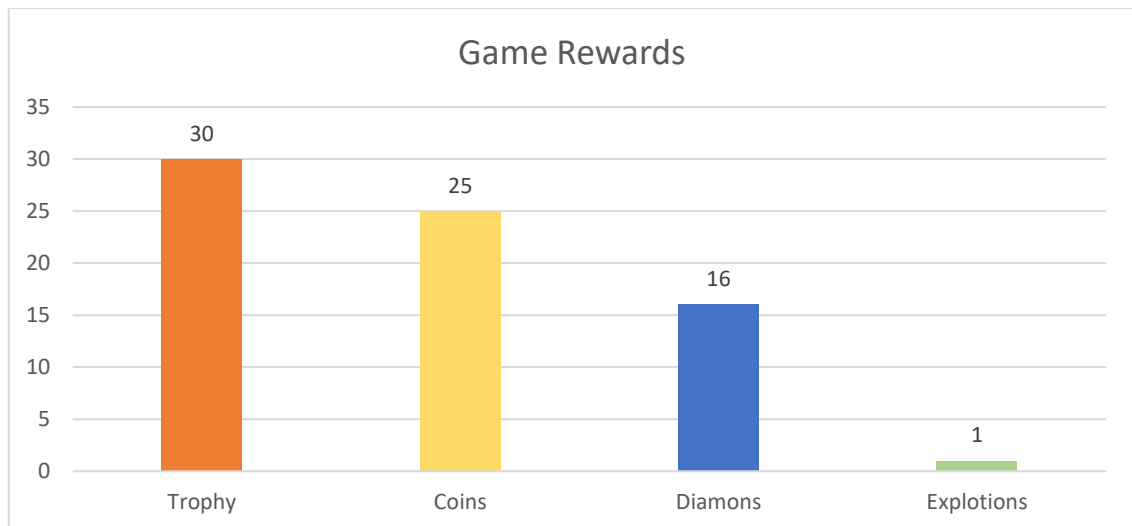


Σχήμα 2.7.1 : Γραφική παράσταση για το συναίσθημα που προτιμούν τα παιδιά [15].

2.8 Είδος παιχνιδιών που αρέσουν στα παιδιά

Τα αγαπημένα χαρακτηριστικά ενός παιδιού προσχολικής ηλικίας που περιέχει ένα παιχνίδι είναι τα rewards και όταν το παιχνίδι περιέχει διάφορων ειδών δραστηριότητες. Οι δραστηριότητες που προτιμούν αφορούν την φροντίδα του χαρακτήρα, η αντιστοίχιση χρωμάτων ή αντικειμένων, shooting, η ζωγραφική και να παίρνουν παραγγελίες φαγητού.

Όσον αφορά τα rewards, λόγω του ότι τα παιδιά χάνουν εύκολα το ενδιαφέρον τους, προτιμούν να υπάρχει κάποιο reward το οποίο μπορεί να είναι κάποιο trophy ή high score. Τα rewards μπορούν να δίνονται είτε όταν ολοκληρωθεί κάποιο level, είτε όταν φτάσει σε κάποιο συγκεκριμένο score είτε όταν ολοκληρώνει κάποιο task. Τα trophies είναι τα rewards που προτιμούν περισσότερο τα παιδιά ενώ το reward που δεν τους αρέσει είναι τα explosions [9,15]. Στο Σχήμα 2.8.1 Γραφική παράσταση για τα rewards που προτιμούν τα παιδιά, παρουσιάζονται τα αποτελέσματα για τα rewards.



Σχήμα 2.8.1 Γραφική παράσταση για τα rewards που προτιμούν τα παιδιά.

2.9 Χρώματα που τραβούν την προσοχή των παιδιών

Ένας πολύ σημαντικός παράγοντας στο να είναι ελκυστικό το παιχνίδι είναι τα χρώματα που χρησιμοποιούνται. Τα παιδιά προτιμούν τα φωτεινά χρώματα όπως είναι τα κύρια χρώματα και τα δευτερεύοντα. Τα κύρια χρώματα είναι το κόκκινο, το πράσινο και το μπλε, ενώ τα δευτερεύοντα χρώματα είναι μίξεις των χρωμάτων. Τα κορίτσια φαίνεται να έχουν ιδιαίτερη προτίμηση στο ροζ και στο μωβ, ενώ τα αγόρια στο κόκκινο, στο κίτρινο, στο μπλε και στο πράσινο [9].

Κεφάλαιο 3

Απαιτούμενη γνώση και Τεχνολογίες, και Ανάλυση Απαιτήσεων και Προδιαγραφές

3.1 Unity	34
3.2 Visual Studio	34
3.3 Modelio	35
3.4 Ανάλυση Requirements για σχεδιασμό παιχνιδιών για παιδιά	35
3.5 Εξειδικευμένες αρχές που θα ακολουθήσει το παιχνίδι	39
3.6 Use cases	40
3.6.1 Περιγραφή διαγράμματος	40

3.1 Unity

Το Unity είναι η πλατφόρμα στην οποία θα δημιουργήσουμε το mobile game. Αρχικά, μας δίνεται η δυνατότητα να επιλέξουμε σε τι πλατφόρμα θέλουμε να γίνει develop το project μας. Για τους σκοπούς αυτής της ατομική διπλωματικής εργασίας, το παιχνίδι θα υλοποιηθεί για Android συσκευές.

Μέσω του Unity, δημιουργούμε τις δικές μας σκηνές οι οποίες είναι τα levels, αλλά και χαρακτήρες. Υπάρχει δυνατότητα να κάνουμε import έτοιμα assets και εικόνες που θέλουμε να χρησιμοποιήσουμε, αντί να τα δημιουργήσουμε όλα από το μηδέν.

3.2 Visual Studio

Το Visual Studio χρησιμοποιείται από το Unity για προγραμματισμό. Η γλώσσα που είναι απαραίτητο να γνωρίζουμε για την υλοποίηση ενός mobile game, είναι η C#. Ουσιαστικά, ο προγραμματισμός είναι απαραίτητος για την λειτουργικότητα του παιχνιδιού.

3.3 Modelio

Το Modelio είναι ένα εργαλείο στο οποίο μπορούμε να δημιουργήσουμε διάφορα διαγράμματα. Μέσω του συγκεκριμένου εργαλείου, δημιουργήθηκε ένα διάγραμμα Use case έτσι ώστε να απεικονίσουμε την αλληλεπίδραση των χρηστών με το παιχνίδι της Μεσογειακής διατροφής. Από το συγκεκριμένο διάγραμμα μπορούμε να δούμε όλες τις περιπτώσεις χρήσης από τους χρήστες.

3.4 Ανάλυση Requirements για σχεδιασμό παιχνιδιών για παιδιά

Η κάθε ηλικία έχει διαφορετικές απαιτήσεις στον σχεδιασμό ενός παιχνιδιού. Είναι πολύ σημαντικό να λαμβάνεται υπόψη η ηλικία, τα συναισθήματα και οι γνωστικές δεξιότητες των παιδιών. Οι 19 πιο σημαντικές απαιτήσεις για τον σχεδιασμό ενός παιχνιδιού για ηλικίες 3 μέχρι 5 ετών είναι οι εξής:

1. Quick Access: το παιδί είναι σημαντικό να ανοίγει το παιχνίδι και να μπορεί να ξεκινήσει εντός δευτερολέπτων. Ένα καλό παράδειγμα είναι να υπάρχει ένα μεγάλο κουμπί play στο main menu του παιχνιδιού.
2. Motivation και Reward: Το παιδί χρειάζεται κάτι για να του δώσει motivation να συνεχίζει να παίζει ή για να ξαναπαίξει το παιχνίδι. Αυτό επιτυγχάνεται με την χρήση των rewards. Τα πιο γνωστά rewards σε παιχνίδια είναι όταν ολοκληρώνεται κάποιο level, όταν φτάνει σε συγκεκριμένα scores και το daily login το οποίο σου δίνει κάποιο δώρο για κάθε ημέρα που παίζεις το παιχνίδι με αποτέλεσμα να παρακινεί το παιδί να ξαναμπει στο παιχνίδι.
3. Multiple paths to a correct answer: Είναι καλό να υπάρχουν πολλά paths γιατί τα παιδιά έχουν περιέργεια και θέλουν να ανακαλύψουν κάθε πιθανό σενάριο δοκιμάζοντας μόνοι τους [3].
4. Σχεδίαση: Το design πρέπει να είναι μοναδικό με κάποια basic functionalities και παρόμοια icons έτσι ώστε να είναι εύκολο το interaction με το παιχνίδι. Σκοπός είναι το παιδί να νιώθει οικεία αφού θα αναγνωρίζει αρκετά κουμπιά. Για παράδειγμα, όταν κάποιος βλέπει το κουμπί που απεικονίζεται στην Εικόνα 3.3.1: Play button, αναγνωρίζει πως είναι το play button [3].



Εικόνα 3.4.1: Play button

5. Μεγάλα hot spots: όσο μικρότερα είναι τα παιδιά, τόσο μεγαλύτερα πρέπει να είναι τα hot spots. Ένας καλός τρόπος για να καταλάβουμε αν τα κουμπιά έχουν ικανοποιητικό μέγεθος, είναι η δοκιμή από παιδιά. Αν υπάρχουν πολλά hot spots στην οθόνη, θα πρέπει να υπάρχει ενδιάμεσος χώρος spacing ανάμεσα τους έτσι ώστε τα παιδιά να μην πατήσουν λάθος κουμπί [3].
6. Μνήμη: Όπως ο υπολογιστής διαθέτει μνήμη RAM, ο άνθρωπος διαθέτει μνήμη εργασίας η οποία αποτελεί το μικρότερο σε διάρκεια είδος μνήμης. Τα παιδιά ακόμη προσπαθούν να κάνουν develop αυτή την μνήμη. Ένα παιχνίδι μπορεί να πιέσει λίγο ένα παιδί να χρησιμοποιήσει την συγκεκριμένη μνήμη, όμως είναι σημαντικό να μην γίνεται overload, δηλαδή να μην έχουν να διαχειριστούν πολλά tasks ταυτόχρονα [3].
7. Capture their attention: Τα αντικείμενα που είναι πιο φωτεινά και αστεία, πάντα τραβούν την προσοχή των παιδιών. Σημαντικό είναι να δώσουμε έμφαση σε αυτό που θέλουμε να ασχοληθεί το παιδί, αλλιώς θα είναι απλά distraction. Ακόμη, επειδή τα παιδιά δεν μένουν πολλή ώρα συγκεντρωμένα, το παιχνίδι θα πρέπει να έχει short tasks [3].
8. Μείωση πιθανότητας το παιδί να κάνει in-app purchases: Τα in-app purchases είναι εντός του παιχνιδιού και συνήθως αφορούν αγορά χαρακτήρων και ξεκλείδωμα levels. Ένας πιθανός τρόπος να αποφύγουμε να γίνουν αγορές κατά λάθος, είναι να ζητά επαλήθευση κωδικού του λογαριασμού στο Google Play για επιβεβαίωση [3].
9. Αλληλεπίδραση με άλλους: Τα παιδιά σε αυτές τις ηλικίες ξεκινούν να ενδιαφέρονται για κοινωνικές εμπειρίες, για παράδειγμα, έχουν ανάγκη να δείξουν στους φίλους τους τα νέα τους παιχνίδια, να τους δείξουν τι έφτιαξαν και να παίξουν μαζί. Ένα έξυπνο feature σε παιχνίδια είναι η δυνατότητα να παίζουν

δύο χρήστες στην ίδια συσκευή. Τα παιδιά με αυτό τον τρόπο, ενώ παίζουν και διασκεδάζουν, μαθαίνουν να συνεργάζονται και να αλληλοεπιδρούν με άλλους [4].

10. Δημιουργικότητα: Στα παιδιά προσχολικής ηλικίας, αρέσει να δημιουργούν μόνοι τους αντί να ακολουθούν αυστηρούς κανόνες, για αυτό το παιχνίδι θα πρέπει να έχει χώρο για την δημιουργικότητα και την φαντασία των παιδιών. Για παράδειγμα, μπορούν να δημιουργούν τον δικό τους χαρακτήρα ή να τον ντύνουν με τα ρούχα που τους αρέσουν [4].
11. Εικονικό περιεχόμενο: Ένα παιδί προσχολικής ηλικίας δεν ξέρει να διαβάσει αλλά ούτε και να κάνει αριθμητική. Άρα το παιχνίδι θα πρέπει να περιέχει αρκετό εικονικό περιεχόμενο για να είναι πιο εύκολο για το παιδί να παίξει και να καταλαβαίνει τι βλέπει στην οθόνη.
12. Ισχυρές συναισθηματικές αντιδράσεις: τα παιχνίδια πρέπει να δημιουργούν συναισθήματα, όπως είναι για παράδειγμα ο φόβος και η χαρά, και για να το πετύχουν αυτό πρέπει να υπάρχει ισορροπία στον αριθμό των components. Τα components είναι τα χαρακτηριστικά των χαρακτήρων, το περιβάλλον, τα rewards, τα εμπόδια που μπορεί να υπάρχουν, η αφήγηση του παιχνιδιού αν υπάρχει κάποια πλοκή και η συνεργασία με άλλους παίκτες. [12]. Στην Εικόνα 3.3.2: Το βιντεοπαιχνίδι Journey δημιουργεί στους παίκτες διαφορετικά συναισθήματα λόγω του περιβάλλοντος, μπορούμε να δούμε πως το περιβάλλον μπορεί να επηρεάσει τα συναισθήματα του χρήστη είτε με τα αντικείμενα που υπάρχουν, είτε με τα χρώματα που χρησιμοποιούνται.



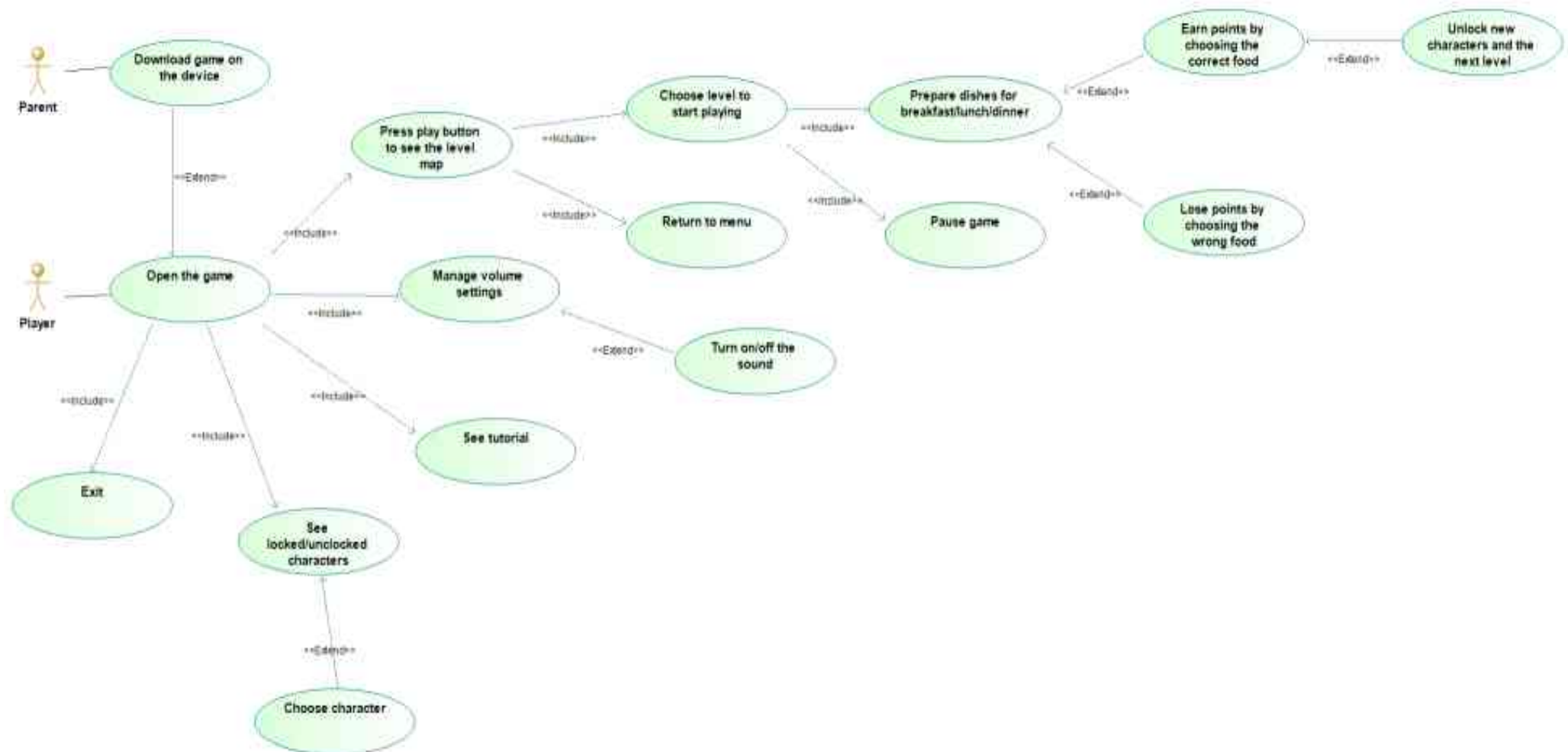
Εικόνα 3.4.2 : Το βιντεοπαιχνίδι Journey δημιουργεί στους παίκτες διαφορετικά συναισθήματα λόγω του περιβάλλοντος [18].

13. Identity: Ο κάθε χρήστης θα πρέπει να νιώθει ότι είναι ένα unique άτομο μέσα στο environment του παιχνιδιού. Η ταυτότητα του παίκτη αντιπροσωπεύεται μέσω κάποιου χαρακτήρα. Μέσω μιας έρευνας που έχει διεξαχθεί, τα παιδιά που είχαν την επιλέξουν ποιο χαρακτήρα θέλουν να τους αντιπροσωπεύει ήταν περισσότερο ικανοποιημένα από τα παιδιά που είχαν να επιλέξουν μόνο μεταξύ ενός τυπικού ανδρικού χαρακτήρα και ενός τυπικού θηλυκού χαρακτήρα [1].
14. Αυξημένη πολυπλοκότητα: Τα περισσότερα παιχνίδια έχουν πολλά levels τα οποία όσο προχωράει ο παίκτης γίνονται πιο δύσκολα. Σκοπός είναι να βελτιωθούν οι ικανότητες του παίκτη [1].
15. Παροχή δραστηριοτήτων που σχετίζονται με το θέμα: Το παιχνίδι που θα υλοποιηθεί αφορά την Μεσογειακή διατροφή άρα θα πρέπει να παρέχονται δραστηριότητες έτσι ώστε τα παιδιά να μπορούν να εξερευνήσουν όλες τις πτυχές [5].
16. Επαναλαμβανόμενα στοιχεία: Τα παιχνίδια που αφορούν την υγεία είναι καλό να περιλαμβάνουν επαναλαμβανόμενα στοιχεία με σκοπό διευκολύνουν κάποιες επαναλαμβανόμενες ενέργειες και να βοηθήσουν τα παιδιά. Για παράδειγμα, τα επαναλαμβανόμενα στοιχεία μπορούν να είναι εικόνες και ήχοι που να βοηθούν στην εξοικείωση των παιδιών [5].
17. Παροχή πρακτικών και ενημερωτικών πληροφοριών: Τα παιχνίδια που αφορούν την υγεία, γίνονται με σκοπό να ενημερώσουν για το συγκεκριμένο πρόβλημα υγείας για αυτό θα πρέπει να προσφέρουν ενημερωτικές αλλά και πρακτικές πληροφορίες με σκοπό να ενημερώσουν τα παιδιά που θα παίζουν το παιχνίδι. Αυτό μπορεί να γίνει με οπτικοακουστικά βοηθήματα [5].
18. Παροχή καθοδήγησης: Το παιχνίδι πρέπει να παρέχει κάποιο είδος βοήθειας, για παράδειγμα μπορεί στο πρώτο level να δείχνει τις κινήσεις που μπορεί ο παίκτης να κάνει. Ακόμη, μπορεί να υπάρχει και σιωπηρή καθοδήγηση, δηλαδή το παιχνίδι να είναι υλοποιημένο με τέτοιο τρόπο που να καθοδηγεί τον παίκτη για το τι πρέπει να κάνει [2].
19. Παροχή feedback: Μια καλή τεχνική που υπάρχει σε αρκετά παιχνίδια είναι η εμφάνιση της προόδου. Κάθε ενέργεια μπορεί να έχει άμεση ανατροφοδότηση. Σκοπός είναι οι παίκτες να μαθαίνουν δεξιότητες για να ξεπεράσουν τις προκλήσεις του παιχνιδιού [2].

3.5 Εξειδικευμένες αρχές που θα ακολουθήσει το παιχνίδι

1. Η πλοήγηση πρέπει να είναι σαφής και απλή.
2. Ταξινόμηση πληροφοριών ανάλογα με το κάθε concept, όπως είναι για παράδειγμα μια ομάδα τροφίμων.
3. Αντικατάσταση κειμένου με οπτικοακουστικό περιεχόμενο.
4. Τα χρώματα πρέπει να είναι ανοιχτά σε ζεστούς τόνους.
5. Ενσωμάτωση ήχων, σχημάτων και χρωμάτων για την δημιουργία animation effects με σκοπό να προσελκύσει την προσοχή των παιδιών.
6. Τα πιο δημοφιλές operation modes είναι το click και το drag and drop. Με αυτό τον τρόπο έχουμε ένα απλό operation mode για το παιχνίδι.
7. Όλα τα operational results πρέπει να είναι ορατά, καθώς επίσης πρέπει να υπάρχει κάποιο feedback [8].
8. Το παιχνίδι πρέπει να περιέχει rewards, έτσι ώστε να τραβούν την προσοχή των παιδιών.
9. Λόγω του ότι το παιχνίδι είναι εκπαιδευτικό, πρέπει να υπάρχουν επαναλαμβανόμενα στοιχεία, όπως είναι οι εικόνες με τα φαγητά που θα εμφανίζονται.

3.6 Use case



Σχήμα 3.6.1 : Use case diagram

3.6.1 Περιγραφή διαγράμματος

Το Σχήμα 3.6.1 : Use case diagram περιγράφει την λειτουργία του παιχνιδιού. Αρχικά, ένας από τους γονείς του παιδιού πρέπει να εγκαταστήσει το παιχνίδι σε συσκευή Android. Αφού εγκατασταθεί το παιχνίδι, ο player ο οποίος είναι το παιδί, μπορεί να ανοίξει το παιχνίδι. Όταν ανοίξει το παιχνίδι, έχει την δυνατότητα να πατήσει το κουμπί “Exit” για να βγει από την εφαρμογή, μπορεί να επιλέξει να δει τους χαρακτήρες, μπορεί να δει το video tutorial, να ρυθμίσει το volume και μπορεί να πατήσει το “Play button” για να μεταβεί στον χάρτη με τα levels.

Στην περίπτωση που επιλέξει να δει τους χαρακτήρες, έχει την δυνατότητα να επιλέξει με ποιο ξεκλειδωμένο χαρακτήρα επιθυμεί να παίξει. Όταν το παιδί πατήσει το Play button το οποίο βρίσκεται κάτω από κάθε χαρακτήρα, ο χαρακτήρας θα εμφανίζεται σε όλα τα levels του παιχνιδιού.

Στην περίπτωση που επιθυμεί να ρυθμίσει το volume, μπορεί να πατήσει το κουμπί με το εικονίδιο του ήχου έτσι ώστε ο ήχος να είναι είτε on είτε off. Στην περίπτωση που ο ήχος είναι on, τότε όταν πατήσει οποιοδήποτε κουμπί, θα ακούγεται ο ήχος. Σε περίπτωση που ο ήχος είναι off, δεν θα ακούγεται όταν πατήσει κάποιο κουμπί.

Αν το παιδί μεταβεί στον χάρτη με τα levels έχει την δυνατότητα να επιστρέψει στο αρχικό menu ή μπορεί να επιλέξει κάποιο level για να ξεκινήσει το παιχνίδι. Αν επιλέξει κάποιο level, έχει την δυνατότητα να κάνει pause το παιχνίδι για να επιστρέψει στο αρχικό menu ή μπορεί να συνεχίσει το παιχνίδι και να επιλέγει φαγητά για κάθε στάδιο. Το κάθε level αποτελείτε από 3 στάδια, το πρωινό, το μεσημεριανό και το βραδινό. Για κάθε στάδιο, όταν επιλέγει φαγητά κερδίζει ή χάνει πόντους ανάλογα με τις επιλογές του. Όταν ολοκληρώσει κάποιο level ανάλογα με το score που έκανε, μπορεί να ξεκλειδώσει καινούργιους χαρακτήρες και το επόμενο level. Οι καινούργιοι χαρακτήρες που έχουν ξεκλειδωθεί, θα εμφανίζονται στο Character selection και θα μπορεί να τους επιλέξει για να παίξει τα επόμενα levels. Όσο αφορά τα levels που έχουν ξεκλειδωθεί, στον χάρτη με τα levels, θα εμφανίζεται ο αριθμός που αντιπροσωπεύει το συγκεκριμένο level, ενώ τα levels που δεν έχουν ξεκλειδωθεί, θα εμφανίζεται το εικονίδιο της κλειδαριάς για να μην μπορεί να παίξει το level.

Κεφάλαιο 4

Σχεδιασμός Συστήματος και Υλοποίηση

4.1 Σχεδιασμός Συστήματος	42
4.2 Υλοποίηση	47
4.2.1 Κουμπιά – Πλοήγηση	47
4.2.2 Ξεκλείδωμα χαρακτήρων και levels	49
4.2.3 Score	50

4.1 Σχεδιασμός Συστήματος

Για τον σχεδιασμό του συστήματος χρησιμοποιήθηκε η μηχανή Unity. Το mobile operating system που επιλέχθηκε για την δημιουργία του παιχνιδιού είναι το Android. Το Unity υποστηρίζει το development παιχνιδιών σε Android. Λόγω του ότι το παιχνίδι αφορά παιδιά προ δημοτικής εκπαίδευσης, το παιχνίδι είναι 2D με απλές λειτουργίες και σαφής πλοήγηση.

Για να μπορεί το παιχνίδι να είναι προσιτό και ελκυστικό για ηλικίες 3 μέχρι 5 ετών, τα χρώματα που επιλέχθηκαν ήταν ανοιχτοί τόνοι, κύρια χρώματα και μίξεις των κύριων χρωμάτων. Τα εικονίδια, τα background και οι χαρακτήρες έχουν όλα φωτεινά χρώματα με σκοπό να προκαλεί στους χρήστες το αίσθημα χαράς.

Ακόμη, τοποθετήθηκαν rewards έτσι ώστε να δίνουν motivation στα παιδιά για να συνεχίσουν να παίζουν το παιχνίδι, καθώς και να τους τραβήξουν την προσοχή. Τα rewards του παιχνιδιού είναι οι χαρακτήρες τους οποίους μπορούν να επιλέγουν για να παίζουν τα διάφορα στάδια. Οι χαρακτήρες που επιλέγηκαν είναι χαρούμενοι και διασκεδαστικοί.



Εικόνα 4.1.1 Χαρακτήρας Μήλο



Εικόνα 4.1.2 Χαρακτήρας Πορτοκάλι



Εικόνα 4.1.3 Χαρακτήρας Αβοκάντο



Εικόνα 4.1.4 Χαρακτήρας Μπανάνα



Εικόνα 4.1.5 Χαρακτήρας Τυρί



Εικόνα 4.1.6 Χαρακτήρας Αυγό



Εικόνα 4.1.6 Χαρακτήρας Σταφύλι



Εικόνα 4.1.7 Χαρακτήρας Nutella



Εικόνα 4.1.8 Χαρακτήρες Παγωτό



Εικόνα 4.1.9 Χαρακτήρας
Ελαιόλαδο

Τα icons που χρησιμοποιήθηκαν για τα κουμπιά, έχουν σκοπό να κάνουν τα παιδιά να νιώθουν οικεία. Για τα κουμπιά χρησιμοποιήθηκε μόνο εικονικό περιεχόμενο για να μπορούν τα παιδιά να τα αναγνωρίζουν, λόγω του ότι δεν ξέρουν να διαβάζουν ακόμη.



Εικόνα 4.1.10 Αρχικό μενού

Για τα φαγητά επιλέχθηκαν απλές εικόνες με σκοπό να τραβήξουν την προσοχή του παιδιού. Ο λόγος που τοποθετήθηκαν αριστερά της οθόνης είναι λόγω του ότι το background περιέχει μόνο λίγες λεπτομέρειες οι οποίες δεν θα αποσπούν την προσοχή των παιδιών από τα φαγητά. Επίσης, έχει τοποθετηθεί ένα πορτοκαλί τραπέζι το οποίο θα έχει όσα τρόφιμα έχουν επιλεγεί. Σκοπός είναι να γνωρίζουν τα παιδιά τις επιλογές που έκαναν για δική τους διευκόλυνση. Τα φαγητά είναι κουμπιά σε panels διαφορετικών canvases, και μπορούν να επιλεγούν πατώντας απλά πάνω τους. Βλέπε Εικόνα 4.1.11 Level 1 Στάδιο 1 – Πρωινό.



Εικόνα 4.1.11 Level 1 Στάδιο 1 - Πρωινό

Το κάθε level περιέχει ένα canvas για το score και δίπλα του ένα εικονίδιο βραβείου έτσι ώστε τα παιδιά να καταλαβαίνουν τι αφορά ο αριθμός στο πάνω μέρος της οθόνης. Από την Εικόνα 4.1.12 Αύξηση score, μπορούμε να δούμε πως όταν το παιδί επιλέξει κάποιο φαγητό και κερδίζει πόντους, θα εμφανίζεται το ποσό που κέρδισε με πράσινο χρώμα και δίπλα το εικονίδιο “+”. Σε περίπτωση που χάσει πόντους, εμφανίζεται το ποσό που αφαιρέθηκε από το score με κόκκινο χρώμα και δίπλα το εικονίδιο “-”. Η αφαίρεση των πόντων απεικονίζεται στην Εικόνα 4.1.13 Μείωση score.



Εικόνα 4.1.12 Αύξηση score



Εικόνα 4.1.13 Μείωση score

Για τον χάρτη με τα levels, όπως φαίνεται και στην Εικόνα 4.1.14 Χάρτης με τα levels, έχει επιλεγεί ένα μονόχρωμο background σε ένα φωτεινό τόνο. Σκοπός ήταν να δώσουμε έμφαση στην πυραμίδα της Μεσογειακής διατροφής και στα κουμπιά τα οποία έχουν έντονο χρώμα για να μην μπερδεύονται με τις πληροφορίες του background έτσι ώστε να μπορεί να τα διακρίνει το παιδί



Εικόνα 4.1.14 Χάρτης με τα levels

Το παιχνίδι σχεδιάστηκε ακολουθώντας έρευνες, άρθρα και βιβλία τα οποία αναγράφονται πιο κάτω. Λήφθηκαν υπόψη οι κανόνες σχεδιασμού παιχνιδιού για παιδιά προσχολικής ηλικίας, τα ενδιαφέροντα των παιδιών και το τι προτιμούν να περιέχει το παιχνίδι, όπως για παράδειγμα, τι χρώματα προτιμούν αλλά τι χαρακτήρες και rewards θεωρούν περισσότερο ελκυστικά.

4.2 Υλοποίηση

Η υλοποίηση του συστήματος αφορά την λειτουργικότητα του παιχνιδιού. Οι κύριες λειτουργίες αφορούν την λειτουργικότητα όλων των κουμπιών, των ήχων που χρησιμοποιήθηκαν, την πλοήγηση μέσα στο παιχνίδι, το ξεκλείδωμα των levels και των χαρακτήρων, την επιλογή χαρακτήρα και ο υπολογισμός των scores.

