



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πτυχιακή Εργασία**

Τίτλος Πτυχιακής Εργασίας	Εκπαιδευτικό Λογισμικό Πρωτοβάθμιας Εκπαίδευσης υπό Μορφή Διαδραστικού Παιχνιδιού  Educational Software for Primary Education in the Form of an Interactive Game
Όνοματεπώνυμο Φοιτητή	Κωνσταντινίδης Παναγιώτης
Πατρώνυμο	Κωνσταντινίδης Δανιήλ
Αριθμός Μητρώου	Π/ 20110
Επιβλέπων	Τσάκωνας Παναγιώτης, Ε.ΔΙ.Π.

Ημερομηνία Παράδοσης Φεβρουάριος 2026



**Copyright**

---

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

## Ευχαριστίες

---

---

Αρχικά θα ήθελα να ευχαριστήσω τους γονείς μου, οι οποίοι στάθηκαν δίπλα μου σε όλη τη διάρκεια των σπουδών μου. Με στήριξαν στις καλές μου στιγμές αλλά και στις πολύ δύσκολες που πίστευα ότι δεν θα τα καταφέρω. Είμαι πραγματικά πολύ τυχερός που τους έχω και τους ευχαριστώ θερμά για όλα.

Επίσης θα ήθελα να ευχαριστήσω αλλά και να αφιερώσω την πτυχιακή στην κοπέλα μου Χριστίνα. Χωρίς την στήριξή της και την βοήθειά της ώστε να με βάλει σε σωστό δρόμο, δεν θα τα είχα καταφέρει ούτε στο πτυχίο μου, αλλά και σε πολλά περισσότερα στην ζωή μου.

---

## Περίληψη

Η παρούσα πτυχιακή εργασία αφορά την ανάπτυξη ενός εκπαιδευτικού λογισμικού για μαθητές πρωτοβάθμιας εκπαίδευσης, με τη μορφή διαδραστικών παιχνιδιών. Στόχος είναι η ενίσχυση της μάθησης μέσα από ευχάριστα και κατανοητά παιχνίδια για παιδιά.

Το λογισμικό περιλαμβάνει εκπαιδευτικά παιχνίδια σε διάφορα μαθήματα πρωτοβάθμιας εκπαίδευσης όπως μαθηματικά, γλώσσα, γεωγραφία, ιστορία, μελέτη περιβάλλοντος αλλά και πληροφορική με ένα παιχνίδι ρομπότ. Κάθε παιχνίδι είναι σχεδιασμένο ώστε να βοηθά τα παιδιά να μαθαίνουν μέσα από απλές δραστηριότητες με εικόνες, ήχους και άμεση αλληλεπίδραση.

Η υλοποίηση πραγματοποιήθηκε με χρήση HTML, CSS και JavaScript επιτρέποντας την εύκολη χρήση του λογισμικού. Τα παιχνίδια στοχεύουν στη βελτίωση της συγκέντρωσης, της μνήμης και της κατανόησης των μαθητών, συμβάλλοντας θετικά στη μαθησιακή διαδικασία.

Λέξεις Κλειδιά: Εκπαιδευτικό Λογισμικό, Πρωτοβάθμια Εκπαίδευση, Διαδραστικά Παιχνίδια, Μάθηση μέσω παιχνιδιού, Εκπαιδευτικά Παιχνίδια, Πληροφορική στην εκπαίδευση

## Abstract

---

This undergraduate thesis focuses on the development of an educational software application for primary school students, designed in the form of interactive games. The aim of this project is to enhance learning through enjoyable and easy-to-understand games suitable for children.

The software includes educational games covering various primary education subjects such as mathematics, language, geography, history, environment studies, as well as informatics through a robotbased game. Each game is designed to help children learn through simple activities using images, sounds, and direct interaction.

The implementation was carried out using HTML, CSS and JavaScript, allowing easy access and use of the software. The games aim to improve students' concentration, memory and comprehension, contributing positively to the learning process.

Key Words: Educational Software, Primary Education, Interactive Games, Game-based Learning, Educational Games, Informatics in Education

## Πίνακας Περιεχομένων

Περίληψη.....	iii
Abstract .....	iii
Εισαγωγή .....	6
<b>1.</b> Εκπαιδευτικό Λογισμικό και Μάθηση μέσω Παιχνιδιού .....	7
<b>1.1</b> Εκπαιδευτικό Λογισμικό .....	7
<b>1.2</b> Μάθηση μέσω Παιχνιδιού .....	7

<b>1.3</b>	<b>Διαδραστικά Παιχνίδια στην Πρωτοβάθμια εκπαίδευση</b>	
	.....	8
<b>2.</b>	<b>Ανάλυση και Στόχοι της εφαρμογής</b>	<b>8</b>
<b>2.1</b>	<b>Σκοπός της Εφαρμογής</b>	<b>8</b>
<b>2.2</b>	<b>Στόχοι της Εφαρμογής</b>	<b>9</b>
<b>2.3</b>	<b>Γνωστικά Αντικείμενα</b>	<b>9</b>
<b>2.4</b>	<b>Ομάδα-Στόχος και Χρήστες της εφαρμογής</b>	<b>10</b>
<b>2.5</b>	<b>Εκπαιδευτική Αξία</b>	<b>10</b>
<b>3.</b>	<b>Ανάλυση και Στόχοι της εφαρμογής</b>	<b>11</b>
<b>3.1</b>	<b>Γενική Δομή και Πλοήγηση</b>	<b>11</b>
<b>3.2</b>	<b>Σχεδιασμός Διεπαφής</b>	<b>11</b>
<b>3.3</b>	<b>Ρόλος Εικόνων και Γραφικών Στοιχείων</b>	<b>12</b>
<b>3.4</b>	<b>Χρήση Ήχων στην εφαρμογή</b>	<b>12</b>
<b>3.5</b>	<b>Ενιαία Εμπειρία Χρήστη σε όλα τα Επίπεδα</b>	<b>12</b>
<b>4.</b>	<b>Περιγραφή Εκπαιδευτικών Παιχνιδιών</b>	<b>13</b>
<b>4.1</b>	<b>Παιχνίδια Μαθηματικών</b>	<b>13</b>
<b>4.2</b>	<b>Παιχνίδια Γλώσσας</b>	<b>13</b>
<b>4.3</b>	<b>Παιχνίδια Γεωγραφίας</b>	<b>13</b>
<b>4.4</b>	<b>Παιχνίδια Ιστορίας</b>	<b>14</b>
<b>4.5</b>	<b>Παιχνίδια Μελέτη Περιβάλλοντος</b>	<b>14</b>
<b>4.6</b>	<b>Παιχνίδι Ρομπότ</b>	<b>14</b>
<b>4.7</b>	<b>Συνολική Εκπαιδευτική Προσέγγιση</b>	<b>15</b>

<b>5.</b>	<b>Περιγραφή Εκπαιδευτικών Παιχνιδιών .....</b>	<b>15</b>
<b>5.1</b>	<b>Τεχνολογίες Ανάπτυξης .....</b>	<b>15</b>
<b>5.2</b>	<b>Δομή των Αρχείων .....</b>	<b>15</b>
<b>5.3</b>	<b>Υλοποίηση της Διεπαφής Χρήστη .....</b>	<b>16</b>
<b>5.4</b>	<b>Λειτουργικότητα και Αλληλεπίδραση .....</b>	<b>16</b>

5.5	Χρήση JavaScript .....	16
<hr/>		
5.6	Εκτέλεση της Εφαρμογής.....	17
6.	Ανάλυση και Επεξήγηση Κώδικα .....	17
6.1	Κεντρικό Μενού της Εφαρμογής (menu.html) .....	17
6.1.1	Συμπέρασμα Μενού .....	19
6.2	Μαθηματικά .....	19
6.2.1	Κατανομή φρούτων (math1.html) .....	19
6.2.2	Πράξεις με Πολλαπλές Επιλογές (math2.html) .....	23
6.2.3	Sudoku (math3.html) .....	25
6.2.4	Συμπέρασμα Μαθηματικών .....	28
6.3	Γλώσσα .....	29
6.3.1	Λέξεις στην σωστή Εικόνα (language1.html) .....	29
6.3.2	Παιχνίδι Ήχου με Εικόνες (language2.html).....	31
6.3.3	Σωστή ορθογραφία (language3.html) .....	33
6.3.4	Συμπέρασμα Γλώσσας .....	34
6.4	Μελέτη Περιβάλλοντος .....	34
6.4.1	Σκουπίδια στην Θάλασσα (environment1.html) .....	34
6.4.2	Σκουπίδια στον Σωστό Κάδο (environment2.html) .....	38
6.4.2	Φαγητά στην Σωστή Κατηγορία (environment3.html) .....	40
6.4.3	Συμπέρασμα Μελέτης Περιβάλλοντος .....	42
6.5	Ιστορία .....	43
6.5.1	Ποιος είναι ο Εικονιζόμενος (history1.html) .....	43
6.5.2	Πότε Έζησαν οι Προσωπικότητες (history2.html) .....	45
6.5.3	Συμπέρασμα Ιστορίας .....	48
6.6	Γεωγραφία .....	49
6.6.1	Βρες τις Χώρες στον Χάρτη (geography1.html) .....	49
6.6.2	Σωστή Χώρα και Πρωτεύουσα (environment2.html) .....	52
6.6.3	Συμπέρασμα Γεωγραφίας .....	54
6.7	Ρομπότ (robot_game.html) .....	54
6.7.1	Συμπέρασμα Ρομπότ .....	63
7.	Συμπεράσματα .....	64
8.	Μελλοντικές Επεκτάσεις .....	65
	Βιβλιογραφία .....	66

## Εισαγωγή

Η εξέλιξη της τεχνολογίας τα τελευταία χρόνια έχει επηρεάσει σημαντικά τον τομέα της εκπαίδευσης με αποτέλεσμα να δημιουργούνται νέες και σύγχρονες μορφές μάθησης. Στη πρωτοβάθμια εκπαίδευση, η αξιοποίηση της τεχνολογίας μπορεί να συμβάλει ουσιαστικά στη βελτίωση της μαθησιακής διαδικασίας, κάνοντας τη μάθηση πιο ελκυστική και αποτελεσματική για τους μαθητές.

Τα εκπαιδευτικά παιχνίδια συνδυάζουν τη μάθηση με το παιχνίδι, με αποτέλεσμα να γίνεται πιο ελκυστική η μάθηση για τα παιδιά. Όταν τα παιδιά μαθαίνουν παίζοντας, συμμετέχουν πιο ενεργά και δείχνουν μεγαλύτερη συγκέντρωση. Η χρήση εικόνων ήχων και απλής αλληλεπίδρασης βοηθά τα παιδιά να κατανοούν καλύτερα το περιεχόμενο και να απολαμβάνουν τη μαθησιακή διαδικασία.

Σκοπός της πτυχιακής, είναι η δημιουργία ενός εκπαιδευτικού λογισμικού για μαθητές δημοτικού με τη μορφή διαδραστικών παιχνιδιών. Το λογισμικό περιλαμβάνει παιχνίδια από βασικά μαθήματα πρωτοβάθμιας εκπαίδευσης καθώς και ένα παιχνίδι με ρομπότ που εισάγει τα παιδιά στη πληροφορική.

Η εφαρμογή έχει σχεδιαστεί ώστε να είναι εύκολη και κατανοητή για χρήση από παιδιά. Δεν απαιτείται εγκατάσταση και μπορεί να χρησιμοποιηθεί κατευθείαν μέσω του υπολογιστή ή από φορητές συσκευές. Στόχος είναι να βοηθήσει τα παιδιά να βελτιώσουν τη συγκέντρωσή τους και την κατανόηση των μαθημάτων με έναν πιο ενδιαφέρον τρόπο.

Η παρούσα πτυχιακή εργασία περιλαμβάνει κεφάλαια που παρουσιάζουν μία γενική ιδέα της εφαρμογής, τον τρόπο σχεδιασμού καθώς και τα συμπεράσματα που προκύπτουν από τη συνολική προσπάθεια.

## 1. Εκπαιδευτικό Λογισμικό και Μάθηση μέσω Παιχνιδιού

### 1.1 Εκπαιδευτικό Λογισμικό

Το εκπαιδευτικό λογισμικό αποτελεί ένα από τα βασικότερα εργαλεία της σύγχρονης εκπαίδευσης. Πρόκειται για εφαρμογές και προγράμματα που έχουν σχεδιαστεί με σκοπό να βοηθήσουν τους/τις μαθητές/-τριες να μάθουν πιο εύκολα, πιο γρήγορα και με μεγαλύτερο ενδιαφέρον. Σε αντίθεση με τις παραδοσιακές μεθόδους διδασκαλίας, το εκπαιδευτικό λογισμικό προσφέρει τη δυνατότητα ενεργού συμμετοχής του μαθητή στη μαθησιακή διαδικασία.

Στην πρωτοβάθμια εκπαίδευση, το εκπαιδευτικό λογισμικό μπορεί να λειτουργήσει υποστηρικτικά προς το μάθημα στην αίθουσα διδασκαλίας. Τα παιδιά έρχονται σε επαφή με τη γνώση μέσα από εικόνες και ήχους με απλές δραστηριότητες, γεγονός που κάνει τη μάθηση πιο κατανοητή και ευχάριστη. Με αυτόν τον τρόπο, αποφεύγεται η μονοτονία και αυξάνεται το ενδιαφέρον.

Ένα σημαντικό πλεονέκτημα του εκπαιδευτικού λογισμικού είναι ότι μπορεί να προσαρμοστεί στις ανάγκες κάθε μαθητή. Κάθε παιδί έχει τον δικό του ρυθμό μάθησης και το λογισμικό επιτρέπει την επανάληψη δραστηριοτήτων χωρίς πίεση ή άγχος. Επιπλέον, τα παιδιά μπορούν να μαθαίνουν κάνοντας λάθη, χωρίς φόβο, καθώς το λάθος αντιμετωπίζεται ως μέρος της μαθησιακής διαδικασίας.

Στη σύγχρονη εποχή, που η τεχνολογία αποτελεί κομμάτι της καθημερινής ζωής των παιδιών, η χρήση εκπαιδευτικού λογισμικού είναι σημαντική. Τα παιδιά είναι ήδη εξοικειωμένα με τις ψηφιακές συσκευές, γεγονός που καθιστά την αξιοποίηση της τεχνολογίας στην εκπαίδευση όχι μόνο χρήσιμη, αλλά απαραίτητη.

## 1.2 Μάθηση μέσω Παιχνιδιού

Η μάθηση μέσω παιχνιδιού αποτελεί μία από τις πιο αποτελεσματικές μορφές εκπαίδευσης. Το παιχνίδι είναι βασικό στοιχείο της παιδικής ζωής και συνδέεται άμεσα με τη χαρά, τη δημιουργικότητα και την εξερεύνηση. Όταν η μάθηση συνδυάζεται με το παιχνίδι, τα παιδιά μαθαίνουν χωρίς να το εκλαμβάνουν ως υποχρέωση.

Μέσα από το παιχνίδι τα παιδιά αναπτύσσουν δεξιότητες όπως η συγκέντρωση, η παρατηρητικότητα και η μνήμη. Τα εκπαιδευτικά παιχνίδια τους δίνουν τη δυνατότητα να συμμετέχουν ενεργά, να δοκιμάζουν, να πειραματίζονται και να μαθαίνουν μέσα από τη δράση. Αυτή η μορφή μάθησης είναι πιο φυσική και πιο κοντά στον τρόπο με τον οποίο τα παιδιά αντιλαμβάνονται τον κόσμο.

Η μάθηση μέσω παιχνιδιού συμβάλλει επίσης στη μείωση άγχους που συχνά συνοδεύει τη σχολική διαδικασία. Τα παιδιά αισθάνονται πιο άνετα, δεν φοβούνται να κάνουν λάθη και αποκτούν μεγαλύτερη αυτοπεποίθηση. Παράλληλα, το παιχνίδι ενισχύει τη θετική στάση απέναντι στη μάθηση και το σχολείο γενικότερα.

## 1.3 Διαδραστικά Παιχνίδια στην Πρωτοβάθμια εκπαίδευση

Τα διαδραστικά παιχνίδια αποτελούν μία σύγχρονη μορφή εκπαιδευτικού λογισμικού που βασίζεται στην ενεργό συμμετοχή του χρήστη. Σε αυτά τα παιχνίδια, τα παιδιά δεν είναι απλοί παρατηρητές, αλλά αλληλεπιδρούν άμεσα με το περιεχόμενο, λαμβάνοντας αποφάσεις και εκτελώντας ενέργειες.

Στην πρωτοβάθμια εκπαίδευση, τα διαδραστικά παιχνίδια μπορούν να καλύψουν πολλά διαφορετικά γνωστικά αντικείμενα, όπως τα μαθηματικά, τη γλώσσα, τη γεωγραφία κ.ά. Μέσα από απλές δραστηριότητες, τα παιδιά εξασκούνται σε βασικές έννοιες και αποκτούν γνώσεις με τρόπο κατανοητό και ευχάριστο.

Η διαδραστικότητα βοηθά τα παιδιά να παραμένουν συγκεντρωμένα και να ενδιαφέρονται περισσότερο για το εκπαιδευτικό περιεχόμενο. Η άμεση ανταπόκριση του παιχνιδιού, όπως τα μηνύματα επιβράβευσης ή διόρθωσης, ενισχύει τη μαθησιακή διαδικασία και βοηθά τα παιδιά να κατανοούν καλύτερα τα λάθη τους.

Επιπλέον, τα διαδραστικά παιχνίδια ενισχύουν την αυτονομία των μαθητών. Τα παιδιά μπορούν να χρησιμοποιηθούν την εφαρμογή μόνα τους, χωρίς καθοδήγηση, γεγονός που τα βοηθά να αναπτύξουν την υπευθυνότητα και την αυτοπεποίθησή τους.

Συνολικά τα διαδραστικά παιχνίδια αποτελούν ένα πολύτιμο εργαλείο για την πρωτοβάθμια εκπαίδευση, καθώς συνδυάζουν τη μάθηση με τη διασκέδαση και προσφέρουν ένα ευχάριστο και αποτελεσματικό μαθησιακό περιβάλλον.