4.2.1 Κουμπιά – Πλοήγηση

Στο αρχικό μενού υπάρχουν 5 κουμπιά, το κουμπί για έξοδο από το παιχνίδι, το “Play button” το οποίο σε οδηγεί στον χάρτη με τα levels, το κουμπί που σε οδηγεί στο scene με τους χαρακτήρες, το κουμπί για το volume και το κουμπί για το “Tutorial”. Το κάθε κουμπί σε οδηγεί σε κάποιο διαφορετικό scene εκτός από το “Exit button”.

Για τους χαρακτήρες έχει δημιουργηθεί κώδικας ο οποίος αναγνωρίζει ποιοι χαρακτήρες είναι ξεκλειδωμένοι και ποιοι όχι. Υπάρχουν κουμπιά για navigation τα οποία είναι βελάκια δεξιά και αριστερά της οθόνης για να μπορεί ο χρήστης να δει τους χαρακτήρες. Βλέπε Εικόνα 4.2.1.1 Κλειδωμένος Χαρακτήρας. Όσοι χαρακτήρες είναι ξεκλειδωμένοι εμφανίζονται κανονικά και στο κάτω μέρος της οθόνης υπάρχει το κουμπί “Play” το οποίο όταν το πατήσουμε επιλέγουμε τον χαρακτήρα με τον οποίο θέλουμε να παίξουμε. Στην περίπτωση που κάποιοι χαρακτήρες δεν έχουν ακόμα ξεκλειδωθεί, εμφανίζεται εικονίδιο κλειδαριάς και δεν υπάρχει το κουμπί “Play”.



Εικόνα 4.2.1.1 Κλειδωμένος Χαρακτήρας

Όταν πατήσουμε το κουμπί για το tutorial μας οδηγεί σε ένα scene που περιέχει ένα επεξηγηματικό video με σκοπό να εξηγήσει πως παίζεται το παιχνίδι με οπτικοακουστικό περιεχόμενο. Μπορούμε να επιστρέψουμε στο αρχικό μενού πατώντας το κουμπί με το εικονίδιο “x” που βρίσκεται δεξιά στο πάνω μέρος της οθόνης.



Εικόνα 4.2.1.2 Screenshot από το video του tutorial

Ακόμη, για κάθε level έχουμε ένα script. Επειδή έχουμε ένα κουμπί για κάθε φαγητό, όταν πατηθεί κάποιο φαγητό καλείτε μια function η οποία κάνει `SetActive(false)` το canvas που περιέχει το φαγητό με σκοπό να μην εμφανίζεται στα αριστερά και το εμφανίζει στο τραπέζι. Επίσης, καλείτε και μια function η οποία αφορά το score. Αν για παράδειγμα, ο χρήστης επιλέξει κάποιο γαλακτοκομικό, καλείτε η function για τα γαλακτοκομικά με σκοπό να προσθέσει ή να αφαιρέσει πόντους ανάλογα με τις επιλογές του χρήστη. Όλα τα κουμπιά είναι συνδεδεμένα με μια function σε διαφορετικό κώδικα, η οποία ενεργοποιεί τον ήχο. Ο ήχος ακούγεται κάθε φορά που θα πατηθεί οποιοδήποτε κουμπί. Επιπρόσθετα, τα φαγητά που βρίσκονται στο τραπέζι είναι και αυτά κουμπιά τα οποία όταν πατηθούν φεύγουν από το τραπέζι και εμφανίζεται ξανά το canvas στα αριστερά της οθόνης. Αν για παράδειγμα ο χρήστης επιλέξει ένα ποτήρι νερό, κερδίζει 5 πόντους. Αν αποφασίσει πως θέλει να φύγει αυτή την επιλογή από το τραπέζι, πατώντας το ποτήρι με το νερό, το νερό θα επιστρέψει στις επιλογές στα αριστερά και οι πόντοι που κέρδισε θα αφαιρεθούν.

Για την εμφάνιση των score είτε στα levels είτε στον χάρτη με τα levels υπάρχει διαφορετικός κώδικας ο οποίος είναι συνδεδεμένος με το text mesh pro έτσι ώστε να ανανεώνεται με κάθε αλλαγή. Στην περίπτωση που προστεθούν πόντοι στο score, θα εμφανίζεται το ποσό που προστέθηκε με πράσινα γράμματα και το εικονίδιο “+” και σε περίπτωση που αφαιρεθούν πόντοι, εμφανίζεται το ποσό που αφαιρείται με κόκκινο χρώμα και το εικονίδιο “-“ δίπλα έτσι ώστε να καταλαβαίνει ο χρήστης αν η επιλογή που έκανε είναι σωστή ή όχι.

Με τα πράσινα κουμπιά στα δεξιά στο κάτω μέρος της οθόνης σε όλα τα στάδια είναι για να προχωρήσουμε στο επόμενο στάδιο. Το πράσινο κουμπί σε όλα τα στάδια με το βραδινό, είναι για την ολοκλήρωση του level , την αποθήκευση του score και την πλοήγηση στον χάρτη με τα levels.

Σε κάθε στάδιο, στο πάνω μέρος της οθόνης, δίπλα από το canvas του score, υπάρχει το “Pause button” το οποίο όταν το πατήσουμε, εμφανίζει ένα pop-up window το οποίο περιέχει ακόμα 2 κουμπιά. Τα κουμπιά στο “Pause menu” είναι το “Resume button” το οποίο όταν πατηθεί επιστρέφει στο στάδιο για να συνεχίσουμε και το “Home button” που μας οδηγεί στο αρχικό μενού. Σε περίπτωση που επιθυμούμε να επιστρέψουμε στο αρχικό μενού πατώντας το συγκεκριμένο κουμπί του “Pause menu”, δεν θα γίνει αποθήκευση του score του συγκεκριμένου level.

Τέλος, έχουμε τα κουμπιά του χάρτη με τα levels. Το κάθε κουμπί αντιπροσωπεύει ένα level. Το κάθε level αντιπροσωπεύει μια ημέρα της εβδομάδας. Για παράδειγμα, το level 1 είναι η Δευτέρα. Πατώντας κάποιο από τα κουμπιά γίνεται πλοήγηση σε συγκεκριμένο level.

4.2.2 Ξεκλείδωμα χαρακτήρων και levels

Οι χαρακτήρες και τα levels ξεκλειδώνονται ανάλογα με το highscore. Για αυτό έχει δημιουργηθεί κώδικας ο οποίος ελέγχει αν ολοκληρώνονται τα προηγούμενα levels και το highscore. Επίσης, γίνεται έλεγχος αν το level έχει ήδη ξεκλειδωθεί έτσι ώστε αν κάποιος παίζει ξανά το προηγούμενο level και έχει πάρει χαμηλότερο score να μην κλειδωθεί ξανά το επόμενο level. Σκοπός είναι η διαδικασία να γίνεται μια φορά.

Η ίδια διαδικασία γίνεται και για τους χαρακτήρες, όμως για τους χαρακτήρες έχει προστεθεί κώδικας ο οποίος όταν επιλέξεις ένα χαρακτήρα, ο συγκεκριμένος χαρακτήρας να εμφανίζεται και στα στάδια. Τόσο στα γεύματα όσο και στους χαρακτήρες, με την βοήθεια κώδικα, το πρόγραμμα κρύβει κάποια canvas αλλά και τους υπόλοιπους χαρακτήρες και εμφανίζει μόνο αυτό που επιλέχθηκε.

4.2.3 Score

Όπως αναφέρθηκε, ο κώδικας για το score έχει διαφορετικά functions για κάθε κατηγορία φαγητών. Ο κώδικας ελέγχει όλες τις επιλογές που έγιναν σε κάθε level και ελέγχει αν οι ποσότητες κάθε κατηγορίας είναι εντός ορίων ή έχουν ξεπεράσει το όριο. Για παράδειγμα, τα guidelines της Μεσογειακής διατροφής για τα όσπρια είναι πως πρέπει να τρώμε όσπρια 3 φορές την εβδομάδα. Για κάθε επιλογή που κάνει ο χρήστης, υπολογίζεται το score. Για να υπολογιστεί το highscore, υπολογίζουμε το άθροισμα των scores όλων των levels.

Ο κώδικας για το score, έχει υλοποιηθεί ακολουθώντας τις σημειώσεις της Δρ. Έλενας Φιλίππου η οποία είναι Κλινική Διατροφολόγος. Στις σημειώσεις αναγράφονταν οι ποσότητες κάθε κατηγορίας φαγητού και κατά πόσο οι ποσότητες έπρεπε να καταναλώνονται καθημερινά ή εβδομαδιαία.

Κεφάλαιο 5

Αξιολόγηση Συστήματος – Αποτελέσματα - Συζήτηση

5.1 Αξιολόγηση Συστήματος	51
5.2 Αποτελέσματα	52
5.3 Συζήτηση	55

5.1 Αξιολόγηση Συστήματος

Μετά την υλοποίηση του συστήματος, το παιχνίδι δόθηκε σε διάφορες ηλικίες για αξιολόγηση, και όχι μόνο σε ηλικίες 3 μέχρι 5 ετών. Οι έφηβοι και οι ενήλικοι έχουν έρθει σε επαφή με διάφορους τύπους παιχνιδιών, για αυτό το feedback που δόθηκε και οι προτεινόμενες ιδέες για να βελτιωθεί το παιχνίδι, ήταν περισσότερο λεπτομερείς.

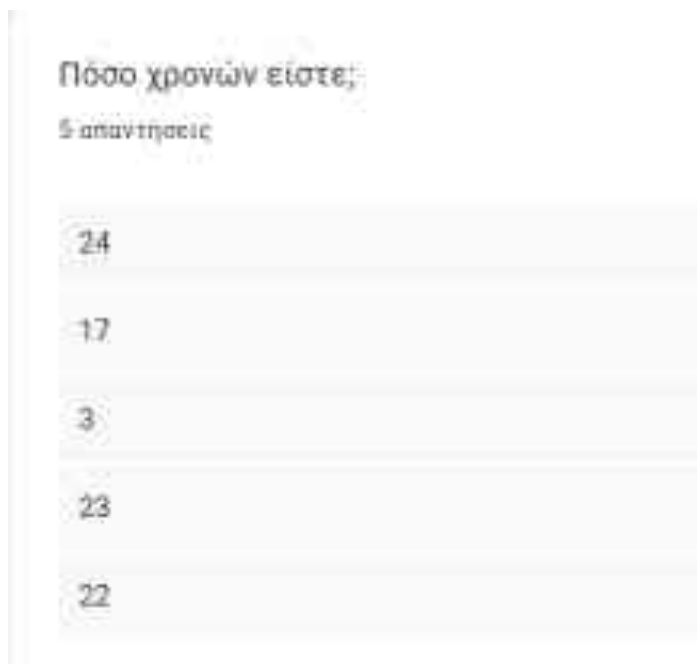
Για την αξιολόγηση, δημιουργήθηκε ένα questionnaire με την χρήση Google Forms. Το questionnaire συμπληρώθηκε από μερικούς χρήστες αφότου έπαιξαν το παιχνίδι. Οι ερωτήσεις που περιείχε το questionnaire αφορούσαν την ηλικία του χρήστη, τι δεν του έχει αρέσει στο παιχνίδι, τι του έχει αρέσει και ιδέες που πρότειναν για βελτίωση.

Το questionnaire βρίσκεται στο link : <https://forms.gle/QY9fFuNdhpM2NNrD7> .

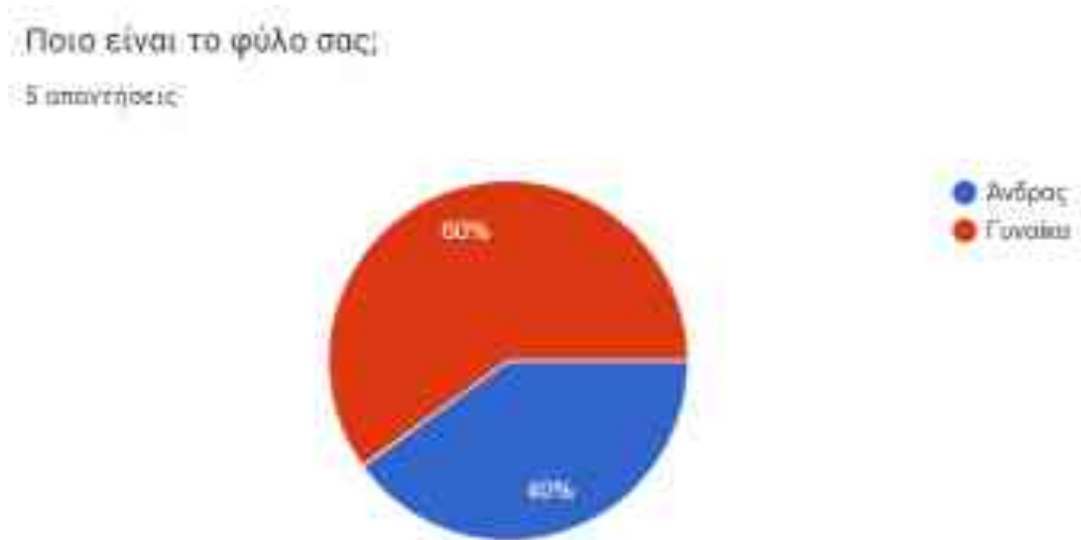
Οι εισηγήσεις για το παιχνίδι, δεν λήφθηκαν μόνο από το questionnaire για αυτό και τα αποτελέσματα πιο κάτω παρουσιάζονται διαφορετικά. Αρκετές εισηγήσεις ήταν σημαντικές για την βελτίωση του παιχνιδιού και υλοποιήθηκαν με επιτυχία. Για κάθε καινούργια εισήγηση που υλοποιήθηκε, λόγω του ότι στο Unity το παιχνίδι πρέπει να γίνει build, υπήρξαν αρκετές εκδόσεις μέχρι να φθάσουμε στην τελική. Οι εισηγήσεις που δεν έχουν υλοποιηθεί, μπορούν να υλοποιηθούν σε μεταγενέστερο στάδιο.

5.2 Αποτελέσματα

Τα αποτελέσματα που πήρα από το questionnaire ήταν τα εξής:



Εικόνα 5.2.1 Απαντήσεις για την ηλικία



Εικόνα 5.2.2 Απαντήσεις για το φύλο

Περιγράψτε τι σας έχει άρεσει στο παιχνίδι:

5 απαντήσεις

Οι χαρακτήρες

Οι χαρακτήρες και το εύρος των φαγητών που μπορώ να επιλέξω

- Η ποικιλία χαρακτήρων, ο τρόπος απόκτησής τους και τα animations τους
- Το παιχνίδι έμμεσα συμβάλλει στην εκμάθηση των παιδιών για την υγιεινή διατροφή τους

Οι χαρακτήρες του παιχνιδιού
Το ότι είναι απλό και όχι περίπλοκο

Οι εικόνες

Εικόνα 5.2.3 Απαντήσεις για το τι άρεσε στους χρήστες

Περιγράψτε τι δεν σας έχει άρεσει στο παιχνίδι:

5 απαντήσεις

Οι εικόνες των φαγητών

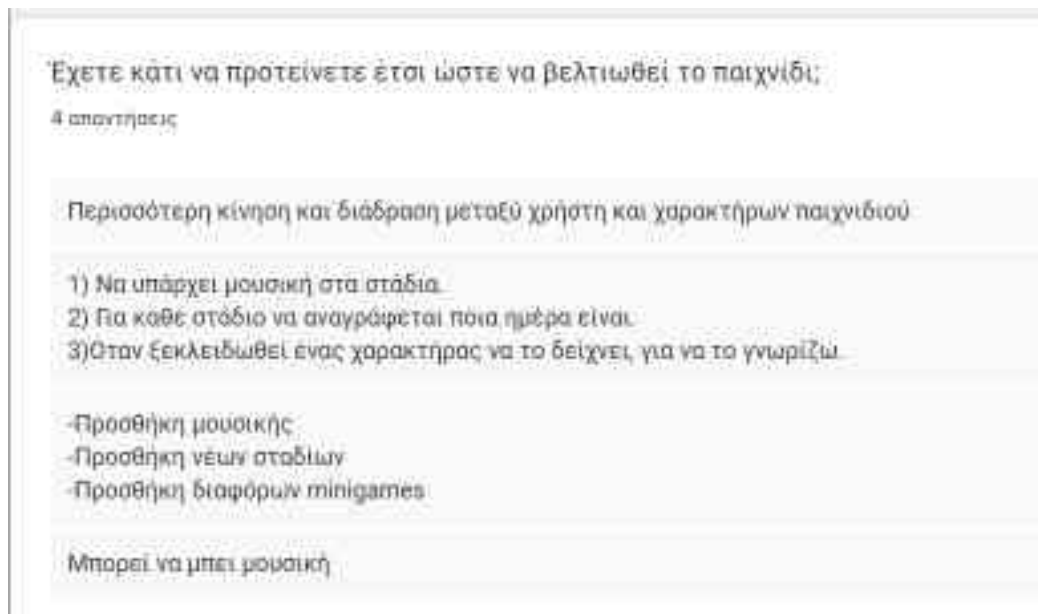
Θα προτιμούσα οι εικόνες των φαγητών να ήταν μεγαλύτερες για να φαίνονται καλύτερα.

-Ελάχιστα στάδια
-Μερικές εικόνες δεν ήταν ξεκάθαρες

Κάποιες εικόνες ήταν λίγο μικρές

Κατά την αύξηση ή μείωση του score δεν εμφανίζει το + και - εικονίδιο αλλά το ερωτηματικό (?).
Το ανοίξα στο tablet το παιχνίδι και κοβεται λίγο, δεν είναι ακριβώς ευθυγραμμισμένο με την οθόνη του tablet.

Εικόνα 5.2.4 Απαντήσεις για το τι δεν άρεσε στους χρήστες



Εικόνα 5.2.5 Εισηγήσεις

Οι υπόλοιπες εισηγήσεις που πήρα ήταν οι εξής:

1. Αντικατάσταση εικονιδίων: Τα εικονίδια “x” και “v” θα ήταν καλύτερα να αντικατασταθούν με φατσούλες.
2. Ακουστικό περιεχόμενο στους τίτλους: Να υπάρχει επιλογή να πατούν πάνω στην λέξη και να διαβάζεται από την εφαρμογή.
3. Συχνό feedback: Να υπάρχει πιο συχνό feedback για να κατανοήσουν τα παιδιά την αιτιολόγηση των επιλογών τους.
4. Απλό background: Το background των σταδίων μπορεί να αντικατασταθεί με πιο απλό και λιγότερες λεπτομέρειες.
5. Ενημερωτικό video: Θα μπορούσε να υπάρχει ένα ενημερωτικό video στην αρχή για εισαγωγή που θα αφορά την Μεσογειακή διατροφή.
6. Εικόνες φαγητών: Τα φαγητά που πρέπει να καταναλώνονται περισσότερο, όπως είναι τα λαχανικά και τα φρούτα, μπορούν να εμφανίζονται σε όλα τα στάδια.
7. Αντικατάσταση τραπεζιού: μπορεί να γίνει αντικατάσταση του τραπεζιού με ένα μεγάλο πιάτο έτσι ώστε να ταιριάζει με τον τίτλο του παιχνιδιού που είναι Prepare your dish.
8. Χρώμα για κάθε κατηγορία φαγητού: Η κάθε κατηγορία φαγητού να έχει διαφορετικό χρώμα. Για παράδειγμα, το panel των φρούτων και των λαχανικών μπορεί να είναι πράσινο, ενώ το panel για τα ψάρια μπορεί να είναι μπλε.

5.3 Συζήτηση

Πολλές από τις εισηγήσεις έχουν ληφθεί υπόψη και έχουν υλοποιηθεί έτσι ώστε να βελτιωθεί το παιχνίδι. Σκοπός είναι να καλυφθούν όλες οι απαιτήσεις για να έχουμε ένα ολοκληρωμένο εκπαιδευτικό παιχνίδι το οποίο τα παιδιά θα θεωρούν διασκεδαστικό και ευχάριστο αλλά ταυτόχρονα να μαθαίνουν για την Μεσογειακή διατροφή.

Το παιχνίδι μπορεί να επεκταθεί. Υπάρχει η δυνατότητα να γίνουν πολλές αλλαγές μέσω του Unity όσον αφορά τον σχεδιασμό του παιχνιδιού, όπως για παράδειγμα να προστεθούν περισσότερα levels.

Η πιο κοινή εισήγηση που έγινε ήταν για τις εικόνες των φαγητών. Ο λόγος που δεν έχει υλοποιηθεί ακόμη η συγκεκριμένη εισήγηση είναι λόγω του ότι πρέπει να βρούμε καινούργια assets ή και ακόμα να βρεθούν εικόνες από το διαδίκτυο και να γίνει επεξεργασία κάθε εικόνας έτσι ώστε το background να γίνει transparent και να αφαιρεθούν τυχόν επιπρόσθετα αντικείμενα που μπορεί να περιέχει η εικόνα. Είναι αρκετά χρονοβόρο να βρούμε ή να δημιουργήσουμε δικά μας πιάτα, για αυτό και η εισήγηση αυτή μπορεί να γίνει σε μεταγενέστερο στάδιο.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Εργασία

6.1 Συμπεράσματα	56
6.2 Μελλοντική Εργασία	57

6.1 Συμπεράσματα

Το παιχνίδι έχει πετύχει σε μέγιστο βαθμό τον σκοπό του, αφού όλες οι προ απαιτούμενες απαιτήσεις του έχουν ολοκληρωθεί με επιτυχία, όμως υπάρχουν ακόμη περιθώρια βελτίωσης όπως μια επέκταση του παιχνιδιού και τους κώδικες που έχουν υλοποιηθεί.

Τα φαγητά έχουν στοιχηθεί ομοιόμορφα σε όλα τα levels του παιχνιδιού στο αριστερό μέρος της οθόνης και αντίστοιχα το τραπέζι στα δεξιά με σκοπό κάθε level του παιχνιδιού να έχει ίδια δομή. Όσον αφορά τους κώδικες που έχουν δημιουργηθεί για την λειτουργικότητα του παιχνιδιού, είναι σωστά δομημένοι και έχουν χρησιμοποιηθεί επεξηγηματικά ονόματα για τους κώδικες, μεταβλητές και τα functions. Για παράδειγμα, ο κώδικας για το τελευταίο level ονομάζεται FoodSunday.cs.

Ακόμη, έχουν δοθεί επεξηγηματικά ονόματα στα αντικείμενα, τις εικόνες και τα κουμπιά έτσι ώστε ο οποιοσδήποτε που θα παραλάβει το project με σκοπό την βελτίωση του να μπορέσει με ευκολία να αναγνωρίσει ποιο αντικείμενο είναι το καθένα. Τα scenes για τα levels έχουν εξίσου απλά και επεξηγηματικά ονόματα.

Το παιχνίδι δεν χρησιμοποιεί περίπλοκες λειτουργίες γιατί σκοπός ήταν όταν τα παιδιά έρθουν σε επαφή με το παιχνίδι, να μπορούν να ξεκινήσουν εύκολα χωρίς να δυσκολευτούν. Επιπρόσθετα, το πιο σημαντικό κριτήριο για την υλοποίηση και τον σχεδιασμό του παιχνιδιού ήταν το παιχνίδι να είναι διασκεδαστικό αλλά ταυτόχρονα να

μαθαίνει στα παιδιά για την Μεσογειακή διατροφή έτσι ώστε να μάθουν να δημιουργούν υγιεινά και ισορροπημένα γεύματα.

6.2 Μελλοντική Εργασία

Αν και το παιχνίδι εξυπηρετεί τον σκοπό του, μπορούν να προστεθούν περισσότερα features που να το κάνουν καλύτερο και πιο ελκυστικό για τα παιδιά. Οι πιο κάτω ιδέες μπορούν να υλοποιηθούν αργότερα:

1. Transition για κάθε level: Το κάθε level αντιπροσωπεύει μια ημέρα, όπως για παράδειγμα το πρώτο level είναι η Δευτέρα. Θα μπορούσε, αφού επιλεγεί το συγκεκριμένο level, να υπάρχει κάποιο transition που να παρουσιάζει πως έχει ξημερώσει και είναι Δευτέρα. Είναι ένα καλό feature που βοηθά τους χρήστες να γνωρίζουν σε ποιο σημείο βρίσκονται.
2. Transition για reward: Σε αρκετά παιχνίδια, όταν ο χρήστης κερδίσει κάποιο reward, όπως είναι οι χαρακτήρες, εμφανίζεται σαν pop-up window ότι ο χρήστης έχει κερδίσει ένα συγκεκριμένο βραβείο. Αυτό γίνεται για ενημέρωση του χρήστη, έτσι ώστε να γνωρίζει ότι έχει κερδίσει το συγκεκριμένο αντικείμενο και πως μπορεί πλέον να το χρησιμοποιήσει.
3. Μουσική: Για κάθε level μπορεί να παίζει μουσική, την οποία μπορεί να δυναμώσει και να χαμηλώσει ο χρήστης με ένα slide bar.
4. Περισσότεροι χαρακτήρες από διαφορετικές κατηγορίες για rewards: Το παιχνίδι ήδη περιέχει 10 χαρακτήρες, όπου οι περισσότεροι είναι φρούτα. Μπορούν να δημιουργηθούν περισσότεροι χαρακτήρες, οι οποίοι θα καλύπτουν όλες τις κατηγορίες φαγητών. Για παράδειγμα, ένα χαρακτήρα που θα αντιπροσωπεύει το κόκκινο κρέας.
5. Tutorial: Το tutorial του παιχνιδιού, είναι ένα video με οπτικοακουστικό περιεχόμενο. Μια καλή ιδέα, είναι να μπορεί ο χρήστης να παίζει ένα level δοκιμαστικά, χωρίς να επηρεάζεται το highscore του.
6. Αντικατάσταση εικονιδίων του χ και ν: Τα εικονίδια του χ και ν μπορούν να αντικατασταθούν με φατσούλες οι οποίες θα προσδιορίζουν στα παιδιά την χρήση τους.

7. Πρόσθετο ακουστικό βοήθημα: Λόγω της ιδιαιτερότητας της ηλικίας των χρηστών, ένα ακουστικό βοήθημα θα μπορέσει να απλοποιήσει την πλοήγηση των παιδιών στο παιχνίδι.
8. Background: Ένα πιο απλό background χωρίς λεπτομέρειες το οποίο δεν θα αποσπά την προσοχή του παιδιού ούτως ώστε να εστιάσει όλη την προσοχή του στην ολοκλήρωση του παιχνιδιού.
9. Video εισαγωγής για σωστό προσδιορισμό ποσότητας φαγητών: Να προστεθεί ένα video εισαγωγής με στόχο να εξηγεί στα παιδιά ποιες είναι οι ιδανικές ποσότητες φαγητών πάντα σύμφωνα με τις προδιαγραφές της μεσογειακής διατροφής.
10. Εικόνες φαγητών: Οι τροφές που είναι απαραίτητο να καταναλώνονται περισσότερο κατά την διάρκεια μιας εβδομάδας να εμφανίζονται συχνότερα στις επιλογές του παιχνιδιού.

Βιβλιογραφία

- [1] Annetta, Leonard A. "The "I's" have it: A framework for serious educational game design." Review of general psychology 14.2, pp. 105-113, 2010.
- [2] Atsusi Hirumi, Bob Appelman, Lloyd Rieber, Richard Van Eck. "Preparing instructional designers for game-based learning: Part 1.", pp. 27-37, 2010.
- [3] Fisher, Carla, "Designing games for children: Developmental, usability, and design considerations for making games for kids." Routledge, 2014.
- [4] Funday Factory, "Game Design for Kids: A paper on how to design age-appropriate game experiences for kids of all ages", [Online]. Available: [age-appropriate-game-design-for-children.pdf \(fundayfactory.com\)](http://fundayfactory.com/age-appropriate-game-design-for-children.pdf) .
- [5] Høiseth, Marikken, Michail N. Giannakos, and Letizia Jaccheri. "derived guidelines for designing toddlers' healthcare games." CHI'13 Extended Abstracts on Human Factors in Computing Systems., pp. 451-456, 2013.
- [6] Janarthanan, Vasudevan., "Serious video games: Games for education and health." In 2012 Ninth International Conference on Information Technology-New Generations, pp. 875-878. IEEE, 2012.
- [7] Kato, Pamela M., "Video games in health care: Closing the gap." Review of general psychology, vol 14, no. 2 : pp. 113-121, 2010.
- [8] Ni, Qi, and Yonghai Yu. "Research on Educational Mobile Games and the effect it has on the Cognitive Development of Preschool Children." 2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC). IEEE, pp. 165-169, 2015.

- [9] Rustina, Y., Krianto, T. and Ayubi, D., "Developing a health education game for preschoolers: What should we consider?." *Enfermería Clínica*, 28, pp.1-4, 2018.
- [10] SensorTower, Top Grossing Game Apps, Cyprus, "Top App Store Rankings for Android", [Online]. Available: <https://app.sensortower.com/android/rankings/top/mobile/cyprus/game?date=2022-07-07>
- [11] Sifa, Rafet, Fabian Hadiji, Julian Runge, Anders Drachen, Kristian Kersting, and Christian Bauckhage. "Predicting purchase decisions in mobile free-to-play games." In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 11, no. 1, pp. 79-85. 2015.
- [12] Squire, Kurt. "Video games in education." *Int. J. Intell. Games & Simulation* 2.1, pp. 49-62, 2005.
- [13] Statcounter Global Stats., "Mobile Operating System Market Share Worldwide", [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide> .
- [14] Statcounter Global Stats., "Mobile Operating System Market Share Cyprus", [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/cyprus>.
- [15] Sudarmilah, Endah, Adhi Susanto, Ridi Ferdiana, and Neila Ramdhani. "Developing a game for preschoolers: What character, emotion and reward will tend to hack preschoolers?." In *2015 International Conference on Data and Software Engineering (ICoDSE)*, pp. 89-92. IEEE, 2015.

- [16] Vishak Vishnu, “11 Best Fun And Educational Android Games For Kids In 2022“, [Online]. Available: <https://techlog360.com/educational-android-games-for-kids/> .
- [17] Wikipedia, Bronkie the Bronchiasaurus, [Online]. Available: [https://en.wikipedia.org/wiki/Bronkie the Bronchiasaurus](https://en.wikipedia.org/wiki/Bronkie_the_Bronchiasaurus) .
- [18] Yuan, Si-Si, et al. "The Design of Buttons in MMO Mobile Game's Battle Interface." International Conference on Human-Computer Interaction. Springer, Cham, 2021.
- [19] Κουρτίδου Φιρούζα, “Μεσογειακή Διατροφή,” [Online]. Available: <https://www.healthline.gr/μεσογειακή-διατροφή-2/> .

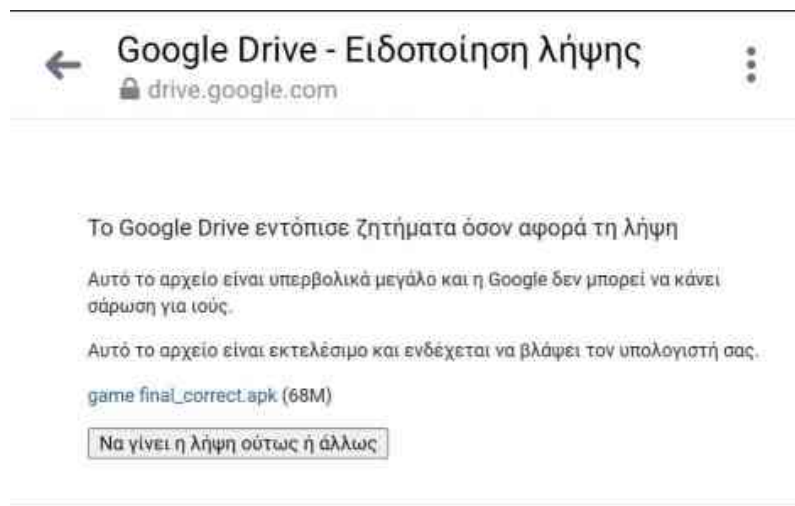
Παράρτημα Α – Οδηγίες Εγκατάστασης

Το παιχνίδι είναι διαθέσιμο σε αυτό τον σύνδεσμο:

https://drive.google.com/drive/folders/1Urvpl0CxSKTMETyd-wmSoEW_NN3Z5Lla?usp=share_link

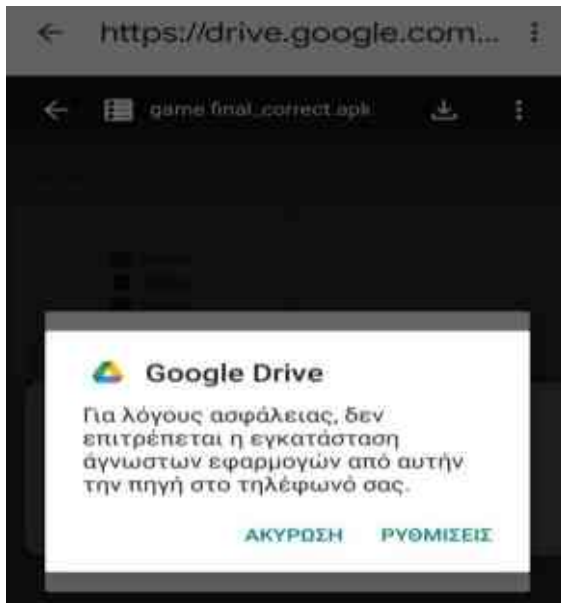
Όταν ανοίξουμε τον πιο πάνω σύνδεσμο, θα μεταβούμε σε ένα φάκελο με το όνομα “ΑΔΕ Παιχνίδι” στο One Drive. Το αρχείο “game final_correct.apk” είναι το παιχνίδι. Η εγκατάσταση μπορεί να γίνει μόνο σε Android συσκευές.

Όταν πατήσουμε στο αρχείο “game final_correct.apk” θα μας εμφανιστεί το εξής μήνυμα:

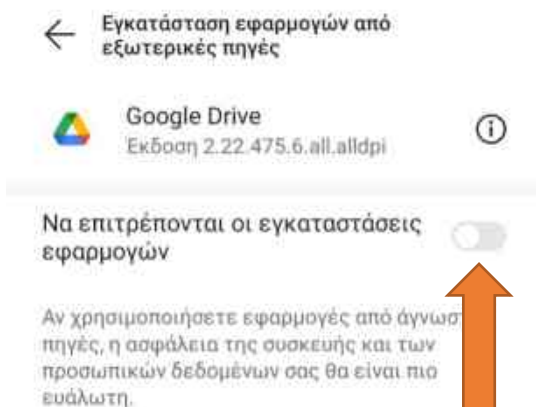


Ο λόγος που εμφανίζεται αυτό το μήνυμα, είναι γιατί το αρχείο είναι μεγάλο και δεν μπορεί να γίνει σάρωση για ιούς. Το αρχείο δεν περιέχει ιούς ή διαφημίσεις και είναι ασφαλές για την συσκευή μας. Όταν εμφανιστεί αυτό το μήνυμα, πατήστε το κουμπί «Να γίνει η λήψη ούτως ή άλλως».

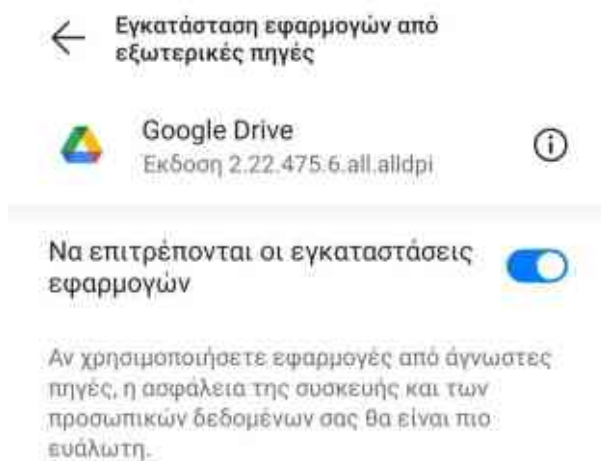
Έπειτα, θα εμφανιστεί ξανά το εξής μήνυμα και θα πρέπει να πατήσουμε το κουμπί “Ρυθμίσεις” για να επιτρέψουμε την εγκατάσταση της εφαρμογής.



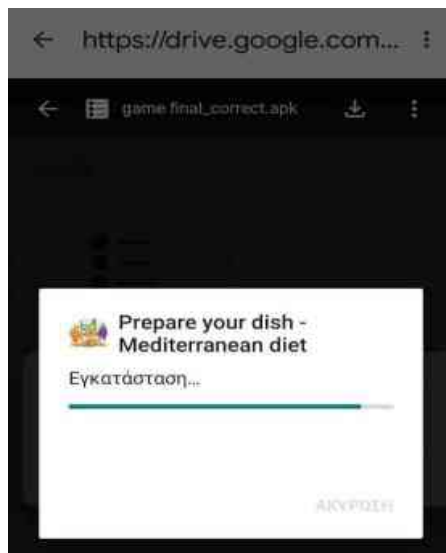
Αφού πατήσουμε το κουμπί “Ρυθμίσεις”, θα πατήσουμε μια φορά το κουμπί που φαίνεται στην εικόνα για να επιτρέπονται οι εγκαταστάσεις εφαρμογών.



Όταν ενεργοποιήσουμε την ρύθμιση, το κουμπί θα έχει την εξής μορφή:



Αφού ενεργοποιήσουμε την ρύθμιση, επιστρέφουμε πίσω και επιλέγουμε το κουμπί “Εγκατάσταση” έτσι ώστε να ξεκινήσει η εγκατάσταση της εφαρμογής στην συσκευή.



Παράρτημα Β – Οδηγίες Χρήσης

Το παιχνίδι ξεκινάει από το αρχικό μενού. Το μενού έχει την εξής μορφή:



1. Το βελάκι 1 δείχνει το κουμπί “Play” το οποίο μας οδηγεί στον χάρτη με τα levels.
2. Το βελάκι 2 δείχνει το κουμπί “Επιλογή χαρακτήρα”.
3. Το βελάκι 3 δείχνει το κουμπί “Ήχος” το οποίο είναι είτε για να ακούγεται ο ήχος των κουμπιών, είτε για να μην ακούγεται.
4. Το βελάκι 4 δείχνει το κουμπί “Tutorial”. Το συγκεκριμένο κουμπί εμφανίζει ένα βίντεο που εξηγεί πως παίζεται το παιχνίδι.
5. Το βελάκι 5 δείχνει το κουμπί “Εξοδος” το οποίο είναι για να κλείσει η εφαρμογή.

Χάρτης με τα levels:



Αρχικά όλα τα στάδια είναι κλειδωμένα εκτός από το πρώτο, και μπορούν να ξεκλειδωθούν για συγκεκριμένα highscores. Για παράδειγμα, το στάδιο 2 θα ξεκλειδωθεί όταν έχουμε score μεγαλύτερο ή ίσο του 10 στο στάδιο 1.

Ταυτόχρονα ξεκλειδώνονται και χαρακτήρες με τους οποίους μπορούμε να επιλέξουμε να παίξουμε. Αρχικά είναι κλειδωμένοι όλοι οι χαρακτήρες εκτός από το μήλο.



Οι χαρακτήρες που είναι κλειδωμένοι θα εμφανίζονται με την εξής μορφή και δεν θα μπορούν να επιλεγθούν:



Όταν ξεκλειδωθεί κάποιος χαρακτήρας, όπως για παράδειγμα το πορτοκάλι, θα φαίνεται όπως απεικονίζεται στην εικόνα:



Με τα βελάκια που φαίνονται δεξιά και αριστερά στο κάτω μέρος της οθόνης, μπορούμε να δούμε τους χαρακτήρες. Πατώντας το κουμπί ‘Play’ που φαίνεται κάτω από τον χαρακτήρα, επιλέγουμε τον συγκεκριμένο χαρακτήρα για να παίξουμε και μεταβαίνουμε στον χάρτη με τα levels για να επιλέξουμε επίπεδο.

Πως παίζεται το παιχνίδι:

Το παιχνίδι περιέχει 7 levels, ένα για κάθε ημέρα της εβδομάδας. Το κάθε level έχει 3 στάδια, το Πρωινό, το Μεσημεριανό και το Βραδινό. Σκοπός είναι να ετοιμάσουμε το τραπέζι με τα φαγητά που επιλέξαμε.

Όπως φαίνεται και στην φωτογραφία, για κάθε στάδιο, έχουμε στην διάθεση μας 10 φαγητά. Μπορούμε να επιλέξουμε όσα φαγητά θέλουμε. Όταν επιλέξουμε κάποιο φαγητό που βρίσκεται στα αριστερά της οθόνης, το φαγητό θα εμφανιστεί στο τραπέζι.



Για κάθε επιλογή μας, βλέπουμε το score να αυξάνεται ή να μειώνεται. Κάθε φορά που το score αυξάνεται θα εμφανίζεται με πράσινο χρώμα και το σύμβολο +, κατά πόσο αυξήθηκε το score, έτσι ώστε τα παιδιά να καταλαβαίνουν κατά πόσο η επιλογή τους ήταν σωστή ή όχι.



Σε περίπτωση που μειωθεί το score, θα εμφανίζεται με κόκκινο χρώμα και το σύμβολο ‘-’, κατά πόσο μειώθηκε το score:



Αν θέλουμε να φύγουμε κάποιο φαγητό από το τραπέζι, μπορούμε να πατήσουμε στο φαγητό. Αν, για παράδειγμα, έχουμε επιλέξει το ποτήρι με το νερό, το νερό έχει τοποθετηθεί στο τραπέζι και έχουμε κερδίσει 5 πόντους.



Σε περίπτωση που τελικά δεν θέλουμε το ποτήρι με το νερό, πατώντας πάνω στο νερό το επιστρέφουμε στα αριστερά της οθόνης και χάνουμε τους πόντους που κερδίσαμε.



Για να προχωρήσουμε στο επόμενο στάδιο πρέπει να πατήσουμε το πράσινο κουμπί που βρίσκεται κάτω δεξιά στην οθόνη.

Σε περίπτωση που πατήσουμε το κουμπί “Παύση”, το οποίο εμφανίζεται σε όλα τα στάδια στο πάνω δεξιά μέρος στην οθόνη, έχουμε 2 επιλογές:



1. Το βελάκι 1 δείχνει το κουμπί “Συνέχεια”, το οποίο όταν το πατήσουμε μπορούμε να συνεχίσουμε το παιχνίδι.
2. Το βελάκι 2 δείχνει το κουμπί “Μενού”, το οποίο μας μεταφέρει στο αρχικό μενού. Σε περίπτωση που πατήσουμε αυτό το κουμπί για να βγούμε από το συγκεκριμένο στάδιο, δεν θα γίνει αποθήκευση των πόντων που κερδίσαμε.

Παράρτημα Γ – Κώδικες – Τεχνικός Οδηγός

Κώδικας Manager.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using UnityEngine.Audio;

public class Manager : MonoBehaviour
{
    public GameObject Button;
    public GameObject Button_2;

    // Start is called before the first frame update
    void Start()
    {

    }

    //load the educational video scene
    public void LoadSceneVideo()
    {
        SceneManager.LoadScene(11);
    }

    //load level 1
    public void LoadSceneMonday(string sceneName)
    {
        SceneManager.LoadScene(2);
    }

    //load level 2
    public void LoadSceneTuesday(string sceneName)
    {
        SceneManager.LoadScene(3);
    }

    //load level 3
    public void LoadSceneWednesday(string sceneName)
    {
        SceneManager.LoadScene(4);
    }

    //load level 4
    public void LoadSceneThursday(string sceneName)
    {
        SceneManager.LoadScene(5);
    }

    //load level 5
    public void LoadSceneFriday(string sceneName)
    {
```

```

        SceneManager.LoadScene(6);
    }

    //load level 6
    public void LoadSceneSaturday(string sceneName)
    {
        SceneManager.LoadScene(7);
    }

    //load level 7
    public void LoadSceneSunday(string sceneName)
    {
        SceneManager.LoadScene(8);
    }

    //load level map
    public void LoadSceneLevelMap(string sceneName)
    {
        SceneManager.LoadScene(1);
    }

    //load home menu
    public void LoadSceneHome(string sceneName)
    {
        SceneManager.LoadScene(0);
    }

    //load tutorial video scene
    public void LoadSceneTutorial()
    {
        SceneManager.LoadScene(10);
    }

    //load character selection scene
    public void LoadSceneCharacterSelection()
    {
        SceneManager.LoadScene(9);
    }
}

```

Πότε καλείται η κάθε function:

Ο κώδικας αυτός είναι υπεύθυνος για την πλοήγηση σε όλα τα scenes του παιχνιδιού. Κάθε scene έχει ένα αριθμό που το αντιπροσωπεύει. Η συνάρτηση LoadSceneVideo() καλείτε όταν πατήσουμε το κουμπί του level 1 που υπάρχει στον χάρτη με τα levels και μας οδηγεί στο εκπαιδευτικό video που αφορά την Μεσογειακή διατροφή. Η συνάρτηση LoadSceneMonday() καλείτε όταν πατήσουμε το κουμπί Play που υπάρχει στο εισαγωγικό βίντεο και μας μεταφέρει στο πρώτο level. Η συνάρτηση LoadSceneTuesday() καλείτε όταν πατήσουμε το κουμπί για το level 2. Η συνάρτηση LoadSceneWednesday() καλείτε όταν πατήσουμε το κουμπί για το level 3. Η συνάρτηση LoadSceneThursday() καλείτε όταν πατήσουμε το κουμπί για το level 4. Η συνάρτηση

LoadSceneFriday() καλείτε όταν πατήσουμε το κουμπί για το level 5. Η συνάρτηση LoadSceneSaturday() καλείτε όταν πατήσουμε το κουμπί για το level 6. Η συνάρτηση LoadSceneSunday() καλείτε όταν πατήσουμε το κουμπί για το level 7. Η συνάρτηση LoadSceneLevelMap() καλείτε όταν πατήσουμε το κουμπί Play το οποίο υπάρχει στο αρχικό μενού και μας μεταφέρει στον χάρτη με τα levels. Η συνάρτηση LoadSceneHome() καλείτε όταν πατήσουμε το κουμπί Home το οποίο βρίσκεται στο scene του Character Selection, στο Pause menu κάθε σταδίου και στον χάρτη με τα levels και είναι υπεύθυνη για να μας μεταφέρει στο scene του αρχικού μενού. Η συνάρτηση LoadSceneTutorial() καλείτε όταν πατήσουμε το κουμπί “?” το οποίο βρίσκεται στο αρχικό μενού και μας μεταφέρει στο scene του tutorial video. Η συνάρτηση LoadSceneCharacterSelection() καλείτε όταν πατήσουμε το κουμπί με τους χαρακτήρες ως εικονίδιο, το οποίο βρίσκεται στο αρχικό μενού, και μας μεταφέρει στο scene επιλογής χαρακτήρων.

Κώδικας PauseManager.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class PauseManager : MonoBehaviour
{
    public GameObject Button;
    public GameObject Button_2;
    //panel of pause menu
    public GameObject Panel;

    //show the pause menu
    public void ShowPause()
    {
        Panel.SetActive(true);
    }

    //hide the pause menu
    public void HidePause()
    {
        Panel.SetActive(false);
    }

    //go to home menu
    public void LoadSceneHome(string sceneName)
    {
```

```

        SceneManager.LoadScene(0);
    }
}

```

Πότε καλείται η κάθε function:

Σε κάθε στάδιο, στο πάνω δεξί μέρος της οθόνης υπάρχει το κουμπί “Pause” το οποίο όταν το πατήσουμε, καλεί την συνάρτηση ShowPause() έτσι ώστε να μας εμφανίσει το panel του pause. Στο panel του pause, υπάρχουν 2 κουμπιά, το Resume και το Home. Όταν πατήσουμε το κουμπί Resume, καλείτε η συνάρτηση HidePause() ενώ όταν πατήσουμε το κουμπί Home, καλείτε η συνάρτηση LoadSceneHome().

Κώδικας ButtonSound.cs :

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Audio;

public class ButtonSound : MonoBehaviour
{
    //state variable:state of the button sound (from PlayerPrefs)
    public static int state;
    //button: sound of the button
    public AudioSource button;
    //soundVolume: volume of the sound
    public float soundVolume = 1f;

    //function Play: get the state from PlayerPrefs. If its 1 then the sound is
    on and
    //if its 0, the sound is off.
    public void Play()
    {
        state = PlayerPrefs.GetInt("state");
        if (state == 1)
        {
            button.volume = soundVolume;
            button.Play();
            state = 1;
        }
        else
        {
            state = 0;
        }
    }

    // Start is called before the first frame update
    void Start()
    {

```

```

    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Play() καλείτε από όλα τα κουμπιά του παιχνιδιού έτσι ώστε να ακούγεται ήχος όταν τα πατήσουμε. Η συγκεκριμένη συνάρτηση ελέγχει αν το state του ήχου είναι 1 ή 0. Σε περίπτωση που είναι 1 ο ήχος είναι on ενώ αν είναι 0, ο ήχος είναι off και δεν ακούγεται.

Κώδικας ChangeSound.cs :

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ChangeSound : MonoBehaviour {

    //state variable: state of volume
    public static int state;

    //function Change: change the state of the buttons' sound
    public void Change()
    {
        //get the variable state from PlayerPrefs
        state=PlayerPrefs.GetInt("state");
        //if its 1 then the sound is on and we want to turn it off
        if (state == 1)
        {
            //set the variable to 0
            PlayerPrefs.SetInt("state",0);
        }
        //if its 0 then the sound is off and we want to turn it on
        else
        {
            PlayerPrefs.SetInt("state", 1);
        }
    }

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()

```

```

    {
    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Change() καλείτε από το κουμπί Volume το οποίο βρίσκεται στο αρχικό μενού και στον χάρτη με τα levels και είναι υπεύθυνο για το αν θα ακούγεται ο ήχος των κουμπιών. Η συνάρτηση είναι υπεύθυνη για να αλλάζουμε το state του ήχου. Αν ο ήχος είναι on, γίνεται off.

Κώδικας VolumeButtonsLevelMap:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class VolumeButtonsLevelMap : MonoBehaviour
{
    //soundOn button
    public GameObject soundOn;
    //soundOff button
    public GameObject soundOff;
    //state of volume
    public static int state;
    //Change Buttons : show the correct button for the volume.
    public void ChangeButtons()
    {
        //get the state from PlayerPrefs
        state = PlayerPrefs.GetInt("state");
        //if the state is 1, show the soundOn button and hide the soundOff button
        if (state == 1)
        {
            soundOn.SetActive(true);
            soundOff.SetActive(false);
        }
        //if the state is 0, show the soundOff button and hide the soundOn button
        else
        {
            soundOn.SetActive(false);
            soundOff.SetActive(true);
        }
    }

    // Start is called before the first frame update
    void Start()
    {
        //when we enter the scene, we need to know what is the state of the sound
        //because we want the correct button to show
        state = PlayerPrefs.GetInt("state");
        if (state == 1)
        {

```

```

        soundOn.SetActive(true);
        soundOff.SetActive(false);
    }
    else
    {
        soundOn.SetActive(false);
        soundOff.SetActive(true);
    }
}

// Update is called once per frame
void Update()
{
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση ChangeButtons() καλείτε όταν πατηθεί το κουμπί για το Volume το οποίο βρίσκεται στον χάρτη με τα levels και σκοπός της είναι να εμφανίζει το σωστό κουμπί του volume, δηλαδή αν ο ήχος είναι off ή on, ελέγχοντας το state του ήχου. Η συνάρτηση Start() καλείτε μια φορά από μόνη της, όταν ξεκινήσει το scene του χάρτη με τα levels. Η συγκεκριμένη συνάρτηση ελέγχει και αυτή το state του ήχου για να εμφανίσει το σωστό κουμπί κατά την είσοδο μας στο scene.

Κώδικας StartButtonSound.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StartButtonSound : MonoBehaviour
{
    //button soundOn
    public GameObject soundOn;

    // Start is called before the first frame update
    void Start()
    {
        //if its active, the sound is on
        if (soundOn.activeSelf)
        {
            PlayerPrefs.SetInt("state", 1);
        }
        //if the button is not active, the sound is off
        else
        {
            PlayerPrefs.SetInt("state", 0);
        }
    }
}

```

```

    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείται μια φορά από μόνη της όταν εισέλθουμε σε κάποιο scene. Σκοπός της συνάρτησης είναι να ελέξει αν το κουμπί που δείχνει απεικονίζει πως ο ήχος είναι on έτσι ώστε να κάνει set το variable state του ήχου.

Κώδικας SelectedCharacterForLevels.cs :

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class SelectedCharacterForLevels : MonoBehaviour
{
    //character apple
    public GameObject Apple;
    //character orange
    public GameObject Orange;
    //character avocado
    public GameObject Avocado;
    //character egg
    public GameObject Egg;
    //character banana
    public GameObject Banana;
    //character cheese
    public GameObject Cheese;
    //character grape
    public GameObject Grape;
    //character olive oil
    public GameObject OliveOil;
    //character icecream
    public GameObject IceCream;
    //character nutella
    public GameObject Nutella;
    //The parentObject that has all characters
    public GameObject AllCharacters;
    //flag integer value for characters
    public static int Character;

    // size and position of character at level 1 for dinner
    public void dinnerMondayScaleTransform()
    {
        AllCharacters.transform.Translate(900, 200, 595);
        AllCharacters.transform.localScale += new Vector3(87.1f, 76f, 0f);
    }
}

```

```

}

//size and position of character for all the dinners
public void dinnerScaleTransform()
{
    AllCharacters.transform.Translate(-200, 100, 595);
    AllCharacters.transform.localScale += new Vector3(40f,40f,0f);
}

//size and position of character for all the lunches
public void lunchScaleTransform()
{
    AllCharacters.transform.Translate(1050f, 0.2f, 0f);
    AllCharacters.transform.localScale += new Vector3(40f, 20f, 0f);
}

// Start is called before the first frame update
//Show the selected character
void Start()
{
    Character=PlayerPrefs.GetInt("Character");
    //Apple selected
    if (Character == 1)
    {
        Apple.SetActive(true);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Orange selected
    else if (Character == 2)
    {
        Apple.SetActive(false);
        Orange.SetActive(true);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Avocado selected
    else if (Character == 3)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(true);
        Egg.SetActive(false);
    }
}

```

```

        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Banana selected
    else if (Character == 4)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(true);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Cheese selected
    else if (Character == 5)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(true);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Egg selected
    else if (Character == 6)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(true);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Grape selected
    else if (Character == 7)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(true);
    }

```

```

        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
    //Icecream selected
    else if (Character == 8)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(true);
        Nutella.SetActive(false);
    }
    //Nutella selected
    else if (Character == 9)
    {
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(true);
    }
    //olive oil selected
    else if (Character == 10)
    {
        AllCharacters.transform.Translate(-0.2f, -0.2f, 0f);
        Apple.SetActive(false);
        Orange.SetActive(false);
        Avocado.SetActive(false);
        Egg.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Grape.SetActive(false);
        OliveOil.SetActive(true);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
    }
}

// Update is called once per frame
void Update()
{
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση dinnerMondayScaleTransform() καλείτε από το πράσινο κουμπί που βρίσκεται στο Μεσημεριανό στο level 1. Η συνάρτηση dinnerScaleTransform() καλείτε από το πράσινο κουμπί που βρίσκεται στο Μεσημεριανό όλων των σταδίων εκτός από του level 1. Η συνάρτηση lunchScaleTransform() καλείτε από το πράσινο κουμπί το οποίο βρίσκεται στο Πρωϊνό όλων των levels. Αυτές οι συναρτήσεις αλλάζουν το μέγεθος και την θέση του χαρακτήρα για κάθε στάδιο έτσι ώστε ο χαρακτήρας να εμφανίζει στην σωστή θέση σε κάθε canvas. Η συνάρτηση Start() καλείτε μια φορά από μόνη της όταν εισέλθουμε στα scenes των levels και σκοπός της είναι να εμφανίζει στα levels, μόνο το χαρακτήρα που έχει επιλεγεί από Character Selection scene.

Κώδικας NextMeal.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class NextMeal : MonoBehaviour
{
    public GameObject Breakfast;
    public GameObject Lunch;
    public GameObject Dinner;

    //show canvas lunch
    public void Next_meal()
    {
        Breakfast.SetActive(false);
        Lunch.SetActive(true);
        Dinner.SetActive(false);
    }

    //show canvas dinner
    public void DinnerMeal()
    {
        Breakfast.SetActive(false);
        Lunch.SetActive(false);
        Dinner.SetActive(true);
    }

    //complete level and go to the level map
    public void LoadSceneLevelMap()
    {
        SceneManager.LoadScene(1);
    }

    // Start is called before the first frame update
    void Start()
    {
        Breakfast.SetActive(true);
    }
}
```

```

        Lunch.SetActive(false);
        Dinner.SetActive(false);
    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Next_meal() καλείτε όταν πατήσουμε το πράσινο κουμπί που βρίσκεται στο Πρωινό όλων των levels για να εμφανίσει το Μεσημεριανό. Η συνάρτηση DinnerMeal() καλείτε όταν πατήσουμε το πράσινο κουμπί που βρίσκεται στο Βραδινό όλων των levels για να εμφανίσει το Βραδινό. Η συνάρτηση LoadSceneLevelMap() καλείτε όταν πατήσουμε το πράσινο κουμπί στο Βραδινό όλων των levels και μας μεταφέρει στον χάρτη με τα levels, και η συνάρτηση Start() καλείτε μια φορά από μόνη της όταν εισέλθουμε σε οποιοδήποτε level. Η συγκεκριμένη συνάρτηση επειδή καλείτε κατά την είσοδο μας στο scene, κρύβει το canvas του Μεσημεριανού και του Βραδινού και εμφανίζει μόνο το canvas του Πρωϊνού.

Κώδικας ExitManager.cs :

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ExitManager : MonoBehaviour
{
    // exit from the game
    public void Exit(){
        Application.Quit();
    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Exit() καλείτε από το κουμπί Exit το οποίο βρίσκεται στο αρχικό μενού και οδηγεί στην έξοδο του παιχνιδιού.

Κώδικας GetHighscore.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class GetHighscore : MonoBehaviour
{
    public int highscore;
    public int level1;
    public int level2;
    public int level3;
    public int level4;
    public int level5;
    public int level6;

    public int flag1;
    public int flag2;
    public int flag3;
    public int flag4;
    public int flag5;
    public int flag6;
    public int flag7;

    public TextMeshProUGUI HighscoreText;

    public GameObject Lock_level2;
    public GameObject ButtonLevel2;
    public GameObject Lock_level3;
    public GameObject ButtonLevel3;
    public GameObject Lock_level4;
    public GameObject ButtonLevel4;
    public GameObject Lock_level5;
    public GameObject ButtonLevel5;
    public GameObject Lock_level6;
    public GameObject ButtonLevel6;
    public GameObject Lock_level7;
    public GameObject ButtonLevel7;

    public static int unlock2 = 0;
    public static int unlock3 = 0;
    public static int unlock4 = 0;
    public static int unlock5 = 0;
    public static int unlock6 = 0;
    public static int unlock7 = 0;

    // Start is called before the first frame update
    //unlock levels (based the highscore) and show the highscore at the top of
    //the screen
    void Start()
    {
        highscore = PlayerPrefs.GetInt("HighScore");
        HighscoreText.text = highscore.ToString();

        level1 = PlayerPrefs.GetInt("Monday score");
        level2 = PlayerPrefs.GetInt("Tuesday score");
        level3 = PlayerPrefs.GetInt("Wednesday score");
        level4 = PlayerPrefs.GetInt("Thursday score");
        level5 = PlayerPrefs.GetInt("Friday score");
    }
}
```

```

level6 = PlayerPrefs.GetInt("Saturday score");

flag1= PlayerPrefs.GetInt("flag1");
flag2 = PlayerPrefs.GetInt("flag2");
flag3 = PlayerPrefs.GetInt("flag3");
flag4 = PlayerPrefs.GetInt("flag4");
flag5 = PlayerPrefs.GetInt("flag5");
flag6 = PlayerPrefs.GetInt("flag6");
flag7 = PlayerPrefs.GetInt("flag7");

unlock2 = PlayerPrefs.GetInt("unlock2");
unlock3 = PlayerPrefs.GetInt("unlock3");
unlock4 = PlayerPrefs.GetInt("unlock4");
unlock5 = PlayerPrefs.GetInt("unlock5");
unlock6 = PlayerPrefs.GetInt("unlock6");
unlock7 = PlayerPrefs.GetInt("unlock7");

if (unlock2 == 1)
{
    //show level 2
    Lock_level2.SetActive(false);
    ButtonLevel2.SetActive(true);
}
if (unlock3==1)
{
    //show level 3
    Lock_level3.SetActive(false);
    ButtonLevel3.SetActive(true);
}
if (unlock4==1)
{
    //show level 4
    Lock_level4.SetActive(false);
    ButtonLevel4.SetActive(true);
}
if (unlock5 == 1)
{
    //show level 5
    Lock_level5.SetActive(false);
    ButtonLevel5.SetActive(true);
}
if (unlock6 == 1)
{
    //show level 6
    Lock_level6.SetActive(false);
    ButtonLevel6.SetActive(true);
}
if (unlock7 == 1)
{
    //show level 7
    Lock_level7.SetActive(false);
    ButtonLevel7.SetActive(true);
}

if (unlock7 == 0)
{
    if (highscore != 0)

```

```

    {
        if (level1 >= 10 && flag1 == 1 && unlock2==0)
        {
            //show level 2
            Lock_level2.SetActive(false);
            ButtonLevel2.SetActive(true);
            PlayerPrefs.SetInt("unlock2",1);
        }
        if (level2 >= 10 && flag2 == 1 && unlock3 == 0)
        {
            //show level 3
            Lock_level3.SetActive(false);
            ButtonLevel3.SetActive(true);
            PlayerPrefs.SetInt("unlock3", 1);
        }
        if (level3 >= 10 && flag3 == 1 && unlock4 == 0)
        {
            //show level 4
            Lock_level4.SetActive(false);
            ButtonLevel4.SetActive(true);
            PlayerPrefs.SetInt("unlock4", 1);
        }
        if (level4 >= 10 && flag4 == 1 && unlock5 == 0)
        {
            //show level 5
            Lock_level5.SetActive(false);
            ButtonLevel5.SetActive(true);
            PlayerPrefs.SetInt("unlock5", 1);
        }
        if (level5 >= 10 && flag5 == 1 && unlock6 == 0)
        {
            //show level 6
            Lock_level6.SetActive(false);
            ButtonLevel6.SetActive(true);
            PlayerPrefs.SetInt("unlock6", 1);
        }
        if (level6 >= 10 && flag6 == 1)
        {
            //show level 7
            Lock_level7.SetActive(false);
            ButtonLevel7.SetActive(true);
            PlayerPrefs.SetInt("unlock7", 1);
        }
    }
}

// Update is called once per frame
void Update()
{
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μια φορά από μόνη της όταν εισέλθουμε στο scene του χάρτη με τα levels και σκοπός του είναι να εμφανίσει το highscore και τα levels που ξεκλειδώθηκαν. Σε περίπτωση που το level δεν έχει ξεκλειδωθεί, εμφανίζει το εικονίδιο της κλειδαριάς έτσι ώστε να αποτρέψει τον χρήστη από το να παίζει το συγκεκριμένο level.