## 2. Ανάλυση και Στόχοι της εφαρμογής

### 2.1 Σκοπός της Εφαρμογής

Ο βασικός σκοπός της εφαρμογής είναι η δημιουργία ενός εκπαιδευτικού εργαλείου που θα βοηθά μαθητές πρωτοβάθμιας εκπαίδευσης να μαθαίνουν μέσα από το παιχνίδι. Η εφαρμογή έχει σχεδιαστεί έτσι ώστε να προσφέρει μία εναλλακτική μορφή μάθησης, διαφορετική από την παραδοσιακή διδασκαλία, η οποία πολλές φορές κουράζει ή δυσκολεύει τα παιδιά.

Η εφαρμογή απευθύνεται σε μαθητές δημοτικού σχολείου και έχει στόχο να ενισχύσει τη μάθηση με απλό και ευχάριστο τρόπο. Μέσα από διαδραστικά παιχνίδια, τα παιδιά έρχονται σε επαφή με βασικές γνώσεις των σχολικών μαθημάτων, χωρίς να αισθάνονται ότι εξετάζονται.

Ένας ακόμη σημαντικός σκοπός της εφαρμογής είναι να βοηθήσει τα παιδιά να αποκτήσουν θετική στάση απέναντι στη μάθηση. Όταν η γνώση παρουσιάζεται με παιγνιώδη τρόπο, τα παιδιά δείχνουν μεγαλύτερο ενδιαφέρον και συμμετοχή. Με αυτόν τον τρόπο, η μάθηση γίνεται μέρος της καθημερινής τους δραστηριότητας και όχι μία υποχρέωση.

Επιπλέον, η εφαρμογή έχει σχεδιαστεί με τρόπο ώστε να προσφέρει άμεση ανατροφοδότηση στον χρήστη μέσω μηνυμάτων επιβράβευσης, διόρθωσης και δυνατοτήτων επανάληψης των δραστηριοτήτων ενισχύοντας έτσι τη μαθησιακή διαδικασία.

Τέλος, η εφαρμογή στοχεύει στο να χρησιμοποιηθεί τόσο στο σχολικό περιβάλλον όσο και στο σπίτι. Έτσι, οι μαθητές έχουν τη δυνατότητα να εξασκούνται και εκτός τάξης, με τη βοήθεια των γονιών τους ή και μόνοι τους.

## 2.2 Στόχοι της Εφαρμογής

Η ανάπτυξη της εφαρμογής βασίστηκε σε συγκεκριμένους εκπαιδευτικούς και παιδαγωγικούς στόχους. Οι στόχοι αυτοί αφορούν τόσο τη γνωστική ανάπτυξη των μαθητών όσο και τη γενικότερη εμπειρία χρήσης της εφαρμογής.

Ένας από τους βασικούς στόχους είναι η βελτίωση της συγκέντρωσης των παιδιών. Μέσα από σύντομα και απλά παιχνίδια, οι μαθητές καλούνται να εστιάσουν τη προσοχή τους σε συγκεκριμένες δραστηριότητες, γεγονός που τους βοηθά να αναπτύξουν καλύτερη συγκέντρωση.

Επιπλέον, η εφαρμογή στοχεύει στην ενίσχυση της μνήμης και της κατανόησης. Τα παιδιά επαναλαμβάνουν έννοιες, εικόνες και ήχους, κάτι που συμβάλλει στην καλύτερη εμπέδωση της γνώσης. Η επανάληψη γίνεται με φυσικό τρόπο, χωρίς πίεση ή άγχος.

Ένας ακόμη σημαντικός στόχος είναι η ανάπτυξη της λογικής σκέψης. Ιδιαίτερα σε παιχνίδια που απαιτούν σωστή σειρά ενεργειών ή επιλογή σωστής απάντησης, τα παιδιά μαθαίνουν να σκέφτονται πριν ενεργήσουν και να αξιολογούν τις επιλογές.

Ακόμη ένας επιπλέον στόχος είναι η ενίσχυση της αυτονομίας των μαθητών, καθώς τα παιδιά μπορούν να πλοηγούνται μόνοι τους στα παιχνίδια, να χρησιμοποιούν βοήθεια όπου χρειάζεται και να επαναλαμβάνουν τις δραστηριότητες όσες φορές επιθυμούν.

Τέλος, η εφαρμογή έχει στόχο να είναι φιλική, απλή και ευχάριστη. Όλα τα παιχνίδια έχουν σχεδιαστεί έτσι ώστε να μπορούν να χρησιμοποιηθούν εύκολα από παιδιά μικρής ηλικίας, χωρίς την ανάγκη πολύπλοκων οδηγιών.

## 2.3 Γνωστικά Αντικείμενα

Η εφαρμογή περιλαμβάνει εκπαιδευτικά παιχνίδια που καλύπτουν διάφορα μαθήματα της πρωτοβάθμιας εκπαίδευσης. Η επιλογή των μαθημάτων έγινε με βάση το σχολικό πρόγραμμα του δημοτικού και τις βασικές γνώσεις που πρέπει να αποκτήσουν οι μαθητές.

Τα μαθήματα που καλύπτονται είναι τα εξής:

- Μαθηματικά : Τα παιδιά εξασκούνται σε βασικές αριθμητικές πράξεις και απλές μαθηματικές έννοιες.
- Γλώσσα : Τα παιδιά βελτιώνονται στην ανάγνωση λέξεων και στην αναγνώριση εικόνων και ήχων.

- Γεωγραφία : Τα παιδιά μαθαίνουν τις χώρες, τους χάρτες και τις πρωτεύουσες.
- Ιστορία : Τα παιδιά μαθαίνουν ιστορικά γεγονότα και γνωστές ιστορικές προσωπικότητες μέσα από εικόνες και απλές ασκήσεις.
- Μελέτη Περιβάλλοντος : Τα παιδιά μαθαίνουν για το περιβάλλον και την διαχείριση απορριμμάτων.
- Πληροφορική : Μέσω ενός παιχνιδιού με ρομπότ, τα παιδιά μαθαίνουν την λογική των εντολών και της ακολουθίας εντολών.
- 

Η ποικιλία των μαθημάτων επιτρέπει στα παιδιά να έρθουν σε επαφή με διαφορετικά αντικείμενα, διατηρώντας το ενδιαφέρον τους αλλά και αποφεύγοντας τη μονοτονία.

Τα παιχνίδια είναι διαχωρισμένα σε επίπεδα ανά τάξη ώστε το περιεχόμενο να είναι κατάλληλο για την ηλικία και το γνωστικό επίπεδο των μαθητών.

## 2.4 Ομάδα-Στόχος και Χρήστες της εφαρμογής

Η εφαρμογή έχει σχεδιαστεί με κύρια ομάδα-στόχο τους μαθητές δημοτικού. Το περιεχόμενο και ο τρόπος παρουσίασης είναι προσαρμοσμένα στην ηλικία και τις ανάγκες των παιδιών του δημοτικού σχολείου.

Παράλληλα, η εφαρμογή μπορεί να χρησιμοποιηθεί και από εκπαιδευτικούς, ως συμπληρωματικό εργαλείο διδασκαλίας μέσα στην τάξη. Οι δάσκαλοι μπορούν να αξιοποιήσουν τα παιχνίδια για επανάληψη ή εμπέδωση της ύλης με πιο ευχάριστο τρόπο.

Επιπλέον, η εφαρμογή απευθύνεται και στους γονείς, οι οποίοι μπορούν να τη χρησιμοποιούν στο σπίτι μαζί με τα παιδιά τους. Με αυτόν τον τρόπο, οι γονείς έχουν τη δυνατότητα να συμμετέχουν ενεργά στη μαθησιακή διαδικασία και να βοηθούν τα παιδιά τους.

## 2.5 Εκπαιδευτική Αξία

Η εφαρμογή βασίζεται στον συνδυασμό της μάθησης και του παιχνιδιού. Τα παιδιά δεν περιορίζονται στην παρακολούθηση, αλλά συμμετέχουν ενεργά σε κάθε δραστηριότητα.

Μέσα από την εφαρμογή, οι μαθητές μαθαίνουν να δοκιμάζουν, να κάνουν λάθη και να προσπαθούν ξανά. Δεν τιμωρούνται για τα λάθη τους με αποτέλεσμα να ενισχύεται η αυτοπεποίθησή τους ώστε να δοκιμάσουν ξανά, μέχρι να καταφέρουν την επίλυση του προβλήματος.

Η ύπαρξη λειτουργιών όπως βοήθεια, λύση και επανάληψη επιτρέπει στα παιδιά να μαθαίνουν με τον δικό τους ρυθμό, ενισχύοντας τη βιωματική μάθηση και τη θετική εμπειρία χρήσης.

Συνολικά, η εφαρμογή συμβάλλει στη δημιουργία ενός ευχάριστου μαθησιακού περιβάλλοντος, το οποίο μπορεί να λειτουργήσει υποστηρικτικά προς τη σχολική εκπαίδευση και να ενισχύσει τη γνώση των μαθητών με σύγχρονο και αποτελεσματικό τρόπο.

## 3. Ανάλυση και Στόχοι της εφαρμογής

### 3.1 Γενική Δομή και Πλοήγηση

Ο σχεδιασμός της εφαρμογής βασίστηκε στη λογική της Απλότητας και της ευκολίας στην χρήση, ώστε να μπορεί να χρησιμοποιηθεί χωρίς δυσκολία από μαθητές πρωτοβάθμιας εκπαίδευσης.

Από την πρώτη στιγμή, στόχος ήταν ο χρήστης να μπορεί να καταλάβει εύκολα πώς λειτουργεί η εφαρμογή και πώς μπορεί να μεταβεί από το ένα παιχνίδι στο άλλο.

Η εφαρμογή ξεκινά με ένα κεντρικό μενού, στο οποίο εμφανίζονται οι βασικές κατηγορίες μαθημάτων. Μέσα από το μενού, ο χρήστης μπορεί να επιλέξει το μάθημα που τον ενδιαφέρει και στη συνέχεια να οδηγηθεί σε αντίστοιχα εκπαιδευτικά παιχνίδια. Μπορεί να επιλέξει ανάμεσα σε 2 ή 3 διαφορετικά επίπεδα πρωτοβάθμιας εκπαίδευσης. Η πλοήγηση είναι ξεκάθαρη και δεν απαιτεί ιδιαίτερες οδηγίες, καθώς όλα τα στοιχεία είναι εμφανή και κατανοητά.

Ιδιαίτερη προσοχή δόθηκε στη χρήση κουμπιών επιστροφής και βοήθειας. Με αυτόν τον τρόπο, ο χρήστης μπορεί ανά πάσα στιγμή να επιστρέψει στο προηγούμενο μενού ή να λάβει οδηγίες για το παιχνίδι που παίζει. Οι οδηγίες είναι γραμμένες με εύκολα και λίγα λόγια ώστε να είναι κατανοητές για παιδιά μικρής ηλικίας.

Η συνολική δομή της εφαρμογής είναι ενιαία σε όλα τα παιχνίδια, γεγονός που βοηθά τα παιδιά να εξοικειωθούν γρήγορα με τον τρόπο λειτουργίας της. Αφού μάθουν πώς λειτουργεί ένα παιχνίδι, μπορούν εύκολα να παίξουν και τα υπόλοιπα.

Σε κάθε παιχνίδι παρέχονται κουμπιά για επιστροφή στο μενού, εμφάνιση βοήθειας και επανάληψη της δραστηριότητάς ώστε ο χρήστης να έχει τον έλεγχο ανά πάσα στιγμή.

### 3.2 Σχεδιασμός Διεπαφής

Η διεπαφή χρήστη της εφαρμογής σχεδιάστηκε με γνώμονα την ηλικία των παιδιών που θα τη χρησιμοποιούν. Τα χρώματα είναι έντονα και ευχάριστα, ενώ τα γραφικά στοιχεία είναι απλά και κατανοητά. Αποφεύχθηκαν πολύπλοκα σχέδια και μικρά γράμματα, ώστε το περιβάλλον να μην κουράζει τα παιδιά.

Κάθε οθόνη περιλαμβάνει μόνο τα απαραίτητα στοιχεία, προκειμένου να αποφεύγεται το μπερδεμα. Τα κουμπιά έχουν ξεκάθαρες ενδείξεις και είναι τοποθετημένα σε σταθερές θέσεις, ώστε τα παιδιά να τα αναγνωρίζουν εύκολα. Αυτή η σταθερότητα στη σχεδίαση βοηθά στη δημιουργία αίσθησης ασφάλειας και οικειότητας.

Επιπλέον, η εφαρμογή έχει σχεδιαστεί ώστε να μπορεί να χρησιμοποιηθεί χωρίς τη συνεχή παρουσία ενήλικα. Τα παιδιά μπορούν να αλληλεπιδρούν μόνα τους με το περιβάλλον, κάτι που ενισχύει την αυτονομία τους. Οι οδηγίες είναι απλές και εμφανίζονται με τρόπο που δεν διακόπτει το παιχνίδι.

Σημαντικό στοιχείο του σχεδιασμού αποτελεί και η αποφυγή περιττών πληροφοριών. Ο στόχος κάθε παιχνιδιού είναι ξεκάθαρος και παρουσιάζεται με απλό τρόπο, ώστε τα παιδιά να γνωρίζουν τι πρέπει να κάνουν χωρίς να μπερδεύονται.

Η διεπαφή παραμένει ομοιόμορφη σε όλα τα παιχνίδια με κοινή διάταξη κουμπιών και παρόμοια αισθητική, διευκολύνοντας της εξοικείωση των χρηστών με το περιβάλλον της εφαρμογής.

### 3.3 Ρόλος Εικόνων και Γραφικών Στοιχείων

Οι εικόνες και τα γραφικά στοιχεία παίζουν καθοριστικό ρόλο στην λειτουργία της εφαρμογής. Στην πρωτοβάθμια εκπαίδευση, η οπτική πληροφορία είναι ιδιαίτερα σημαντική, καθώς βοηθά τα παιδιά να κατανοούν πιο εύκολα το περιεχόμενο.

Στην εφαρμογή, οι εικόνες χρησιμοποιούνται τόσο για διακοσμητικούς όσο και για εκπαιδευτικούς σκοπούς. Για παράδειγμα, σε παιχνίδια γλώσσας και γεωγραφίας, οι εικόνες

βοηθούν τα παιδιά να συνδέουν λέξεις και έννοιες με οπτικά στοιχεία. Αυτό διευκολύνει την απομνημόνευση και την κατανόηση.

Σε αρκετά παιχνίδια, οι εικόνες αποτελούν βασικό στοιχείο αλληλεπίδρασης καθώς ο χρήστης καλείται να σύρει, να επιλέξει ή να αντιστοιχίσει γραφικά αντικείμενα με συγκεκριμένες έννοιες ή κατηγορίες.

### **3.4 Χρήση Ήχων στην εφαρμογή**

Στην εφαρμογή δεν έχει χρησιμοποιηθεί μουσική, ώστε να αποφεύγεται η απόσπαση της προσοχής των μαθητών. Η επιλογή έγινε με στόχο τη δημιουργία ενός ήρεμου και κατάλληλου περιβάλλοντος μάθησης.

Οι ήχοι χρησιμοποιούνται μόνο όπου είναι απαραίτητο, κυρίως στο παιχνίδι της γλώσσας, βοηθώντας τα παιδιά να συνδέουν εικόνες με ακουστικά ερεθίσματα. Οι ήχοι είναι σύντομοι και δεν κουράζουν τον χρήστη.

Η περιορισμένη χρήση ήχων λειτουργεί υποστηρικτικά στη μαθησιακή διαδικασία και ενισχύει τη συγκέντρωση των παιδιών.

### **3.5 Ενιαία Εμπειρία Χρήστη σε όλα τα Επίπεδα**

Ένα βασικό στοιχείο σχεδιασμού της εφαρμογής είναι η ενιαία εμπειρία χρήστη σε όλα τα εκπαιδευτικά παιχνίδια. Παρόλο που τα παιχνίδια αφορούν διαφορετικά μαθήματα, ο τρόπος λειτουργίας τους ακολουθεί κοινή λογική.

Η χρήση παρόμοιων κουμπιών, χρωμάτων και τρόπων αλληλεπίδρασης βοηθά τα παιδιά να προσαρμόζονται γρήγορα. Έτσι, μειώνεται ο χρόνος που χρειάζονται για να κατανοήσουν το κάθε παιχνίδι και μπορούν να επικεντρωθούν στη μάθηση.

Η ενιαία αυτή προσέγγιση συμβάλλει στη συνολική ποιότητα της εφαρμογής και δημιουργεί ένα ολοκληρωμένο και φιλικό περιβάλλον για τους χρήστες. Οι κοινές λειτουργίες όπως βοήθεια, λύση και επανάληψη εμφανίζονται με τον ίδιο τρόπο σε όλα τα επίπεδα, προσφέροντας συνεπή και προβλέψιμη εμπειρία χρήσης.

## **4. Περιγραφή Εκπαιδευτικών Παιχνιδιών**

### **4.1 Παιχνίδια Μαθηματικών**

Τα παιχνίδια μαθηματικών έχουν σχεδιαστεί με στόχο να βοηθήσουν τους μαθητές να εξασκηθούν σε βασικές αριθμητικές έννοιες με απλό και ευχάριστο τρόπο. Μέσα από τις δραστηριότητες αυτές, τα παιδιά καλούνται να λύσουν απλές πράξεις και να επιλέξουν τη σωστή απάντηση.

Η μορφή των παιχνιδιών είναι φιλική προς τα παιδιά και βασίζεται στη λογική της άμεσης αλληλεπίδρασης. Οι μαθητές βλέπουν την ερώτηση στην οθόνη και καλούνται να επιλέξουν την απάντηση μέσα από διαθέσιμες επιλογές. Στην περίπτωση σωστής απάντησης, το παιχνίδι συνεχίζεται, ενώ σε περίπτωση λάθους εμφανίζεται σχετικό μήνυμα.