Κώδικας ScoreScript.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
using UnityEngine.SceneManagement;

public class ScoreScript : MonoBehaviour
{
    public static int score;
    public static int OldHighscore=0;
    public static int NewHighscore = 0;
    public static int Monday = 0;
    public static int Tuesday = 0;
    public static int Wednesday = 0;
    public static int Thursday = 0;
    public static int Friday = 0;
    public static int Saturday = 0;
    public static int Sunday = 0;
    public static int vegetablesAndFruits=0;
    public static int Redmeat=0;
    public static int dairy=0;
    public static int legumes=0;
    public static int eggs=0;
    public static int sweets=0;
    public static int fish=0;
    public static int chicken=0;
    public static int flag = 0;
    public TextMeshProUGUI scoreText;
```

```

public static int Veget_Fruits_day;
public static int Redmeat_day;
public static int dairy_day;
public static int legumes_day;
public static int eggs_day;
public static int sweets_day;
public static int fish_day;
public static int chicken_day;

```

```

public TextMeshProUGUI txt;
public float moveSpeed = 100.0f;
public float alphaValue = 1.0f;

```

```

// Start is called before the first frame update
//for each level, set the score to 0. Get the previous selections of the current level
//because we want to start the level from the begging.

```

```

void Start()

```

```

{

```

```

    score = 0;

```

```

    Scene currentScene = SceneManager.GetActiveScene();

```

```

    int buildIndex = currentScene.buildIndex;

```

```

    if (buildIndex == 2)

```

```

    {

```

```

        Veget_Fruits_day=PlayerPrefs.GetInt("Veg_Fruit_1");

```

```

        Redmeat_day=PlayerPrefs.GetInt("Redmeat1");

```

```

        dairy_day=PlayerPrefs.GetInt("dairy1");

```

```

        legumes_day=PlayerPrefs.GetInt("legumes1");

```

```

        eggs_day=PlayerPrefs.GetInt("eggs1");

```

```

        sweets_day=PlayerPrefs.GetInt("sweets1");

```

```

        fish_day=PlayerPrefs.GetInt("fish1");

```

```

        chicken_day=PlayerPrefs.GetInt("chicken1");

```

```

    }

```

```

    if (buildIndex == 3)

```

```

    {

```

```

        Veget_Fruits_day = PlayerPrefs.GetInt("Veg_Fruit_2");

```

```

Redmeat_day = PlayerPrefs.GetInt("Redmeat2");
dairy_day = PlayerPrefs.GetInt("dairy2");
legumes_day = PlayerPrefs.GetInt("legumes2");
eggs_day = PlayerPrefs.GetInt("eggs2");
sweets_day = PlayerPrefs.GetInt("sweets2");
fish_day = PlayerPrefs.GetInt("fish2");
chicken_day = PlayerPrefs.GetInt("chicken2");
}
if (buildIndex == 4)
{
    Veget_Fruits_day = PlayerPrefs.GetInt("Veg_Fruit_3");
    Redmeat_day = PlayerPrefs.GetInt("Redmeat3");
    dairy_day = PlayerPrefs.GetInt("dairy3");
    legumes_day = PlayerPrefs.GetInt("legumes3");
    eggs_day = PlayerPrefs.GetInt("eggs3");
    sweets_day = PlayerPrefs.GetInt("sweets3");
    fish_day = PlayerPrefs.GetInt("fish3");
    chicken_day = PlayerPrefs.GetInt("chicken3");

}
if (buildIndex == 5)
{
    Veget_Fruits_day = PlayerPrefs.GetInt("Veg_Fruit_4");
    Redmeat_day = PlayerPrefs.GetInt("Redmeat4");
    dairy_day = PlayerPrefs.GetInt("dairy4");
    legumes_day = PlayerPrefs.GetInt("legumes4");
    eggs_day = PlayerPrefs.GetInt("eggs4");
    sweets_day = PlayerPrefs.GetInt("sweets4");
    fish_day = PlayerPrefs.GetInt("fish4");
    chicken_day = PlayerPrefs.GetInt("chicken4");
}
if (buildIndex == 6)
{
    Veget_Fruits_day = PlayerPrefs.GetInt("Veg_Fruit_5");
    Redmeat_day = PlayerPrefs.GetInt("Redmeat5");
    dairy_day = PlayerPrefs.GetInt("dairy5");
    legumes_day = PlayerPrefs.GetInt("legumes5");
    eggs_day = PlayerPrefs.GetInt("eggs5");
    sweets_day = PlayerPrefs.GetInt("sweets5");
    fish_day = PlayerPrefs.GetInt("fish5");

```

```

        chicken_day = PlayerPrefs.GetInt("chicken5");
    }
    if (buildIndex == 7)
    {
        Veget_Fruits_day = PlayerPrefs.GetInt("Veg_Fruit_6");
        Redmeat_day = PlayerPrefs.GetInt("Redmeat6");
        dairy_day = PlayerPrefs.GetInt("dairy6");
        legumes_day = PlayerPrefs.GetInt("legumes6");
        eggs_day = PlayerPrefs.GetInt("eggs6");
        sweets_day = PlayerPrefs.GetInt("sweets6");
        fish_day = PlayerPrefs.GetInt("fish6");
        chicken_day = PlayerPrefs.GetInt("chicken6");
    }
    if (buildIndex == 8)
    {
        Veget_Fruits_day = PlayerPrefs.GetInt("Veg_Fruit_7");
        Redmeat_day = PlayerPrefs.GetInt("Redmeat7");
        dairy_day = PlayerPrefs.GetInt("dairy7");
        legumes_day = PlayerPrefs.GetInt("legumes7");
        eggs_day = PlayerPrefs.GetInt("eggs7");
        sweets_day = PlayerPrefs.GetInt("sweets7");
        fish_day = PlayerPrefs.GetInt("fish7");
        chicken_day = PlayerPrefs.GetInt("chicken7");
    }
    vegetablesAndFruits = vegetablesAndFruits - Veget_Fruits_day;
    if (vegetablesAndFruits < 0)
    {
        vegetablesAndFruits = 0;
    }
    Redmeat = Redmeat - Redmeat_day;
    if (Redmeat < 0)
    {
        Redmeat = 0;
    }
    dairy = dairy - dairy_day;
    if (dairy < 0)
    {
        dairy = 0;
    }
    legumes = legumes - legumes_day;

```

```

    if (legumes < 0)
    {
        legumes = 0;
    }
    eggs=eggs - eggs_day;
    if (eggs < 0)
    {
        eggs = 0;
    }
    sweets = sweets - sweets_day;
    if (sweets < 0)
    {
        sweets = 0;
    }
    fish = fish - fish_day;
    if (fish < 0)
    {
        fish = 0;
    }
    chicken = chicken - chicken_day;
    if (chicken < 0)
    {
        chicken = 0;
    }

    Veget_Fruits_day =0;
    Redmeat_day=0;
    dairy_day=0;
    legumes_day=0;
    eggs_day=0;
    sweets_day=0;
    fish_day=0;
    chicken_day=0;
    updateScoreUI();
}

```

//add points to the score and call function ShowAnimationScore() to show the feedback on the screen

```

public void addScore(int xAmount)
{

```

```

        score += xAmount;
        updateScoreUI();
        ShowAnimationScore(xAmount,1);
    }

//reduce points from the score and call function ShowAnimationScore() to show the feedback on the screen
public void ReduceScore(int yAmount)
{
    score -= yAmount;
    ShowAnimationScore(yAmount, 0);
    updateScoreUI();
}

//if the score is smaller than 0, set the score variable to 0 and show the score on the screen
void updateScoreUI()
{
    if (score < 0)
    {
        score = 0;
    }
    scoreText.text = score.ToString();
}

//show feedback for the score. If we gain points, the feedback has green color and if we lose points,
//the feedback has red color
void ShowAnimationScore(int xAmount, int add)
{
    if (alphaValue <= 0.0f)
    {
        txt.transform.Translate(new Vector2(0, 0));
        txt.enabled = false;
    }
    else
    {
        //txt.transform.Translate(new Vector2(0, moveSpeed) * Time.deltaTime);

        alphaValue -= Time.deltaTime;
    }
}

```

```

        if (add == 1)
        {
            txt.color = new Color32(0, 255, 0, 255);
            txt.text = "+" + xAmount.ToString();
        }
        else
        {
            txt.color = new Color32(255, 0, 0, 255);
            txt.text = "-" + xAmount.ToString();
        }
    }
}

```

//for each level, save the user's selection with the PlayerPrefs and update highscore

```

public void ScoreWhenDone()
{
    Scene currentScene = SceneManager.GetActiveScene();
    int buildIndex = currentScene.buildIndex;

    if (buildIndex == 2)
    {
        PlayerPrefs.SetInt("Monday score", 0);
        PlayerPrefs.SetInt("Monday score", score);
        PlayerPrefs.SetInt("flag1", 1);
        PlayerPrefs.SetInt("Veg_Fruit_1", Veget_Fruits_day);
        PlayerPrefs.SetInt("Redmeat1", Redmeat_day);
        PlayerPrefs.SetInt("dairy1", dairy_day);
        PlayerPrefs.SetInt("legumes1", legumes_day);
        PlayerPrefs.SetInt("eggs1", eggs_day);
        PlayerPrefs.SetInt("sweets1", sweets_day);
        PlayerPrefs.SetInt("fish1", fish_day);
        PlayerPrefs.SetInt("chicken1", chicken_day);
    }
    if (buildIndex == 3)
    {
        PlayerPrefs.SetInt("Tuesday score", 0);
        PlayerPrefs.SetInt("Tuesday score", score);
    }
}

```

```

PlayerPrefs.SetInt("flag2", 1);
PlayerPrefs.SetInt("Veg_Fruit_2", Veget_Fruits_day);
PlayerPrefs.SetInt("Redmeat2", Redmeat_day);
PlayerPrefs.SetInt("dairy2", dairy_day);
PlayerPrefs.SetInt("legumes2", legumes_day);
PlayerPrefs.SetInt("eggs2", eggs_day);
PlayerPrefs.SetInt("sweets2", sweets_day);
PlayerPrefs.SetInt("fish2", fish_day);
PlayerPrefs.SetInt("chicken2", chicken_day);
}
if (buildIndex == 4)
{
    PlayerPrefs.SetInt("Wednesday score", 0);
    PlayerPrefs.SetInt("Wednesday score", score);
    PlayerPrefs.SetInt("flag3", 1);
    PlayerPrefs.SetInt("Veg_Fruit_3", Veget_Fruits_day);
    PlayerPrefs.SetInt("Redmeat3", Redmeat_day);
    PlayerPrefs.SetInt("dairy3", dairy_day);
    PlayerPrefs.SetInt("legumes3", legumes_day);
    PlayerPrefs.SetInt("eggs3", eggs_day);
    PlayerPrefs.SetInt("sweets3", sweets_day);
    PlayerPrefs.SetInt("fish3", fish_day);
    PlayerPrefs.SetInt("chicken3", chicken_day);

}
if (buildIndex == 5)
{
    PlayerPrefs.SetInt("Thursday score", 0);
    PlayerPrefs.SetInt("Thursday score", score);
    PlayerPrefs.SetInt("flag4", 1);
    PlayerPrefs.SetInt("Veg_Fruit_4", Veget_Fruits_day);
    PlayerPrefs.SetInt("Redmeat4", Redmeat_day);
    PlayerPrefs.SetInt("dairy4", dairy_day);
    PlayerPrefs.SetInt("legumes4", legumes_day);
    PlayerPrefs.SetInt("eggs4", eggs_day);
    PlayerPrefs.SetInt("sweets4", sweets_day);
    PlayerPrefs.SetInt("fish4", fish_day);
    PlayerPrefs.SetInt("chicken4", chicken_day);
}
if (buildIndex == 6)

```

```

{
    PlayerPrefs.SetInt("Friday score", 0);
    PlayerPrefs.SetInt("Friday score", score);
    PlayerPrefs.SetInt("flag5", 1);
    PlayerPrefs.SetInt("Veg_Fruit_5", Veget_Fruits_day);
    PlayerPrefs.SetInt("Redmeat5", Redmeat_day);
    PlayerPrefs.SetInt("dairy5", dairy_day);
    PlayerPrefs.SetInt("legumes5", legumes_day);
    PlayerPrefs.SetInt("eggs5", eggs_day);
    PlayerPrefs.SetInt("sweets5", sweets_day);
    PlayerPrefs.SetInt("fish5", fish_day);
    PlayerPrefs.SetInt("chicken5", chicken_day);
}
if (buildIndex == 7)
{
    PlayerPrefs.SetInt("Saturday score", 0);
    PlayerPrefs.SetInt("Saturday score", score);
    PlayerPrefs.SetInt("flag6", 1);
    PlayerPrefs.SetInt("Veg_Fruit_6", Veget_Fruits_day);
    PlayerPrefs.SetInt("Redmeat6", Redmeat_day);
    PlayerPrefs.SetInt("dairy6", dairy_day);
    PlayerPrefs.SetInt("legumes6", legumes_day);
    PlayerPrefs.SetInt("eggs6", eggs_day);
    PlayerPrefs.SetInt("sweets6", sweets_day);
    PlayerPrefs.SetInt("fish6", fish_day);
    PlayerPrefs.SetInt("chicken6", chicken_day);
}
if (buildIndex==8)
{
    PlayerPrefs.SetInt("Sunday score", 0);
    PlayerPrefs.SetInt("Sunday score", score);
    PlayerPrefs.SetInt("flag7", 1);
    PlayerPrefs.SetInt("Veg_Fruit_7", Veget_Fruits_day);
    PlayerPrefs.SetInt("Redmeat7", Redmeat_day);
    PlayerPrefs.SetInt("dairy7", dairy_day);
    PlayerPrefs.SetInt("legumes7", legumes_day);
    PlayerPrefs.SetInt("eggs7", eggs_day);
    PlayerPrefs.SetInt("sweets7", sweets_day);
    PlayerPrefs.SetInt("fish7", fish_day);
    PlayerPrefs.SetInt("chicken7", chicken_day);
}

```

```

    }
    OldHighscore = PlayerPrefs.GetInt("HighScore");
    Monday= PlayerPrefs.GetInt("Monday score");
    Tuesday = PlayerPrefs.GetInt("Tuesday score");
    Wednesday = PlayerPrefs.GetInt("Wednesday score");
    Thursday = PlayerPrefs.GetInt("Thursday score");
    Friday = PlayerPrefs.GetInt("Friday score");
    Saturday = PlayerPrefs.GetInt("Saturday score");
    Sunday = PlayerPrefs.GetInt("Sunday score");
    NewHighscore = Monday + Tuesday + Wednesday + Thursday + Friday + Saturday + Sunday;

    if (NewHighscore>OldHighscore)
    {
        OldHighscore = NewHighscore;
        PlayerPrefs.SetInt("HighScore", 0);
        PlayerPrefs.SetInt("HighScore", OldHighscore);
        PlayerPrefs.Save();
    }
}

```

//for the vegetables and the fruits we can gain 4 points but only if we don't exceed limit, or else we
//lose 4 points

```

public void VegetablesAndFruits(int xAmount)
{
    xAmount = 4;

    Veget_Fruits_day++;
    vegetablesAndFruits++;
    //5 servings of fruits/vegetables everyday. (week: max 35)
    if (vegetablesAndFruits <=35)
    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

```

//for the red meat we can gain 1 point but only if we don't exceed limit, or else we

```

//lose 1 point
public void meat(int xAmount)
{
    xAmount = 1;
    Redmeat_day++;
    Redmeat++;
    //1 time a week
    if (Redmeat <=1)
    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

//for the dairy we can gain 2 points but only if we don't exceed limit, or else we
//lose 2 points
public void Dairy_(int xAmount)
{
    xAmount = 2;
    dairy_day++;
    dairy++;
    //3-4 servings of dairy everyday (week: max 28)
    if (dairy <=28)
    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

//for the legumes we can gain 3 points but only if we don't exceed limit, or else we
//lose 3 points
public void Legumes_(int xAmount)
{
    xAmount = 3;
    legumes_day++;

```

```

    legumes++;
    //3 times a week
    if (legumes <=3)
    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

//for the eggs we can gain 2 points but only if we don't exceed limit, or else we
//lose 2 points
public void Eggs(int xAmount)
{
    xAmount = 2;
    eggs_day++;
    eggs++;
    //max 7 times a week
    if (eggs<=7)
    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

//for the fish we can gain 2 points but only if we don't exceed limit, or else we
//lose 2 points
public void Fish(int tuna)
{
    int xAmount = 2;

    if (tuna == 1)
    {
        addScore(xAmount);
    }
    else
    {

```

```

        fish_day++;
        fish++;
        //max 2 times a week
        if (fish <= 2)
        {
            addScore(xAmount);
        }
        else
        {
            ReduceScore(xAmount);
        }
    }

}

//for the sweets we can gain 1 point but only if we don't exceed limit, or else we
//lose 1 point
public void sweets_(int xAmount)
{
    xAmount = 1;
    sweets_day++;
    sweets++;
    //max 2 times a week
    if (sweets<=2)
    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

//for the chicken we can gain 2 points but only if we don't exceed limit, or else we
//lose 2 points
public void Chicken_(int xAmount)
{
    xAmount = 2;
    chicken_day++;
    chicken++;
    //max 3 times a week
    if (chicken <=3)

```

```

    {
        addScore(xAmount);
    }
    else
    {
        ReduceScore(xAmount);
    }
}

```

//remove vegetables and fruits from the table: lose/gain points

```

public void ReverseVegetablesAndFruits()
{
    int xAmount = 4;
    Veget_Fruits_day--;
    vegetablesAndFruits--;
    //5 servings of fruits/vegetables everyday. (week: max 35)
    if (vegetablesAndFruits <= 35)
    {
        ReduceScore(xAmount);

    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }

    }
}

```

//remove red meat from the table: lose/gain points

```

public void Reversemeat()
{
    int xAmount = 1;
    Redmeat_day--;
    Redmeat--;
    //1 time a week
    if (Redmeat <= 1)
    {

```

```

        ReduceScore(xAmount);

    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }

    }
}

//remove dairy from the table: lose/gain points
public void ReverseDairy_()
{
    int xAmount = 2;
    dairy_day--;
    dairy--;
    //3-4 servings of dairy everyday (week: max 28)
    if (dairy <= 28)
    {
        ReduceScore(xAmount);
    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }

    }
}

//remove legumes from the table: lose/gain points
public void ReverseLegumes_()
{
    int xAmount = 3;
    legumes_day--;
    legumes--;
    //3 times a week
    if (legumes <= 3)

```

```

    {
        ReduceScore(xAmount);
    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }
    }
}

//remove eggs from the table: lose/gain points
public void ReverseEggs()
{
    int xAmount = 2;
    eggs_day--;
    eggs--;
    //max 7 times a week
    if (eggs <= 7)
    {
        ReduceScore(xAmount);
    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }
    }
}

//remove fish from the table: lose/gain points
public void ReverseFish(int tuna)
{
    int xAmount = 2;
    if (tuna == 1)
    {
        ReduceScore(xAmount);
    }
}

```

```

    }
    else
    {
        fish_day--;
        fish--;
        //max 2 times a week
        if (fish <= 2)
        {
            ReduceScore(xAmount);
        }
        else
        {
            if (score != 0)
            {
                addScore(xAmount);
            }
        }
    }
}
}

```

//remove sweets from the table: lose/gain points

```

public void ReverseSweets_()
{
    int xAmount = 1;
    sweets_day--;
    sweets--;
    //max 2 times a week
    if (sweets <= 2)
    {
        ReduceScore(xAmount);
    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }
    }
}

```

```

    }
}

//remove chicken from the table: lose/gain points
public void ReverseChicken_()
{
    int xAmount = 2;
    chicken_day--;
    chicken--;
    //max 3 times a week
    if (chicken <= 3)
    {
        ReduceScore(xAmount);

    }
    else
    {
        if (score != 0)
        {
            addScore(xAmount);
        }

    }
}

```

```

//remove fries from the table: lose/gain points
public void ReverseFries()
{
    int xAmount = 2;
    if (score != 0)
    {
        addScore(xAmount);
    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μια φορά από μόνη της όταν εισέλθουμε σε οποιοδήποτε level και σκοπός της είναι να μηδενίσει το score του level που βρισκόμαστε έτσι ώστε να

μπορούμε να παίζουμε το level από την αρχή και σε περίπτωση που έχουμε ξαναπαίξει το παιχνίδι, βρίσκει τις προηγούμενες μας επιλογές και τις αναιρεί. Η συνάρτηση `addScore()` καλείτε από τις υπόλοιπες συναρτήσεις οι οποίες είναι για κάθε κατηγορία φαγητού και σκοπός της είναι να προσθέτει πόντους στο `score`. Η συνάρτηση `ReduceScore()` καλείτε από τις υπόλοιπες συναρτήσεις οι οποίες είναι για κάθε κατηγορία φαγητού και σκοπός της είναι να αφαιρεί πόντους από το `score`. Η συνάρτηση `UpdateScoreUI()` καλείτε στις συναρτήσεις `addScore()` και `ReduceScore()`. Η συνάρτηση `ShowAnimationScore()` καλείτε στις συναρτήσεις `addScore()` και `ReduceScore()`. Η συνάρτηση `ScoreWhenDone()` καλείτε από το πράσινο κουμπί που υπάρχει στο Βραδινό όλων των levels. Η συνάρτηση `VegetablesAndFruits()` καλείτε από τα κουμπιά που είναι εικόνες φρούτων και λαχανικών και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `meat()` καλείτε από τα κουμπιά που είναι εικόνες κόκκινου κρέατος και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `Dairy()` καλείτε από τα κουμπιά που είναι εικόνες γαλακτοκομικών προϊόντων και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `legumes()` καλείτε από τα κουμπιά που είναι εικόνες οσπρίων και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `eggs()` καλείτε από τα κουμπιά που είναι εικόνες αυγών και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `fish()` καλείτε από τα κουμπιά που είναι εικόνες ψαριών και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `sweets()` καλείτε από τα κουμπιά που είναι εικόνες γλυκών και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `Chicken()` καλείτε από τα κουμπιά που είναι εικόνες που είναι κοτόπουλο και βρίσκονται στα αριστερά της οθόνης σε κάθε level. Η συνάρτηση `ReverseVegetablesAndFruits()` καλείτε από τα κουμπιά που είναι εικόνες φρούτων και λαχανικών και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση `Reversemeat()` καλείτε από τα κουμπιά που είναι εικόνες κόκκινου κρέατος και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση `ReverseDairy()` καλείτε από τα κουμπιά που είναι εικόνες γαλακτοκομικών προϊόντων και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση `Reverselegumes()` καλείτε από τα κουμπιά που είναι εικόνες οσπρίων και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση `Reverseeggs()` καλείτε από τα κουμπιά που είναι εικόνες αυγών και βρίσκονται στο τραπέζι σε κάθε level. . Η συνάρτηση `Reversefish()` καλείτε από τα κουμπιά που είναι εικόνες ψαριών και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση `Reversesweets()` καλείτε από τα κουμπιά που είναι εικόνες γλυκών και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση `ReverseChicken()` καλείτε από

τα κουμπιά που είναι εικόνες που είναι κοτόπουλο και βρίσκονται στο τραπέζι σε κάθε level. Η συνάρτηση Reversefries() καλείτε από τα κουμπιά που είναι εικόνες που είναι πατάτες και βρίσκονται στο τραπέζι σε κάθε level.

Κώδικας CharacterSelection.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class CharacterSelection : MonoBehaviour
{
    public GameObject Apple;
    public static int flagApple=1;
    public GameObject Orange;
    public static int flagOrange = 0;
    public static int unlockOrange = 0;
    public GameObject Avocado;
    public static int flagAvocado = 0;
    public static int unlockAvocado = 0;
    public GameObject Egg;
    public static int flagEgg = 0;
    public static int unlockEgg = 0;
    public GameObject Banana;
    public static int flagBanana = 0;
    public static int unlockBanana = 0;
    public GameObject Cheese;
    public static int flagCheese = 0;
    public static int unlockCheese = 0;
    public GameObject Grape;
    public static int flagGrape = 0;
    public static int unlockGrape = 0;
    public GameObject OliveOil;
    public static int flagOliveOil = 0;
    public static int unlockOliveOil = 0;
    public GameObject IceCream;
    public static int flagIceCream = 0;
    public static int unlockIceCream = 0;
```

```
public GameObject Nutella;  
public static int flagNutella = 0;  
public static int unlockNutella = 0;
```

```
public GameObject rightApple;  
public GameObject rightOrange;  
public GameObject leftOrange;  
public GameObject rightAvocado;  
public GameObject leftAvocado;  
public GameObject rightBanana;  
public GameObject leftBanana;  
public GameObject rightCheese;  
public GameObject leftCheese;  
public GameObject rightEgg;  
public GameObject leftEgg;  
public GameObject rightGrape;  
public GameObject leftGrape;  
public GameObject rightIceCream;  
public GameObject leftIceCream;  
public GameObject rightNutella;  
public GameObject leftNutella;  
public GameObject leftOliveOil;
```

```
public GameObject selectApple;  
public GameObject selectOrange;  
public GameObject selectAvocado;  
public GameObject selectBanana;  
public GameObject selectCheese;  
public GameObject selectEgg;  
public GameObject selectGrape;  
public GameObject selectIceCream;  
public GameObject selectNutella;  
public GameObject selectOliveOil;
```

```
public GameObject LockOrange;  
public GameObject LockAvocado;  
public GameObject LockBanana;  
public GameObject LockCheese;  
public GameObject LockEgg;  
public GameObject LockGrape;
```

```
public GameObject LockIceCream;
public GameObject LockNutella;
public GameObject LockOliveOil;
```

```
public int highscore;
public static int unlock2 = 0;
public static int unlock3 = 0;
public static int unlock4 = 0;
public static int unlock5 = 0;
public static int unlock6 = 0;
public static int unlock7 = 0;
public static int unlock8 = 0;
public static int unlock9 = 0;
public static int unlock10 = 0;
```

```
//show the apple character, the arrow and the play button
```

```
public void ShowSelectionApple()
{
    rightApple.SetActive(true);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(false);
    leftGrape.SetActive(false);
    rightIceCream.SetActive(false);
    leftIceCream.SetActive(false);
    rightNutella.SetActive(false);
    leftNutella.SetActive(false);
    leftOliveOil.SetActive(false);

    selectApple.SetActive(true);
    selectOrange.SetActive(false);
    selectAvocado.SetActive(false);
```

```

selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);

```

```

Apple.SetActive(true);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);

```

```

}

```

//show the orange character, the arrows and the play button

```

public void ShowSelectionOrange()
{
    if (unlock2 == 1)
    {
        Orange.SetActive(true);
        LockOrange.SetActive(false);
        selectOrange.SetActive(true);
    }
    else
    {
        LockOrange.SetActive(true);
        Orange.SetActive(false);
        selectOrange.SetActive(false);
    }
    rightApple.SetActive(false);
    rightOrange.SetActive(true);
    leftOrange.SetActive(true);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);

```

```
leftBanana.SetActive(false);
rightCheese.SetActive(false);
leftCheese.SetActive(false);
rightEgg.SetActive(false);
leftEgg.SetActive(false);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
LockAvocado.SetActive(false);
LockBanana.SetActive(false);
LockCheese.SetActive(false);
LockEgg.SetActive(false);
LockGrape.SetActive(false);
LockIceCream.SetActive(false);
LockNutella.SetActive(false);
```

```

        LockOliveOil.SetActive(false);
    }
//show the avocado character, the arrows and the play button
public void ShowSelectionAvocado()
{
    if (unlock3 == 1)
    {
        Avocado.SetActive(true);
        LockAvocado.SetActive(false);
        selectAvocado.SetActive(true);
    }
    else
    {
        LockAvocado.SetActive(true);
        Avocado.SetActive(false);
        selectAvocado.SetActive(false);
    }
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(true);
    leftAvocado.SetActive(true);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(false);
    leftGrape.SetActive(false);
    rightIceCream.SetActive(false);
    leftIceCream.SetActive(false);
    rightNutella.SetActive(false);
    leftNutella.SetActive(false);
    leftOliveOil.SetActive(false);

    selectApple.SetActive(false);
    selectOrange.SetActive(false);
    selectBanana.SetActive(false);
    selectCheese.SetActive(false);

```

```
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
LockOrange.SetActive(false);
LockBanana.SetActive(false);
LockCheese.SetActive(false);
LockEgg.SetActive(false);
LockGrape.SetActive(false);
LockIceCream.SetActive(false);
LockNutella.SetActive(false);
LockOliveOil.SetActive(false);
```

```
}
```

//show the banana character, the arrows and the play button

```
public void ShowSelectionBanana()
{
    if (unlock4 == 1)
    {
        Banana.SetActive(true);
        LockBanana.SetActive(false);
        selectBanana.SetActive(true);
    }
    else
    {
        LockBanana.SetActive(true);
        Banana.SetActive(false);
        selectBanana.SetActive(false);
    }
}
```

```
rightApple.SetActive(false);
rightOrange.SetActive(false);
leftOrange.SetActive(false);
rightAvocado.SetActive(false);
leftAvocado.SetActive(false);
rightBanana.SetActive(true);
leftBanana.SetActive(true);
rightCheese.SetActive(false);
leftCheese.SetActive(false);
rightEgg.SetActive(false);
leftEgg.SetActive(false);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
LockAvocado.SetActive(false);
```

```

        LockOrange.SetActive(false);
        LockCheese.SetActive(false);
        LockEgg.SetActive(false);
        LockGrape.SetActive(false);
        LockIceCream.SetActive(false);
        LockNutella.SetActive(false);
        LockOliveOil.SetActive(false);
    }
//show the cheese character, the arrows and the play button
public void ShowSelectionCheese()
{
    if (unlock5 == 1)
    {
        Cheese.SetActive(true);
        LockCheese.SetActive(false);
        selectCheese.SetActive(true);
    }
    else
    {
        LockCheese.SetActive(true);
        Cheese.SetActive(false);
        selectCheese.SetActive(false);
    }
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(true);
    leftCheese.SetActive(true);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(false);
    leftGrape.SetActive(false);
    rightIceCream.SetActive(false);
    leftIceCream.SetActive(false);
    rightNutella.SetActive(false);
    leftNutella.SetActive(false);

```

```
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);  
selectOrange.SetActive(false);  
selectAvocado.SetActive(false);  
selectBanana.SetActive(false);  
selectEgg.SetActive(false);  
selectGrape.SetActive(false);  
selectIceCream.SetActive(false);  
selectNutella.SetActive(false);  
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);  
Avocado.SetActive(false);  
Orange.SetActive(false);  
Banana.SetActive(false);  
Egg.SetActive(false);  
Grape.SetActive(false);  
IceCream.SetActive(false);  
Nutella.SetActive(false);  
OliveOil.SetActive(false);
```

```
LockAvocado.SetActive(false);  
LockBanana.SetActive(false);  
LockOrange.SetActive(false);  
LockEgg.SetActive(false);  
LockGrape.SetActive(false);  
LockIceCream.SetActive(false);  
LockNutella.SetActive(false);  
LockOliveOil.SetActive(false);
```

```
}
```

```
//show the egg character, the arrows and the play button
```

```
public void ShowSelectionEgg()  
{  
    if (unlock6 == 1)  
    {  
        Egg.SetActive(true);  
        LockEgg.SetActive(false);  
        selectEgg.SetActive(true);  
    }  
}
```