Σε ορισμένα παιχνίδια μαθηματικών, οι δραστηριότητες περιλαμβάνουν και αλληλεπίδραση με αντικείμενα στην οθόνη, όπως η μετακίνηση ή αντιστοίχιση στοιχείων, ενισχύοντας τη βιωματική μάθηση. Τα παιχνίδια είναι οργανωμένα σε επίπεδα ανάλογα με την

τάξη, ώστε το περιεχόμενο να προσαρμόζεται στο γνωστικό επίπεδο των μαθητών. Ο στόχος των παιχνιδιών και η ενίσχυση της αυτοπεποίθησης των μαθητών χωρίς πίεση.

## 4.2 Παιχνίδια Γλώσσας

Τα παιχνίδια γλώσσας βοηθούν τα παιδιά να έρθουν σε επαφή με λέξεις, εικόνες και ήχους, ενισχύοντας τη γλωσσική τους ανάπτυξη. Οι δραστηριότητες είναι σχεδιασμένες ώστε να είναι απλές και κατανοητές, κατάλληλες για μαθητές δημοτικού.

Μέσα από τα παιχνίδια αυτά, τα παιδιά καλούνται να αναγνωρίσουν λέξεις, να τις αντιστοιχίσουν με εικόνες ή να επιλέξουν τη σωστή απάντηση με βάση ακουστικά ερεθίσματα. Η χρήση εικόνων και ήχων βοηθά τα παιδιά να συνδέουν τη γραπτή και την προφορική μορφή της γλώσσας.

Σε ορισμένες δραστηριότητες, τα παιδιά λαμβάνουν άμεση ακουστική καθοδήγηση, γεγονός που ενισχύει την κατανόηση και τη σωστή προφορά των λέξεων.

Τα παιχνίδια γλώσσας συμβάλλουν στη βελτίωση του λεξιλογίου, της μνήμης και της κατανόησης, ενώ παράλληλα κάνουν τη μάθηση πιο ευχάριστη.

## 4.3 Παιχνίδια Γεωγραφίας

Τα παιχνίδια γεωγραφίας έχουν ως στόχο να βοηθήσουν τα παιδιά να γνωρίσουν χώρες, χάρτες και βασικά γεωγραφικά στοιχεία. Οι δραστηριότητες βασίζονται στην οπτική ανάγνωση και στην απλή αλληλεπίδραση.

Οι μαθητές καλούνται να αντιστοιχίσουν χώρες με τη σωστή θέση στον χάρτη ή να αναγνωρίσουν γεωγραφικά στοιχεία μέσα από εικόνες. Η διαδικασία αυτή βοηθά τα παιδιά να αποκτήσουν βασικές γνώσεις γεωγραφίας με τρόπο κατανοητό και παιγνιώδη.

Η χρήση εικόνων και χαρτών κάνει το μάθημα πιο ενδιαφέρον και βοηθά στην καλύτερη απομνημόνευση των πληροφοριών.

## 4.4 Παιχνίδια Ιστορίας

Τα παιχνίδια ιστορίας στοχεύουν στη γνωριμία των μαθητών με ιστορικά στοιχεία και γεγονότα, μέσα από απλές και κατανοητές δραστηριότητες. Το περιεχόμενο παρουσιάζεται με τρόπο προσαρμοσμένο στην ηλικία των παιδιών.

Μέσα από εικόνες και βασικές πληροφορίες, τα παιδιά μαθαίνουν να αναγνωρίζουν ιστορικές περιόδους και πρόσωπα. Οι δραστηριότητες είναι σχεδιασμένες ώστε να μην απαιτούν αποστήθιση, αλλά κατανόηση και παρατήρηση.

Οι μαθητές καλούνται επίσης να αναγνωρίσουν χώρες και πρωτεύουσες μέσα από επιλογές, ενισχύοντας τη σύνδεση εικόνας και γνώσης.

Τα παιχνίδια ιστορίας βοηθούν τα παιδιά να αποκτήσουν μία πρώτη επαφή με το μάθημα της ιστορίας με ευχάριστο τρόπο.

## 4.5 Παιχνίδια Μελέτη Περιβάλλοντος

Τα παιχνίδια μελέτης περιβάλλοντος έχουν ως στόχο να ευαισθητοποιήσουν τα παιδιά σχετικά με το περιβάλλον και τη σωστή συμπεριφορά απέναντι του. Μέσα από απλές δραστηριότητες τα παιδιά υιοθετούν βασικές στάσεις όπως η ανακύκλωση και η προστασία του περιβάλλοντος.

Οι μαθητές καλούνται να επιλέξουν τη σωστή κατηγορία για διάφορα αντικείμενα ή να ολοκληρώσουν δραστηριότητες που σχετίζονται με την καθημερινή ζωή. Με αυτόν τον τρόπο, τα παιδιά κατανοούν τη σημασία της σωστής περιβαλλοντικής συμπεριφοράς.

Σε ορισμένα παιχνίδια η αλληλεπίδραση γίνεται μέσω μετακίνησης αντικειμένων στην οθόνη, ενισχύοντας τη συμμετοχή και τη βιωματική κατανόηση των εννοιών.

Τα παιχνίδια λειτουργούν ως μέσο εκπαίδευσης και παράλληλα δημιουργούν οικολογική συνείδηση από μικρή ηλικία.

#### **4.6 Παιχνίδι Ρομπότ**

Το παιχνίδι με το ρομπότ αποτελεί μία εισαγωγή των μαθητών στις βασικές έννοιες της πληροφορικής και της αλγοριθμικής σκέψης. Μέσα από το παιχνίδι αυτό, τα παιδιά μαθαίνουν να χρησιμοποιούν απλές εντολές και να τις δομούν με σωστή σειρά.

Οι μαθητές καλούνται να καθοδηγήσουν το ρομπότ σε συγκεκριμένη διαδρομή επιλέγοντας τις κατάλληλες κινήσεις. Η δραστηριότητα αυτή βοηθά τα παιδιά να κατανοήσουν την έννοια της ακολουθίας εντολών και της αιτιότητας.

Οι εντολές εκτελούνται επιτρέποντας στα παιδιά να παρατηρούν άμεσα το αποτέλεσμα των επιλογών τους και να διορθώνουν τυχόν λάθη μέσω επανάληψης της δραστηριότητας.

Το παιχνίδι ρομπότ ενισχύει τη λογική σκέψη, τη συγκέντρωση και την ικανότητα επίλυσης προβλημάτων, επιτελώντας ένα σημαντικό εργαλείο εισαγωγής στην πληροφορική.

#### **4.7 Συνολική Εκπαιδευτική Προσέγγιση**

Όλα τα εκπαιδευτικά παιχνίδια της εφαρμογής έχουν σχεδιαστεί με κοινή φιλοσοφία και στόχο τη μάθηση μέσα από το παιχνίδι. Παρότι αφορούν διαφορετικά μαθήματα, ακολουθούν παρόμοιο τρόπο λειτουργίας, ώστε τα παιδιά να εξοικειώνονται εύκολα.

Η ποικιλία των παιχνιδιών βοηθά στη διατήρηση του ενδιαφέροντος των μαθητών και προσφέρει μία ολοκληρωμένη εμπειρία. Μέσα από την εφαρμογή, τα παιδιά μαθαίνουν παίζοντας και αναπτύσσουν βασικές δεξιότητες με ευχάριστο και κατανοητό τρόπο.

Η ύπαρξη κοινών λειτουργιών, όπως βοήθεια, λύση και επανάληψη σε όλα τα παιχνίδια, ενισχύει τη συνοχή της εφαρμογής και τη θετική εμπειρία χρήσης.

### **5. Περιγραφή Εκπαιδευτικών Παιχνιδιών**

#### **5.1 Τεχνολογίες Ανάπτυξης**

Η υλοποίηση της εκπαιδευτικής εφαρμογής πραγματοποιήθηκε με τη χρήση τεχνολογιών διαδικτύου, οι οποίες επιτρέπουν την εύκολη πρόσβαση και χρήση της εφαρμογής χωρίς την ανάγκη εγκατάστασης. Συγκεκριμένα, χρησιμοποιήθηκαν οι τεχνολογίες HTML, CSS, και JavaScript.

Η HTML χρησιμοποιήθηκε για τη δομή των σελίδων και τη δημιουργία των βασικών στοιχείων της εφαρμογής, όπως κουμπιά, εικόνες, τίτλοι και περιοχές παιχνιδιού. Κάθε παιχνίδι υλοποιήθηκε σε ξεχωριστό αρχείο HTML, γεγονός που βοηθά στην καλύτερη οργάνωση της εφαρμογής.

Η CSS χρησιμοποιήθηκε για τον σχεδιασμό και τη μορφοποίηση της εφαρμογής. Μέσω της CSS ορίστηκαν τα χρώματα, οι θέσεις των στοιχείων, τα μεγέθη γραμματοσειρών και η γενική εμφάνιση κάθε παιχνιδιού. Ο στόχος ήταν η δημιουργία ενός φιλικού και ευχάριστου περιβάλλοντος για παιδιά.

Η JavaScript χρησιμοποιήθηκε για τη λειτουργικότητα των παιχνιδιών. Μέσω της JavaScript υλοποιήθηκε η αλληλεπίδραση του χρήστη με την εφαρμογή, όπως η επιλογή απαντήσεων, η εμφάνιση μηνυμάτων σωστού ή λάθους και η κίνηση αντικειμένων.

Ένας ακόμη λόγος που χρησιμοποιήθηκε η JavaScript ήταν για την υλοποίηση πιο σύνθετων λειτουργιών, όπως η επανάληψη παιχνιδιών, η εμφάνιση βοήθειας και λύσεων, καθώς και ο έλεγχος της ροής των δραστηριοτήτων.

## 5.2 Δομή των Αρχείων

Η εφαρμογή είναι οργανωμένη σε φακέλους, ώστε να υπάρχει τάξη στη δομή της. Υπάρχει φάκελος που περιέχει όλα τα αρχεία HTML, φάκελος με τις εικόνες και φάκελος με τα ηχητικά αποσπάσματα.

Κάθε αρχείο περιλαμβάνει τον κώδικα της δραστηριότητας και λειτουργεί αυτόνομα. Αυτό σημαίνει ότι κάθε παιχνίδι μπορεί να εκτελεστεί ανεξάρτητα, ενώ ταυτόχρονα συνδέεται με τα υπόλοιπα μέσω του κεντρικού μενού.

Η επιλογή να μην χρησιμοποιηθούν ξεχωριστά αρχεία CSS έγινε για λόγους απλότητας. Έτσι όλος ο κώδικας που αφορά ένα παιχνίδι βρίσκεται συγκεντρωμένος σε ένα αρχείο, διευκολύνοντας τη συντήρηση και τη κατανόησή του. Η δομή αυτή επιτρέπει εύκολη προσθήκη νέων παιχνιδιών ή επιπέδων, χωρίς να απαιτούνται αλλαγές στον υπάρχοντα κώδικα των υπολοίπων δραστηριοτήτων.

## 5.3 Υλοποίηση της Διεπαφής Χρήστη

Η διεπαφή χρήστη υλοποιήθηκε με στόχο να είναι απλή και κατανοητή για παιδιά δημοτικού. Τα στοιχεία της οθόνης τοποθετήθηκαν σε σταθερές θέσεις, ώστε οι μαθητές να γνωρίζουν πάντα πού βρίσκονται τα βασικά κουμπιά.

Χρησιμοποιήθηκαν μεγάλα και ευανάγνωστα κουμπιά, με έντονα χρώματα, ώστε να ξεχωρίζουν εύκολα. Επίσης, δόθηκε έμφαση στην καθαρή παρουσίαση του παιχνιδιού, χωρίς περιττά στοιχεία που θα μπορούσαν να αποσπάσουν την προσοχή.

Σε όλα τα παιχνίδια υπάρχει παρόμοια λογική σχεδίασης γεγονός που βοηθά τα παιδιά να εξοικειώνονται γρήγορα με το περιβάλλον της εφαρμογής και να τη χρησιμοποιούν χωρίς δυσκολία.

Σε όλα τα παιχνίδια χρησιμοποιούνται κοινά κουμπιά πλοήγησης, όπως επιστροφή, βοήθεια, λύση και επανάληψη, τα οποία βρίσκονται σε σταθερές θέσεις στην οθόνη.

## 5.4 Λειτουργικότητα και Αλληλεπίδραση

Η λειτουργικότητα της εφαρμογής βασίζεται στην άμεση αλληλεπίδραση του χρήστη με τα παιχνίδια. Οι μαθητές χρησιμοποιούν το ποντίκι για να επιλέγουν απαντήσεις, να μετακινούν αντικείμενα ή να εκτελούν εντολές, ανάλογα με το είδος του παιχνιδιού.

Σε πολλές δραστηριότητες χρησιμοποιείται η τεχνική του drag and drop, η οποία επιτρέπει στα παιδιά να μεταφέρουν αντικείμενα στη σωστή θέση. Αυτή η μορφή αλληλεπίδρασης είναι ιδιαίτερα κατανοητή και ελκυστική για μαθητές μικρής ηλικίας.

Στα παιχνίδια με μετακίνηση αντικειμένων, ο μηχανισμός γίνεται με δύο διαδοχικά (όχι διπλό) κλικ ώστε να είναι πιο εύκολη για τα παιδιά μικρής ηλικίας.

Η εφαρμογή παρέχει άμεση ανατροφοδότηση στον χρήστη. Όταν μία ενέργεια είναι σωστή, εμφανίζεται κατάλληλο μήνυμα επιβράβευσης, ενώ σε περίπτωση λάθους ενημερώνεται ο μαθητής ώστε να προσπαθήσει ξανά. Με αυτόν τον τρόπο, η μάθηση γίνεται μέσα από δοκιμή και επανάληψη.

## 5.5 Χρήση JavaScript

Η JavaScript αποτελεί βασικό εργαλείο για τη λειτουργία των παιχνιδιών. Μέσω αυτής ελέγχεται η ροή του παιχνιδιού, η εμφάνιση και απόκρυψη στοιχείων, καθώς και η σωστή ή λανθασμένη επιλογή του χρήστη.

Ο κώδικας είναι γραμμένος με απλή λογική και κατανοητή δομή, ώστε να μπορεί εύκολα να επεκταθεί ή να τροποποιηθεί. Χρησιμοποιούνται μεταβλητές για την αποθήκευση τιμών και συναρτήσεις για την εκτέλεση συγκεκριμένων ενεργειών, όπως ο έλεγχος της απάντησης ή η επανεκκίνηση του παιχνιδιού.

Η επιλογή απλής υλοποίησης επιτρέπει στην εφαρμογή να λειτουργεί ομαλά σε όλους τους σύγχρονους φυλλομετρητές χωρίς ιδιαίτερες απαιτήσεις σε υπολογιστική ισχύ.

Σε ειδικές περιπτώσεις, όπως το παιχνίδι με το ρομπότ, η JavaScript χρησιμοποιείται για τη διαχείριση ακολουθίας εντολών και την εκτέλεση τους με συγκεκριμένη σειρά.

## 5.6 Εκτέλεση της Εφαρμογής

Η εφαρμογή μπορεί να εκτελεστεί εύκολα χωρίς εγκατάσταση. Ο χρήστης χρειάζεται μόνο έναν σύγχρονο φυλλομετρητή και πρόσβαση στα αρχεία της εφαρμογής. Ανοίγοντας το κεντρικό αρχείο μενού (menu.html), μπορεί να μεταφερθεί σε όλα τα παιχνίδια.

Η απλότητα στην εκτέλεση καθιστά την εφαρμογή κατάλληλη τόσο για χρήση στο σχολικό περιβάλλον όσο και στο σπίτι. Δεν απαιτούνται ειδικές ρυθμίσεις ή τεχνικές γνώσεις για τη χρήση της.

## 6. Ανάλυση και Επεξήγηση Κώδικα

### 6.1 Κεντρικό Μενού της Εφαρμογής (menu.html)

Το κεντρικό μενού είναι η πρώτη οθόνη που βλέπει ο χρήστης και λειτουργεί ως πύλη προς τα μαθήματα της εφαρμογής. Από εδώ ο μαθητής επιλέγει το μάθημα που θέλει και μεταφέρεται στην σωστή σελίδα. Η σελίδα ξεκινά με τα βασικά στοιχεία HTML και ορίζει ότι η γλώσσα είναι Ελληνικά.

Στο <head> δηλώνονται η κωδικοποίηση χαρακτήρων, η προσαρμογή σε διαφορετικές οθόνες και ο τίτλος της σελίδας.

```
<!DOCTYPE html>
<html lang="el">
<head>
  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Διαδραστικά Παιχνίδια Πρωτοβάθμιας Εκπαίδευσης</title>
```

#### 1 Κεντρικό μενού: βασική δομή HTML και ρυθμίσεις head

Ο τίτλος της εφαρμογής εμφανίζεται στην οθόνη με τη χρήση του tag <h1>. Χρησιμοποιείται αλλαγή γραμμής <br> ώστε ο τίτλος να εμφανίζεται σε περισσότερες γραμμές και να είναι πιο ευανάγνωστος. Τα κουμπιά του μενού τοποθετούνται μέσα σε ένα container, ώστε να ελέγχεται εύκολα η διάταξή τους. Κάθε κουμπί του μενού είναι ένα στοιχείο <button> το οποίο περιέχει εικόνα αντί για απλό κείμενο. Η επιλογή αυτή κάνει το μενού πιο ελκυστικό και κατανοητό για παιδιά.

```
<body>
  <h1>Διαδραστικά Παιχνίδια <br> Πρωτοβάθμιας <br> Εκπαίδευσης</h1>

  <div class="menu-container">
    <button class="menu-btn" onclick="window.location.href='math.html'">
      
    </button>

    <button class="menu-btn" onclick="window.location.href='language.html'">
      
    </button>

    <button class="menu-btn" onclick="window.location.href='environment.html'">
      
    </button>

    <button class="menu-btn" onclick="window.location.href='history.html'">
      
    </button>

    <button class="menu-btn" onclick="window.location.href='geography.html'">
      
    </button>

    <button class="menu-btn" onclick="window.location.href='robot.html'">
      
    </button>
  </div>
</body>
```

## 2 Τίτλος εφαρμογής και διάταξη κουμπιών με εικόνες στο κεντρικό μενού

Η πλοήγηση γίνεται με χρήση της window.location.href η οποία μεταφέρει τον χρήστη στη σελίδα του αντίστοιχου μαθήματος. Στο CSS ορίζεται ότι η σελίδα καταλαμβάνει όλη την οθόνη, έχει φόντο εικόνα και όλα τα στοιχεία είναι κεντραρισμένα. Το container των κουμπιών εμφανίζει τα κουμπιά σε κατακόρυφη διάταξη και ορίζει κενό μεταξύ τους. Το κάθε κουμπί έχει συγκεκριμένες διαστάσεις και αφαιρείται το προεπιλεγμένο στυλ του browser, καθώς το οπτικό μέρος του κουμπιού είναι η εικόνα. Η εικόνα μέσα στο κουμπί προσαρμόζεται σωστά στο μέγεθος του κουμπιού και έχει στρογγυλεμένες γωνίες. Τέλος, υπάρχει ένα απλό εφέ κατά το πέρασμα του ποντικιού, ώστε ο χρήστης να καταλαβαίνει ότι πρόκειται για διαδραστικό κουμπί.

```

<style>
  body {
    margin: 0;
    height: 100vh;
    background: url("../photos/classroom.jpg") no-repeat center center fixed;
    background-size: cover;
    font-family: "Segoe UI", Roboto, Arial, sans-serif;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
  }

  h1 {
    position: absolute;
    top: 13%;
    left: 50%;
    transform: translateX(-50%);
    color: black;
    font-size: 1.2em;
    text-align: center;
    margin: 0;
  }

  .menu-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    gap: 8%;
    margin-top: 100px;
  }

  .menu-btn {
    width: 350px;
    height: 50px;
    padding: 0;
    background: none;
    border: none;
    cursor: pointer;
    transition: transform 0.25s;
  }

  .menu-btn img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    border-radius: 12px;
  }

  .menu-btn:hover {
    transform: scale(1.05);
  }

```

### 3 Πλοήγηση με JavaScript και μορφοποίηση κουμπιών μέσω CSS

#### 6.1.1 Συμπέρασμα Μενού

Το menu.html αποτελεί μία απλή και οργανωμένη υλοποίηση κεντρικού μενού.

- Χρησιμοποιεί εικόνες ώστε το περιβάλλον να είναι φιλικό.
- Παρέχει ξεκάθαρη πλοήγηση.
- Διατηρεί ομοιόμορφο στυλ με την χρήση CSS.

#### 6.2 Μαθηματικά

Τα παιχνίδια μαθηματικών καλύπτουν διαφορετικά επίπεδα δυσκολίας. Κοινός στόχος είναι η εξάσκηση των μαθητών μέσα από άμεση αλληλεπίδραση, έλεγχο σωστού/λ'θους και δυνατότητα βοήθειας και επανάληψης.

### 6.2.1 Κατανομή φρούτων (math1.html)

Στο παιχνίδι αυτό ο μαθητής πρέπει να μοιράσει τα φρούτα στα παιδιά σύμφωνα με απαιτήσεις που δημιουργούνται δυναμικά. Στο HTML τα φρούτα δηλώνονται ως εικόνες με data-type και data-slot, ώστε η JavaScript να τα χειρίζεται εύκολα.

Αντίστοιχα, τα παιδιά είναι εικόνες με ώστε να γνωρίζουμε σε ποιον δίνεται ένα φρούτο. Υπάρχουν επίσης κοινά κουμπιά πλοήγησης και λειτουργιών (Πίσω, Βοήθεια, Menu, Λύση, Ξαναπαίξε) με το κουμπί επανάληψης να είναι αρχικά κρυφό.

```
<body>
<button type="button" class="back-btn" onclick="window.location.href='math.html'">Πίσω</button>
<button type="button" class="help-btn" onclick="toggleHelp()">Βοήθεια</button>
<button type="button" class="menu-btn" onclick="window.location.href='menu.html'">Menu</button>
<button type="button" class="hint-btn" id="hintBtn">Λύση</button>
<button type="button" class="replay-btn" id="replayBtn">Ξαναπαίξε</button>

<div class="help-overlay" id="helpOverlay">
  Με το ποντίκι, κάνε κλικ στα φρούτα <br> και σύρε τα στα παιδιά που τα θέλουν.
</div>

<div class="title">Μοίρασε σωστά τα φρούτα </div>

<div class="task-box" id="taskBox"></div>

<div class="fruits" id="fruitsRow">
  <div class="fruit-group-bg bg-left"></div>
  <div class="fruit-group-bg bg-center"></div>
  <div class="fruit-group-bg bg-right"></div>
  
  
  
  
  
  
  
  
  
</div>

<div class="name name-left">Γιώργος</div>
<div class="name name-center">Νίκος</div>
<div class="name name-right">Γιάννης</div>





<div class="basket basket-left" id="basket-george"></div>
<div class="basket basket-center" id="basket-nick"></div>
<div class="basket basket-right" id="basket-john"></div>
<div class="message" id="message"></div>
```

#### 4 Δομή HTML παιχνιδιού κατανομής φρούτων και βασικά στοιχεία διεπαφής

Οι δοκιμασίες δεν είναι στατικές. Δημιουργούνται με τυχαίο τρόπο ώστε σε κάθε επανάληψη εκτέλεσης το παιχνίδι να είναι διαφορετικό. Η συνάρτηση generateRequirements() μοιράζει συνολικά 3 φρούτα ανά είδος σε 3 παιδιά και ελέγχει ότι κάθε παιδί παίρνει τουλάχιστον 1 φρούτο.

```

function randomSplitOfThree() {
  const a = randInt(0, 3);
  const b = randInt(0, 3-a);
  const c = 3 - a - b;
  return [a,b,c];
}

function generateRequirements() {
  for(let tries=0; tries<200; tries++){
    const [ag, an, aj] = randomSplitOfThree();
    const [bg, bn, bj] = randomSplitOfThree();
    const [og, on, oj] = randomSplitOfThree();

    const totals = {
      george: ag + bg + og,
      nick:   an + bn + on,
      john:   aj + bj + oj
    };

    if (totals.george >= 1 && totals.nick >= 1 && totals.john >= 1){
      requirements = {
        george: { apple: ag, banana: bg, orange: og },
        nick:   { apple: an, banana: bn, orange: on },
        john:   { apple: aj, banana: bj, orange: oj }
      };
      return;
    }
  }
}

```

##### 5 Δημιουργία τυχαίων απαιτήσεων με τη generateRequirements()

Η μετακίνηση υλοποιείται με δύο κλικ. Με το πρώτο κλικ ο μαθητής πατάει το φρούτο και με το δεύτερο αφήνει το φρούτο στο παιδί που θέλει να το δώσει.

```

fruit.addEventListener("click", e => {
  if (gameOver) return;

  if (!draggedFruit) {
    pickFruit(fruit, e);
    e.stopPropagation();
    return;
  }

  dropFruit(e);
  e.stopPropagation();
});

```

##### 6 Μετακίνηση φρούτων με σύστημα δύο κλικ (επιλογή–τοποθέτηση)

Στην dropFruit γίνεται έλεγχος αν το φρούτο επιτρέπεται βάσει των requirements και αν δεν έχει ήδη καλυφθεί η ποσότητα. Αν είναι σωστό, αυξάνεται το collected, μπαίνει εικόνα στο καλάθι του παιδιού και αφαιρείται το επιλεγμένο φρούτο από την οθόνη.

```

if (requirements[kidName][fruitType] > collected[kidName][fruitType]) {
  collected[kidName][fruitType]++;

  const basket = document.getElementById(`basket-${kidName}`);
  const newFruit = document.createElement("img");
  newFruit.src = draggedFruit.src;
  basket.appendChild(newFruit);
  draggedFruit.remove();
  showMessage("Σωστό", "limegreen", 2000);
  placed = true;
  checkCompletion();
} else {
  showMessage("Λάθος", "red", 2000);
  resetFruit(draggedFruit);
}
}

```

#### 7 Έλεγχος σωστής τοποθέτησης φρούτου και ενημέρωση προόδου (dropFruit)

Η ολοκλήρωση ελέγχεται με every() πάνω σε όλα τα παιδιά και όλα τα φρούτα. Αν ολοκληρωθεί, το παιχνίδι κλειδώνει, αποκρύπτεται το κουμπί αποκάλυψης της λύσης και εμφανίζεται το κουμπί της επανάληψης του παιχνιδιού.

```

function checkCompletion() {
  const done = Object.keys(requirements).every(kid =>
    Object.keys(requirements[kid]).every(fruit =>
      collected[kid][fruit] >= requirements[kid][fruit]
    )
  );

  if (done) {
    gameOver = true;
    if (msgTimeout) clearTimeout(msgTimeout);
    message.style.opacity = 0;
    replayBtn.style.display = 'inline-block';
    hintBtn.style.display = 'none';
    if (draggedFruit) {
      resetFruit(draggedFruit);
      draggedFruit = null;
    }
  }
}

```

#### 8 Έλεγχος ολοκλήρωσης παιχνιδιού και εμφάνιση επανάληψης

Η restartGame() επαναφέρει μεταβλητές, ξαναδημιουργεί απαιτήσεις, καθαρίζει τα καλάθια, ξαναφτιάχνει τα φρούτα και ξαναδένει τα event listeners.

```

function restartGame() {
  gameOver = false;
  if (msgTimeout) clearTimeout(msgTimeout);
  message.textContent = '';
  message.style.opacity = 0;
  replayBtn.style.display = 'none';
  hintBtn.style.display = 'inline-block';
  generateRequirements();
  renderTaskBox();

  collected = {
    george: { apple: 0, banana: 0, orange: 0 },
    nick:   { apple: 0, banana: 0, orange: 0 },
    john:   { apple: 0, banana: 0, orange: 0 }
  };

  document.getElementById('basket-george').innerHTML = '';
  document.getElementById('basket-nick').innerHTML = '';
  document.getElementById('basket-john').innerHTML = '';

  const fruitsRow = document.getElementById('fruitsRow');
  fruitsRow.innerHTML = `
    
    
    
    
    
    
    
    
    
  `;

  const newFruits = Array.from(document.querySelectorAll('.fruit'));
  fruits.length = 0;
  newFruits.forEach(f => fruits.push(f));
  layoutFruits();
  layoutBaskets();

  newFruits.forEach(fruit => {
    fruit.addEventListener("click", e => {
      if (gameOver) return;

      if (!draggedFruit) {
        pickFruit(fruit, e);
        e.stopPropagation();
        return;
      }

      dropFruit(e);
      e.stopPropagation();
    });
  });
}

```

### 9 Επαναφορά παιχνιδιού και αρχικοποίηση νέου γύρου (restartGame)

#### 6.2.2 Πράξεις με Πολλαπλές Επιλογές (math2.html)

Στο παιχνίδι πράξεων ο μαθητής βλέπει μία πράξη και 4 επιλογές απάντησης. Το UI χτίζεται με answersGrid και οι απαντήσεις παράγονται δυναμικά από τη JavaScript.

Το παιχνίδι δημιουργεί 10 ερωτήσεις με generateProblems10() όπου κάθε ερώτηση παράγεται από generateOneProblem()

```
function generateProblems10() {
  const arr = [];
  for (let i = 0; i < 10; i++) {
    arr.push(generateOneProblem());
  }
  return shuffle(arr);
}
```

#### 10 Δημιουργία ερωτήσεων

Η generateOneProblem() επιλέγει τυχαίο τελεστή (+, -, x, /) και φτιάχνει πράξη με διψήφιους αριθμούς. Στη διαίρεση εξασφαλίζει ακέραιο αποτέλεσμα δημιουργώντας το a ως πολλαπλάσιο του b.

```
else { // "/"
  b = randInt(10, 99);
  const k = randInt(1, 9);
  a = b * k;

  let tries = 0;
  while ((a < 10 || a > 99) && tries < 300) {
    b = randInt(10, 99);
    const kk = randInt(1, 9);
    a = b * kk;
    tries++;
  }

  if (a < 10 || a > 99) {
    b = randInt(10, 33);
    const kk = randInt(2, 9);
    a = b * kk;
    while (a > 99) {
      b = randInt(10, 33);
      a = b * kk;
    }
  }

  correct = a / b;
  q = `${a} / ${b}`;
```

#### 11 Τυχαία παραγωγή πράξεων και έλεγχος ακέραιου αποτελέσματος

Για κάθε ερώτηση δημιουργούνται κουμπιά απάντησης. Με το κλικ γίνεται έλεγχος και αν είναι σωστό προχωράει στην επόμενη ερώτηση, αλλιώς εμφανίζεται μήνυμα λάθους.

```

if (ans === current.correct) {
  showMessage("Σωστό", "limegreen", 2000);

  index++;
  if (index >= problems.length) {
    gameOver = true;
    gameBox.style.display = "none";
    messageBox.style.top = "50%";
    showMessage("Συγχαρητήρια, τα κατάφερες!", "limegreen", null);
    replayBtn.style.display = "inline-block";
    hintBtn.style.display = "none";
    return;
  }

  setTimeout(() => {
    renderProblem();
  }, 450);
} else {
  showMessage("Λάθος", "red", 2000);
}

```

#### 12 Έλεγχος απάντησης και μετάβαση στην επόμενη ερώτηση

Το κουμπί της Λύσης εδώ δεν δίνει την πλήρη απάντηση, αλλά λειτουργεί ως hint για την τρέχουσα ερώτηση.

```

function showHintForThisQuestion(){
  if (gameOver) return;
  clearHintLayer();
  const btns = [...answersGrid.querySelectorAll('.answer-btn')];
  const p = problems[index];
  const correctBtn = btns.find(b => Number(b.dataset.value) === p.correct);
  if (correctBtn) correctBtn.classList.add('hint-correct');
}

```

#### 13 Κουμπί «Λύση» ως hint για την τρέχουσα ερώτηση

### 6.2.3 Sudoku (math3.html)

Το sudoku είναι το πιο σύνθετο παιχνίδι στα μαθηματικά. Υλοποιείται ως πλέγμα 9x9 και δημιουργείται δυναμικά μέσω JavaScript.

```

function makeEmptyGrid(){
  return Array.from({length:9}, () => Array(9).fill(0));
}

function isValid(grid, r, c, val){
  for (let x=0; x<9; x++){
    if (grid[r][x] === val) return false;
    if (grid[x][c] === val) return false;
  }
  const br = Math.floor(r/3)*3;
  const bc = Math.floor(c/3)*3;
  for (let rr=br; rr<br+3; rr++){
    for (let cc=bc; cc<bc+3; cc++){
      if (grid[rr][cc] === val) return false;
    }
  }
  return true;
}

function findEmpty(grid){
  for (let r=0; r<9; r++){
    for (let c=0; c<9; c++){
      if (grid[r][c] === 0) return [r,c];
    }
  }
  return null;
}

function solveGrid(grid){
  const spot = findEmpty(grid);
  if (!spot) return true;

  const [r,c] = spot;
  const nums = shuffle([1,2,3,4,5,6,7,8,9].slice());

  for (const v of nums){
    if (isValid(grid, r, c, v)){
      grid[r][c] = v;
      if (solveGrid(grid)) return true;
      grid[r][c] = 0;
    }
  }
  return false;
}

```

#### 14 Sudoku: δυναμική δημιουργία πλέγματος 9×9 με JavaScript

Για να φτιαχτεί το puzzle αφαιρούνται τιμές από την λύση, βάσει GIVENS\_COUNT

```
function buildPuzzleFromSolution(sol) {
  const p = cloneGrid(sol);

  const coords = [];
  for (let r=0; r<9; r++){
    for (let c=0; c<9; c++){
      coords.push([r,c]);
    }
  }
  shuffle(coords);

  const toRemove = 81 - GIVENS_COUNT;
  for (let k=0; k<toRemove; k++){
    const [r,c] = coords[k];
    p[r][c] = 0;
  }
  return p;
}
```

#### 15 Δημιουργία puzzle με αφαίρεση τιμών βάσει GIVENS\_COUNT

Δεν επιτρέπεται απλή πληκτρολόγηση. Δημιουργούνται tokens αριθμών μόνο για τα κενά και ο μαθητής τα τοποθετεί με μηχανισμό δύο κλικ όπως και στα άλλα παιχνίδια.

```
function buildMissingTokens() {
  numbersPanel.innerHTML = "";

  const missing = [];
  for (let r=0; r<9; r++){
    for (let c=0; c<9; c++){
      if (puzzle[r][c] === 0){
        missing.push(solution[r][c]);
      }
    }
  }

  missing.sort((a,b) => a - b);
  missing.forEach((n, idx)=>{
    const d = document.createElement('div');
    d.className = 'num';
    d.textContent = String(n);
    d.dataset.num = String(n);
    d.dataset.used = "false";
    d.dataset.tokenId = "t_" + idx;
    numbersPanel.appendChild(d);
  });

  bindTokensTwoClick();
}
```

#### 16 Τοποθέτηση αριθμών με tokens και μηχανισμό δύο κλικ

Στην απόθεση dropPick γίνονται έλεγχοι για να μην είναι given κελί, να μην είναι ήδη συμπληρωμένο και να ταιριάζει με τη λύση.

```

if (cell){
  const r = Number(cell.dataset.r);
  const c = Number(cell.dataset.c);

  if (givenMask[r][c]) {
    showMessage("Λάθος", "red", 1500);
    resetPick();
    return;
  }

  if (puzzle[r][c] !== 0){
    showMessage("Λάθος", "red", 1500);
    resetPick();
    return;
  }

  if (solution[r][c] === num){
    puzzle[r][c] = num;
    cell.textContent = String(num);
    cell.classList.add('user');

    currentPick.dataset.used = "true";
    currentPick.style.visibility = "hidden";
    currentPick.classList.remove('dragging');
    currentPick.classList.remove('picked');
    currentPick.style.left = "";
    currentPick.style.top = "";
    currentPick = null;

    showMessage("Σωστό", "limegreen", 1200);

    if (checkComplete()){
      gameOver = true;
      numbersPanel.style.display = "none";
      hintBtn.style.display = "none";
      replayBtn.style.display = "inline-block";
      showMessage("Συγχαρητήρια!", "limegreen", null);
    }
    return;
  }
}
}

```

#### 17 Έλεγχοι εγκυρότητας κατά την απόθεση αριθμού (dropPick)

Το κουμπί της Λύσης συμπληρώνει όλο το puzzle και τερματίζει το παιχνίδι. Το κουμπί Ξαναπαίξε ξεκινά νέο sudoku από την αρχή δημιουργώντας νέα παραλλαγή.

```

function solveAll() {
  if (gameOver) return;
  puzzle = cloneGrid(solution);
  givenMask = buildGivenMask(puzzle);
  renderBoard();
  gameOver = true;
  numbersPanel.style.display = "none";
  hintBtn.style.display = "none";
  replayBtn.style.display = "inline-block";
  showMessage("Συγχαρητήρια!", "limegreen", null);
}

function newGame() {
  gameOver = false;
  if (msgTimeout) { clearTimeout(msgTimeout); msgTimeout = null; }
  messageBox.textContent = "";
  messageBox.style.opacity = 0;
  replayBtn.style.display = "none";
  hintBtn.style.display = "inline-block";
  numbersPanel.style.display = "";

  if (currentPick) resetPick();

  const g = makeEmptyGrid();
  solveGrid(g);
  solution = cloneGrid(g);
  puzzle = buildPuzzleFromSolution(solution);
  givenMask = buildGivenMask(puzzle);
  renderBoard();
  buildMissingTokens();
}

hintBtn.addEventListener('click', solveAll);
replayBtn.addEventListener('click', newGame);
window.addEventListener('load', newGame);

```

18 Λύση Sudoku και επανεκκίνηση με νέο puzzle

#### 6.2.4 Συμπέρασμα Μαθηματικών

Τα μαθηματικά παιχνίδια έχουν κοινή λογική σχεδίασης. Έχουν σταθερή πλοήγηση, καθαρή ανατροφοδότηση σωστού και λάθους και δυνατότητα βοήθειας και επανάληψης. Ο κώδικας χρησιμοποιεί data-\* γνωρίσματα για εύκολο χειρισμό στοιχείων, δομές δεδομένων για κανόνες και ελέγχους και μηχανισμό δύο κλικ για απλούστερη αλληλεπίδραση από μικρά παιδιά.