```

else
{
    LockEgg.SetActive(true);
    Egg.SetActive(false);
    selectEgg.SetActive(false);
}
rightApple.SetActive(false);
rightOrange.SetActive(false);
leftOrange.SetActive(false);
rightAvocado.SetActive(false);
leftAvocado.SetActive(false);
rightBanana.SetActive(false);
leftBanana.SetActive(false);
rightCheese.SetActive(false);
leftCheese.SetActive(false);
rightEgg.SetActive(true);
leftEgg.SetActive(true);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);

selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);

Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);

```

```

    Grape.SetActive(false);
    IceCream.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);

    LockAvocado.SetActive(false);
    LockBanana.SetActive(false);
    LockCheese.SetActive(false);
    LockOrange.SetActive(false);
    LockGrape.SetActive(false);
    LockIceCream.SetActive(false);
    LockNutella.SetActive(false);
    LockOliveOil.SetActive(false);
}
//show the grape character, the arrows and the play button
public void ShowSelectionGrape()
{
    if (unlock7 == 1)
    {
        Grape.SetActive(true);
        LockGrape.SetActive(false);
        selectGrape.SetActive(true);
    }
    else
    {
        LockGrape.SetActive(true);
        Grape.SetActive(false);
        selectGrape.SetActive(false);
    }
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);

```

```
rightGrape.SetActive(true);
leftGrape.SetActive(true);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
LockAvocado.SetActive(false);
LockBanana.SetActive(false);
LockCheese.SetActive(false);
LockEgg.SetActive(false);
LockOrange.SetActive(false);
LockIceCream.SetActive(false);
LockNutella.SetActive(false);
LockOliveOil.SetActive(false);
```

```
}
```

```
//show the ice cream, the arrows and the play button
```

```
public void ShowSelectionIceCream()
{
```

```

if (unlock8 == 1)
{
    IceCream.SetActive(true);
    LockIceCream.SetActive(false);
    selectIceCream.SetActive(true);
}
else
{
    LockIceCream.SetActive(true);
    IceCream.SetActive(false);
    selectIceCream.SetActive(false);
}

rightApple.SetActive(false);
rightOrange.SetActive(false);
leftOrange.SetActive(false);
rightAvocado.SetActive(false);
leftAvocado.SetActive(false);
rightBanana.SetActive(false);
leftBanana.SetActive(false);
rightCheese.SetActive(false);
leftCheese.SetActive(false);
rightEgg.SetActive(false);
leftEgg.SetActive(false);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(true);
leftIceCream.SetActive(true);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);

selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);

```

```

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);

    LockAvocado.SetActive(false);
    LockBanana.SetActive(false);
    LockCheese.SetActive(false);
    LockEgg.SetActive(false);
    LockGrape.SetActive(false);
    LockOrange.SetActive(false);
    LockNutella.SetActive(false);
    LockOliveOil.SetActive(false);
}

//show the Nutella character, the arrows and the play button
public void ShowSelectionNutella()
{
    if (unlock9 == 1)
    {
        Nutella.SetActive(true);
        LockNutella.SetActive(false);
        selectNutella.SetActive(true);
    }
    else
    {
        LockNutella.SetActive(true);
        Nutella.SetActive(false);
        selectNutella.SetActive(false);
    }
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);

```

```
rightBanana.SetActive(false);
leftBanana.SetActive(false);
rightCheese.SetActive(false);
leftCheese.SetActive(false);
rightEgg.SetActive(false);
leftEgg.SetActive(false);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(true);
leftNutella.SetActive(true);
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
OliveOil.SetActive(false);
```

```
LockAvocado.SetActive(false);
LockBanana.SetActive(false);
LockCheese.SetActive(false);
LockEgg.SetActive(false);
LockGrape.SetActive(false);
LockIceCream.SetActive(false);
```

```

        LockOrange.SetActive(false);
        LockOliveOil.SetActive(false);
    }
    //show the olive oil, the arrows and the play button
    public void ShowSelectionOliveOil()
    {
        if (unlock10 == 1)
        {
            OliveOil.SetActive(true);
            LockOliveOil.SetActive(false);
            selectOliveOil.SetActive(true);
        }
        else
        {
            LockOliveOil.SetActive(true);
            OliveOil.SetActive(false);
            selectOliveOil.SetActive(false);
        }
        rightApple.SetActive(false);
        rightOrange.SetActive(false);
        leftOrange.SetActive(false);
        rightAvocado.SetActive(false);
        leftAvocado.SetActive(false);
        rightBanana.SetActive(false);
        leftBanana.SetActive(false);
        rightCheese.SetActive(false);
        leftCheese.SetActive(false);
        rightEgg.SetActive(false);
        leftEgg.SetActive(false);
        rightGrape.SetActive(false);
        leftGrape.SetActive(false);
        rightIceCream.SetActive(false);
        leftIceCream.SetActive(false);
        rightNutella.SetActive(false);
        leftNutella.SetActive(false);
        leftOliveOil.SetActive(true);

        selectApple.SetActive(false);
        selectOrange.SetActive(false);
        selectAvocado.SetActive(false);

```

```
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
```

```
LockAvocado.SetActive(false);
LockBanana.SetActive(false);
LockCheese.SetActive(false);
LockEgg.SetActive(false);
LockGrape.SetActive(false);
LockIceCream.SetActive(false);
LockNutella.SetActive(false);
LockOrange.SetActive(false);
```

```
}
```

//when we press the play button of apple character, set others characters' flags to 0 and show only the apple character

```
public void chooseApple()
{
    flagApple = 1;
    Apple.SetActive(true);

    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 0;
```

```
flagNutella = 0;
flagOliveOil = 0;
flagOrange = 0;
```

```
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
PlayerPrefs.SetInt("Character", 1);
PlayerPrefs.Save();
SceneManager.LoadScene(1);
```

```
}
```

//when we press the play button of orange character, set others characters' flags to 0 and show only the orange character

```
public void chooseOrange()
```

```
{
```

```
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 0;
    flagNutella = 0;
    flagOliveOil = 0;
    flagOrange = 1;
```

```
    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(true);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
```

```

Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);

PlayerPrefs.SetInt("Character", 2);
PlayerPrefs.Save();
SceneManager.LoadScene(1);
}

```

//when we press the play button of avocado character, set others characters' flags to 0 and show only the avocado character

```

public void chooseAvocado()
{
    flagApple = 0;
    flagAvocado = 1;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 0;
    flagNutella = 0;
    flagOliveOil = 0;
    flagOrange = 0;

    Apple.SetActive(false);
    Avocado.SetActive(true);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    IceCream.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);

    PlayerPrefs.SetInt("Character", 3);
    PlayerPrefs.Save();
    SceneManager.LoadScene(1);
}

```

//when we press the play button of banana character, set others characters' flags to 0 and show only the banana character

```
public void chooseBanana()
{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 1;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 0;
    flagNutella = 0;
    flagOliveOil = 0;
    flagOrange = 0;

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(true);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    IceCream.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);

    PlayerPrefs.SetInt("Character", 4);
    PlayerPrefs.Save();
    SceneManager.LoadScene(1);
}
```

//when we press the play button of cheese character, set others characters' flags to 0 and show only the cheese character

```
public void chooseCheese()
{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 1;
```

```
flagEgg = 0;
flagGrape = 0;
flagIceCream = 0;
flagNutella = 0;
flagOliveOil = 0;
flagOrange = 0;
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(true);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
PlayerPrefs.SetInt("Character", 5);
PlayerPrefs.Save();
SceneManager.LoadScene(1);
```

```
}
```

//when we press the play button of egg character, set others characters' flags to 0 and show only the egg character

```
public void chooseEgg()
{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 1;
    flagGrape = 0;
    flagIceCream = 0;
    flagNutella = 0;
    flagOliveOil = 0;
    flagOrange = 0;

    Apple.SetActive(false);
    Avocado.SetActive(false);
```

```

        Orange.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Egg.SetActive(true);
        Grape.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
        OliveOil.SetActive(false);

        PlayerPrefs.SetInt("Character", 6);
        PlayerPrefs.Save();
        SceneManager.LoadScene(1);
    }

```

//when we press the play button of grape character, set others characters' flags to 0 and show only the grape character

```

public void chooseGrape()
{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 1;
    flagIceCream = 0;
    flagNutella = 0;
    flagOliveOil = 0;
    flagOrange = 0;

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(true);
    IceCream.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);
}

```

```

        PlayerPrefs.SetInt("Character", 7);
        PlayerPrefs.Save();
        SceneManager.LoadScene(1);
    }
    //when we press the play button of ice cream character, set others characters' flags to 0 and show only the
    ice cream character

```

```

public void chooseIceCream()
{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 1;
    flagNutella = 0;
    flagOliveOil = 0;
    flagOrange = 0;

```

```

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    IceCream.SetActive(true);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);

```

```

        PlayerPrefs.SetInt("Character", 8);
        PlayerPrefs.Save();
        SceneManager.LoadScene(1);
    }
    //when we press the play button of nutella character, set others characters' flags to 0 and show only the
    nutella character

```

```

public void chooseNutella()

```

```

{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 0;
    flagNutella = 1;
    flagOliveOil = 0;
    flagOrange = 0;

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    IceCream.SetActive(false);
    Nutella.SetActive(true);
    OliveOil.SetActive(false);

    PlayerPrefs.SetInt("Character", 9);
    PlayerPrefs.Save();
    SceneManager.LoadScene(1);
}

```

//when we press the play button of olive oil character, set others characters' flags to 0 and show only the olive oil character

```

public void chooseOliveOil()
{
    flagApple = 0;
    flagAvocado = 0;
    flagBanana = 0;
    flagCheese = 0;
    flagEgg = 0;
    flagGrape = 0;
    flagIceCream = 0;
    flagNutella = 0;
}

```

```

    flagOliveOil = 1;
    flagOrange = 0;

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    IceCream.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(true);

    PlayerPrefs.SetInt("Character", 10);
    PlayerPrefs.Save();
    SceneManager.LoadScene(1);
}
// Start is called before the first frame update
//check which characters are not locked and show the selected character
void Start()
{
    highscore = PlayerPrefs.GetInt("HighScore");
    if (highscore >= 10)
    {
        unlock2 = 1;
    }
    if (highscore >= 20)
    {
        unlock3 = 1;
    }
    if (highscore >= 30)
    {
        unlock4 = 1;
    }
    if (highscore >= 40)
    {
        unlock5 = 1;
    }
    if (highscore >= 50)

```

```

{
    unlock6 = 1;
}
if (highscore >=60)
{
    unlock7 = 1;
}
if (highscore >=70)
{
    unlock8 = 1;
}
if (highscore >=80)
{
    unlock9 = 1;
}
if (highscore >=90)
{
    unlock10 = 1;
}

//show the selected character
if (flagApple == 1)
{
    rightApple.SetActive(true);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(false);
    leftGrape.SetActive(false);
    rightIceCream.SetActive(false);
    leftIceCream.SetActive(false);
    rightNutella.SetActive(false);
    leftNutella.SetActive(false);
}

```

```

leftOliveOil.SetActive(false);

selectApple.SetActive(true);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);


Apple.SetActive(true);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
}
else if (flagOrange == 1)
{
    rightApple.SetActive(false);
    rightOrange.SetActive(true);
    leftOrange.SetActive(true);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(false);
    leftGrape.SetActive(false);
    rightIceCream.SetActive(false);

```

```

leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);

selectApple.SetActive(false);
selectOrange.SetActive(true);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);

Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(true);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
}
else if (flagAvocado == 1)
{
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(true);
    leftAvocado.SetActive(true);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);

```

```
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(true);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(true);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
}
```

```
else if (flagBanana == 1)
```

```
{
```

```
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(true);
    leftBanana.SetActive(true);
    rightCheese.SetActive(false);
```

```

    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(false);
    leftGrape.SetActive(false);
    rightIceCream.SetActive(false);
    leftIceCream.SetActive(false);
    rightNutella.SetActive(false);
    leftNutella.SetActive(false);
    leftOliveOil.SetActive(false);

    selectApple.SetActive(false);
    selectOrange.SetActive(false);
    selectAvocado.SetActive(false);
    selectBanana.SetActive(true);
    selectCheese.SetActive(false);
    selectEgg.SetActive(false);
    selectGrape.SetActive(false);
    selectIceCream.SetActive(false);
    selectNutella.SetActive(false);
    selectOliveOil.SetActive(false);

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(true);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(false);
    IceCream.SetActive(false);
    Nutella.SetActive(false);
    OliveOil.SetActive(false);
}
else if (flagCheese == 1)
{
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);

```

```
rightBanana.SetActive(false);
leftBanana.SetActive(false);
rightCheese.SetActive(true);
leftCheese.SetActive(true);
rightEgg.SetActive(false);
leftEgg.SetActive(false);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);
```

```
selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(true);
selectEgg.SetActive(false);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);
```

```
Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(true);
Egg.SetActive(false);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);
```

```
}
else if (flagEgg == 1)
{
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
```

```

leftOrange.SetActive(false);
rightAvocado.SetActive(false);
leftAvocado.SetActive(false);
rightBanana.SetActive(false);
leftBanana.SetActive(false);
rightCheese.SetActive(false);
leftCheese.SetActive(false);
rightEgg.SetActive(true);
leftEgg.SetActive(true);
rightGrape.SetActive(false);
leftGrape.SetActive(false);
rightIceCream.SetActive(false);
leftIceCream.SetActive(false);
rightNutella.SetActive(false);
leftNutella.SetActive(false);
leftOliveOil.SetActive(false);

```

```

selectApple.SetActive(false);
selectOrange.SetActive(false);
selectAvocado.SetActive(false);
selectBanana.SetActive(false);
selectCheese.SetActive(false);
selectEgg.SetActive(true);
selectGrape.SetActive(false);
selectIceCream.SetActive(false);
selectNutella.SetActive(false);
selectOliveOil.SetActive(false);

```

```

Apple.SetActive(false);
Avocado.SetActive(false);
Orange.SetActive(false);
Banana.SetActive(false);
Cheese.SetActive(false);
Egg.SetActive(true);
Grape.SetActive(false);
IceCream.SetActive(false);
Nutella.SetActive(false);
OliveOil.SetActive(false);

```

```

}

```

```

else if (flagGrape == 1)

```

```

{
    rightApple.SetActive(false);
    rightOrange.SetActive(false);
    leftOrange.SetActive(false);
    rightAvocado.SetActive(false);
    leftAvocado.SetActive(false);
    rightBanana.SetActive(false);
    leftBanana.SetActive(false);
    rightCheese.SetActive(false);
    leftCheese.SetActive(false);
    rightEgg.SetActive(false);
    leftEgg.SetActive(false);
    rightGrape.SetActive(true);
    leftGrape.SetActive(true);
    rightIceCream.SetActive(false);
    leftIceCream.SetActive(false);
    rightNutella.SetActive(false);
    leftNutella.SetActive(false);
    leftOliveOil.SetActive(false);

    selectApple.SetActive(false);
    selectOrange.SetActive(false);
    selectAvocado.SetActive(false);
    selectBanana.SetActive(false);
    selectCheese.SetActive(false);
    selectEgg.SetActive(false);
    selectGrape.SetActive(true);
    selectIceCream.SetActive(false);
    selectNutella.SetActive(false);
    selectOliveOil.SetActive(false);

    Apple.SetActive(false);
    Avocado.SetActive(false);
    Orange.SetActive(false);
    Banana.SetActive(false);
    Cheese.SetActive(false);
    Egg.SetActive(false);
    Grape.SetActive(true);
    IceCream.SetActive(false);
    Nutella.SetActive(false);

```

```

        OliveOil.SetActive(false);
    }
    else if (flagIceCream == 1)
    {
        rightApple.SetActive(false);
        rightOrange.SetActive(false);
        leftOrange.SetActive(false);
        rightAvocado.SetActive(false);
        leftAvocado.SetActive(false);
        rightBanana.SetActive(false);
        leftBanana.SetActive(false);
        rightCheese.SetActive(false);
        leftCheese.SetActive(false);
        rightEgg.SetActive(false);
        leftEgg.SetActive(false);
        rightGrape.SetActive(false);
        leftGrape.SetActive(false);
        rightIceCream.SetActive(true);
        leftIceCream.SetActive(true);
        rightNutella.SetActive(false);
        leftNutella.SetActive(false);
        leftOliveOil.SetActive(false);

        selectApple.SetActive(false);
        selectOrange.SetActive(false);
        selectAvocado.SetActive(false);
        selectBanana.SetActive(false);
        selectCheese.SetActive(false);
        selectEgg.SetActive(false);
        selectGrape.SetActive(false);
        selectIceCream.SetActive(true);
        selectNutella.SetActive(false);
        selectOliveOil.SetActive(false);

        Apple.SetActive(false);
        Avocado.SetActive(false);
        Orange.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Egg.SetActive(false);

```

```

        Grape.SetActive(false);
        IceCream.SetActive(true);
        Nutella.SetActive(false);
        OliveOil.SetActive(false);
    }
    else if (flagNutella == 1)
    {
        rightApple.SetActive(false);
        rightOrange.SetActive(false);
        leftOrange.SetActive(false);
        rightAvocado.SetActive(false);
        leftAvocado.SetActive(false);
        rightBanana.SetActive(false);
        leftBanana.SetActive(false);
        rightCheese.SetActive(false);
        leftCheese.SetActive(false);
        rightEgg.SetActive(false);
        leftEgg.SetActive(false);
        rightGrape.SetActive(false);
        leftGrape.SetActive(false);
        rightIceCream.SetActive(false);
        leftIceCream.SetActive(false);
        rightNutella.SetActive(true);
        leftNutella.SetActive(true);
        leftOliveOil.SetActive(false);

        selectApple.SetActive(false);
        selectOrange.SetActive(false);
        selectAvocado.SetActive(false);
        selectBanana.SetActive(false);
        selectCheese.SetActive(false);
        selectEgg.SetActive(false);
        selectGrape.SetActive(false);
        selectIceCream.SetActive(false);
        selectNutella.SetActive(true);
        selectOliveOil.SetActive(false);

        Apple.SetActive(false);
        Avocado.SetActive(false);
        Orange.SetActive(false);

```

```

        Banana.SetActive(false);
        Cheese.SetActive(false);
        Egg.SetActive(false);
        Grape.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(true);
        OliveOil.SetActive(false);
    }
    else if (flagOliveOil == 1)
    {
        rightApple.SetActive(false);
        rightOrange.SetActive(false);
        leftOrange.SetActive(false);
        rightAvocado.SetActive(false);
        leftAvocado.SetActive(false);
        rightBanana.SetActive(false);
        leftBanana.SetActive(false);
        rightCheese.SetActive(false);
        leftCheese.SetActive(false);
        rightEgg.SetActive(false);
        leftEgg.SetActive(false);
        rightGrape.SetActive(false);
        leftGrape.SetActive(false);
        rightIceCream.SetActive(false);
        leftIceCream.SetActive(false);
        rightNutella.SetActive(false);
        leftNutella.SetActive(false);
        leftOliveOil.SetActive(true);

        selectApple.SetActive(false);
        selectOrange.SetActive(false);
        selectAvocado.SetActive(false);
        selectBanana.SetActive(false);
        selectCheese.SetActive(false);
        selectEgg.SetActive(false);
        selectGrape.SetActive(false);
        selectIceCream.SetActive(false);
        selectNutella.SetActive(false);
        selectOliveOil.SetActive(true);
    }

```

```
        Apple.SetActive(false);
        Avocado.SetActive(false);
        Orange.SetActive(false);
        Banana.SetActive(false);
        Cheese.SetActive(false);
        Egg.SetActive(false);
        Grape.SetActive(false);
        IceCream.SetActive(false);
        Nutella.SetActive(false);
        OliveOil.SetActive(true);
    }
}
// Update is called once per frame
void Update(){}
}
```

Πότε καλείται η κάθε function:

Η συνάρτηση ShowSelectionApple() καλείται από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Πορτοκάλι στο Character selection scene. Η συνάρτηση ShowSelectionOrange() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Μήλο και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Αβοκάντο στο Character selection scene. Η συνάρτηση ShowSelectionAvocado() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Πορτοκάλι και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Μπανάνα στο Character selection scene. Η συνάρτηση ShowSelectionBanana() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Αβοκάντο και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Τυρί στο Character selection scene. Η συνάρτηση ShowSelectionCheese() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Μπανάνα και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Αυγό στο Character selection scene. Η συνάρτηση ShowSelectionEgg() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Τυρί και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Σταφύλι στο Character selection scene. Η συνάρτηση ShowSelectionGrape() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Αυγό και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Nutella στο Character selection scene. Η συνάρτηση ShowSelectionIceCream() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Nutella και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Ελαιόλαδο στο Character selection scene. Η συνάρτηση ShowSelectionNutela() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Σταφύλι και από το βελάκι που βρίσκεται στα αριστερά του χαρακτήρα Παγωτό στο Character selection scene. Η συνάρτηση ShowSelectionOliveOil() καλείται από το βελάκι που βρίσκεται στα δεξιά του χαρακτήρα Παγωτό στο Character selection scene. Η συνάρτηση chooseApple() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Μήλο στο Character selection scene. Η συνάρτηση chooseOrange() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Πορτοκάλι στο Character selection scene. Η συνάρτηση chooseAvocado() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Αβοκάντο στο Character selection scene. Η συνάρτηση chooseBanana() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Μπανάνα στο Character selection scene. Η

συνάρτηση chooseCheese() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Τυρί στο Character selection scene. Η συνάρτηση chooseEgg() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Αυγό στο Character selection scene. Η συνάρτηση chooseGrape() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Σταφύλι στο Character selection scene. Η συνάρτηση chooseIceCream() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Παγωτό στο Character selection scene. Η συνάρτηση chooseNutella() καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Nutella στο Character selection scene. Η συνάρτηση chooseOliveOil () καλείτε όταν πατήσουμε το κουμπί Play το οποίο βρίσκεται κάτω από τον χαρακτήρα Ελαιόλαδο στο Character selection scene και η συνάρτηση Start() καλείτε από μόνο της μια φορά όταν εισέλθουμε στο Character Selection scene.

Κώδικας ShowFoodMonday.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShowFoodMonday: MonoBehaviour
{

    //Breakfast's food
    public GameObject Picture_apple;
    public GameObject Picture_icecream;
    public GameObject Picture_toast;
    public GameObject Picture_oatmeal;
    public GameObject Picture_juice;
    public GameObject Picture_dessert;
    public GameObject Picture_cheese;
    public GameObject Picture_cucumber;
    public GameObject Picture_tomato;
    public GameObject Picture_icecream2;

    public GameObject CanvasApple;
    public GameObject CanvasIceCream;
    public GameObject CanvasToast;
```

```
public GameObject CanvasOatmeal;
public GameObject CanvasJuice;
public GameObject CanvasDessert;
public GameObject CanvasCheese;
public GameObject CanvasCucumber;
public GameObject CanvasTomato;
public GameObject CanvasIceCream2;
```

```
//Lunch's food
```

```
public GameObject Picture_tangerines;
public GameObject Picture_milkcereal;
public GameObject Picture_fakes;
public GameObject Picture_oliveoil;
public GameObject Picture_fries;
public GameObject Picture_water;
public GameObject Picture_icecream_;
public GameObject Picture_chips;
public GameObject Picture_egg;
public GameObject Picture_donut;
```

```
public GameObject Canvastangerines;
public GameObject CanvasMilkCereal;
public GameObject CanvasFakes;
public GameObject CanvasOLiveOil;
public GameObject CanvasFries;
public GameObject CanvasWater;
public GameObject CanvasIceCream_;
public GameObject CanvasChips;
public GameObject CanvasEgg;
public GameObject CanvasDonut;
```

```
//Dinner's food
```

```
public GameObject Picture_burger;
public GameObject Picture_strawberry;
public GameObject Picture_water_;
public GameObject Picture_pomeg;
public GameObject Picture_sweet;
public GameObject Picture_sweet2;
public GameObject Picture_peas;
public GameObject Picture_carrot;
```

```

public GameObject Picture_cake;
public GameObject Picture_corn;
public GameObject Picture_lolly;

public GameObject CanvasBurger;
public GameObject CanvasStrawberry;
public GameObject CanvasWater_;
public GameObject CanvasPomeg;
public GameObject CanvasSweet;
public GameObject CanvasPeas;
public GameObject CanvasCarrot;
public GameObject CanvasCake;
public GameObject CanvasCorn;
public GameObject CanvasLolly;

// Start is called before the first frame update
//at the begging of level 1, hide every food that is on the table and only show the food on the left
void Start()
{
    //breakfast table
    Picture_apple.SetActive(false);
    Picture_icecream.SetActive(false);
    Picture_toast.SetActive(false);
    Picture_oatmeal.SetActive(false);
    Picture_juice.SetActive(false);
    Picture_dessert.SetActive(false);
    Picture_cheese.SetActive(false);
    Picture_cucumber.SetActive(false);
    Picture_tomato.SetActive(false);
    Picture_icecream2.SetActive(false);

    //lunch table
    Picture_tangerines.SetActive(false);
    Picture_milkcereal.SetActive(false);
    Picture_fakes.SetActive(false);
    Picture_oliveoil.SetActive(false);
    Picture_fries.SetActive(false);
    Picture_water.SetActive(false);
    Picture_icecream_.SetActive(false);

```

```

Picture_chips.SetActive(false);
Picture_egg.SetActive(false);
Picture_donut.SetActive(false);

//dinner table
Picture_burger.SetActive(false);
Picture_strawberry.SetActive(false);
Picture_water_.SetActive(false);
Picture_pomeg.SetActive(false);
Picture_sweet.SetActive(false);
Picture_sweet2.SetActive(false);
Picture_peas.SetActive(false);
Picture_carrot.SetActive(false);
Picture_cake.SetActive(false);
Picture_corn.SetActive(false);
Picture_lolly.SetActive(false);
}

// Update is called once per frame
void Update()
{

}

//display the tomato on the table and hide the tomato on the left
    public void ShowTomato(){
        Picture_tomato.SetActive(true);
        CanvasTomato.SetActive(false);
    }

//display the tomato on the left and hide the tomato on the table
    public void hideTomato()
    {
        Picture_tomato.SetActive(false);
        CanvasTomato.SetActive(true);
    }

//display the apple on the table and hide the apple on the left
    public void ShowApple()
    {
        Picture_apple.SetActive(true);
        CanvasApple.SetActive(false);
    }

```

```

//display the apple on the left and hide the apple on the table
public void hideApple()
{
    Picture_apple.SetActive(false);
    CanvasApple.SetActive(true);
}

//display the ice cream on the table and hide the ice cream on the left
public void ShowIcecream()
{
    Picture_icecream.SetActive(true);
    CanvasIceCream.SetActive(false);
}

//hide the ice cream on the table and display the ice cream on the left
public void hideIcecream()
{
    Picture_icecream.SetActive(false);
    CanvasIceCream.SetActive(true);
}

//display the toast on the table and hide the toast on the left
public void ShowToast()
{
    Picture_toast.SetActive(true);
    CanvasToast.SetActive(false);
}

//display the toast on the left and hide the toast on the table
public void hideToast()
{
    Picture_toast.SetActive(false);
    CanvasToast.SetActive(true);
}

//display oat meal on the table and hide oat meal on the left
public void ShowOatMeal()
{
    Picture_oatmeal.SetActive(true);
    CanvasOatmeal.SetActive(false);
}

//hide oatmeal on the table and show oatmeal on the left
public void hideOatMeal()
{
    Picture_oatmeal.SetActive(false);

```

```

        CanvasOatmeal.SetActive(true);
    }
//show juice on the table and hide juice on the left
    public void ShowJuice()
    {
        Picture_juice.SetActive(true);
        CanvasJuice.SetActive(false);
    }
//hide juice on the table and show juice on the left
    public void hideJuice()
    {
        Picture_juice.SetActive(false);
        CanvasJuice.SetActive(true);
    }
//show desseert on the table and hide dessert on the left
    public void ShowDessert()
    {
        Picture_dessert.SetActive(true);
        CanvasDessert.SetActive(false);
    }
//hide dessert on the table and show dessert on the left
    public void hideDessert()
    {
        Picture_dessert.SetActive(false);
        CanvasDessert.SetActive(true);
    }
//show cheese on the table and hide cheese on the left
    public void ShowCheese()
    {
        Picture_cheese.SetActive(true);
        CanvasCheese.SetActive(false);
    }
//hide cheese on the table and show cheese on the left
    public void hideCheese()
    {
        Picture_cheese.SetActive(false);
        CanvasCheese.SetActive(true);
    }
//show cucumber on the table and hide cucumber on the left
    public void ShowCucumber()

```

```

{
    Picture_cucumber.SetActive(true);
    CanvasCucumber.SetActive(false);
}

//hide cucumber on the table and show cucumber on the left
public void hideCucumber()
{
    Picture_cucumber.SetActive(false);
    CanvasCucumber.SetActive(true);
}

//show the second ice cream on the table and hide the one on the left
public void ShowIceCream2()
{
    Picture_icecream2.SetActive(true);
    CanvasIceCream2.SetActive(false);
}

//show the second ice cream on the left and hide the one on the table
public void hideIceCream2()
{
    Picture_icecream2.SetActive(false);
    CanvasIceCream2.SetActive(true);
}

//show the tangerines on the table and the others on the left
public void ShowTangerines()
{
    Picture_tangerines.SetActive(true);
    CanvasTangerines.SetActive(false);
}

//hide the tangerines on the table and show the others on the left
public void hideTangerines()
{
    Picture_tangerines.SetActive(false);
    CanvasTangerines.SetActive(true);
}

//show the cereal on the table and hide the cereal on the left
public void ShowCereal()
{
    Picture_milkcereal.SetActive(true);
    CanvasMilkCereal.SetActive(false);
}

```

```

//hide the cereal on the table and show the cereal on the left
public void hideCereal()
{
    Picture_milkcereal.SetActive(false);
    CanvasMilkCereal.SetActive(true);
}

//show the fakes on the table and hide the other one on the left
public void ShowFakes()
{
    Picture_fakes.SetActive(true);
    CanvasFakes.SetActive(false);
}

//hide the fakes on the table and show the one on the left
public void hideFakes()
{
    Picture_fakes.SetActive(false);
    CanvasFakes.SetActive(true);
}

//show the olive oil the table and hide the olive oil on the left
public void ShowOliveOil()
{
    Picture_oliveoil.SetActive(true);
    CanvasOLiveOil.SetActive(false);
}

//show the olive oil on the left and hide the olive oil on the table
public void hideOliveOil()
{
    Picture_oliveoil.SetActive(false);
    CanvasOLiveOil.SetActive(true);
}

//show the fries on the table and hide the fries on the left
public void ShowFries()
{
    Picture_fries.SetActive(true);
    CanvasFries.SetActive(false);
}

//show the fries on the left and hide the fries on the table
public void hideFries()
{
    Picture_fries.SetActive(false);

```

```

        CanvasFries.SetActive(true);
    }
//show the water on the table and hide the water on the left
    public void ShowWater()
    {
        Picture_water.SetActive(true);
        CanvasWater.SetActive(false);
    }
//hide the water on the table and show the water on the left
    public void hideWater()
    {
        Picture_water.SetActive(false);
        CanvasWater.SetActive(true);
    }
// show the third ice cream on the table and hide the ice cream on the left
    public void ShowIceCream__()
    {
        Picture_icecream_.SetActive(true);
        CanvasIceCream_.SetActive(false);
    }
//hide the ice cream on the table and show the ice cream on the left
    public void hideIceCream__()
    {
        Picture_icecream_.SetActive(false);
        CanvasIceCream_.SetActive(true);
    }
//show the chips on the table and hide the chips on the left
    public void ShowChips()
    {
        Picture_chips.SetActive(true);
        CanvasChips.SetActive(false);
    }
//hide the chips on the table and show the chips on the left
    public void hideChips()
    {
        Picture_chips.SetActive(false);
        CanvasChips.SetActive(true);
    }
//show the egg on the table and hide the egg on the left
    public void ShowEgg()