#### 6.3 Γλώσσα

Τα παιχνίδια γλώσσας έχουν σχεδιαστεί με στόχο την ενίσχυση της γλωσσικής κατανόησης των μαθητών μέσα από εικόνες, ήχους και διαδραστικές δραστηριότητες. Κάθε παιχνίδι υλοποιείται σε ξεχωριστό αρχείο HTML και ακολουθεί κοινή λογική λειτουργίας, ώστε οι μαθητές να εξοικειώνονται εύκολα με το περιβάλλον.

### 6.3.1 Λέξεις στην σωστή Εικόνα (language1.html)

Στο παιχνίδι αυτό εμφανίζονται τέσσερις εικόνες επάνω και τέσσερις λέξεις κάτω. Ο μαθητής επιλέγει μία λέξη και την τοποθετεί στη σωστή εικόνα με μηχανισμό δύο κλικ. Οι εικόνες και οι λέξεις προέρχονται από έναν pool αντικειμένων με src, key και label.

```
const pool = [
  {src: '../photos/shoe.jpg', key: 'shoe', label: 'Παπούτσι'},
  {src: '../photos/ball.jpg', key: 'ball', label: 'Μπάλα'},
  {src: '../photos/book.jpg', key: 'book', label: 'Βιβλίο'},
  {src: '../photos/tree.jpg', key: 'tree', label: 'Δέντρο'},
  {src: '../photos/flower.jpg', key: 'flower', label: 'Λουλούδι'},
  {src: '../photos/car.jpg', key: 'car', label: 'Αυτοκίνητο'},
  {src: '../photos/hat.jpg', key: 'hat', label: 'Καπέλο'},
  {src: '../photos/shirt.jpg', key: 'shirt', label: 'Μπλουζά'}
];
```

#### 19 Αντιστοίχιση λέξεων με εικόνες μέσω μηχανισμού δύο κλικ

Το παιχνίδι χωρίζεται σε δύο γύρους των τεσσάρων αντικειμένων. Χρησιμοποιείται roundIndex και λαμβάνονται κάθε φορά τέσσερα στοιχεία από το gameSet. Οι εικόνες και οι λέξεις ανακατεύονται ώστε να μην καταλαμβάνουν πάντα την ίδια θέση.

```
const start = roundIndex * 4;
const pick = gameSet.slice(start, start + 4);

currentRoundKeys = pick.map(x => x.key);

const imagesShuffled = shuffle(pick);
const wordsShuffled = shuffle(pick.map(it => ({type: it.key, label: it.label, id: 'w-'+it.key})));
```

#### 20 Δομή δύο γύρων και τυχαία διάταξη αντικειμένων

Όταν ο μαθητής κάνει κλικ σε μία λέξη, αν δεν κρατάει κάτι, η λέξη ενεργοποιείται η λειτουργία dragging ενώ αν ήδη κρατάει τότε ενεργοποιείται η dropWord(e) για απόθεση.

```
words.forEach(w=>{
  w.addEventListener('click', (e)=>{
    if(w.style.pointerEvents==='none') return;

    if(!dragging){
      dragging = w;
      const r = w.getBoundingClientRect();
      offX = e.clientX - r.left;
      offY = e.clientY - r.top;

      w.style.position='fixed';
      w.style.left=r.left+'px';
      w.style.top=r.top+'px';
      w.style.bottom='';
      w.style.zIndex=1000;

      e.stopPropagation();
      return;
    }

    dropWord(e);
    e.stopPropagation();
  });
});
```

#### 21 Λογική επιλογής και απόθεσης λέξεων (drag-drop με δύο κλικ)

Στη `dropWord` ο έλεγχος γίνεται συγκρίνοντας `wordType = dragging.dataset.type` και `imgKey = im.dataset.key`. Αν είναι ίδια τότε θεωρείται σωστό αλλιώς χαρακτηρίζεται ως λάθος.

```

if(wordType===imgKey){
  placeWordUnderImage(dragging, im);

  const completesRound=(placedThisRound+1===4);
  placed=true;
  placedThisRound+=1;

  if(completesRound){
    if(roundIndex === 0){
      roundIndex = 1;
      dragging = null;
      buildRound();
      return;
    }else{
      endGame();
      return;
    }
  }else{
    showMessage('Σωστό!', 'lightgreen', 1200);
  }
}else{
  showMessage('Λάθος!', 'red', 1200);
  resetWord(dragging);
}
break;

```

## 22 Έλεγχος σωστής αντιστοίχισης λέξης–εικόνας στη `dropWord`

Το κουμπί Λύση καλεί την `hintOne()` η οποία τοποθετεί ένα σωστό ζευγάρι κάθε φορά, επιλέγοντας τυχαία από τις λέξεις που απομένουν. Μετά την ολοκλήρωση και των δύο γύρων, καλείται η `endgame()` που αποκρύπτει τα στοιχεία, εμφανίζει μήνυμα επιτυχίας και δείχνει το κουμπί επανάληψης.

```

const roundsBase = [
  { sound: '../music/ball.wav', correct: 'ball', images: ['apple', 'ball', 'banana', 'book'] },
  { sound: '../music/flower.wav', correct: 'flower', images: ['car', 'flower', 'robot', 'tree'] },
  { sound: '../music/banana.wav', correct: 'banana', images: ['flower', 'hat', 'book', 'banana'] },
  { sound: '../music/car.wav', correct: 'car', images: ['car', 'ball', 'banana', 'apple'] },
  { sound: '../music/hat.wav', correct: 'hat', images: ['robot', 'orange', 'shirt', 'hat'] },
  { sound: '../music/shoe.wav', correct: 'shoe', images: ['tree', 'hat', 'car', 'shoe'] }
];

```

```

msg.textContent = 'Συγχαρητήρια, τα κατάφερες!';
msg.style.color = 'lightgreen';
msg.style.opacity = 1;
replayBtn.style.display = 'inline-block';
hintBtn.style.display = 'none';
}

function restartGame(){
  roundIndex = 0;
  placedThisRound = 0;
  dragging = null;
  offX = 0; offY = 0;
  currentImgs = [];
  currentRoundKeys = null;
  gameSet = shuffle(pool);
  topRow.style.display = '';
  wordsRow.style.display = '';
  titleText.style.display = '';
  replayBtn.style.display = 'none';
  hintBtn.style.display = 'inline-block';
  msg.textContent = '';
  msg.style.opacity = 0;

  buildRound();
}

replayBtn.addEventListener('click', restartGame);
hintBtn.addEventListener('click', hintOne);

window.addEventListener('load', async()=>{
  await preloadingAll;
  gameSet = shuffle(pool);
  buildRound();
});

```

### 23 Λειτουργία hint και τερματισμός παιχνιδιού αντιστοίχισης

#### 6.3.2 Παιχνίδι Ήχου με Εικόνες (language2.html)

Σε αυτό το παιχνίδι ο μαθητής πατά το κουμπί «Ήχος» και έπειτα πρέπει να επιλέξει ανάμεσα σε τέσσερις εικόνες εκείνη που αντιστοιχεί στον ήχο που άκουσε. Υπάρχει audio στοιχείο που φορτώνει το σωστό αρχείο ήχου σε κάθε γύρο.

```
<audio id="soundPlayer"></audio>
```

### 24 Φόρτωση και αναπαραγωγή ήχου ανά γύρο με audio στοιχείο

Ο πίνακας roundsBase ορίζει τους γύρους με sound, correct και images.

### 25 Ορισμός γύρων παιχνιδιού μέσω του πίνακα roundsBase

Με το πάτημα του κουμπιού εκτελείται το playSound() και παίζει το soundPlayer.

```

function playSound(){
  if (!soundPlayer.src || gameOver) return;
  soundPlayer.currentTime = 0;
  soundPlayer.play();
}

```

## 26 Αναπαραγωγή ήχου με τη συνάρτηση playSound()

Κατά το click σε εικόνα καλείται η `handleChoice(name)` και συγκρίνεται η επιλογή με το `round.correct`

```
if (name === round.correct) {
  const isLast = currentRoundIndex === rounds.length - 1;

  if (isLast) {
    endGame();
  } else {
    currentRoundIndex++;
    loadRound();
  }
} else {
  showError();
}
```

## 27 Έλεγχος επιλογής εικόνας με τη `handleChoice()`

Το κουμπί «Λύση» δεν απαντά αντί του χρήστη αλλά τονίζει για λίγο τη σωστή εικόνα με class `hint-correct`. Μετά από 2 δευτερόλεπτα αφαιρείται το outline.

```
const correctEl = currentImages.find(im => im.dataset.name === round.correct);
if (correctEl) correctEl.classList.add('hint-correct');

hintTimer = setTimeout(() => {
  clearHintLayer();

  hintRunning = false;
  hintBtn.disabled = false;
  hintBtn.style.pointerEvents = '';
}, 2000);
```

## 28 Οπτική υπόδειξη σωστής εικόνας με προσωρινό highlight (`hint-correct`)

Όταν ολοκληρωθούν όλοι οι γύροι, καλείται η `endGame()` η οποία αποκρύπτει τις εικόνες, τον τίτλο, το κουμπί του ήχου καθώς και τη λύση, εμφανίζει το κουμπί «Ξαναπαίξε» και δείχνει μήνυμα με Συγχαρητήρια.

```
soundBtn.style.display = 'none';
hintBtn.style.display = 'none';
replayBtn.style.display = 'inline-block';
soundPlayer.pause();
soundPlayer.currentTime = 0;
titleText.style.display = 'none';
showWinPermanent();
```

## 29 Προσωρινό highlight σωστής εικόνας με class hint-correct

### 6.3.3 Σωστή ορθογραφία (language3.html)

Στο παιχνίδι της ορθογραφίας, εμφανίζονται λέξεις από τις οποίες λείπουν κάποια γράμματα και ο μαθητής πρέπει να τοποθετήσει το σωστό γράμμα ή τους σωστούς διφθόγγους.

Τα targets, που είναι οι λέξεις, και τα letters, που είναι οι επιλογές, ορίζονται σε πίνακες. Η σωστή αντιστοίχιση γίνεται μέσω του correctMap που καθορίζει ποιο letter ταιριάζει σε ποιο target.

```
const targetsBase = [
  { key: 'frontist', incomplete: 'Φρονι_στήριο', complete: 'Φροντιστήριο' },
  { key: 'diax', incomplete: 'Διαχ_ριση', complete: 'Διαχείριση' },
  { key: 'oiko', incomplete: '_κοσύστημα', complete: 'Οικοςύστημα' },
  { key: 'anem', incomplete: 'Ανεμόμ_λος', complete: 'Ανερόμυλος' },
  { key: 'hfaist', incomplete: 'Ηφ_στειο', complete: 'Ηφαίστειο' },
  { key: 'elefth', incomplete: 'Ελευθ_ρία', complete: 'Ελευθερία' },
  { key: 'autok', incomplete: 'Αυτοκόλλ_το', complete: 'Αυτοκόλλητο' }
];

const lettersBase = [
  { text: 'ε', letter: 'e' },
  { text: 'αι', letter: 'ai' },
  { text: 'ει', letter: 'ei' },
  { text: 'η', letter: 'h' },
  { text: 'ι', letter: 'i' },
  { text: 'υ', letter: 'u' },
  { text: 'οι', letter: 'oi' }
];

const correctMap = {
  i: 'frontist',
  ei: 'diax',
  oi: 'oiko',
  u: 'anem',
  ai: 'hfaist',
  e: 'elefth',
  h: 'autok'
};
```

## 30 Ορισμός στόχων, επιλογών και σωστής αντιστοίχισης με correctMap

Κατά την απόθεση dropLetter συγκρίνονται το letterKey = dragging.dataset.letter to targetKey = target.dataset.key και το correctTarget = correctMap[letterKey]

```
if(correctTarget === targetKey && target.dataset.filled !== 'true'){
  target.textContent = target.dataset.complete;
  target.dataset.filled = 'true';
  dragging.remove();
  placedCount++;

  if (placedCount === totalPairs) {
    hintBtn.style.display = 'none';
    endGame();
    return;
  } else {
    showMessage('Σωστό!', 'lightgreen', 1200);
  }

  placed = true;
}else{
  showMessage('Λάθος!', 'red', 1200);
  resetLetter(dragging);
  placed = true;
}
break;
```

## 31 Έλεγχος σωστής απόθεσης γράμματος στη dropLetter

Το κουμπί «Λύση» καλεί τη solveOne() και συμπληρώνει μία σωστή λέξη κάθε φορά, αφαιρώντας και το αντίστοιχο γράμμα ή διφθόγγο.

```
hintBtn.addEventListener('click', solveOne);
```

**32** Σταδιακή επίλυση λέξεων με τη solveOne()

Όταν συμπληρωθούν όλα τότε εκτελείται η endGame() που εμφανίζει το κουμπί «Ξαναπαίξε».

```
if (placedCount === totalPairs) {
    hintBtn.style.display = 'none';
    endGame();
}
```

**33** Τερματισμός παιχνιδιού και εμφάνιση κουμπιού επανάληψης (endGame)

### 6.3.4 Συμπέρασμα Γλώσσας

Τα παιχνίδια γλώσσας χρησιμοποιούν κοινό μοντέλο λειτουργίας. Έχουν τυχαία σειρά, άμεσο έλεγχο σωστού και λάθους, κουμπί βοήθειας και λύσης καθώς και δυνατότητα επανάληψης χωρίς ανανέωση της σελίδας. Ο κώδικας βασίζεται σε δομές δεδομένων και σε χειρισμό με δύο κλικ για να είναι εύχρηστος από παιδιά πρωτοβάθμιας εκπαίδευσης.

## 6.4 Μελέτη Περιβάλλοντος

Τα παιχνίδια μελέτης περιβάλλοντος έχουν στόχο να ευαισθητοποιήσουν τους μαθητές σχετικά με τη σωστή περιβαλλοντική συμπεριφορά. Στα παιχνίδια αυτά ο μαθητής μαθαίνει να προστατεύει τη φύση, να διακρίνει τα ανακυκλώσιμα από τα απορρίματα και να κατηγοριοποιεί τρόφιμα. Και τα 3 παιχνίδια έχουν κοινή λογική διεπαφής με κουμπιά Πίσω, Βοήθεια, Menu, Λύση και Ξαναπαίξε.