```

```

    {
        Picture_egg.SetActive(true);
        CanvasEgg.SetActive(false);
    }
//hide the egg on the table and show the egg on the left
public void hideEgg()
{
    Picture_egg.SetActive(false);
    CanvasEgg.SetActive(true);
}
//show the donut on the table and hide the donut on the left
public void ShowDonut()
{
    Picture_donut.SetActive(true);
    CanvasDonut.SetActive(false);
}
//hide the donut on the table and show donut on the left
public void hideDonut()
{
    Picture_donut.SetActive(false);
    CanvasDonut.SetActive(true);
}
//show the burger on the table and hide the burger on the left
public void ShowBurger()
{
    Picture_burger.SetActive(true);
    CanvasBurger.SetActive(false);
}
//hide the burger on the table and show the burger on the left
public void hideBurger()
{
    Picture_burger.SetActive(false);
    CanvasBurger.SetActive(true);
}
//show the strawberry on the table and hide the one on the left
public void ShowStrwaberry()
{
    Picture_strawberry.SetActive(true);
    CanvasStrawberry.SetActive(false);
}

```

```

//show the strawberry on the left and hide the one on the table
public void hideStrwaberry()
{
    Picture_strawberry.SetActive(false);
    CanvasStrawberry.SetActive(true);
}

//show the second water on the table and hide the water on the left
public void ShowWater_()
{
    Picture_water_.SetActive(true);
    CanvasWater_.SetActive(false);
}

//show the water on the left and hide the water on the table
public void hideWater_()
{
    Picture_water_.SetActive(false);
    CanvasWater_.SetActive(true);
}

//show the pomeg on the table and hide the pomeg on the left
public void ShowPomeg()
{
    Picture_pomeg.SetActive(true);
    CanvasPomeg.SetActive(false);
}

//show the pomg on the left and hide the one on the table
public void hidePomeg()
{
    Picture_pomeg.SetActive(false);
    CanvasPomeg.SetActive(true);
}

//show the sweets on the table and hide the sweet on the left
public void ShowSweet()
{
    Picture_sweet.SetActive(true);
    Picture_sweet2.SetActive(true);
    CanvasSweet.SetActive(false);
}

//show the sweet on the left and hide the sweets on the table
public void hideSweet()
{

```

```

        Picture_sweet.SetActive(false);
        Picture_sweet2.SetActive(false);
        CanvasSweet.SetActive(true);
    }
//show the peas on the table and hide the peas on the left
    public void ShowPeas()
    {
        Picture_peas.SetActive(true);
        CanvasPeas.SetActive(false);
    }
//hide the peas on the table and show the peas on the left
    public void hidePeas()
    {
        Picture_peas.SetActive(false);
        CanvasPeas.SetActive(true);
    }
//show the carrot on the table and hide the carrot on the left
    public void ShowCarrot()
    {
        Picture_carrot.SetActive(true);
        CanvasCarrot.SetActive(false);
    }
//hide the carrot on the table and show the carrot on the left
    public void hideCarrot()
    {
        Picture_carrot.SetActive(false);
        CanvasCarrot.SetActive(true);
    }
//show the cake on the table and hide the cake on the left
    public void ShowCake()
    {
        Picture_cake.SetActive(true);
        CanvasCake.SetActive(false);
    }
//hide the cake on the table and show the cake on the left
    public void hideCake()
    {
        Picture_cake.SetActive(false);
        CanvasCake.SetActive(true);
    }

```

```

//show the corn on the table and hide the corn on the left
public void ShowCorn()
{
    Picture_corn.SetActive(true);
    CanvasCorn.SetActive(false);
}

//show the corn on the left and hide the corn on the table
public void hideCorn()
{
    Picture_corn.SetActive(false);
    CanvasCorn.SetActive(true);
}

//show the lolly on the table and hide the lolly on the left
public void ShowLolly()
{
    Picture_lolly.SetActive(true);
    CanvasLolly.SetActive(false);
}

//show the lolly on the left and hide the lolly on the table
public void hideLolly()
{
    Picture_lolly.SetActive(false);
    CanvasLolly.SetActive(true);
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο πρώτο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι. Οι συναρτήσεις Show() καλούνται μόνο από τα φαγητά που βρίσκονται στα αριστερά της οθόνης και είναι οι επιλογές μας, ενώ οι συναρτήσεις hide() καλούνται από τα φαγητά που βρίσκονται πάνω στο τραπέζι. Οι συναρτήσεις ShowTomato() και hideTomato() καλούνται όταν πατήσουμε στο φαγητό "ντομάτα". Οι συναρτήσεις ShowApple() και hideApple() καλούνται όταν πατήσουμε στο φαγητό

"μήλο". Οι συναρτήσεις ShowIceCream() και hideIceCream() καλούνται όταν πατήσουμε στο φαγητό "παγωτό". Οι συναρτήσεις ShowToast() και hideToast() καλούνται όταν πατήσουμε στο φαγητό "toast". Οι συναρτήσεις ShowOatMeal() και hideOatMeal() καλούνται όταν πατήσουμε στο φαγητό "oatmeal". Οι συναρτήσεις ShowJuice() και hideJuice() καλούνται όταν πατήσουμε τον χυμό. Οι συναρτήσεις ShowDessert() και hideDessert() καλούνται όταν πατήσουμε στο γλυκό. Οι συναρτήσεις ShowCheese() και hideCheese() καλούνται όταν πατήσουμε στο φαγητό "τυρί". Οι συναρτήσεις ShowCucumber() και hideCucumber() καλούνται όταν πατήσουμε στο φαγητό "αγγουράκι". Οι συναρτήσεις ShowIcecream() και hideIcecream () καλούνται όταν πατήσουμε στο φαγητό " παγωτό". Οι συναρτήσεις ShowTangerines () και hideTangerines () καλούνται όταν πατήσουμε στο φαγητό "μανταρίνι". Οι συναρτήσεις ShowCereal () και hideCereal () καλούνται όταν πατήσουμε στο φαγητό "δημητριακά". Οι συναρτήσεις ShowFakes () και hideFakes () καλούνται όταν πατήσουμε στο φαγητό "Φακές". Οι συναρτήσεις ShowOliveOil () και hideOliveOil () καλούνται όταν πατήσουμε το "ελαιόλαδο". Οι συναρτήσεις ShowFries () και hideFries () καλούνται όταν πατήσουμε στο φαγητό "τηγανιτές πατάτες". Οι συναρτήσεις ShowWater () και hideWater () καλούνται όταν πατήσουμε το " νερό ". Οι συναρτήσεις ShowChips () και hideChips () καλούνται όταν πατήσουμε στο φαγητό " Chips". Οι συναρτήσεις ShowEgg () και hideEgg () καλούνται όταν πατήσουμε στο φαγητό " αυγό". Οι συναρτήσεις ShowDonut () και hideDonut() καλούνται όταν πατήσουμε στο φαγητό "Donut". Οι συναρτήσεις ShowBurger () και hide Burger () καλούνται όταν πατήσουμε στο φαγητό " Burger". Οι συναρτήσεις ShowStrawberry () και hideStrawberry () καλούνται όταν πατήσουμε στο φαγητό " φράουλα ". Οι συναρτήσεις ShowPomeg () και hidePomeg() καλούνται όταν πατήσουμε στο φαγητό " Ρόδι". Οι συναρτήσεις ShowSweet () και hideSweet () καλούνται όταν πατήσουμε στο φαγητό " γλυκό". Οι συναρτήσεις ShowPeas () και hidePeas () καλούνται όταν πατήσουμε στο φαγητό " αρακάς". Οι συναρτήσεις ShowCarrot () και hideCarrot () καλούνται όταν πατήσουμε στο φαγητό "καρότο". Οι συναρτήσεις ShowCake () και hideCake() καλούνται όταν πατήσουμε στο φαγητό "cake". Οι συναρτήσεις ShowCorn () και hideCorn() καλούνται όταν πατήσουμε στο φαγητό " καλαμπόκι". Οι συναρτήσεις ShowLolly () και hideLolly () καλούνται όταν πατήσουμε στο φαγητό " Γλειφιτζούρι".

Κώδικας ShowFoodTuesday.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShowFoodTuesday : MonoBehaviour
{
    //Breakfast
    public GameObject Picture_banana;
    public GameObject Picture_cereal;
    public GameObject Picture_cake;
    public GameObject Picture_cocacola;
    public GameObject Picture_water;
    public GameObject Picture_sandwich;
    public GameObject Picture_hotchocolate;
    public GameObject Picture_carrot;
    public GameObject Picture_tomato;
    public GameObject Picture_icecream1;

    public GameObject CanvasBanana;
    public GameObject CanvasCereal;
    public GameObject CanvasCake;
    public GameObject CanvasCocaCola;
    public GameObject CanvasWater;
    public GameObject CanvasSandwich;
    public GameObject CanvasHotchocolate;
    public GameObject CanvasCarrot;
    public GameObject CanvasTomato;
    public GameObject CanvasIceCream1;

    //Lunch
    public GameObject Picture_pourgouri;
    public GameObject Picture_salad;
    public GameObject Picture_milkshake;
    public GameObject Picture_oliveoil;
    public GameObject Picture_nutella;
    public GameObject Picture_stake;
    public GameObject Picture_sweet;
    public GameObject Picture_icecream2;
    public GameObject Picture_fruitsalad;
    public GameObject Picture_donut;

    public GameObject CanvasPourgouri;
    public GameObject CanvasSalad;
    public GameObject CanvasMilkshake;
    public GameObject CanvasOLiveOil;
    public GameObject CanvasNutella;
    public GameObject CanvasStake;
    public GameObject CanvasSweet;
    public GameObject CanvasIceCream2;
    public GameObject CanvasFruitSalad;
    public GameObject CanvasDonut;

    //Dinner
    public GameObject Picture_nutella2;
    public GameObject Picture_pizza;
    public GameObject Picture_sweet2;
    public GameObject Picture_cocacola2;
    public GameObject Picture_cucumber;
```

```

public GameObject Picture_water2;
public GameObject Picture_veggies;
public GameObject Picture_yogurt;
public GameObject Picture_hotdog;
public GameObject Picture_milk;

public GameObject CanvasNutella2;
public GameObject CanvasPizza;
public GameObject CanvasSweet2;
public GameObject CanvasCocacola2;
public GameObject CanvasCucumber;
public GameObject CanvasWater2;
public GameObject CanvasVeggies;
public GameObject CanvasYogurt;
public GameObject CanvasHotdog;
public GameObject CanvasMilk;

// Start is called before the first frame update
void Start()
{
    //breakfast table
    Picture_banana.SetActive(false);
    Picture_cereal.SetActive(false);
    Picture_cake.SetActive(false);
    Picture_cocacola.SetActive(false);
    Picture_water.SetActive(false);
    Picture_sandwich.SetActive(false);
    Picture_hotchocolate.SetActive(false);
    Picture_carrot.SetActive(false);
    Picture_tomato.SetActive(false);
    Picture_icecream1.SetActive(false);

    //lunch table
    Picture_pourgouri.SetActive(false);
    Picture_salad.SetActive(false);
    Picture_milkshake.SetActive(false);
    Picture_oliveoil.SetActive(false);
    Picture_nutella.SetActive(false);
    Picture_stake.SetActive(false);
    Picture_sweet.SetActive(false);
    Picture_icecream2.SetActive(false);
    Picture_fruitsalad.SetActive(false);
    Picture_donut.SetActive(false);

    //dinner table
    Picture_nutella2.SetActive(false);
    Picture_pizza.SetActive(false);
    Picture_sweet2.SetActive(false);
    Picture_cocacola2.SetActive(false);
    Picture_cucumber.SetActive(false);
    Picture_water2.SetActive(false);
    Picture_veggies.SetActive(false);
    Picture_yogurt.SetActive(false);
    Picture_hotdog.SetActive(false);
    Picture_milk.SetActive(false);
}

// Update is called once per frame

```

```

void Update()
{

}

//show the banana on the table and hide the banana on the left
public void ShowBanana()
{
    Picture_banana.SetActive(true);
    CanvasBanana.SetActive(false);
}

//show the banana on the left and hide the banana on the table

public void hideBanana()
{
    Picture_banana.SetActive(false);
    CanvasBanana.SetActive(true);
}

//show the cereal on the table and hide the cereal on the left

public void ShowCereal()
{
    Picture_cereal.SetActive(true);
    CanvasCereal.SetActive(false);
}

//show the cereal on the left and hide the cereal on the table
public void hideCereal()
{
    Picture_cereal.SetActive(false);
    CanvasCereal.SetActive(true);
}

//show the cake on the table and hide the one on the left
public void ShowCake()
{
    Picture_cake.SetActive(true);
    CanvasCake.SetActive(false);
}

//show the cake on the left ad hide the one on the table
public void hideCake()
{
    Picture_cake.SetActive(false);
    CanvasCake.SetActive(true);
}

//show the coca cola on the table and hide the one on the left
public void ShowCocaCola()
{
    Picture_cocacola.SetActive(true);
    CanvasCocaCola.SetActive(false);
}

//show the coca cola on the left and hide the one on the table
public void hideCocaCola()
{
    Picture_cocacola.SetActive(false);
    CanvasCocaCola.SetActive(true);
}

//show thw water on the table and hide the one on the left
public void ShowWater()
{
    Picture_water.SetActive(true);

```

```

        CanvasWater.SetActive(false);
    }
//hide the water on the table and show the one on the left
    public void hideWater()
    {
        Picture_water.SetActive(false);
        CanvasWater.SetActive(true);
    }
//show the sandwich on the table and hide the one on the left
    public void ShowSandwich()
    {
        Picture_sandwich.SetActive(true);
        CanvasSandwich.SetActive(false);
    }
//hide sandwich on the table and show the one on the left
    public void hideSandwich()
    {
        Picture_sandwich.SetActive(false);
        CanvasSandwich.SetActive(true);
    }
//show hot chocolate on the table and hide the one on the left
    public void ShowHotchocolate()
    {
        Picture_hotchocolate.SetActive(true);
        CanvasHotchocolate.SetActive(false);
    }
//show hot chocolate on the left and hide the one on the table

    public void hideHotchocolate()
    {
        Picture_hotchocolate.SetActive(false);
        CanvasHotchocolate.SetActive(true);
    }
//show carrot on the table and hide the one on the left

    public void ShowCarrot()
    {
        Picture_carrot.SetActive(true);
        CanvasCarrot.SetActive(false);
    }
//show carrot on the left and hide the one on the table

    public void hideCarrot()
    {
        Picture_carrot.SetActive(false);
        CanvasCarrot.SetActive(true);
    }
//show the tomato on the table and hide the one on the left

    public void ShowTomate()
    {
        Picture_tomato.SetActive(true);
        CanvasTomato.SetActive(false);
    }
//show the tomato on the left and hide the one on the table

    public void hideTomato()
    {
        Picture_tomato.SetActive(false);

```

```

        CanvasTomato.SetActive(true);
    }

    //show the ice cream on the table and hide the one on the left
    public void ShowIceCream1()
    {
        Picture_icecream1.SetActive(true);
        CanvasIceCream1.SetActive(false);
    }
    //show the ice cream on the left and hide the one on the table

    public void hideIceCream1()
    {
        Picture_icecream1.SetActive(false);
        CanvasIceCream1.SetActive(true);
    }
    //show the pourgouri on the table and hide the one on the left
    public void ShowPourgouri()
    {
        Picture_pourgouri.SetActive(true);
        CanvasPourgouri.SetActive(false);
    }
    //show the pourgouri on the left and hide the one on the table

    public void hidePourgouri()
    {
        Picture_pourgouri.SetActive(false);
        CanvasPourgouri.SetActive(true);
    }
    //show the salad on the table and hide the one on the left

    public void ShowSalad()
    {
        Picture_salad.SetActive(true);
        CanvasSalad.SetActive(false);
    }
    //show the salad on the left and hide the one on the table
    public void hideSalad()
    {
        Picture_salad.SetActive(false);
        CanvasSalad.SetActive(true);
    }
    //show the milkshake on the table and hide the one on the left

    public void ShowMilkshake()
    {
        Picture_milkshake.SetActive(true);
        CanvasMilkshake.SetActive(false);
    }
    //show the milkshake on the left and hide the one on the table

    public void hideMilkshake()
    {
        Picture_milkshake.SetActive(false);
        CanvasMilkshake.SetActive(true);
    }
    //show the olive oil on the table and hide the one on the left
    public void Showoliveoil()
    {

```

```

        Picture_oliveoil.SetActive(true);
        CanvasOLiveOil.SetActive(false);
    }
    //show the olive oil  on the left and hide the one on the table
    public void hideoliveoil()
    {
        Picture_oliveoil.SetActive(false);
        CanvasOLiveOil.SetActive(true);
    }
    //show the nutella on the table and hide the one on the left
    public void ShowNutella()
    {
        Picture_nutella.SetActive(true);
        CanvasNutella.SetActive(false);
    }
    //show the nutella on the left and hide the one on the table
    public void hideNutella()
    {
        Picture_nutella.SetActive(false);
        CanvasNutella.SetActive(true);
    }
    //show the stake on the table and hide the one on the left
    public void ShowStake()
    {
        Picture_stake.SetActive(true);
        CanvasStake.SetActive(false);
    }
    //show the stake  on the left and hide the one on the table
    public void hideStake()
    {
        Picture_stake.SetActive(false);
        CanvasStake.SetActive(true);
    }
    //show the sweet on the table and hide the one on the left
    public void ShowSweet()
    {
        Picture_sweet.SetActive(true);
        CanvasSweet.SetActive(false);
    }
    //show the sweet  on the left and hide the one on the table
    public void hideSweet()
    {
        Picture_sweet.SetActive(false);
        CanvasSweet.SetActive(true);
    }
    //show the ice cream on the table and hide the one on the left
    public void ShowIcecream2()
    {
        Picture_icecream2.SetActive(true);
        CanvasIceCream2.SetActive(false);
    }
    //show the ice cream  on the left and hide the one on the table
    public void hideIcecream2()
    {
        Picture_icecream2.SetActive(false);
        CanvasIceCream2.SetActive(true);
    }
    //show the fruit salad on the table and hide the one on the left
    public void ShowFruitSalad()

```

```

    {
        Picture_fruitsalad.SetActive(true);
        CanvasFruitSalad.SetActive(false);
    }

//show the fruit salad on the left and hide the one on the table
public void hideFruitSalad()
{
    Picture_fruitsalad.SetActive(false);
    CanvasFruitSalad.SetActive(true);
}

//show the donut on the table and hide the one on the left
public void ShowDonut()
{
    Picture_donut.SetActive(true);
    CanvasDonut.SetActive(false);
}

//show the donut on the left and hide the one on the table
public void hideDonut()
{
    Picture_donut.SetActive(false);
    CanvasDonut.SetActive(true);
}

//show the nutella on the table and hide the one on the left
public void ShowNutella2()
{
    Picture_nutella2.SetActive(true);
    CanvasNutella2.SetActive(false);
}

//show the nutella on the left and hide the one on the table
public void hideNutella2()
{
    Picture_nutella2.SetActive(false);
    CanvasNutella2.SetActive(true);
}

//show the pizza on the table and hide the one on the left
public void ShowPizza()
{
    Picture_pizza.SetActive(true);
    CanvasPizza.SetActive(false);
}

//show the pizza on the left and hide the one on the table
public void hidePizza()
{
    Picture_pizza.SetActive(false);
    CanvasPizza.SetActive(true);
}

//show the sweet on the table and hide the one on the left
public void ShowSweet2()
{
    Picture_sweet2.SetActive(true);
    CanvasSweet2.SetActive(false);
}

//show the sweet on the left and hide the one on the table
public void hideSweet2()
{
    Picture_sweet2.SetActive(false);
    CanvasSweet2.SetActive(true);
}

```

```

//show the coca cola on the table and hide the one on the left
public void ShowCocacola2()
{
    Picture_cocacola2.SetActive(true);
    CanvasCocacola2.SetActive(false);
}
//show the coca cola on the left and hide the one on the table
public void hideCocacola2()
{
    Picture_cocacola2.SetActive(false);
    CanvasCocacola2.SetActive(true);
}
//show the cucumber on the table and hide the one on the left
public void ShowCucumber()
{
    Picture_cucumber.SetActive(true);
    CanvasCucumber.SetActive(false);
}
//show the cucumber on the left and hide the one on the table
public void hideCucumber()
{
    Picture_cucumber.SetActive(false);
    CanvasCucumber.SetActive(true);
}
//show the water on the table and hide the one on the left
public void ShowWater2()
{
    Picture_water2.SetActive(true);
    CanvasWater2.SetActive(false);
}
//show the water on the left and hide the one on the table
public void hideWater2()
{
    Picture_water2.SetActive(false);
    CanvasWater2.SetActive(true);
}
//show the veggies on the table and hide the one on the left
public void ShowVeggies()
{
    Picture_veggies.SetActive(true);
    CanvasVeggies.SetActive(false);
}
//show the veggies on the left and hide the one on the table
public void hideVeggies()
{
    Picture_veggies.SetActive(false);
    CanvasVeggies.SetActive(true);
}
//show the yogurt on the table and hide the one on the left
public void ShowYogurt()
{
    Picture_yogurt.SetActive(true);
    CanvasYogurt.SetActive(false);
}
//show the yogurt on the left and hide the one on the table
public void hideYogurt()
{
    Picture_yogurt.SetActive(false);
    CanvasYogurt.SetActive(true);
}

```

```

    }
    //show the hot dog on the table and hide the one on the left
    public void ShowHotdog()
    {
        Picture_hotdog.SetActive(true);
        CanvasHotdog.SetActive(false);
    }
    //show the hot dog on the left and hide the one on the table
    public void hideHotdog()
    {
        Picture_hotdog.SetActive(false);
        CanvasHotdog.SetActive(true);
    }
    //show the milk on the table and hide the one on the left
    public void ShowMilk()
    {
        Picture_milk.SetActive(true);
        CanvasMilk.SetActive(false);
    }
    //show the milk on the left and hide the one on the table
    public void hideMilk()
    {
        Picture_milk.SetActive(false);
        CanvasMilk.SetActive(true);
    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο δεύτερο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι. Οι συναρτήσεις Show() καλούνται μόνο από τα φαγητά που βρίσκονται στα αριστερά της οθόνης και είναι οι επιλογές μας, ενώ οι συναρτήσεις hide() καλούνται από τα φαγητά που βρίσκονται πάνω στο τραπέζι. Οι συναρτήσεις ShowBanana() και hideBanana() καλούνται όταν πατήσουμε στο φαγητό "μπανάνα". Οι συναρτήσεις ShowCereal() και hideCereal() καλούνται όταν πατήσουμε στο φαγητό "δημητριακά". Οι συναρτήσεις ShowCake() και hideCake() καλούνται όταν πατήσουμε στο φαγητό "Cake". Οι συναρτήσεις ShowCocacola() και hideCocacola() καλούνται όταν πατήσουμε στην Coca cola. Οι συναρτήσεις ShowWater () και hideWater () καλούνται όταν πατήσουμε το νερό. Οι συναρτήσεις ShowHotChocolate () και hideHotChocolate () καλούνται όταν πατήσουμε το Hot Chocolate. Οι συναρτήσεις ShowCarrot () και hideCarrot () καλούνται όταν πατήσουμε στο φαγητό "καρότο". Οι

συναρτήσεις ShowTomato () και hideTomato () καλούνται όταν πατήσουμε στο φαγητό " ντομάτα". Οι συναρτήσεις ShowIce cream () και hideIce cream () καλούνται όταν πατήσουμε στο φαγητό "παγωτό ". Οι συναρτήσεις ShowPourgouri () και hidePourgouri () καλούνται όταν πατήσουμε στο φαγητό "Πουργούρι". Οι συναρτήσεις Show Salad () και hide Salad () καλούνται όταν πατήσουμε στο φαγητό " σαλάτα". Οι συναρτήσεις ShowMilkshake () και hideMilkshake () καλούνται όταν πατήσουμε στο φαγητό " Milkshake". Οι συναρτήσεις ShowOliveOil () και hideOliveOil () καλούνται όταν πατήσουμε το "ελαιόλαδο". Οι συναρτήσεις ShowNutella () και hideNutella() καλούνται όταν πατήσουμε στο φαγητό "Nutella". Οι συναρτήσεις ShowSteak () και hideSteak () καλούνται όταν πατήσουμε στο φαγητό "steak". Οι συναρτήσεις ShowSweet () και hideSweet () καλούνται όταν πατήσουμε στο γλυκό. Οι συναρτήσεις ShowFruitSalad () και hideFruitSalad () καλούνται όταν πατήσουμε στο φαγητό "φρουτοσαλάτα". Οι συναρτήσεις ShowDonut () και hideDonut () καλούνται όταν πατήσουμε στο φαγητό "donut". Οι συναρτήσεις ShowPizza () και hidePizza () καλούνται όταν πατήσουμε στο φαγητό "pizza". Οι συναρτήσεις ShowCucumber () και hideCucumber () καλούνται όταν πατήσουμε στο φαγητό "αγγουράκι". Οι συναρτήσεις ShowVeggies () και hideVeggies () καλούνται όταν πατήσουμε στο φαγητό "βραστά λαχανικά". Οι συναρτήσεις ShowYogurt () και hideYogurt () καλούνται όταν πατήσουμε στο φαγητό "γιαούρτι". Οι συναρτήσεις ShowHotdog () και hideHotdog () καλούνται όταν πατήσουμε στο φαγητό "hot dog". Οι συναρτήσεις ShowMilk () και hideMilk () καλούνται όταν πατήσουμε στο γάλα. Ακολουθούμε την ίδια διαδικασία και στους κώδικες που ακολουθούν.

Κώδικας ShowFoodWednesday.cs:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShowFoodWednesday : MonoBehaviour
{
    //Breakfast's food
    public GameObject bread_picture;
    public GameObject breadCanvas;
    public GameObject cereal_picture;
    public GameObject cerealCanvas;
    public GameObject burger_picture;
    public GameObject burgerCanvas;
    public GameObject egg_picture;
    public GameObject eggCanvas;
    public GameObject tomato_picture;
```

```

public GameObject tomatoCanvas;
public GameObject cheese_picture;
public GameObject cheeseCanvas;
public GameObject cucumber_picture;
public GameObject cucumberCanvas;
public GameObject carrot_picture;
public GameObject carrotCanvas;
public GameObject yogurt_picture;
public GameObject yogurtCanvas;
public GameObject lemonade_picture;
public GameObject lemonadeCanvas;

//Lunch's food
public GameObject pear_picture;
public GameObject pearCanvas;
public GameObject louvi_picture;
public GameObject louviCanvas;
public GameObject friedchicken_picture;
public GameObject friedchickenCanvas;
public GameObject oil_picture;
public GameObject oilCanvas;
public GameObject tuna_picture;
public GameObject tunaCanvas;
public GameObject water_picture;
public GameObject waterCanvas;
public GameObject kiwi_picture;
public GameObject kiwiCanvas;
public GameObject icecream_picture;
public GameObject icecreamCanvas;
public GameObject banana_picture;
public GameObject bananaCanvas;
public GameObject salad_picture;
public GameObject saladCanvas;

//Dinner's food
public GameObject pastitsio_picture;
public GameObject pastitsioCanvas;
public GameObject chocolates_picture;
public GameObject chocolatesCanvas;
public GameObject sprite_picture;
public GameObject spriteCanvas;
public GameObject lemon_picture;
public GameObject lemonCanvas;
public GameObject carrot;
public GameObject carrot2Canvas;
public GameObject sweet_picture;
public GameObject sweetCanvas;
public GameObject watermelon_picture;
public GameObject watermelonCanvas;
public GameObject jellybear_picture;
public GameObject jellybearCanvas;
public GameObject oliveoil;
public GameObject oliveoilCanvas;
public GameObject milk_picture;
public GameObject milkCanvas;

void Start()
{
    //breakfast table

```

```

bread_picture.SetActive(false);
cereal_picture.SetActive(false);
burger_picture.SetActive(false);
egg_picture.SetActive(false);
tomato_picture.SetActive(false);
cheese_picture.SetActive(false);
cucumber_picture.SetActive(false);
carrot_picture.SetActive(false);
yogurt_picture.SetActive(false);
lemonade_picture.SetActive(false);

//lunch table
pear_picture.SetActive(false);
louvi_picture.SetActive(false);
friedchicken_picture.SetActive(false);
oil_picture.SetActive(false);
tuna_picture.SetActive(false);
water_picture.SetActive(false);
kiwi_picture.SetActive(false);
icecream_picture.SetActive(false);
banana_picture.SetActive(false);
salad_picture.SetActive(false);

//dinner table
pastitsio_picture.SetActive(false);
chocolates_picture.SetActive(false);
sprite_picture.SetActive(false);
lemon_picture.SetActive(false);
carrot.SetActive(false);
sweet_picture.SetActive(false);
watermelon_picture.SetActive(false);
jellybear_picture.SetActive(false);
oliveoil.SetActive(false);
milk_picture.SetActive(false);

}
//show the bread on the table and hide the one on the left
public void ShowBread()
{
    bread_picture.SetActive(true);
    breadCanvas.SetActive(false);
}
//show the bread on the left and hide the one on the table
public void hideBread()
{
    bread_picture.SetActive(false);
    breadCanvas.SetActive(true);
}
//show the cereal on the table and hide the one on the left
public void ShowCereal()
{
    cereal_picture.SetActive(true);
    cerealCanvas.SetActive(false);
}
//show the cereal on the left and hide the one on the table
public void hideCereal()
{
    cereal_picture.SetActive(false);
    cerealCanvas.SetActive(true);
}

```

```

    }
    //show the burger on the table and hide the one on the left
    public void ShowBurger()
    {
        burger_picture.SetActive(true);
        burgerCanvas.SetActive(false);
    }
    //show the burger on the left and hide the one on the table
    public void hideBurger()
    {
        burger_picture.SetActive(false);
        burgerCanvas.SetActive(true);
    }
    //show the egg on the table and hide the one on the left
    public void ShowEgg()
    {
        egg_picture.SetActive(true);
        eggCanvas.SetActive(false);
    }
    //show the egg on the left and hide the one on the table
    public void hideEgg()
    {
        egg_picture.SetActive(false);
        eggCanvas.SetActive(true);
    }
    //show the tomato on the table and hide the one on the left

    public void ShowTomato()
    {
        tomato_picture.SetActive(true);
        tomatoCanvas.SetActive(false);
    }
    //show the tomato on the left and hide the one on the table
    public void hideTomato()
    {
        tomato_picture.SetActive(false);
        tomatoCanvas.SetActive(true);
    }
    //show the cheese on the table and hide the one on the left
    public void ShowCheese()
    {
        cheese_picture.SetActive(true);
        cheeseCanvas.SetActive(false);
    }
    //show the cheese on the left and hide the one on the table
    public void hideCheese()
    {
        cheese_picture.SetActive(false);
        cheeseCanvas.SetActive(true);
    }
    //show the cucumber on the table and hide the one on the left
    public void ShowCucumber()
    {
        cucumber_picture.SetActive(true);
        cucumberCanvas.SetActive(false);
    }
    //show the cucumber on the left and hide the one on the table
    public void hideCucumber()
    {