### 6.4.1 Σκουπίδια στην Θάλασσα (environment1.html)

Το παιχνίδι αυτό εμφανίζει σκουπίδια μέσα στην θάλασσα. Ο μαθητής πρέπει να κάνει κλικ πάνω στα σκουπίδια για να τα μαζέψει. Η βασική ιδέα στο HTML είναι ότι τα σκουπίδια είναι εικόνες με class trash. Εκτός από τα σκουπίδια εμφανίζονται και ψάρια, τα οποία δεν πρέπει να πατηθούν. Αυτά ξεχωρίζουν μέσω της class fish.

```

















```

**34** Δομή HTML: σκουπίδια (trash) και ψάρια (fish) ως διαφορετικά στοιχεία

Κατά την έναρξη του παιχνιδιού, όλα τα αντικείμενα τοποθετούνται σε τυχαίες θέσεις ώστε να μην πέφτουν το ένα πάνω από το άλλο. Αυτό γίνεται με τη συνάρτηση randomizePositionsNoOverlap

```

function randomizePositionsNoOverlap(elements) {
  const itemW = 120;
  const itemH = 120;
  const gap = 10;

  const W = window.innerWidth;
  const H = window.innerHeight;

  const minX = 10;
  const maxX = Math.max(10, W - itemW - 10);

  const minY = 90;
  const maxY = Math.max(90, H - itemH - 140);

  const placed = [];

  function overlaps(r, others) {
    return others.some(o =>
      r.x < o.x + o.w + gap &&
      r.x + r.w + gap > o.x &&
      r.y < o.y + o.h + gap &&
      r.y + r.h + gap > o.y
    );
  }

  elements.forEach(el => {
    let ok = false;

    for (let attempt = 0; attempt < 2000; attempt++) {
      const x = Math.floor(minX + Math.random() * (maxX - minX + 1));
      const y = Math.floor(minY + Math.random() * (maxY - minY + 1));
      const rect = { x, y, w: itemW, h: itemH };

      if (!overlaps(rect, placed)) {
        el.style.left = x + 'px';
        el.style.top = y + 'px';
        placed.push(rect);
        ok = true;
        break;
      }
    }

    if (!ok) {
      const x = Math.floor(minX + Math.random() * (maxX - minX + 1));
      const y = Math.floor(minY + Math.random() * (maxY - minY + 1));
      el.style.left = x + 'px';
      el.style.top = y + 'px';
      placed.push({ x, y, w: itemW, h: itemH });
    }
  });
}

```

### 35 Τυχαία τοποθέτηση αντικειμένων χωρίς επικάλυψη (randomizePositionsNoOverlap)

Τα σκουπίδια έχουν click event. Αν ο μαθητής πατήσει σκουπίδι, αυτό αφαιρείται από την οθόνη, μειώνεται ο μετρητής remainingTrash και εμφανίζεται μήνυμα «Μπράβο». Αν ο μαθητής πατήσει ψάρι, εμφανίζεται μήνυμα λάθους.

```

trashItems.forEach(t => {
  t.style.pointerEvents = 'auto';
  t.addEventListener('click', () => {
    if (gameOver) return;

    if (lastHintEl === t) {
      t.classList.remove('hint-glow');
      lastHintEl = null;
      hintBusy = false;
    }

    t.remove();
    remainingTrash--;

    if (remainingTrash > 0) {
      showMsg("Μπράβο!", 2, '#22c55e');
    } else {
      endGame();
    }
  });
});

fishItems.forEach(f => {
  f.style.pointerEvents = 'auto';
  f.addEventListener('click', () => {
    if (gameOver) return;
    showMsg("Λάθος, προσιάτεψε τα ψάρια.", 5, 'red');
  });
});

```

### 36 Αλληλεπίδραση με σκουπίδια και ψάρια μέσω click events

Στο τέλος εμφανίζεται μήνυμα επιτυχίας και ενεργοποιείται το κουμπί επανάληψης. Επίσης τα ψάρια απενεργοποιούνται για να μην είναι δυνατό να γίνει κλικ σε αυτά μετά την λήξη.

```

function endGame() {
  gameOver = true;

  if (timeoutRef) {
    clearTimeout(timeoutRef);
    timeoutRef = null;
  }

  if (lastHintEl) {
    lastHintEl.classList.remove('hint-glow');
    lastHintEl = null;
  }
  hintBusy = false;

  messageBox.textContent = "Συγχαρητήρια, μάζεψε όλα τα σκουπίδια!";
  messageBox.style.color = '#22c55e';
  messageBox.style.opacity = 1;

  const fishItems = Array.from(document.querySelectorAll('.fish'));
  fishItems.forEach(f => {
    f.style.pointerEvents = 'none';
  });

  hintBtn.style.display = 'none';
  replayBtn.style.display = 'inline-block';
}

```

### 37 Ολοκλήρωση παιχνιδιού, απενεργοποίηση ψαριών και εμφάνιση επανάληψης

Το κουμπί της λύσης δεν ολοκληρώνει το παιχνίδι αλλά εστιάζει σε ένα σκουπίδι για 2 δευτερόλεπτα ώστε να βοηθήσει τον μαθητή να το εντοπίσει.

```
function showNextHint(){
  if (gameOver) return;
  if (hintBusy) return;

  if (lastHintEl) {
    lastHintEl.classList.remove('hint-glow');
    lastHintEl = null;
  }

  let tries = 0;
  while (tries < hintTrashQueue.length) {
    const idx = hintIndex % hintTrashQueue.length;
    const el = hintTrashQueue[idx];
    hintIndex = (hintIndex + 1) % hintTrashQueue.length;

    if (document.body.contains(el)) {
      el.classList.add('hint-glow');
      lastHintEl = el;

      hintBusy = true;
      setTimeout(() => {
        if (lastHintEl) lastHintEl.classList.remove('hint-glow');
        lastHintEl = null;
        hintBusy = false;
      }, 2000);

      return;
    }

    tries++;
  }
}
```

### 38 Υπόδειξη σκουπιδιού με προσωρινό highlight μέσω κουμπιού «Λύση»

#### 6.4.2 Σκουπίδια στον Σωστό Κάδο (environment2.html)

Σε αυτό το παιχνίδι υπάρχουν δύο κάδοι, ένας ανακύκλωσης και ένας απλός κάδος και μία σειρά αντικειμένων. Κάθε κάδος έχει data-bin ώστε η JavaScript να ξέρει ποιος είναι ποιος.

Τα αντικείμενα δημιουργούνται δυναμικά μέσα στο trashRow από έναν πίνακα trashTypesBase. Ο κανόνας για το ποια αντικείμενα πάνε προς ανακύκλωση ορίζεται σε Set.

### 39 Δυναμική δημιουργία αντικειμένων και κανόνες ανακύκλωσης με Set

Όταν ο μαθητής βάλει ένα αντικείμενο σε έναν κάδο, γίνεται έλεγχος με βάση το recycleSet. Αν είναι σωστό, το αντικείμενο αφαιρείται. Αν είναι λάθος επιστρέφει στη θέση του.

```
<div class="bins-row">
  
  
</div>

<div class="trash-row" id="trashRow"></div>
<div class="message" id="messageBox"></div>

<script>
  const helpOverlay = document.getElementById('helpOverlay');
  const trashRow = document.getElementById('trashRow');
  const messageBox = document.getElementById('messageBox');
  const bins = Array.from(document.querySelectorAll('.bin-img'));
  const hintBtn = document.getElementById('hintBtn');
  const replayBtn = document.getElementById('replayBtn');
  const recycleSet = new Set(['can', 'bottle', 'book']);
  const trashTypesBase = ['banana', 'apple', 'book', 'can', 'bottle', 'orange'];

  const correctBin = recycleSet.has(trashType) ? 'recycle' : 'bin';

  if (binType === correctBin){
    clearAllHints();
    draggedTrash.remove();
    remaining--;
    showMessage('Σωστό', 'limegreen', 2000);

    if (remaining === 0){
      gameOver = true;
      hintBtn.style.display = 'none';
      replayBtn.style.display = 'inline-block';
      showWinPermanent();
    }
  } else {
    showMessage('Λάθος Κάδος', 'red', 3000);
    resetTrash(draggedTrash);
  }
</script>
```

### 40 Έλεγχος σωστής απόθεσης αντικειμένων βάσει recycleSet

Η μετακίνηση υλοποιείται με δύο διαδοχικά κλικ. Με το πρώτο γίνεται σύλληψη του αντικειμένου και με το δεύτερο γίνεται η απόθεση. Επίσης υπάρχει κατάλληλος κώδικας στο event mousemove, ώστε το αντικείμενο να ακολουθεί την κίνηση του δρομέα του ποντικιού. Το κουμπί της λύσης εμφανίζει για λίγα δευτερόλεπτα μία μικρή ετικέτα κάτω από ένα τυχαίο αντικείμενο, γράφοντας αν πρέπει να πάει στην ανακύκλωση ή στον κάδο.

```

function bindDragHandlers(items){
  items.forEach(item => {
    item.addEventListener('click', (e)=>{
      if (gameOver) return;

      if (!draggedTrash){
        pickTrash(item, e);
        e.stopPropagation();
        return;
      }

      dropTrash(e);
      e.stopPropagation();
    });
  });

  requestAnimationFrame(()=>layoutTrashSlots());
  window.addEventListener('resize', ()=>{ requestAnimationFrame(()=>layoutTrashSlots()); });
}

document.addEventListener('mousemove', (e)=>{
  if (!draggedTrash || gameOver) return;
  draggedTrash.style.left = (e.clientX - offsetX) + 'px';
  draggedTrash.style.top = (e.clientY - offsetY) + 'px';
});

document.addEventListener('click', (e)=>{
  if (!draggedTrash || gameOver) return;
  dropTrash(e);
});

function solveHintRandom(){
  if (gameOver) return;
  const remainingItems = Array.from(document.querySelectorAll('.trash-item'));
  if (!remainingItems.length) return;

  const pick = remainingItems[Math.floor(Math.random() * remainingItems.length)];
  showHintUnderTrash(pick);
}

```

#### 41 Μετακίνηση αντικειμένων με δύο κλικ και οπτική υπόδειξη κάδου μέσω «Λύσης»

Στο τέλος εμφανίζεται μήνυμα επιτυχίας και το κουμπί επανάληψης.

```

if (remaining === 0){
  gameOver = true;
  hintBtn.style.display = 'none';
  replayBtn.style.display = 'inline-block';
  showWinPermanent();
}

```

#### 42 Τερματισμός παιχνιδιού και εμφάνιση κουμπιού επανάληψης

### 6.4.2 Φαγητά στην Σωστή Κατηγορία (environment3.html)

Σε αυτό το παιχνίδι ο μαθητής κατηγοριοποιεί τρόφιμα σε 5 κατηγορίες. Οι κατηγορίες δηλώνονται ως εικόνες με data-cat

#### 43 Ορισμός κατηγοριών μέσω εικόνων με data-cat

Ο βασικός κανόνας του παιχνιδιού είναι ο πίνακας `foodToCategory`, που ορίζει σε ποια κατηγορία ανήκει κάθε τρόφιμο.

```






const foodToCategory = {
  cucumber: 'vegetables',
  pepper: 'vegetables',
  lettuce: 'vegetables',

  apple: 'fruits',
  banana: 'fruits',
  orange: 'fruits',

  chicken: 'meat',
  steak: 'meat',
  duck: 'meat',

  lobster: 'fish',
  fishfood: 'fish',
  octopus: 'fish',

  fakes: 'legumes',
  fasolakia: 'legumes',
  giants: 'legumes'
};
```

#### 44 Κανόνας αντιστοίχισης τροφίμων σε κατηγορίες με `foodToCategory`

Τα τρόφιμα προκύπτουν από λίστα `foodListBase`, ανακατεύονται και μοιράζονται σε δύο σειρές, ώστε να χωράνε στην οθόνη.

```
const list = shuffle(foodListBase.slice());
const top = list.slice(0, 7);
const bottom = list.slice(7);
```

#### 45 Δημιουργία και διάταξη τροφίμων από το `foodListBase`

Η μετακίνηση γίνεται με κλικ, όπως και στα άλλα παιχνίδια.

```
item.addEventListener('click', (e) => {
  if (gameOver) return;

  if (!draggedFood) {
    pickFood(item, e);
    e.stopPropagation();
    return;
  }
});
```

#### 46 Μετακίνηση αντικειμένων με μηχανισμό δύο κλικ

Κατά την απόθεση, συγκρίνεται το data-cat της κατηγορίας με τη σωστή κατηγορία που προκύπτει από το foodToCategory.

```
const foodKey = draggedFood.dataset.food;
const correctCategory = foodToCategory[foodKey];

for (const cat of categories) {
  const rect = cat.getBoundingClientRect();
  if (e.clientX > rect.left && e.clientX < rect.right &&
    e.clientY > rect.top && e.clientY < rect.bottom) {

    const catKey = cat.dataset.cat;

    if (catKey === correctCategory) {
      clearAllHints();
      draggedFood.remove();
      remaining--;
      showMessage('Σωστό', 'limegreen', 2000);

      if (remaining === 0) {
        gameOver = true;
        categoriesRow.style.display = 'none';
        hintBtn.style.display = 'none';
        replayBtn.style.display = 'inline-block';
        showWinPermanent();
      }
    } else {
      showMessage('Λάθος', 'red', 3000);
      resetFood(draggedFood);
    }
  }
}
```

#### 47 Έλεγχος σωστής κατηγορίας τροφίμου κατά την απόθεση

Το κουμπί της λύσης εμφανίζει ετικέτα hint κάτω από ένα τυχαίο τρόφιμο με τη σωστή κατηγορία.

```
function hintRandomFood() {
  if (gameOver) return;
  const remainingFoods = Array.from(document.querySelectorAll('.food-item'));
  if (!remainingFoods.length) return;

  const pick = remainingFoods[Math.floor(Math.random() * remainingFoods.length)];
  showHintUnderFood(pick);
}
```

#### 48 Υπόδειξη σωστής κατηγορίας τροφίμου με ετικέτα hint

Όταν όλα τα τρόφιμα τοποθετηθούν σωστά, εμφανίζεται μήνυμα επιτυχίας και ενεργοποιείται το κουμπί επανάληψης

```
if (remaining === 0) {
  gameOver = true;
  categoriesRow.style.display = 'none';
  hintBtn.style.display = 'none';
  replayBtn.style.display = 'inline-block';
  showWinPermanent();
}
```

## 49 Ολοκλήρωση παιχνιδιού κατηγοριοποίησης και ενεργοποίηση επανάληψης

### 6.4.3 Συμπέρασμα Μελέτης Περιβάλλοντος

Τα παιχνίδια μελέτης περιβάλλοντος ακολουθούν κοινή λογική. Έχουν απλή αλληλεπίδραση, σαφείς κανόνες, άμεση ανατροφοδότηση σωστού και λάθους, λειτουργία λύσης και δυνατότητα επανάληψης. Με αυτόν τον τρόπο οι μαθητές μαθαίνουν βασικές αρχές προστασίας του περιβάλλοντος και σωστής κατηγοριοποίησης με ευχάριστο και παιγνιώδη τρόπο.

## 6.5 Ιστορία

Τα παιχνίδια ιστορίας έχουν στόχο να βοηθήσουν τους μαθητές να γνωρίσουν ιστορικά πρόσωπα και χρονολογίες με διαδραστικό τρόπο. Υπάρχει κοινή λογική διεπαφής, όπως στα υπόλοιπα παιχνίδια, και άμεση ανατροφοδότηση σωστού και λάθους.

### 6.5.1 Ποιος είναι ο Εικονιζόμενος (history1.html)

Σε αυτό το παιχνίδι εμφανίζεται μία εικόνα ενός προσώπου και ο μαθητής πρέπει να επιλέξει σωστά το όνομα από τα κουμπιά επιλογών στο κάτω μέρος. Η εικόνα εμφανίζεται στο characterContainer και αλλάζει σε κάθε γύρο με βάση τα δεδομένα του παιχνιδιού.

Οι επιλογές ονομάτων είναι κουμπιά με data-key ώστε η JavaScript να ξέρει ποιο πρόσωπο διάλεξε ο μαθητής.

```
<div class="character-container" id="characterContainer">
  
</div>

<div class="names-row" id="namesRow">
  <button class="name-btn" data-key="napoleon">Ναπολέον</button>
  <button class="name-btn" data-key="alexander">Μέγας Αλέξανδρος</button>
  <button class="name-btn" data-key="zeus">Δίας</button>
  <button class="name-btn" data-key="hercules">Ηρακλής</button>
  <button class="name-btn" data-key="washington">Ουάσιγκτον</button>
  <button class="name-btn" data-key="kolokotronis">Κολοκοτρώνης</button>
</div>
```

### 50 Επιλογή ονομάτων μέσω κουμπιών με data-key

Οι γύροι ορίζονται σε πίνακα stagesBase όπου κάθε γύρος έχει εικόνα και το σωστό key.

```
const stagesBase = [
  { img: 'zeus.jpg', correct: 'zeus' },
  { img: 'kolokotronis.jpg', correct: 'kolokotronis' },
  { img: 'hercules.jpg', correct: 'hercules' },
  { img: 'washington.jpg', correct: 'washington' },
  { img: 'alexander.jpg', correct: 'alexander' },
  { img: 'napoleon.jpg', correct: 'napoleon' }
];
```

### 51 Ορισμός γύρων αναγνώρισης μέσω πίνακα stagesBase

Στην αρχή γίνεται ανακάτεμα για να μην εμφανίζονται με την ίδια σειρά.

```
stages = shuffle(stagesBase.slice());
```

### 52 Τυχαία αναδιάταξη γύρων στην αρχή του παιχνιδιού

Όταν πατηθεί ένα κουμπί, συγκρίνεται το data-key του κουμπιού με το correct του τρέχοντος γύρου.

```

const key = btn.dataset.key;
const currentStage = stages[stageIndex];

if (key === currentStage.correct) {
  handleCorrect();
} else {
  handleWrong();
}

```

### 53 Έλεγχος σωστής επιλογής με σύγκριση data-key και correct γύρου

Σε σωστή απάντηση εμφανίζεται μήνυμα «Σωστό» και μετά από δύο δευτερόλεπτα γίνεται μετάβαση στον επόμενο γύρο. Σε λάθος απάντηση εμφανίζεται «Λάθος»

```

showMessage('Σωστό', 'limegreen', 2000);

if (stageIndex === stages.length - 1) {
  nextTimeout = setTimeout(() => {
    finishGame();
  }, 2000);
  return;
}

nextTimeout = setTimeout(() => {
  goNextStage();
}, 2000);
}

function handleWrong() {
  if (gameOver) return;
  showMessage('Λάθος', 'red', 2000);
}

```

### 54 Εμφάνιση μηνύματος σωστής ή λάθος απάντησης και μετάβαση στον επόμενο γύρο

Το κουμπί λύσης δεν επιλέγει αντί του μαθητή αλλά υποδεικνύει τη σωστή επιλογή με πράσινο outline στο σωστό κουμπί. Όταν ολοκληρωθούν όλοι οι γύροι, καλείται η finishGame() η οποία απενεργοποιεί την εικόνα, τις επιλογές και το κουμπί λύσης, εμφανίζει το κουμπί «Ξαναπαίξε» και εμφανίζει μήνυμα επιτυχίας.

```

function showHintCorrect(){
  if (gameOver) return;
  clearHint();
  const currentStage = stages[stageIndex];
  const correctBtn = buttons.find(b => b.dataset.key === currentStage.correct);
  if (correctBtn) correctBtn.classList.add('hint-correct');
}

function finishGame(){
  gameOver = true;
  characterBox.style.display = 'none';
  namesRow.style.display = 'none';
  hintBtn.style.display = 'none';
  replayBtn.style.display = 'inline-block';

  messageBox.classList.add('center');
  showMessage('Συγχαρητήρια τους βρήκες όλους!', 'limegreen', null);
}

```

## 55 Υπόδειξη σωστής επιλογής και τελικός τερματισμός παιχνιδιού (finishGame)

### 6.5.2 Πότε Έζησαν οι Προσωπικότητες (history2.html)

Σε αυτό το παιχνίδι ο μαθητής πρέπει να ταιριάζει κάθε προσωπικότητα με τη σωστή χρονολογία. Επάνω υπάρχουν οι κάρτες των προσώπων και κάτω τα πλαίσια των χρονολογιών. Κάθε εικόνα έχει data-id ώστε να γίνεται σωστός έλεγχος.

```
<div class="person-card">
  <div class="person-name">Κολοκοτρώνης</div>
  
</div>

<div class="person-card">
  <div class="person-name">Κωνσταντίνος Παλαιολόγος</div>
  
</div>

<div class="person-card">
  <div class="person-name">Ναπολέον</div>
  
</div>

<div class="person-card">
  <div class="person-name">Καραγκούνης</div>
  
</div>

<div class="person-card">
  <div class="person-name">Όμηρος</div>
  
</div>

<div class="person-card">
  <div class="person-name">Μεταξάς</div>
  
</div>

<div class="person-card">
  <div class="person-name">Μέγας Αλέξανδρος</div>
  
</div>
```

### 56 Αναγνώριση εικόνων μέσω data-id για σωστό έλεγχο

Κάθε θέση χρονολογίας έχει data-date ώστε να είναι συγκεκριμένος ο στόχος.

```

<div class="date-slot">
  <div class="drop-area" data-date="800bc"></div>
  <p class="date-label">800π.χ.</p>
</div>

<div class="date-slot">
  <div class="drop-area" data-date="336bc"></div>
  <p class="date-label">336π.χ.</p>
</div>

<div class="date-slot">
  <div class="drop-area" data-date="1453"></div>
  <p class="date-label">1453</p>
</div>

<div class="date-slot">
  <div class="drop-area" data-date="1799"></div>
  <p class="date-label">1799</p>
</div>

<div class="date-slot">
  <div class="drop-area" data-date="1821"></div>
  <p class="date-label">1821</p>
</div>

<div class="date-slot">
  <div class="drop-area" data-date="1940"></div>
  <p class="date-label">1940</p>
</div>

<div class="date-slot">
  <div class="drop-area" data-date="2004"></div>
  <p class="date-label">2004</p>
</div>

```

### 57 Ορισμός στόχων χρονολογίας με data-date

Ο βασικός κανόνας του παιχνιδιού είναι ο πίνακας correctMap, που λέει ποιο πρόσωπο ταιριάζει σε ποια χρονολογία.

```

const correctMap = {
  omiros: "800bc",
  alexander: "336bc",
  palaiologos: "1453",
  napoleon: "1799",
  kolokotronis: "1821",
  metaxas: "1940",
  karagounis: "2004"
};

```

### 58 Αντιστοίχιση προσώπων με χρονολογίες μέσω του πίνακα correctMap

Η μετακίνηση γίνεται με μηχανισμό διαδοχικών κλικ όπως και στα υπόλοιπα παιχνίδια. Με το πρώτο πιάνει την εικόνα και με το δεύτερο την αφήνει στην χρονολογία.

```
img.addEventListener('click', (e)=>{
  if (gameOver) return;

  if (!draggedImg){
    pick(img, e);
    e.stopPropagation();
    return;
  }

  drop(e);
  e.stopPropagation();
});
```

#### 59 Μετακίνηση εικόνων σε χρονολογίες με μηχανισμό δύο κλικ

Κατά το σύρσιμο, η εικόνα ακολουθεί την κίνηση του ποντικιού.

```
document.addEventListener('mousemove', (e)=>{
  if (!draggedImg || gameOver) return;
  draggedImg.style.left = (e.clientX - offsetX) + 'px';
  draggedImg.style.top = (e.clientY - offsetY) + 'px';
});
```

#### 60 Ιχνηλάτηση κίνησης του δρομέα του ποντικιού με mousemove

Όταν ο μαθητής αφήσει την εικόνα πάνω σε drop-area, γίνεται έλεγχος χρονολογίας, αντιπαραβολή με τη σωστή και αν το drop-area είναι κενό. Αν είναι σωστό τότε εμφανίζεται μήνυμα σωστού, η εικόνα κλειδώνεται (locked) μέσα στο drop-area και η αρχική εικόνα αποκρύπτεται.

```
if (targetDate === correctDate && dropArea.childElementCount === 0) {
  showMessage('Σωστό', 'limegreen', 2000);

  const locked = document.createElement('img');
  locked.src = draggedImg.src;
  locked.alt = draggedImg.alt;
  locked.className = 'locked-img';
  locked.setAttribute('draggable', 'false');
  dropArea.innerHTML = '';
  dropArea.appendChild(locked);
  draggedImg.style.visibility = 'hidden';
  placed[draggedId] = true;
  placedNow = true;
} else {
  showMessage('Λάθος', 'red', 2000);
}
```

#### 61 Έλεγχος σωστής απόθεσης εικόνας σε χρονολογία και κλειδωμα θέσης

Το κουμπί της λύσης συμπληρώνει ένα σωστό ζευγάρι κάθε φορά. Επιλέγει τυχαία ένα πρόσωπο που δεν έχει τοποθετηθεί ακόμη και το βάζει στην σωστή χρονολογία.

```
function solveOne() {
  if (gameOver) return;

  const remaining = getRemainingIds();
  if (!remaining.length) return;

  const pickId = remaining[Math.floor(Math.random() * remaining.length)];
  const correctDate = correctMap[pickId];

  const drop = dropAreas.find(a => a.dataset.date === correctDate);
  if (!drop || drop.childElementCount !== 0) return;

  const img = document.querySelector(`.person-img[data-id="${pickId}"]`);
  if (!img) return;

  const locked = document.createElement('img');
  locked.src = img.src;
  locked.alt = img.alt;
  locked.className = 'locked-img';
  locked.setAttribute('draggable', 'false');
  drop.innerHTML = '';
  drop.appendChild(locked);
  img.style.visibility = 'hidden';
  placed[pickId] = true;
  checkAllDone();
}
```

## 62 Σταδιακή επίλυση αντιστοίχισης προσώπου-χρονολογίας μέσω «Λύσης»

Όταν όλα τοποθετηθούν σωστά, αποκρύπτεται η λύση, εμφανίζεται κουμπί επανάληψης και εμφανίζεται μήνυμα επιτυχίας.

```
gameOver = true;
hintBtn.style.display = 'none';
replayBtn.style.display = 'inline-block';
peopleRow.style.display = 'none';
showMessage("Συγχαρητήρια!", "limegreen", null);
```

## 63 Ολοκλήρωση παιχνιδιού και εμφάνιση επανάληψης με μήνυμα επιτυχίας

### 6.5.3 Συμπέρασμα Ιστορίας

Τα παιχνίδια ιστορίας χρησιμοποιούν δύο διαφορετικούς μηχανισμούς μάθησης. Χρησιμοποιούν αναγνώριση προσώπου με επιλογή ονόματος και αντιστοίχιση προσώπου με χρονολογία. Και στα δύο υπάρχει τυχαία σειρά, άμεση ανατροφοδότηση σωστού και λάθους, λειτουργία λύσης και δυνατότητα επανάληψης, γεγονός που ενισχύει την εξάσκηση των μαθητών χωρίς πίεση.

## 6.6 Γεωγραφία

Τα παιχνίδια γεωγραφίας έχουν στόχο να βοηθήσουν τους μαθητές να γνωρίσουν χώρες, χάρτες, σημαίες και πρωτεύουσες μέσα από διαδραστικές δραστηριότητες. Και στα δύο

παιχνίδια χρησιμοποιείται άμεση ανατροφοδότηση σωστού και λάθους καθώς και κουμπιά βοήθειας και επανάληψης.

### 6.6.1 Βρες τις Χώρες στον Χάρτη (geography1.html)

Σε αυτό το παιχνίδι ο μαθητής βλέπει έναν χάρτη της Ευρώπης, όπου μία χώρα εμφανίζεται τονισμένη με πράσινο χρώμα. Ο στόχος είναι να επιλέξει το σωστό όνομα χώρας από τις διαθέσιμες επιλογές.

Η εικόνα του χάρτη εμφανίζεται σε div με background-image, ώστε να μπορεί να αλλάξει δυναμικά.

```
<div class="map-bg" id="mapBg"></div>
```

#### 64 Εμφάνιση χάρτη με δυναμικό background-image σε div

Η αλλαγή της εικόνας γίνεται μέσω JavaScript.

```
function setMapImage(src) {
    mapBg.style.backgroundImage = `url("${src}")`;
}
```

#### 65 Δυναμική αλλαγή εικόνας χάρτη μέσω JavaScript

Οι χώρες που χρησιμοποιούνται στο παιχνίδι ορίζονται στον πίνακα pool, όπου κάθε αντικείμενο περιλαμβάνει το όνομα της χώρας και το αρχείο εικόνας του χάρτη με την χώρα τονισμένη. Υπάρχει επίσης λίστα με όλες τις χώρες της Ευρώπης.

```
const pool = [
  { country: "Ελλάδα", file: "../photos/europe_greece.jpg" },
  { country: "Γαλλία", file: "../photos/europe_france.jpg" },
  { country: "Ισπανία", file: "../photos/europe_spain.jpg" },
  { country: "Πορτογαλία", file: "../photos/europe_portugal.jpg" },
  { country: "Γερμανία", file: "../photos/europe_germany.jpg" },
  { country: "Μεγάλη Βρετανία", file: "../photos/europe_england.jpg" },
  { country: "Ιταλία", file: "../photos/europe_italy.jpg" }
];

const europeCountries = [
  "Αλβανία", "Ανδόρα", "Αυστρία", "Λευκορωσία", "Βέλγιο", "Βοσνία & Ερζεγοβίνη", "Βουλγαρία", "Κροατία", "Κύπρος", "Τσεχία",
  "Δανία", "Εσθονία", "Φινλανδία", "Γαλλία", "Γερμανία", "Ελλάδα", "Ουγγαρία", "Ισλανδία", "Ιρλανδία", "Ιταλία",
  "Κόσοβο", "Λετονία", "Λιχτενστάιν", "Λιθουανία", "Λουξεμβούργο", "Μάλτα", "Μολδαβία", "Μονακό", "Μαυροβούνιο", "Ολλανδία",
  "Βόρεια Μακεδονία", "Νορβηγία", "Πολωνία", "Πορτογαλία", "Ρουμανία", "Ρωσία", "Άγιος Μαρίνος", "Σερβία", "Σλοβακία", "Σλοβενία",
  "Ισπανία", "Σουηδία", "Ελβετία", "Ουκρανία", "Μεγάλη Βρετανία", "Βατικανό"
];
```

#### 66 Ορισμός χωρών και επιλογών μέσω pool και λίστας ευρωπαϊκών χωρών

Για κάθε γύρο δημιουργούνται κουμπιά επιλογών χωρών στο κάτω μέρος της οθόνης. Η σωστή απάντηση αναμειγνύεται με λανθασμένες ώστε να μην έχει πάντα την ίδια θέση.

```
function buildOptions() {
  const correct = current;
  const others = europeCountries.filter(c => c !== correct);
  shuffle(others);

  const CHOICES = 7;
  const pick = [correct, ...others.slice(0, Math.max(CHOICES - 1, 1))];
  shuffle(pick);

  wordsBar.innerHTML = "";
  pick.forEach(name => {
    const b = document.createElement("button");
    b.className = "word-chip";
    b.dataset.country = name;
    b.textContent = name;
    b.addEventListener("click", () => onAnswerClick(name));
    wordsBar.appendChild(b);
  });
}
```

#### 67 Δημιουργία επιλογών χωρών και τυχαία διάταξη σωστής απάντησης

Όταν ο μαθητής πατήσει μία χώρα, γίνεται σύγκριση με τη σωστή απάντηση.

```
function onAnswerClick(country) {
  if (gameOver || locked) return;
  if (!current) return;

  if (country === current) {
    locked = true;
    clearHintGlow();
    showMessage("Σωστό", "limegreen", 2000);

    solved[current] = true;

    setTimeout(() => {
      locked = false;
      if (!getRemaining().length) {
        finishGame();
      } else {
        pickNext();
      }
    }, 2000);
  } else {
    showMessage("Λάθος", "red", 2000);
  }
}
```

#### 68 Έλεγχος σωστής επιλογής χώρας με σύγκριση απαντήσεων

Όταν ολοκληρωθούν όλες οι χώρες το παιχνίδι τερματίζεται.

```
function finishGame() {
  gameOver = true;
  locked = false;
  clearHintGlow();

  mapBg.style.display = "none";
  pageTitle.style.display = "none";

  hintBtn.style.display = 'none';
  replayBtn.style.display = 'inline-block';
  showMessage("Συγχαρητήρια!", "limegreen", null);
}
```

### 69 Τερματισμός παιχνιδιού μετά την ολοκλήρωση όλων των χωρών

Το κουμπί της λύσης υποδεικνύει τη σωστή χώρα με πράσινο περίγραμμα στο αντίστοιχο κουμπί. Η επανεκκίνηση επαναφέρει όλες τις μεταβλητές και ξεκινά νέο παιχνίδι.

```
function hintNow(){
  if (gameOver || locked) return;
  clearHintGlow();
  if (!current) return;
  const btn = wordsBar.querySelector(`.word-chip[data-country="${current}"]`);
  if (btn) btn.classList.add("hint-correct");
}

function restartGame(){
  gameOver = false;
  locked = false;

  if (msgTimeout) { clearTimeout(msgTimeout); msgTimeout = null; }
  messageBox.textContent = '';
  messageBox.style.opacity = 0;
  hintBtn.style.display = 'inline-block';
  replayBtn.style.display = 'none';
  mapBg.style.display = "block";
  pageTitle.style.display = "block";
  solved = {};
  allQuizCountries.forEach(c => solved[c] = false);
  clearHintGlow();
  pickNext();
}
```

### 70 Υπόδειξη σωστής χώρας και επανεκκίνηση παιχνιδιού

#### 6.6.2 Σωστή Χώρα και Πρωτεύουσα (environment2.html)

Σε αυτό το παιχνίδι ο μαθητής βλέπει μία σημαία και καλείται να επιλέξει τη σωστή χώρα και πρωτεύουσα ανάμεσα σε 3 επιλογές για το καθένα. Η εικόνα της σημαίας εμφανίζεται σε στοιχείο <img>

## 71 Εμφάνιση σημαίας μέσω στοιχείου <img>

Οι γύροι του παιχνιδιού ορίζονται στον πίνακα `stagesBase`. Κάθε γύρος περιέχει το αρχείο της σημαίας, την σωστή χώρα, την σωστή πρωτεύουσα και επιλογές.

```

const stagesBase = [
  { flag:"australia.jpg", countryCorrect:"Αυστραλία",
  { flag:"england.jpg", countryCorrect:"Αγγλία",
  { flag:"denmark.jpg", countryCorrect:"Δανία",
  { flag:"germany.jpg", countryCorrect:"Γερμανία",
  { flag:"greece.jpg", countryCorrect:"Ελλάδα",
  { flag:"italy.jpg", countryCorrect:"Ιταλία",
  { flag:"japan.jpg", countryCorrect:"Ιαπωνία",
  { flag:"serbia.jpg", countryCorrect:"Σερβία",
  { flag:"slovenia.jpg", countryCorrect:"Σλοβενία",
  { flag:"spain.jpg", countryCorrect:"Ισπανία",
  { flag:"usa.jpg", countryCorrect:"Η.Π.Α.",
];
```

## 72 Ορισμός γύρων σημαίας-χώρας-πρωτεύουσας μέσω πίνακα `stagesBase`

Πριν ξεκινήσει το παιχνίδι, οι επιλογές ανακατεύονται.

```
function buildStages() {
  stages = shuffle(stagesBase.map(s => ({
    ...s,
    countryOptions: shuffle([...s.countryOptions]),
    capitalOptions: shuffle([...s.capitalOptions])
  })));
}
```

## 73 Τυχαία αναδιάταξη επιλογών πριν την έναρξη του παιχνιδιού

Ο μαθητής επιλέγει χώρα και πρωτεύουσα πατώντας στα αντίστοιχα κουμπιά. Οι επιλογές αποθηκεύονται σε μεταβλητές.

```
let selectedCountry = null;
let selectedCapital = null;
```

## 74 Επιλογή χώρας και πρωτεύουσας και αποθήκευση επιλογών σε μεταβλητές

```
function highlightCorrect() {
  clearHint();
  const s = stages[index];
  const countryBtns = Array.from(countryOptions.querySelectorAll('.option-btn'));
  const capitalBtns = Array.from(capitalOptions.querySelectorAll('.option-btn'));
  const c1 = countryBtns.find(b => b.textContent === s.countryCorrect);
  const c2 = capitalBtns.find(b => b.textContent === s.capitalCorrect);

  if (c1) c1.classList.add('hint-correct');
  if (c2) c2.classList.add('hint-correct');
}
```

Ο έλεγχος γίνεται με το κουμπί «Υποβολή»

```

submitBtn.addEventListener('click', () => {

  if (!selectedCountry && !selectedCapital) {
    showMessage("Διάλεξε Χώρα\nκαι Πρωτεύουσα", "red", 2000);
    return;
  }
  if (!selectedCountry && selectedCapital) {
    showMessage("Διάλεξε Χώρα", "red", 2000);
    return;
  }
  if (selectedCountry && !selectedCapital) {
    showMessage("Διάλεξε Πρωτεύουσα", "red", 2000);
    return;
  }

  const s = stages[index];
  const wrongCountry = (selectedCountry !== s.countryCorrect);
  const wrongCapital = (selectedCapital !== s.capitalCorrect);

  if (!wrongCountry && !wrongCapital) {
    clearHint();
    showMessage("Σωστό", "limegreen", 2000);
    nextStage();
    return;
  }

  if (wrongCountry && wrongCapital) {
    showMessage("Λάθος Χώρα\nηΛάθος Πρωτεύουσα", "red", 2300);
  } else if (wrongCountry) {
    showMessage("Λάθος Χώρα", "red", 2300);
  } else {
    showMessage("Λάθος Πρωτεύουσα", "red", 2300);
  }
});

```

#### 75 Έλεγχος απαντήσεων μέσω κουμπιού «Υποβολή»

Το κουμπί της λύσης υποδεικνύει τη σωστή χώρα και πρωτεύουσα με πράσινο περίγραμμα.