```

```

        cucumber_picture.SetActive(false);
        cucumberCanvas.SetActive(true);
    }

    //show the carrot on the table and hide the one on the left
    public void ShowCarrot()
    {
        carrot_picture.SetActive(true);
        carrotCanvas.SetActive(false);
    }
    //show the carrot on the left and hide the one on the table
    public void hideCarrot()
    {
        carrot_picture.SetActive(false);
        carrotCanvas.SetActive(true);
    }
    //show the yogurt on the table and hide the one on the left
    public void ShowYogurt()
    {
        yogurt_picture.SetActive(true);
        yogurtCanvas.SetActive(false);
    }
    //show the yogurt on the left and hide the one on the table
    public void hideYogurt()
    {
        yogurt_picture.SetActive(false);
        yogurtCanvas.SetActive(true);
    }
    //show the lemonade on the table and hide the one on the left
    public void ShowLemonade()
    {
        lemonade_picture.SetActive(true);
        lemonadeCanvas.SetActive(false);
    }
    //show the lemonade on the left and hide the one on the table
    public void hideLemonade()
    {
        lemonade_picture.SetActive(false);
        lemonadeCanvas.SetActive(true);
    }
    //show the pear on the table and hide the one on the left
    public void ShowPearLunch()
    {
        pear_picture.SetActive(true);
        pearCanvas.SetActive(false);
    }
    //show the pear on the left and hide the one on the table
    public void hidePearLunch()
    {
        pear_picture.SetActive(false);
        pearCanvas.SetActive(true);
    }
    //show the louvia on the table and hide the one on the left
    public void ShowLouviLunch()
    {
        louvi_picture.SetActive(true);
        louviCanvas.SetActive(false);
    }

```

```

    }
    //show the louvia on the left and hide the one on the table
    public void hideLouviLunch()
    {
        louvi_picture.SetActive(false);
        louviCanvas.SetActive(true);
    }
    //show the chicken on the table and hide the one on the left
    public void ShowFriedChickenLunch()
    {
        friedchicken_picture.SetActive(true);
        friedchickenCanvas.SetActive(false);
    }
    //show the chicken on the left and hide the one on the table
    public void hideFriedChickenLunch()
    {
        friedchicken_picture.SetActive(false);
        friedchickenCanvas.SetActive(true);
    }
    //show the olive oil on the table and hide the one on the left
    public void ShowOilLunch()
    {
        oil_picture.SetActive(true);
        oilCanvas.SetActive(false);
    }
    //show the olive oil on the left and hide the one on the table
    public void hideOilLunch()
    {
        oil_picture.SetActive(false);
        oilCanvas.SetActive(true);
    }
    //show the tuna on the table and hide the one on the left
    public void ShowTunaLunch()
    {
        tuna_picture.SetActive(true);
        tunaCanvas.SetActive(false);
    }
    //show the tuna on the left and hide the one on the table
    public void hideTunaLunch()
    {
        tuna_picture.SetActive(false);
        tunaCanvas.SetActive(true);
    }
    //show the water on the table and hide the one on the left
    public void ShowWaterLunch()
    {
        water_picture.SetActive(true);
        waterCanvas.SetActive(false);
    }
    //show the water on the left and hide the one on the table
    public void hideWaterLunch()
    {
        water_picture.SetActive(false);
        waterCanvas.SetActive(true);
    }
    //show the kiwi on the table and hide the one on the left
    public void ShowKiwiLunch()
    {
        kiwi_picture.SetActive(true);

```

```

        kiwiCanvas.SetActive(false);
    }
    //show the kiwi on the left and hide the one on the table
    public void hideKiwiLunch()
    {
        kiwi_picture.SetActive(false);
        kiwiCanvas.SetActive(true);
    }
    //show the ice cream on the table and hide the one on the left
    public void ShowIcecreamLunch()
    {
        icecream_picture.SetActive(true);
        icecreamCanvas.SetActive(false);
    }
    //show the ice cream on the left and hide the one on the table
    public void hideIcecreamLunch()
    {
        icecream_picture.SetActive(false);
        icecreamCanvas.SetActive(true);
    }
    //show the banana on the table and hide the one on the left
    public void ShowBananaLunch()
    {
        banana_picture.SetActive(true);
        bananaCanvas.SetActive(false);
    }
    //show the banana on the left and hide the one on the table

    public void hideBananaLunch()
    {
        banana_picture.SetActive(false);
        bananaCanvas.SetActive(true);
    }

    //show the salad on the table and hide the one on the left
    public void ShowSaladLunch()
    {
        salad_picture.SetActive(true);
        saladCanvas.SetActive(false);
    }
    //show the salad on the left and hide the one on the table
    public void hideSaladLunch()
    {
        salad_picture.SetActive(false);
        saladCanvas.SetActive(true);
    }
    //show the pastitsio on the table and hide the one on the left
    public void ShowPastitsioDinner()
    {
        pastitsio_picture.SetActive(true);
        pastitsioCanvas.SetActive(false);
    }
    //show the pastitsio on the left and hide the one on the table
    public void hidePastitsioDinner()
    {
        pastitsio_picture.SetActive(false);
        pastitsioCanvas.SetActive(true);
    }
    //show the chocolates on the table and hide the one on the left

```

```

public void ShowChocolatesDinner()
{
    chocolates_picture.SetActive(true);
    chocolatesCanvas.SetActive(false);
}
//show the chocolates on the left and hide the one on the table
public void hideChocolatesDinner()
{
    chocolates_picture.SetActive(false);
    chocolatesCanvas.SetActive(true);
}
//show the sprite on the table and hide the one on the left
public void ShowSpriteDinner()
{
    sprite_picture.SetActive(true);
    spriteCanvas.SetActive(false);
}
//show the sprite on the left and hide the one on the table
public void hideSpriteDinner()
{
    sprite_picture.SetActive(false);
    spriteCanvas.SetActive(true);
}
//show the lemon on the table and hide the one on the left
public void ShowLemonDinner()
{
    lemon_picture.SetActive(true);
    lemonCanvas.SetActive(false);
}
//show the lemon on the left and hide the one on the table
public void hideLemonDinner()
{
    lemon_picture.SetActive(false);
    lemonCanvas.SetActive(true);
}
//show the carrot on the table and hide the one on the left
public void ShowCarrotDinner()
{
    carrot.SetActive(true);
    carrot2Canvas.SetActive(false);
}
//show the carrot on the left and hide the one on the table
public void hideCarrotDinner()
{
    carrot.SetActive(false);
    carrot2Canvas.SetActive(true);
}
//show the sweet on the table and hide the one on the left
public void ShowSweetDinner()
{
    sweet_picture.SetActive(true);
    sweetCanvas.SetActive(false);
}
//show the sweet on the left and hide the one on the table
public void hideSweetDinner()
{
    sweet_picture.SetActive(false);
    sweetCanvas.SetActive(true);
}

```

```

//show the watermelon on the table and hide the one on the left
public void ShowWatermelonDinner()
{
    watermelon_picture.SetActive(true);
    watermelonCanvas.SetActive(false);
}
//show the watermelon on the left and hide the one on the table
public void hideWatermelonDinner()
{
    watermelon_picture.SetActive(false);
    watermelonCanvas.SetActive(true);
}
//show the jelly bear on the table and hide the one on the left
public void ShowJellyBearDinner()
{
    jellybear_picture.SetActive(true);
    jellybearCanvas.SetActive(false);
}
//show the jelly bear on the left and hide the one on the table
public void hideJellyBearDinner()
{
    jellybear_picture.SetActive(false);
    jellybearCanvas.SetActive(true);
}
//show the olive oil on the table and hide the one on the left
public void ShowOliveOilDinner()
{
    oliveoil.SetActive(true);
    oliveoilCanvas.SetActive(false);
}
//show the olive oil on the left and hide the one on the table
public void hideOliveOilDinner()
{
    oliveoil.SetActive(false);
    oliveoilCanvas.SetActive(true);
}
//show the milk on the table and hide the one on the left
public void ShowMilkDinner()
{
    milk_picture.SetActive(true);
    milkCanvas.SetActive(false);
}
//show the milk on the left and hide the one on the table
public void hideMilkDinner()
{
    milk_picture.SetActive(false);
    milkCanvas.SetActive(true);
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο τρίτο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα

αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι.

Κώδικας ShowFoodThursday.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShowFoodThursday : MonoBehaviour
{
    //breakfast
    public GameObject wholegrains;
    public GameObject wholegrainsCanvas;
    public GameObject berries;
    public GameObject berriesCanvas;
    public GameObject milkshake;
    public GameObject milkshakeCanvas;
    public GameObject cookies;
    public GameObject cookiesCanvas;
    public GameObject sandwich;
    public GameObject sandwichCanvas;
    public GameObject chips;
    public GameObject chipsCanvas;
    public GameObject mochi;
    public GameObject mochiCanvas;
    public GameObject pear;
    public GameObject pearCanvas;
    public GameObject tomato;
    public GameObject tomatoCanvas;
    public GameObject water;
    public GameObject waterCanvas;

    //Lunch
    public GameObject fishLunch;
    public GameObject fishLunchCanvas;
    public GameObject chocolatesLunch;
    public GameObject chocolatesLunchCanvas;
    public GameObject potatoLunch;
    public GameObject potatoLunchCanvas;
    public GameObject oliveoilLunch;
    public GameObject oliveoilLunchCanvas;
    public GameObject nutellaLunch;
    public GameObject nutellaLunchCanvas;
    public GameObject waterLunch;
    public GameObject waterLunchCanvas;
    public GameObject greenbeansLunch;
    public GameObject greenbeansLunchCanvas;
    public GameObject marshmallowLunch;
    public GameObject marshmallowLunchCanvas;
    public GameObject spriteLunch;
    public GameObject spriteLunchCanvas;
    public GameObject cherryLunch;
    public GameObject cherryLunchCanvas;

    //Dinner
    public GameObject donutDinner;
```

```

public GameObject donutCanvasDinner;
public GameObject burgerDinner;
public GameObject burgerCanvasDinner;
public GameObject noodlesDinner;
public GameObject noodlesCanvasDinner;
public GameObject pancakesDinner;
public GameObject pancakesCanvasDinner;
public GameObject pomegDinner;
public GameObject pomegCanvasDinner;
public GameObject cornDinner;
public GameObject cornCanvasDinner;
public GameObject eggDinner;
public GameObject eggCanvasDinner;
public GameObject dessertDinner;
public GameObject dessertCanvasDinner;
public GameObject oliveoilDinner;
public GameObject oliveoilCanvasDinner;
public GameObject milkDinner;
public GameObject milkCanvasDinner;

```

// Start is called before the first frame update

```
void Start()
```

```
{
```

```
    //breakfast
```

```
    wholegrains.SetActive(false);
```

```
    berries.SetActive(false);
```

```
    milkshake.SetActive(false);
```

```
    cookies.SetActive(false);
```

```
    sandwich.SetActive(false);
```

```
    chips.SetActive(false);
```

```
    mochi.SetActive(false);
```

```
    pear.SetActive(false);
```

```
    tomato.SetActive(false);
```

```
    water.SetActive(false);
```

```
    //lunch
```

```
    fishLunch.SetActive(false);
```

```
    chocolatesLunch.SetActive(false);
```

```
    potatoLunch.SetActive(false);
```

```
    oliveoilLunch.SetActive(false);
```

```
    nutellaLunch.SetActive(false);
```

```
    waterLunch.SetActive(false);
```

```
    greenbeansLunch.SetActive(false);
```

```
    marshmallowLunch.SetActive(false);
```

```
    spriteLunch.SetActive(false);
```

```
    cherryLunch.SetActive(false);
```

```
    //dinner
```

```
    donutDinner.SetActive(false);
```

```
    burgerDinner.SetActive(false);
```

```
    noodlesDinner.SetActive(false);
```

```
    pancakesDinner.SetActive(false);
```

```
    pomegDinner.SetActive(false);
```

```
    cornDinner.SetActive(false);
```

```
    eggDinner.SetActive(false);
```

```
    dessertDinner.SetActive(false);
```

```
    oliveoilDinner.SetActive(false);
```

```
    milkDinner.SetActive(false);
```

```

    }
    //show the milk on the table and hide the one on the left
    public void ShowMilkDinner()
    {
        milkDinner.SetActive(true);
        milkCanvasDinner.SetActive(false);
    }
    //show the milk on the left and hide the one on the table
    public void hideMilkDinner()
    {
        milkDinner.SetActive(false);
        milkCanvasDinner.SetActive(true);
    }
    //show the olive oil on the table and hide the one on the left
    public void ShowOliveOilDinner()
    {
        oliveoilDinner.SetActive(true);
        oliveoilCanvasDinner.SetActive(false);
    }
    //show the olive oil on the left and hide the one on the table
    public void hideOliveOilDinner()
    {
        oliveoilDinner.SetActive(false);
        oliveoilCanvasDinner.SetActive(true);
    }
    //show the dessert on the table and hide the one on the left
    public void ShowDessertDinner()
    {
        dessertDinner.SetActive(true);
        dessertCanvasDinner.SetActive(false);
    }
    //show the dessert on the left and hide the one on the table
    public void hideDessertDinner()
    {
        dessertDinner.SetActive(false);
        dessertCanvasDinner.SetActive(true);
    }
    //show the egg on the table and hide the one on the left
    public void ShowEggDinner()
    {
        eggDinner.SetActive(true);
        eggCanvasDinner.SetActive(false);
    }
    //show the egg on the left and hide the one on the table
    public void hideEggDinner()
    {
        eggDinner.SetActive(false);
        eggCanvasDinner.SetActive(true);
    }
    //show the corn on the table and hide the one on the left
    public void ShowCornDinner()
    {
        cornDinner.SetActive(true);
        cornCanvasDinner.SetActive(false);
    }
    //show the corn on the left and hide the one on the table
    public void hideCornDinner()
    {
        cornDinner.SetActive(false);
    }

```

```

        cornCanvasDinner.SetActive(true);
    }
    //show the pomegr on the table and hide the one on the left
    public void ShowPomegDinner()
    {
        pomegDinner.SetActive(true);
        pomegCanvasDinner.SetActive(false);
    }
    //show the pomegr on the left and hide the one on the table
    public void hidePomegDinner()
    {
        pomegDinner.SetActive(false);
        pomegCanvasDinner.SetActive(true);
    }
    //show the pomegr on the table and hide the one on the left
    public void ShowPancakesDinner()
    {
        pancakesDinner.SetActive(true);
        pancakesCanvasDinner.SetActive(false);
    }
    //show the pomegr on the left and hide the one on the table
    public void hidePancakesDinner()
    {
        pancakesDinner.SetActive(false);
        pancakesCanvasDinner.SetActive(true);
    }
    //show the noodles on the table and hide the one on the left
    public void ShowNoodlesDinner()
    {
        noodlesDinner.SetActive(true);
        noodlesCanvasDinner.SetActive(false);
    }
    //show the noodles on the left and hide the one on the table
    public void hideNoodlesDinner()
    {
        noodlesDinner.SetActive(false);
        noodlesCanvasDinner.SetActive(true);
    }
    //show the burger on the table and hide the one on the left
    public void ShowBurgerDinner()
    {
        burgerDinner.SetActive(true);
        burgerCanvasDinner.SetActive(false);
    }
    //show the burger on the left and hide the one on the table
    public void hideBurgerDinner()
    {
        burgerDinner.SetActive(false);
        burgerCanvasDinner.SetActive(true);
    }

    //show the donut on the table and hide the one on the left
    public void ShowDonutDinner()
    {
        donutDinner.SetActive(true);
        donutCanvasDinner.SetActive(false);
    }
    //show the donut on the left and hide the one on the table
    public void hideDonutDinner()

```

```

    {
        donutDinner.SetActive(false);
        donutCanvasDinner.SetActive(true);
    }
//show the cherry on the table and hide the one on the left
public void ShowCherryLunch()
{
    cherryLunch.SetActive(true);
    cherryLunchCanvas.SetActive(false);
}
//show the cherry on the left and hide the one on the table
public void hideCherryLunch()
{
    cherryLunch.SetActive(false);
    cherryLunchCanvas.SetActive(true);
}
//show the soda on the table and hide the one on the left
public void ShowSpriteLunch()
{
    spriteLunch.SetActive(true);
    spriteLunchCanvas.SetActive(false);
}
//show the soda on the left and hide the one on the table
public void hideSpriteLunch()
{
    spriteLunch.SetActive(false);
    spriteLunchCanvas.SetActive(true);
}
//show the marshmallow on the table and hide the one on the left
public void ShowMarshmallowLunch()
{
    marshmallowLunch.SetActive(true);
    marshmallowLunchCanvas.SetActive(false);
}
//show the marshmallow on the left and hide the one on the table
public void hideMarshmallowLunch()
{
    marshmallowLunch.SetActive(false);
    marshmallowLunchCanvas.SetActive(true);
}
//show the green beans on the table and hide the one on the left
public void ShowGreenbeansLunch()
{
    greenbeansLunch.SetActive(true);
    greenbeansLunchCanvas.SetActive(false);
}
//show the green beans on the left and hide the one on the table
public void hideGreenbeansLunch()
{
    greenbeansLunch.SetActive(false);
    greenbeansLunchCanvas.SetActive(true);
}
//show the water on the table and hide the one on the left
public void ShowWaterLunch()
{
    waterLunch.SetActive(true);
    waterLunchCanvas.SetActive(false);
}
//show the water on the left and hide the one on the table

```

```

public void hideWaterLunch()
{
    waterLunch.SetActive(false);
    waterLunchCanvas.SetActive(true);
}
//show the nutella on the table and hide the one on the left
public void ShowNutellaLunch()
{
    nutellaLunch.SetActive(true);
    nutellaLunchCanvas.SetActive(false);
}
//show the nutella on the left and hide the one on the table
public void hideNutellaLunch()
{
    nutellaLunch.SetActive(false);
    nutellaLunchCanvas.SetActive(true);
}
//show the olive oil on the table and hide the one on the left
public void ShowOliveOilLunch()
{
    oliveoilLunch.SetActive(true);
    oliveoilLunchCanvas.SetActive(false);
}
//show the olive oil on the left and hide the one on the table
public void hideOliveOilLunch()
{
    oliveoilLunch.SetActive(false);
    oliveoilLunchCanvas.SetActive(true);
}
//show the potato on the table and hide the one on the left
public void ShowPotatoLunch()
{
    potatoLunch.SetActive(true);
    potatoLunchCanvas.SetActive(false);
}
//show the potato on the left and hide the one on the table
public void hidePotatoLunch()
{
    potatoLunch.SetActive(false);
    potatoLunchCanvas.SetActive(true);
}
//show the chocolates on the table and hide the one on the left
public void ShowChocolatesLunch()
{
    chocolatesLunch.SetActive(true);
    chocolatesLunchCanvas.SetActive(false);
}
//show the chocolates on the left and hide the one on the table
public void hideChocolatesLunch()
{
    chocolatesLunch.SetActive(false);
    chocolatesLunchCanvas.SetActive(true);
}
//show the fish on the table and hide the one on the left
public void ShowFishLunch()
{
    fishLunch.SetActive(true);
    fishLunchCanvas.SetActive(false);
}

```

```

//show the fish on the left and hide the one on the table
public void hideFishLunch()
{
    fishLunch.SetActive(false);
    fishLunchCanvas.SetActive(true);
}

//show the water on the table and hide the one on the left
public void ShowWater()
{
    water.SetActive(true);
    waterCanvas.SetActive(false);
}

//show the water on the left and hide the one on the table
public void hideWater()
{
    water.SetActive(false);
    waterCanvas.SetActive(true);
}

//show the tomato on the table and hide the one on the left
public void ShowTomato()
{
    tomato.SetActive(true);
    tomatoCanvas.SetActive(false);
}

//show the tomato on the left and hide the one on the table
public void hideTomato()
{
    tomato.SetActive(false);
    tomatoCanvas.SetActive(true);
}

//show the pear on the table and hide the one on the left
public void ShowPear()
{
    pear.SetActive(true);
    pearCanvas.SetActive(false);
}

//show the pear on the left and hide the one on the table
public void hidePear()
{
    pear.SetActive(false);
    pearCanvas.SetActive(true);
}

//show the mochi on the table and hide the one on the left
public void ShowMochi()
{
    mochi.SetActive(true);
    mochiCanvas.SetActive(false);
}

//show the mochi on the left and hide the one on the table
public void hideMochi()
{
    mochi.SetActive(false);
    mochiCanvas.SetActive(true);
}

//show the chips on the table and hide the one on the left
public void ShowChips()
{
    chips.SetActive(true);
}

```

```

        chipsCanvas.SetActive(false);
    }
    //show the chips on the left and hide the one on the table
    public void hideChips()
    {
        chips.SetActive(false);
        chipsCanvas.SetActive(true);
    }
    //show the sandwich on the table and hide the one on the left
    public void ShowSandwich()
    {
        sandwich.SetActive(true);
        sandwichCanvas.SetActive(false);
    }
    //show the sandwich on the left and hide the one on the table
    public void hideSandwich()
    {
        sandwich.SetActive(false);
        sandwichCanvas.SetActive(true);
    }
    //show the cookies on the table and hide the one on the left
    public void ShowCookies()
    {
        cookies.SetActive(true);
        cookiesCanvas.SetActive(false);
    }
    //show the cookies on the left and hide the one on the table
    public void hideCookies()
    {
        cookies.SetActive(false);
        cookiesCanvas.SetActive(true);
    }
    //show the milkshake on the table and hide the one on the left
    public void ShowMilkshake()
    {
        milkshake.SetActive(true);
        milkshakeCanvas.SetActive(false);
    }
    //show the milkshake on the left and hide the one on the table
    public void hideMilkshake()
    {
        milkshake.SetActive(false);
        milkshakeCanvas.SetActive(true);
    }
    //show the berries on the table and hide the one on the left
    public void ShowBerries()
    {
        berries.SetActive(true);
        berriesCanvas.SetActive(false);
    }
    //show the berries on the left and hide the one on the table
    public void hideBerries()
    {
        berries.SetActive(false);
        berriesCanvas.SetActive(true);
    }
    //show the whole grains on the table and hide the one on the left
    public void ShowWholeGrains()
    {

```

```

        wholegrains.SetActive(true);
        wholegrainsCanvas.SetActive(false);
    }

//show the whole grains on the left and hide the one on the table
public void hideWholeGrains()
{
    wholegrains.SetActive(false);
    wholegrainsCanvas.SetActive(true);
}

// Update is called once per frame
void Update()
{

}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο τέταρτο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι.

Κώδικας ShowFoodFriday.cs :

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShowFoodFriday : MonoBehaviour
{
    //breakfast
    public GameObject yogurt;
    public GameObject yogurtCanvas;
    public GameObject icecream;
    public GameObject icecreamCanvas;
    public GameObject peach;
    public GameObject peachCanvas;
    public GameObject egg;
    public GameObject eggCanvas;
    public GameObject sandwich;
    public GameObject sandwichCanvas;
    public GameObject lemon;
    public GameObject lemonCanvas;
    public GameObject oatmeal;
    public GameObject oatmealCanvas;
    public GameObject pepper;
    public GameObject pepperCanvas;
    public GameObject cake;
}

```

```

public GameObject cakeCanvas;
public GameObject juice;
public GameObject juiceCanvas;

//lunch
public GameObject potatoesLunch;
public GameObject potatoesCanvasLunch;
public GameObject veggiesLunch;
public GameObject veggiesCanvasLunch;
public GameObject strawberryLunch;
public GameObject strawberryCanvasLunch;
public GameObject oliveoilLunch;
public GameObject oliveoilCanvasLunch;
public GameObject cocacolaLunch;
public GameObject cocaCanvasLunch;
public GameObject icecreamLunch;
public GameObject icecreamCanvasLunch;
public GameObject dessertLunch;
public GameObject dessertCanvasLunch;
public GameObject chickenLunch;
public GameObject chickenCanvasLunch;
public GameObject saladLunch;
public GameObject saladCanvasLunch;
public GameObject peasLunch;
public GameObject peasCanvasLunch;

//dinner
public GameObject saladDinner;
public GameObject saladCanvasDinner;
public GameObject cheesecakeDinner;
public GameObject cheesecakeCanvasDinner;
public GameObject biscuitsDinner;
public GameObject biscuitsCanvasDinner;
public GameObject friedChickenDinner;
public GameObject friedChickenCanvasDinner;
public GameObject orangeDinner;
public GameObject orangeCanvasDinner;
public GameObject milkDinner;
public GameObject milkCanvasDinner;
public GameObject donutDinner;
public GameObject donutCanvasDinner;
public GameObject souvlakiaDinner;
public GameObject souvlakiaCanvasDinner;
public GameObject yogurtDinner;
public GameObject yogurtCanvasDinner;
public GameObject berriesDinner;
public GameObject berriesCanvasDinner;

// Start is called before the first frame update
void Start()
{
    //breakfast
    yogurt.SetActive(false);
    icecream.SetActive(false);
    peach.SetActive(false);
    egg.SetActive(false);

```

```

sandwich.SetActive(false);
lemon.SetActive(false);
oatmeal.SetActive(false);
pepper.SetActive(false);
cake.SetActive(false);
juice.SetActive(false);

//lunch
potatoesLunch.SetActive(false);
veggiesLunch.SetActive(false);
strawberryLunch.SetActive(false);
oliveoilLunch.SetActive(false);
cocacolaLunch.SetActive(false);
icecreamLunch.SetActive(false);
dessertLunch.SetActive(false);
chickenLunch.SetActive(false);
saladLunch.SetActive(false);
peasLunch.SetActive(false);

//dinner
saladDinner.SetActive(false);
cheesecakeDinner.SetActive(false);
biscuitsDinner.SetActive(false);
friedChickenDinner.SetActive(false);
orangeDinner.SetActive(false);
milkDinner.SetActive(false);
donutDinner.SetActive(false);
souvlaKiaDinner.SetActive(false);
yogurtDinner.SetActive(false);
berriesDinner.SetActive(false);
}
//show the berries on the table and hide the one on the left
public void ShowBerriesDinner()
{
    berriesDinner.SetActive(true);
    berriesCanvasDinner.SetActive(false);
}
//show the berries on the left and hide the one on the table
public void HideDinnerberries()
{
    berriesDinner.SetActive(false);
    berriesCanvasDinner.SetActive(true);
}
//show the yogurt on the table and hide the one on the left
public void ShowYogurtDinner()
{
    yogurtDinner.SetActive(true);
    yogurtCanvasDinner.SetActive(false);
}
//show the yogurt on the left and hide the one on the table
public void HideDinneryogurt()
{
    yogurtDinner.SetActive(false);
    yogurtCanvasDinner.SetActive(true);
}
//show the souvlakia on the table and hide the one on the left
public void ShowSouvlakiaDinner()
{
    souvlakiaDinner.SetActive(true);

```

```

        souvlakiaCanvasDinner.SetActive(false);
    }
    //show the souvlakia on the left and hide the one on the table
    public void HideDinnersouvlakia()
    {
        souvlakiaDinner.SetActive(false);
        souvlakiaCanvasDinner.SetActive(true);
    }
    //show the donut on the table and hide the one on the left
    public void ShowDonutDinner()
    {
        donutDinner.SetActive(true);
        donutCanvasDinner.SetActive(false);
    }
    //show the donut on the left and hide the one on the table
    public void HideDinnerdonut()
    {
        donutDinner.SetActive(false);
        donutCanvasDinner.SetActive(true);
    }
    //show the milk on the table and hide the one on the left
    public void ShowMilkDinner()
    {
        milkDinner.SetActive(true);
        milkCanvasDinner.SetActive(false);
    }
    //show the milk on the left and hide the one on the table
    public void HideDinnermilk()
    {
        milkDinner.SetActive(false);
        milkCanvasDinner.SetActive(true);
    }
    //show the orange on the table and hide the one on the left
    public void ShowOrangeDinner()
    {
        orangeDinner.SetActive(true);
        orangeCanvasDinner.SetActive(false);
    }
    //show the orange on the left and hide the one on the table
    public void HideDinnerorange()
    {
        orangeDinner.SetActive(false);
        orangeCanvasDinner.SetActive(true);
    }
    //show the chicken on the table and hide the one on the left
    public void ShowFriedChicken()
    {
        friedChickenDinner.SetActive(true);
        friedChickenCanvasDinner.SetActive(false);
    }
    //show the chicken on the left and hide the one on the table
    public void HideDinnerfriedchicken()
    {
        friedChickenDinner.SetActive(false);
        friedChickenCanvasDinner.SetActive(true);
    }
    //show the biscuits on the table and hide the one on the left
    public void ShowBiscuitsDinner()
    {

```

```

        biscuitsDinner.SetActive(true);
        biscuitsCanvasDinner.SetActive(false);
    }

    //show the biscuits on the left and hide the one on the table
    public void HideDinnerbiscuits()
    {
        biscuitsDinner.SetActive(false);
        biscuitsCanvasDinner.SetActive(true);
    }

    //show the cheeseckae on the table and hide the one on the left
    public void ShowCheeseCakeDinner()
    {
        cheesecakeDinner.SetActive(true);
        cheesecakeCanvasDinner.SetActive(false);
    }

    //show the cheesecake on the left and hide the one on the table
    public void HideDinnercheesecake()
    {
        cheesecakeDinner.SetActive(false);
        cheesecakeCanvasDinner.SetActive(true);
    }

    //show the salad on the table and hide the one on the left
    public void ShowSaladDinner()
    {
        saladDinner.SetActive(true);
        saladCanvasDinner.SetActive(false);
    }

    //show the salad on the left and hide the one on the table
    public void HideDinnersalad()
    {
        saladDinner.SetActive(false);
        saladCanvasDinner.SetActive(true);
    }

    //show the peas on the table and hide the one on the left
    public void ShowPeasLunch()
    {
        peasLunch.SetActive(true);
        peasCanvasLunch.SetActive(false);
    }

    //show the peas on the left and hide the one on the table
    public void HideLunchpeas()
    {
        peasLunch.SetActive(false);
        peasCanvasLunch.SetActive(true);
    }

    //show the salad on the table and hide the one on the left
    public void ShowSaladLunch()
    {
        saladLunch.SetActive(true);
        saladCanvasLunch.SetActive(false);
    }

    //show the salad on the left and hide the one on the table
    public void HideLunchsalad()
    {
        saladLunch.SetActive(false);
        saladCanvasLunch.SetActive(true);
    }

    //show the chicken on the table and hide the one on the left

```

```

public void ShowChickenLunch()
{
    chickenLunch.SetActive(true);
    chickenCanvasLunch.SetActive(false);
}
//show the chicken on the left and hide the one on the table
public void HideLunchchicken()
{
    chickenLunch.SetActive(false);
    chickenCanvasLunch.SetActive(true);
}
//show the dessert on the table and hide the one on the left
public void ShowDessertLunch()
{
    dessertLunch.SetActive(true);
    dessertCanvasLunch.SetActive(false);
}
//show the dessert on the left and hide the one on the table
public void HideLunchdessert()
{
    dessertLunch.SetActive(false);
    dessertCanvasLunch.SetActive(true);
}
//show the ice cream on the table and hide the one on the left
public void ShowIcecreamLunch()
{
    icecreamLunch.SetActive(true);
    icecreamCanvasLunch.SetActive(false);
}
//show the ice cream on the left and hide the one on the table
public void HideLunchicecream()
{
    icecreamLunch.SetActive(false);
    icecreamCanvasLunch.SetActive(true);
}
//show the coca cola on the table and hide the one on the left
public void ShowCocaColaLunch()
{
    cocacolaLunch.SetActive(true);
    cocaCanvasLunch.SetActive(false);
}
//show the coca cola on the left and hide the one on the table
public void HideLunchcocacola()
{
    cocacolaLunch.SetActive(false);
    cocaCanvasLunch.SetActive(true);
}
//show the olive oil on the table and hide the one on the left
public void ShowOliveoilLunch()
{
    oliveoilLunch.SetActive(true);
    oliveoilCanvasLunch.SetActive(false);
}
//show the olive oil on the left and hide the one on the table
public void HideLuncholiveoil()
{
    oliveoilLunch.SetActive(false);
    oliveoilCanvasLunch.SetActive(true);
}

```

```

//show the strawberry on the table and hide the one on the left
public void ShowStrawberryLunch()
{
    strawberryLunch.SetActive(true);
    strawberryCanvasLunch.SetActive(false);
}
//show the strawberry on the left and hide the one on the table
public void HideLunchstrawberry()
{
    strawberryLunch.SetActive(false);
    strawberryCanvasLunch.SetActive(true);
}
//show the veggies on the table and hide the one on the left
public void ShowVeggiesLunch()
{
    veggiesLunch.SetActive(true);
    veggiesCanvasLunch.SetActive(false);
}
//show the veggies on the left and hide the one on the table
public void HideLunchveggies()
{
    veggiesLunch.SetActive(false);
    veggiesCanvasLunch.SetActive(true);
}
//show the potato on the table and hide the one on the left
public void ShowPotatoesLunch()
{
    potatoesLunch.SetActive(true);
    potatoesCanvasLunch.SetActive(false);
}
//show the potato on the left and hide the one on the table
public void HideLunchpotatoes()
{
    potatoesLunch.SetActive(false);
    potatoesCanvasLunch.SetActive(true);
}
//show the juice on the table and hide the one on the left
public void ShowJuice()
{
    juice.SetActive(true);
    juiceCanvas.SetActive(false);
}
//show the juice on the left and hide the one on the table
public void Hidejuice()
{
    juice.SetActive(false);
    juiceCanvas.SetActive(true);
}
//show the cake on the table and hide the one on the left
public void ShowCake()
{
    cake.SetActive(true);
    cakeCanvas.SetActive(false);
}
//show the cake on the left and hide the one on the table
public void Hidecake()
{
    cake.SetActive(false);
    cakeCanvas.SetActive(true);
}

```

```

    }
    //show the pepper on the table and hide the one on the left
    public void ShowPepper()
    {
        pepper.SetActive(true);
        pepperCanvas.SetActive(false);
    }
    //show the pepper on the left and hide the one on the table
    public void Hidepepper()
    {
        pepper.SetActive(false);
        pepperCanvas.SetActive(true);
    }
    //show the oatmeal on the table and hide the one on the left
    public void ShowOatmeal()
    {
        oatmeal.SetActive(true);
        oatmealCanvas.SetActive(false);
    }

    //show the oatmeal on the left and hide the one on the table
    public void Hideoatmeal()
    {
        oatmeal.SetActive(false);
        oatmealCanvas.SetActive(true);
    }
    //show the lemon on the table and hide the one on the left
    public void ShowLemon()
    {
        lemon.SetActive(true);
        lemonCanvas.SetActive(false);
    }

    //show the lemon on the left and hide the one on the table
    public void HideLemon()
    {
        lemon.SetActive(false);
        lemonCanvas.SetActive(true);
    }
    //show the sandwich on the table and hide the one on the left
    public void ShowSandwich()
    {
        sandwich.SetActive(true);
        sandwichCanvas.SetActive(false);
    }
    //show the sandwich on the left and hide the one on the table
    public void Hidesandwich()
    {
        sandwich.SetActive(false);
        sandwichCanvas.SetActive(true);
    }
    //show the egg on the table and hide the one on the left
    public void ShowEgg()
    {
        egg.SetActive(true);
        eggCanvas.SetActive(false);
    }
    //show the egg on the left and hide the one on the table
    public void Hideegg()

```

```

    {
        egg.SetActive(false);
        eggCanvas.SetActive(true);
    }
//show the peach on the table and hide the one on the left
public void ShowPeach()
{
    peach.SetActive(true);
    peachCanvas.SetActive(false);
}
//show the peach on the left and hide the one on the table
public void Hidepeach()
{
    peach.SetActive(false);
    peachCanvas.SetActive(true);
}
//show the ice cream on the table and hide the one on the left
public void Showicecream()
{
    icecream.SetActive(true);
    icecreamCanvas.SetActive(false);
}
//show the ice cream on the left and hide the one on the table
public void Hideicecream()
{
    icecream.SetActive(false);
    icecreamCanvas.SetActive(true);
}
//show the yogurt on the table and hide the one on the left
public void ShowYogurt()
{
    yogurt.SetActive(true);
    yogurtCanvas.SetActive(false);
}

//show the yogurt on the left and hide the one on the table
public void HideYogurt()
{
    yogurt.SetActive(false);
    yogurtCanvas.SetActive(true);
}

// Update is called once per frame
void Update()
{
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο πέμπτο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα

αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι.

Κώδικας FoodSaturday.cs :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FoodSunday : MonoBehaviour
{
    //breakfast
    public GameObject smoothie;
    public GameObject smoothieCanvas;
    public GameObject cake;
    public GameObject cakeCanvas;
    public GameObject cheese;
    public GameObject cheeseCanvas;
    public GameObject tomato;
    public GameObject tomatoCanvas;
    public GameObject cucumber;
    public GameObject cucumberCanvas;
    public GameObject milk;
    public GameObject milkCanvas;
    public GameObject chocolates;
    public GameObject chocolatesCanvas;
    public GameObject avocadotoast;
    public GameObject avocadotoastCanvas;
    public GameObject yogurt;
    public GameObject yogurtCanvas;
    public GameObject toast;
    public GameObject toastCanvas;

    //lunch
    public GameObject octopusLunch;
    public GameObject octopusCanvasLunch;
    public GameObject burgerLunch;
    public GameObject burgerCanvasLunch;
    public GameObject pepperLunch;
    public GameObject pepperCanvasLunch;
    public GameObject oliveoilLunch;
    public GameObject oliveoilCanvasLunch;
    public GameObject cornLunch;
    public GameObject cornCanvasLunch;
    public GameObject saladLunch;
    public GameObject saladCanvasLunch;
    public GameObject orangeLunch;
    public GameObject orangeCanvasLunch;
    public GameObject chickenLunch;
    public GameObject chickenCanvasLunch;
    public GameObject icecreamLunch;
    public GameObject icecreamCanvasLunch;
    public GameObject potatoLunch;
    public GameObject potatoCanvasLunch;

    //dinner
    public GameObject darkchocolateDinner;
    public GameObject darkchocolateCanvasDinner;
```

```

public GameObject pastaDinner;
public GameObject pastaCanvasDinner;
public GameObject spriteDinner;
public GameObject spriteCanvasDinner;
public GameObject cookiesDinner;
public GameObject cookiesCanvasDinner;
public GameObject cucumberDinner;
public GameObject cucumberCanvasDinner;
public GameObject marshmallowDinner;
public GameObject marshmallowCanvasDinner;
public GameObject milkDinner;
public GameObject milkCanvasDinner;
public GameObject milkshakeDinner;
public GameObject milkshakeCanvasDinner;
public GameObject cheeseDinner;
public GameObject cheeseCanvasDinner;
public GameObject waterDinner;
public GameObject waterCanvasDinner;

```

// Start is called before the first frame update

```

void Start()
{
    //breakfast
    smoothie.SetActive(false);
    cake.SetActive(false);
    cheese.SetActive(false);
    tomato.SetActive(false);
    cucumber.SetActive(false);
    milk.SetActive(false);
    chocolates.SetActive(false);
    avocadotoast.SetActive(false);
    yogurt.SetActive(false);
    toast.SetActive(false);

    //lunch
    octopusLunch.SetActive(false);
    burgerLunch.SetActive(false);
    pepperLunch.SetActive(false);
    oliveoilLunch.SetActive(false);
    cornLunch.SetActive(false);
    saladLunch.SetActive(false);
    orangeLunch.SetActive(false);
    chickenLunch.SetActive(false);
    icecreamLunch.SetActive(false);
    potatoLunch.SetActive(false);

    //dinner
    darkchocolateDinner.SetActive(false);
    pastaDinner.SetActive(false);
    spriteDinner.SetActive(false);
    cookiesDinner.SetActive(false);
    cucumberDinner.SetActive(false);
    marshmallowDinner.SetActive(false);
    milkDinner.SetActive(false);
    cheeseDinner.SetActive(false);
    waterDinner.SetActive(false);
}

```

```

        milkshakeDinner.SetActive(false);
    }
    //show the smoothie on the table and hide the one on the left
    public void ShowSmoothie()
    {
        smoothie.SetActive(true);
        smoothieCanvas.SetActive(false);
    }
    //show the smoothie on the left and hide the one on the table
    public void HideSmoothie()
    {
        smoothie.SetActive(false);
        smoothieCanvas.SetActive(true);
    }

    //show the cake on the table and hide the one on the left
    public void ShowCake()
    {
        cake.SetActive(true);
        cakeCanvas.SetActive(false);
    }
    //show the cake on the left and hide the one on the table
    public void HideCake()
    {
        cake.SetActive(false);
        cakeCanvas.SetActive(true);
    }

    //show the cheese on the table and hide the one on the left
    public void Showcheese()
    {
        cheese.SetActive(true);
        cheeseCanvas.SetActive(false);
    }
    //show the cheese on the left and hide the one on the table
    public void HideCheese()
    {
        cheese.SetActive(false);
        cheeseCanvas.SetActive(true);
    }

    //show the tomato on the table and hide the one on the left
    public void Showtomato()
    {
        tomato.SetActive(true);
        tomatoCanvas.SetActive(false);
    }
    //show the tomato on the left and hide the one on the table
    public void Hidetomato()
    {
        tomato.SetActive(false);
        tomatoCanvas.SetActive(true);
    }

    //show the cucmber on the table and hide the one on the left
    public void Showcucumber()
    {
        cucumber.SetActive(true);
        cucumberCanvas.SetActive(false);
    }
    //show the cucumber on the left and hide the one on the table

```

```

public void Hidecucumber()
{
    cucumber.SetActive(false);
    cucumberCanvas.SetActive(true);
}
//show the milk on the table and hide the one on the left
public void Showmilk()
{
    milk.SetActive(true);
    milkCanvas.SetActive(false);
}
//show the milk on the left and hide the one on the table
public void Hidemilk()
{
    milk.SetActive(false);
    milkCanvas.SetActive(true);
}
//show the chocolates on the table and hide the one on the left
public void Showchocolates()
{
    chocolates.SetActive(true);
    chocolatesCanvas.SetActive(false);
}
//show the chocolates on the left and hide the one on the table
public void HideChocolates()
{
    chocolates.SetActive(false);
    chocolatesCanvas.SetActive(true);
}
//show the avocado on the table and hide the one on the left
public void Showavocadotoast()
{
    avocadotoast.SetActive(true);
    avocadotoastCanvas.SetActive(false);
}
//show the avocado on the left and hide the one on the table
public void Hideavocadotoast()
{
    avocadotoast.SetActive(false);
    avocadotoastCanvas.SetActive(true);
}
//show the yogurt on the table and hide the one on the left
public void Showyogurt()
{
    yogurt.SetActive(true);
    yogurtCanvas.SetActive(false);
}
//show the yogurt on the left and hide the one on the table
public void Hideyogurt()
{
    yogurt.SetActive(false);
    yogurtCanvas.SetActive(true);
}
//show the toast on the table and hide the one on the left
public void Showcatoast()
{
    toast.SetActive(true);
    toastCanvas.SetActive(false);
}

```

```

//show the toast on the left and hide the one on the table
public void Hidetoast()
{
    toast.SetActive(false);
    toastCanvas.SetActive(true);
}
//show the octopus on the table and hide the one on the left
public void ShowLunchoctopus()
{
    octopusLunch.SetActive(true);
    octopusCanvasLunch.SetActive(false);
}
//show the octopus on the left and hide the one on the table
public void HideLunchoctopus()
{
    octopusLunch.SetActive(false);
    octopusCanvasLunch.SetActive(true);
}
//show the burger on the table and hide the one on the left
public void ShowLunchburger()
{
    burgerLunch.SetActive(true);
    burgerCanvasLunch.SetActive(false);
}
//show the burger on the left and hide the one on the table
public void HideLunchburger()
{
    burgerLunch.SetActive(false);
    burgerCanvasLunch.SetActive(true);
}
//show the pepper on the table and hide the one on the left
public void ShowLunchpepper()
{
    pepperLunch.SetActive(true);
    pepperCanvasLunch.SetActive(false);
}
//show the pepper on the left and hide the one on the table
public void HideLunchpepper()
{
    pepperLunch.SetActive(false);
    pepperCanvasLunch.SetActive(true);
}
//show the olive oil on the table and hide the one on the left
public void ShowLuncholiveoil()
{
    oliveoilLunch.SetActive(true);
    oliveoilCanvasLunch.SetActive(false);
}
//show the olive oil on the left and hide the one on the table
public void HideLuncholiveoil()
{
    oliveoilLunch.SetActive(false);
    oliveoilCanvasLunch.SetActive(true);
}
//show the corn on the table and hide the one on the left
public void ShowLunchcorn()
{
    cornLunch.SetActive(true);
    cornCanvasLunch.SetActive(false);
}

```

```

    }
    //show the corn on the left and hide the one on the table
    public void HideLunchcorn()
    {
        cornLunch.SetActive(false);
        cornCanvasLunch.SetActive(true);
    }
    //show the salad on the table and hide the one on the left
    public void ShowLunchsalad()
    {
        saladLunch.SetActive(true);
        saladCanvasLunch.SetActive(false);
    }
    //show the salad on the left and hide the one on the table
    public void HideLunchsalad()
    {
        saladLunch.SetActive(false);
        saladCanvasLunch.SetActive(true);
    }
    //show the orange on the table and hide the one on the left
    public void ShowLunchorange()
    {
        orangeLunch.SetActive(true);
        orangeCanvasLunch.SetActive(false);
    }
    //show the orange on the left and hide the one on the table
    public void HideLunchorange()
    {
        orangeLunch.SetActive(false);
        orangeCanvasLunch.SetActive(true);
    }
    //show the chicken on the table and hide the one on the left
    public void ShowLunchchicken()
    {
        chickenLunch.SetActive(true);
        chickenCanvasLunch.SetActive(false);
    }
    //show the chicken on the left and hide the one on the table
    public void HideLunchchicken()
    {
        chickenLunch.SetActive(false);
        chickenCanvasLunch.SetActive(true);
    }
    //show the ice cream on the table and hide the one on the left
    public void ShowLunchicecream()
    {
        icecreamLunch.SetActive(true);
        icecreamCanvasLunch.SetActive(false);
    }
    //show the ice cream on the left and hide the one on the table
    public void HideLunchicecream()
    {
        icecreamLunch.SetActive(false);
        icecreamCanvasLunch.SetActive(true);
    }
    //show the potato on the table and hide the one on the left
    public void ShowLunchpotato()
    {
        potatoLunch.SetActive(true);
    }

```

```

        potatoCanvasLunch.SetActive(false);
    }
    //show the potato on the left and hide the one on the table
    public void HideLunchpotato()
    {
        potatoLunch.SetActive(false);
        potatoCanvasLunch.SetActive(true);
    }
    //show the dark chocolate on the table and hide the one on the left
    public void ShowDinnerdarkchocolate()
    {
        darkchocolateDinner.SetActive(true);
        darkchocolateCanvasDinner.SetActive(false);
    }
    //show the dark chocolate on the left and hide the one on the table
    public void HideDinnerdarkchocolate()
    {
        darkchocolateDinner.SetActive(false);
        darkchocolateCanvasDinner.SetActive(true);
    }
    //show the pasta on the table and hide the one on the left
    public void ShowDinnerPasta()
    {
        pastaDinner.SetActive(true);
        pastaCanvasDinner.SetActive(false);
    }
    //show the pasta on the left and hide the one on the table
    public void HideDinnerpasta()
    {
        pastaDinner.SetActive(false);
        pastaCanvasDinner.SetActive(true);
    }
    //show the sprite on the table and hide the one on the left
    public void ShowDinnersprite()
    {
        spriteDinner.SetActive(true);
        spriteCanvasDinner.SetActive(false);
    }
    //show the sprite on the left and hide the one on the table
    public void HideDinnersprite()
    {
        spriteDinner.SetActive(false);
        spriteCanvasDinner.SetActive(true);
    }
    //show the cookies on the table and hide the one on the left
    public void ShowDinnercookies()
    {
        cookiesDinner.SetActive(true);
        cookiesCanvasDinner.SetActive(false);
    }
    //show the cookies on the left and hide the one on the table
    public void HideDinnercookies()
    {
        cookiesDinner.SetActive(false);
        cookiesCanvasDinner.SetActive(true);
    }
    //show the cucumber on the table and hide the one on the left
    public void ShowDinnercucumber()
    {

```

```

        cucumberDinner.SetActive(true);
        cucumberCanvasDinner.SetActive(false);
    }
    //show the cucumber on the left and hide the one on the table
    public void HideDinnercucumber()
    {
        cucumberDinner.SetActive(false);
        cucumberCanvasDinner.SetActive(true);
    }
    //show the marshmallow on the table and hide the one on the left
    public void ShowDinnermarshmallow()
    {
        marshmallowDinner.SetActive(true);
        marshmallowCanvasDinner.SetActive(false);
    }
    //show the marshmallow on the left and hide the one on the table
    public void HideDinnermarshmallow()
    {
        marshmallowDinner.SetActive(false);
        marshmallowCanvasDinner.SetActive(true);
    }
    //show the milk on the table and hide the one on the left
    public void ShowDinnermilk()
    {
        milkDinner.SetActive(true);
        milkCanvasDinner.SetActive(false);
    }
    //show the milk on the left and hide the one on the table
    public void HideDinnermilk()
    {
        milkDinner.SetActive(false);
        milkCanvasDinner.SetActive(true);
    }
    //show the milkshake on the table and hide the one on the left
    public void ShowDinnermilkshake()
    {
        milkshakeDinner.SetActive(true);
        milkshakeCanvasDinner.SetActive(false);
    }
    //show the milkshake on the left and hide the one on the table
    public void HideDinnermilkshake()
    {
        milkshakeDinner.SetActive(false);
        milkshakeCanvasDinner.SetActive(true);
    }
    //show the cheese on the table and hide the one on the left
    public void ShowDinnercheese()
    {
        cheeseDinner.SetActive(true);
        cheeseCanvasDinner.SetActive(false);
    }
    //show the cheese on the left and hide the one on the table
    public void HideDinnercheese()
    {
        cheeseDinner.SetActive(false);
        cheeseCanvasDinner.SetActive(true);
    }
    //show the water on the table and hide the one on the left
    public void ShowDinnerwater()

```

```

    {
        waterDinner.SetActive(true);
        waterCanvasDinner.SetActive(false);
    }

//show the water on the left and hide the one on the table
public void HideDinnerwater()
{
    waterDinner.SetActive(false);
    waterCanvasDinner.SetActive(true);
}

// Update is called once per frame
void Update()
{
}
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο έκτο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι.

Κώδικας FoodSunday.cs :

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FoodSunday : MonoBehaviour
{
    //breakfast
    public GameObject smoothie;
    public GameObject smoothieCanvas;
    public GameObject cake;
    public GameObject cakeCanvas;
    public GameObject cheese;
    public GameObject cheeseCanvas;
    public GameObject tomato;
    public GameObject tomatoCanvas;
    public GameObject cucumber;
    public GameObject cucumberCanvas;
    public GameObject milk;
    public GameObject milkCanvas;
    public GameObject chocolates;
    public GameObject chocolatesCanvas;
    public GameObject avocadotoast;
    public GameObject avocadotoastCanvas;
    public GameObject yogurt;
}

```

```

public GameObject yogurtCanvas;
public GameObject toast;
public GameObject toastCanvas;

//lunch
public GameObject octopusLunch;
public GameObject octopusCanvasLunch;
public GameObject burgerLunch;
public GameObject burgerCanvasLunch;
public GameObject pepperLunch;
public GameObject pepperCanvasLunch;
public GameObject oliveoilLunch;
public GameObject oliveoilCanvasLunch;
public GameObject cornLunch;
public GameObject cornCanvasLunch;
public GameObject saladLunch;
public GameObject saladCanvasLunch;
public GameObject orangeLunch;
public GameObject orangeCanvasLunch;
public GameObject chickenLunch;
public GameObject chickenCanvasLunch;
public GameObject icecreamLunch;
public GameObject icecreamCanvasLunch;
public GameObject potatoLunch;
public GameObject potatoCanvasLunch;

//dinner
public GameObject darkchocolateDinner;
public GameObject darkchocolateCanvasDinner;
public GameObject pastaDinner;
public GameObject pastaCanvasDinner;
public GameObject spriteDinner;
public GameObject spriteCanvasDinner;
public GameObject cookiesDinner;
public GameObject cookiesCanvasDinner;
public GameObject cucumberDinner;
public GameObject cucumberCanvasDinner;
public GameObject marshmallowDinner;
public GameObject marshmallowCanvasDinner;
public GameObject milkDinner;
public GameObject milkCanvasDinner;
public GameObject milkshakeDinner;
public GameObject milkshakeCanvasDinner;
public GameObject cheeseDinner;
public GameObject cheeseCanvasDinner;
public GameObject waterDinner;
public GameObject waterCanvasDinner;

// Start is called before the first frame update
void Start()
{
    //breakfast
    smoothie.SetActive(false);
    cake.SetActive(false);
    cheese.SetActive(false);
    tomato.SetActive(false);

```

```

cucumber.SetActive(false);
milk.SetActive(false);
chocolates.SetActive(false);
avocadotoast.SetActive(false);
yogurt.SetActive(false);
toast.SetActive(false);

//lunch
octopusLunch.SetActive(false);
burgerLunch.SetActive(false);
pepperLunch.SetActive(false);
oliveoilLunch.SetActive(false);
cornLunch.SetActive(false);
saladLunch.SetActive(false);
orangeLunch.SetActive(false);
chickenLunch.SetActive(false);
icecreamLunch.SetActive(false);
potatoLunch.SetActive(false);

//dinner
darkchocolateDinner.SetActive(false);
pastaDinner.SetActive(false);
spriteDinner.SetActive(false);
cookiesDinner.SetActive(false);
cucumberDinner.SetActive(false);
marshmallowDinner.SetActive(false);
milkDinner.SetActive(false);
cheeseDinner.SetActive(false);
waterDinner.SetActive(false);
milkshakeDinner.SetActive(false);

}
//show smoothie on the table
public void ShowSmoothie()
{
    smoothie.SetActive(true);
    smoothieCanvas.SetActive(false);
}
//hide smoothie from the table
public void HideSmoothie()
{
    smoothie.SetActive(false);
    smoothieCanvas.SetActive(true);
}

//show cake on the table
public void ShowCake()
{
    cake.SetActive(true);
    cakeCanvas.SetActive(false);
}
//hide cake from the table
public void HideCake()
{
    cake.SetActive(false);
    cakeCanvas.SetActive(true);
}

//show cheese on the table
public void Showcheese()

```

```

    {
        cheese.SetActive(true);
        cheeseCanvas.SetActive(false);
    }
//hide cheese from the table
public void HideCheese()
{
    cheese.SetActive(false);
    cheeseCanvas.SetActive(true);
}
//show tomato on the table
public void Showtomato()
{
    tomato.SetActive(true);
    tomatoCanvas.SetActive(false);
}
//hide tomato from the table
public void Hidetomato()
{
    tomato.SetActive(false);
    tomatoCanvas.SetActive(true);
}
//show cucumber on the table
public void Showcucumber()
{
    cucumber.SetActive(true);
    cucumberCanvas.SetActive(false);
}
//hide cucumber from the table
public void Hidecucumber()
{
    cucumber.SetActive(false);
    cucumberCanvas.SetActive(true);
}
//show milk on the table
public void Showmilk()
{
    milk.SetActive(true);
    milkCanvas.SetActive(false);
}
//hide milk from the table
public void Hidemilk()
{
    milk.SetActive(false);
    milkCanvas.SetActive(true);
}
//show chocolates on the table
public void Showchocolates()
{
    chocolates.SetActive(true);
    chocolatesCanvas.SetActive(false);
}
//hide chocolates from the table
public void HideChocolates()
{
    chocolates.SetActive(false);
    chocolatesCanvas.SetActive(true);
}
//show avocado toast on the table

```

```

public void Showcavocadotoast()
{
    avocadotoast.SetActive(true);
    avocadotoastCanvas.SetActive(false);
}
// hide avocado toast from the table
public void Hideavocadotoast()
{
    avocadotoast.SetActive(false);
    avocadotoastCanvas.SetActive(true);
}
//show yogurt on the table
public void Showyogurt()
{
    yogurt.SetActive(true);
    yogurtCanvas.SetActive(false);
}
//hide yogurt from the table
public void Hideyogurt()
{
    yogurt.SetActive(false);
    yogurtCanvas.SetActive(true);
}
//show toast on the table
public void Showcatoast()
{
    toast.SetActive(true);
    toastCanvas.SetActive(false);
}
//hide toast from the table
public void Hidetoast()
{
    toast.SetActive(false);
    toastCanvas.SetActive(true);
}
//show octopus on the table
public void ShowLunchoctopus()
{
    octopusLunch.SetActive(true);
    octopusCanvasLunch.SetActive(false);
}
//hide octopus from the table
public void HideLunchoctopus()
{
    octopusLunch.SetActive(false);
    octopusCanvasLunch.SetActive(true);
}
//show burger on the table
public void ShowLunchburger()
{
    burgerLunch.SetActive(true);
    burgerCanvasLunch.SetActive(false);
}
//hide burger from the table
public void HideLunchburger()
{
    burgerLunch.SetActive(false);
    burgerCanvasLunch.SetActive(true);
}

```

```

// show pepper on the table
public void ShowLunchpepper()
{
    pepperLunch.SetActive(true);
    pepperCanvasLunch.SetActive(false);
}
//hide pepper from the table
public void HideLunchpepper()
{
    pepperLunch.SetActive(false);
    pepperCanvasLunch.SetActive(true);
}
//show olive olive on the table
public void ShowLuncholiveoil()
{
    oliveoilLunch.SetActive(true);
    oliveoilCanvasLunch.SetActive(false);
}
//hide olive oil from the table
public void HideLuncholiveoil()
{
    oliveoilLunch.SetActive(false);
    oliveoilCanvasLunch.SetActive(true);
}
//show corn on the table
public void ShowLunchcorn()
{
    cornLunch.SetActive(true);
    cornCanvasLunch.SetActive(false);
}
//hide corn from the table
public void HideLunchcorn()
{
    cornLunch.SetActive(false);
    cornCanvasLunch.SetActive(true);
}
//show salad on the table
public void ShowLunchsalad()
{
    saladLunch.SetActive(true);
    saladCanvasLunch.SetActive(false);
}
//hide salad from the table
public void HideLunchsalad()
{
    saladLunch.SetActive(false);
    saladCanvasLunch.SetActive(true);
}
//show orange on the table
public void ShowLunchorange()
{
    orangeLunch.SetActive(true);
    orangeCanvasLunch.SetActive(false);
}
//hide orange from the table
public void HideLunchorange()
{
    orangeLunch.SetActive(false);
    orangeCanvasLunch.SetActive(true);
}

```

```

    }
    //show chicken on the table
    public void ShowLunchchicken()
    {
        chickenLunch.SetActive(true);
        chickenCanvasLunch.SetActive(false);
    }
    //hide chicken from the table
    public void HideLunchchicken()
    {
        chickenLunch.SetActive(false);
        chickenCanvasLunch.SetActive(true);
    }
    //show ice cream on the table
    public void ShowLunchicecream()
    {
        icecreamLunch.SetActive(true);
        icecreamCanvasLunch.SetActive(false);
    }
    //hide icecream from the table
    public void HideLunchicecream()
    {
        icecreamLunch.SetActive(false);
        icecreamCanvasLunch.SetActive(true);
    }
    //show potato on the table
    public void ShowLunchpotato()
    {
        potatoLunch.SetActive(true);
        potatoCanvasLunch.SetActive(false);
    }
    //hide potato from the table
    public void HideLunchpotato()
    {
        potatoLunch.SetActive(false);
        potatoCanvasLunch.SetActive(true);
    }
    //show dark chocolate on the table
    public void ShowDinnerdarkchocolate()
    {
        darkchocolateDinner.SetActive(true);
        darkchocolateCanvasDinner.SetActive(false);
    }
    //hide darkchocolate from the table
    public void HideDinnerdarkchocolate()
    {
        darkchocolateDinner.SetActive(false);
        darkchocolateCanvasDinner.SetActive(true);
    }
    //show pasta on the table
    public void ShowDinnerPasta()
    {
        pastaDinner.SetActive(true);
        pastaCanvasDinner.SetActive(false);
    }
    //hide pasta from the table
    public void HideDinnerpasta()
    {
        pastaDinner.SetActive(false);
    }

```

```

        pastaCanvasDinner.SetActive(true);
    }
//show soda on the table
    public void ShowDinnersprite()
    {
        spriteDinner.SetActive(true);
        spriteCanvasDinner.SetActive(false);
    }
//hide soda from the table
    public void HideDinnersprite()
    {
        spriteDinner.SetActive(false);
        spriteCanvasDinner.SetActive(true);
    }
//show cookies on the table
    public void ShowDinnercookies()
    {
        cookiesDinner.SetActive(true);
        cookiesCanvasDinner.SetActive(false);
    }
//hide cookies from the table
    public void HideDinnercookies()
    {
        cookiesDinner.SetActive(false);
        cookiesCanvasDinner.SetActive(true);
    }
//show cucumber on the table
    public void ShowDinnercucumber()
    {
        cucumberDinner.SetActive(true);
        cucumberCanvasDinner.SetActive(false);
    }
//hide cucumber from the table
    public void HideDinnercucumber()
    {
        cucumberDinner.SetActive(false);
        cucumberCanvasDinner.SetActive(true);
    }
//show marshmallow on the table
    public void ShowDinnermarshmallow()
    {
        marshmallowDinner.SetActive(true);
        marshmallowCanvasDinner.SetActive(false);
    }
//hide marshmallow from the table
    public void HideDinnermarshmallow()
    {
        marshmallowDinner.SetActive(false);
        marshmallowCanvasDinner.SetActive(true);
    }
//show milk on the table
    public void ShowDinnermilk()
    {
        milkDinner.SetActive(true);
        milkCanvasDinner.SetActive(false);
    }
//hide milk from the table
    public void HideDinnermilk()
    {

```

```

        milkDinner.SetActive(false);
        milkCanvasDinner.SetActive(true);
    }
    //show milkshake on the table
    public void ShowDinnermilkshake()
    {
        milkshakeDinner.SetActive(true);
        milkshakeCanvasDinner.SetActive(false);
    }

    //hide milkshake from the table
    public void HideDinnermilkshake()
    {
        milkshakeDinner.SetActive(false);
        milkshakeCanvasDinner.SetActive(true);
    }

    //show cheese on the table
    public void ShowDinnercheese()
    {
        cheeseDinner.SetActive(true);
        cheeseCanvasDinner.SetActive(false);
    }

    //hide dinner from the table
    public void HideDinnercheese()
    {
        cheeseDinner.SetActive(false);
        cheeseCanvasDinner.SetActive(true);
    }

    //show water on the table
    public void ShowDinnerwater()
    {
        waterDinner.SetActive(true);
        waterCanvasDinner.SetActive(false);
    }

    //hide water from the table
    public void HideDinnerwater()
    {
        waterDinner.SetActive(false);
        waterCanvasDinner.SetActive(true);
    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

Πότε καλείται η κάθε function:

Η συνάρτηση Start() καλείτε μόνο μια φορά από μόνη της όταν εισέλθουμε στο έβδομο level. Για το κάθε φαγητό κάθε σταδίου, έχουμε 2 functions την Show και την Hide. Η Show() καλείται όταν πατήσουμε στο φαγητό που βρίσκεται στις επιλογές στα αριστερά ενώ το Hide() καλείται όταν πατήσουμε στο ίδιο φαγητό αλλά όταν βρίσκεται στο τραπέζι.