#### 76 Υπόδειξη σωστής χώρας και πρωτεύουσας με πράσινο περίγραμμα

Όταν ολοκληρωθούν όλοι οι γύροι, εμφανίζεται μήνυμα επιτυχίας και το κουμπί επανάληψης.

```
function finishGame() {
    flagImg.style.display = "none";
    countryTitle.style.display = "none";
    countryOptions.style.display = "none";
    capitalTitle.style.display = "none";
    capitalOptions.style.display = "none";
    submitBtn.style.display = "none";
    finalMessage.style.display = "block";
    replayBtn.style.display = "inline-block";
    hintBtn.style.display = "none";
    clearHint();
}
```

77 Ολοκλήρωση παιχνιδιού και εμφάνιση κουμπιού επανάληψης

### 6.6.3 Συμπέρασμα Γεωγραφίας

Τα παιχνίδια γεωγραφίας συνδυάζουν χάρτες, εικόνες και δοκιμασίες πολλαπλής επιλογής με σκοπό να ενισχύσουν τη δεξιότητα αντιστοίχισης χωρών και πρωτευουσών. Ο κώδικας βασίζεται σε πίνακες δεδομένων, τυχαία σειρά επιλογών και άμεση ανατροφοδότηση, προσφέροντας ένα ευχάριστο και αποτελεσματικό μαθησιακό περιβάλλον για μαθητές πρωτοβάθμιας εκπαίδευσης.

## 6.7 Ρομπότ (robot\_game.html)

Το παιχνίδι με το ρομπότ αποτελεί εισαγωγή των μαθητών στις βασικές έννοιες της πληροφορικής, όπως η ακολουθία εντολών, η αλγοριθμική σκέψη και η επίλυση προβλημάτων. Ο μαθητής καλείται να οδηγήσει το ρομπότ από την αφετηρία μέχρι την έξοδο, επιλέγοντας σωστή σειρά κινήσεων.

Αποθηκεύει τις εντολές του μαθητή σε έναν πίνακα queue. Κάθε πάτημα σε κουμπί κατεύθυνσης προσθέτει μία τιμή. Η εισαγωγή εντολών απενεργοποιείται όταν το παιχνίδι έχει τελειώσει ή όταν τρέχει ήδη η εκτέλεση ή η λύση ώστε να μην αλλοιώνεται η ακολουθία ενώ εκτελείται.

```
const queue = [];
let gameOver = false;
let runningTimer = null;
let solving = false;
let reachedBySolution = false;
```

78 Αποθήκευση εντολών κίνησης σε πίνακα queue

Η εμφάνιση της ουράς γίνεται από τη renderQueue(). Δημιουργείται δυναμικά ένα grid κειμένων που εμφανίζει αριθμημένα βήματα. Αυτό γίνεται ώστε να βλέπει ο μαθητής την σειρά εκτέλεσης.

```
function renderQueue(listOverride) {
  const list = Array.isArray(listOverride) ? listOverride : queue;

  if(list.length === 0){
    queueBox.textContent = 'Καμία Εντολή';
    return;
  }
  const wrap = document.createElement('div');
  wrap.className = 'queue-grid';
  list.forEach((dir, i)=>{
    const item = document.createElement('div');
    item.className = 'q-item';
    item.textContent = (i + 1) + '. ' + (labelByDir[dir] || dir);
    wrap.appendChild(item);
  });
  queueBox.innerHTML = '';
  queueBox.appendChild(wrap);
}
```

### 79 Εμφάνιση και αρίθμηση εντολών με τη renderQueue()

Επιπλέον, επιτρέπει διόρθωση εντολών είτε με πλήρη διαγραφή είτε με αφαίρεση της τελευταίας εντολής. Έτσι, ο μαθητής μπορεί να διορθώσει ή να αλλάξει την διαδρομή του ρομπότ αν το επιθυμεί.

```
clearBtn.addEventListener('click', ()=>{
  if(gameOver || solving || runBtn.disabled) return;
  clearQueueToNone();
});

clearOneBtn.addEventListener('click', ()=>{
  if(gameOver || solving || runBtn.disabled) return;
  if(queue.length === 0) return;
  queue.pop();
  renderQueue();
});
```

### 80 Διόρθωση ουράς εντολών με διαγραφή ή αναίρεση τελευταίας κίνησης

Ο λαβύρινθος αναπαριστάται ως πίνακας 19x19. Κάθε κελί είναι είτε τοίχος είτε διάδρομος (τιμές 1 και 0). Η έξοδος και η εκκίνηση ορίζονται ως σταθερές θέσεις μέσα στο πλέγμα και το ρομπότ ξεκινά από την εκκίνηση.

Η θέση του ρομπότ ενημερώνεται οπτικά με τη setRobotInGrid(). Εκεί γίνεται μεταροπή από συντεταγμένες κελιού σε πραγματική θέση pixels πάνω στον λαβύρινθο. Αυτό επιτρέπει στο ρομπότ να εμφανίζεται πάντα σωστά σε ένα κελί.

```

const SIZE = 19;
let grid = [];
let start = { r: SIZE - 2, c: SIZE - 2 };
let exit = { r: 1, c: 1 };
let robotPos = { r: start.r, c: start.c };

function inBounds(r,c){ return r >= 0 && r < SIZE && c >= 0 && c < SIZE; }
function isWall(r,c){ return grid[r][c] == 1; }

function cellSizePx(){
  return mazeBox.clientWidth / SIZE;
}

function setRobotInGrid(){
  const s = cellSizePx();
  robotToken.style.width = s + 'px';
  robotToken.style.height = s + 'px';
  robotToken.style.left = Math.round(robotPos.c * s) + 'px';
  robotToken.style.top = Math.round(robotPos.r * s) + 'px';
}

```

#### 81 Αναπαράσταση λαβύρινθου 19×19 και ακριβής τοποθέτηση ρομπότ στο πλέγμα

Κάθε νέα παρτίδα δημιουργεί διαφορετικό λαβύρινθο μέσω της `generateMaze19()`. Η λογική είναι ότι αρχικοποιεί όλο το πλέγμα ως τοίχους, ανοίγει τα κελιά σε περιττές συντεταγμένες ώστε να υπάρχουν βασικοί κόμβοι διαδρόμου και χρησιμοποιεί στοίβα και πίνακα επισκέψεων ώστε να σκάψει διαδρόμους με βήμα δύο κελιών, αφαιρώντας τον ενδιάμεσο τοίχο. Το αποτέλεσμα είναι ένας λαβύρινθος με συνδεδεμένες διαδρομές.

```

function generateMaze19(){
  grid = Array.from({length: SIZE}, ()=> Array(SIZE).fill(1));

  for(let r=1;r<SIZE-1;r++){
    for(let c=1;c<SIZE-1;c++){
      if(r%2===1 && c%2===1) grid[r][c] = 0;
    }
  }

  const stack = [];
  const visited = Array.from({length: SIZE}, ()=> Array(SIZE).fill(false));

  const startCell = { r: 1, c: 1 };
  stack.push(startCell);
  visited[startCell.r][startCell.c] = true;

  const dirs = [
    { dr: 0, dc: 2 },
    { dr: 0, dc: -2 },
    { dr: 2, dc: 0 },
    { dr: -2, dc: 0 }
  ];

  while(stack.length){
    const cur = stack[stack.length - 1];
    const options = [];

    for(const d of dirs){
      const nr = cur.r + d.dr;
      const nc = cur.c + d.dc;
      if(nr <= 0 || nr >= SIZE-1 || nc <= 0 || nc >= SIZE-1) continue;
      if(!visited[nr][nc]){
        options.push({ nr, nc, wr: cur.r + d.dr/2, wc: cur.c + d.dc/2 });
      }
    }

    if(options.length === 0){
      stack.pop();
      continue;
    }

    const pick = shuffle(options)[0];
    grid[pick.wr][pick.wc] = 0;
    visited[pick.nr][pick.nc] = true;
    stack.push({ r: pick.nr, c: pick.nc });
  }

  exit = { r: 1, c: 1 };
  start = { r: SIZE - 2, c: SIZE - 2 };
  grid[exit.r][exit.c] = 0;
  grid[start.r][start.c] = 0;
}

```

## 82 Δημιουργία λαβύρινθου με αλγόριθμο σκαψίματος διαδρόμων (generateMaze19)

Η ensureNeighborOpen εγγυάται ότι η έξοδος και η εκκίνηση δεν θα είναι κλεισμένες από τοίχους. Η κίνηση ελέγχεται πρώτα από την isValidNextCell(). Υπολογίζει το επόμενο κελί με βάση την κατεύθυνση και απορρίπτει μία κίνηση αν οδηγεί εκτός πλέγματος ή πάνω σε τοίχο. Επίσης ελέγχει (και επιστρέφει την πληροφορία) αν το κελί είναι η έξοδος.

Έπειτα, η stepMove() εφαρμόζει την κίνηση. Ενημερώνει τη θέση, μετακινεί οπτικά το ρομπότ και αν έφτασε στην έξοδο καλεί έλεγχο νίκης. Αν η κίνηση είναι άκυρη, εμφανίζει μήνυμα λάθους και επιστρέφει false.

```

const ensureNeighborOpen = (r,c)=>{
  const neigh = [
    {r:r-1,c},
    {r:r+1,c},
    {r,c:c-1},
    {r,c:c+1}
  ].filter(p=> inBounds(p.r,p.c));
  if(neigh.some(p=> !isWall(p.r,p.c))) return;
  const p = neigh[Math.floor(Math.random()*neigh.length)];
  grid[p.r][p.c] = 0;
};

ensureNeighborOpen(exit.r, exit.c);
ensureNeighborOpen(start.r, start.c);
}

window.addEventListener('resize', ()=> setRobotInGrid());

function isValidNextCell(r,c,dir){
  let nr = r, nc = c;
  if(dir === 'up') nr -= 1;
  if(dir === 'down') nr += 1;
  if(dir === 'left') nc -= 1;
  if(dir === 'right') nc += 1;

  if(!inBounds(nr,nc)) return { ok:false, r, c, isExit:false };
  if(isWall(nr,nc)) return { ok:false, r, c, isExit:false };

  const isExit = (nr === exit.r && nc === exit.c);
  return { ok:true, r:nr, c:nc, isExit };
}

function stepMove(dir){
  if(gameOver) return false;

  const v = isValidNextCell(robotPos.r, robotPos.c, dir);
  if(!v.ok){ showError(); return false; }

  robotPos.r = v.r;
  robotPos.c = v.c;
  setRobotInGrid();

  if(v.isExit) checkWin();
  return true;
}

```

### 83 Έλεγχος εγκυρότητας κίνησης και εκτέλεση βήματος με isValidNextCell() και stepMove()

Όταν ο μαθητής πατήσει το κουμπί «Τρέξε» το παιχνίδι κλειδώνει τα κουμπιά και εκτελεί τις κινήσεις με setInterval ανά MOVE\_DELAY ms. Σε κάθε βήμα τονίζει χρωματικά την εντολή που εκτελείται, εμφανίζει την τρέχουσα εντολή στο label εκτέλεσης και επιχειρεί να μετακινήσει το ρομπότ. Αν έχει προγραμματιστεί λάθος κίνηση, τότε η εκτέλεση σταματά και η ουρά καθαρίζεται ώστε ο μαθητής να ξαναπροσπαθήσει.

```

runBtn.addEventListener('click', ()=>{
  if(queue.length === 0 || runBtn.disabled || gameOver || solving) return;

  reachedBySolution = false;
  hideStopBtn();

  const moves = [...queue];

  lockUIForRun();
  hideExec();
  setActiveQueueIndex(null);

  let i = 0;
  runningTimer = setInterval(()=>{
    if(gameOver){
      clearInterval(runningTimer);
      runningTimer = null;
      hideExec();
      setActiveQueueIndex(null);
      return;
    }

    setActiveQueueIndex(i);
    showExec(moves[i]);

    const ok = stepMove(moves[i]);
    if(!ok){
      clearInterval(runningTimer);
      runningTimer = null;
      hideExec();
      setActiveQueueIndex(null);
      forceUnlockUI();
      clearQueueToNone();
      return;
    }

    i++;
    if(i >= moves.length){
      clearInterval(runningTimer);
      runningTimer = null;
      hideExec();
      setActiveQueueIndex(null);
      forceUnlockUI();
      clearQueueToNone();
    }
  }, MOVE_DELAY);
});

```

#### 84 Εκτέλεση εντολών με setInterval και επισήμανση τρέχουσας κίνησης

Η checkWin() ενεργοποιείται όταν το ρομπότ φτάσει στο κελί της εξόδου. Τότε δηλώνει την λήξη του παιχνιδιού, κρύβει τα κουμπιά, εμφανίζει το κουμπί «Ξαναπαίξε» και δείχνει μήνυμα επιτυχίας εκτός αν ο τερματισμός έγινε με την χρήση του κουμπιού της λύσης. Τότε εμφανίζει μήνυμα «Δοκίμασε Ξανά»

```

function checkWin(){
  if(gameOver) return;
  if(robotPos.r === exit.r && robotPos.c === exit.c){
    gameOver = true;

    setActiveQueueIndex(null);

    if(reachedBySolution){
      showTryAgain();
    }else{
      winMsg.classList.add('show');
      setTimeout(()=> winMsg.classList.remove('show'), 5000);
    }

    const controlsBar = document.querySelector('.controls');
    if (controlsBar) controlsBar.style.display = 'none';

    runBtn.style.display = 'none';
    stopSolveBtn.style.display = 'none';
    clearBtn.style.display = 'none';
    clearOneBtn.style.display = 'none';
    queueBox.style.display = 'none';
    hideExec();
    clearQueueToNone();
    controlButtons.forEach(b=> b.disabled = true);
    runBtn.disabled = true;
    hintBtn.style.display = 'none';
    replayTopBtn.style.display = 'inline-block';

    if (runningTimer) { clearInterval(runningTimer); runningTimer = null; }
    solving = false;
    hideStopBtn();
  }
}

```

#### 85 Έλεγχος νίκης ρομπότ και εμφάνιση κατάλληλου μηνύματος τερματισμού (checkWin)

Το κουμπί της λύσης δεν δίνει έτοιμο μονοπάτι. Αντίθετα, υπολογίζει δυναμικά τη διαδρομή από την τρέχουσα θέση προς την έξοδο με αναζήτηση κατά πλάτος (BFS). Αυτό γίνεται γιατί σε πλέγμα με ίσο κόστος βήματος βρίσκει τη συντομότερη διαδρομή.

Η bfsPath() κρατά τα κελιά από τα οποία έχει ήδη περάσει το ρομπότ (visited) για να μην ξαναπερνά από τα ίδια και parent για να ανακατασκευάσει στο τέλος την διαδρομή που οδήγησε στον στόχο.

```

function bfsPath(startR, startC, goalR, goalC){
  const q = [{ r:startR, c:startC }];
  const visited = Array.from({length: SIZE}, ()=> Array(SIZE).fill(false));
  const parent = new Map();
  visited[startR][startC] = true;

  const dirs = [
    { d:'up', dr:-1, dc:0 },
    { d:'down', dr:1, dc:0 },
    { d:'left', dr:0, dc:-1 },
    { d:'right', dr:0, dc:1 }
  ];

  while(q.length){
    const cur = q.shift();
    if(cur.r === goalR && cur.c === goalC) break;

    for(const x of dirs){
      const nr = cur.r + x.dr;
      const nc = cur.c + x.dc;
      if(!inBounds(nr,nc)) continue;
      if(visited[nr][nc]) continue;
      if(isWall(nr,nc)) continue;
      visited[nr][nc] = true;
      const key = `${nr},${nc}`;
      parent.set(key, { prev: `${cur.r},${cur.c}`, dir: x.d });
      q.push({ r:nr, c:nc });
    }
  }

  const goalKey = `${goalR},${goalC}`;
  if(!parent.has(goalKey) && !(startR === goalR && startC === goalC)) return null;

  const path = [];
  if(startR === goalR && startC === goalC) return path;

  let k = goalKey;
  while(k !== `${startR},${startC}`){
    const p = parent.get(k);
    if(!p) return null;
    path.push(p.dir);
    k = p.prev;
  }
  path.reverse();
  return path;
}

```

## 86 Υπολογισμός συντομότερης λύσης με BFS και ανακατασκευή διαδρομής (bfsPath)

Η `runSolveToEnd()` καλεί το BFS, γεμίζει την ουρά με το αποτέλεσμα και εκκινεί εκτέλεση με την ίδια λογική χρονισμού. Εδώ τίθεται `reachedBySolution=true`, ώστε όταν φτάσει στην έξοδο να εμφανιστεί «Δοκίμασε Ξανά» αντί για «Συγχαρητήρια».

Υπάρχει επίσης κουμπί «Σταμάτα» που διακόπτει την αυτόματη λύση το οποίο σταματά το timer, ξεκλειδώνει το UI και καθαρίζει την ουρά.

```

function runSolveToEnd() {
  if(gameOver || solving) return;

  if (runningTimer) { clearInterval(runningTimer); runningTimer = null; }

  reachedBySolution = true;
  solving = true;

  const moves = bfsPath(robotPos.r, robotPos.c, exit.r, exit.c);
  if(!moves || moves.length === 0){
    solving = false;
    hideStopBtn();
    forceUnlockUI();
    reachedBySolution = false;
    setActiveQueueIndex(null);
    return;
  }

  queue.length = 0;
  queue.push(...moves);
  renderQueue(queue);

  runMovesSequence(moves);
}

hintBtn.addEventListener('click', ()=>{
  runSolveToEnd();
});

stopSolveBtn.addEventListener('click', ()=>{
  if(!solving) return;
  if (runningTimer) { clearInterval(runningTimer); runningTimer = null; }
  solving = false;
  reachedBySolution = false;
  hideExec();
  hideStopBtn();
  setActiveQueueIndex(null);
  forceUnlockUI();
  clearQueueToNone();
});

```

### 87 Αυτόματη εκτέλεση λύσης με BFS, έλεγχος τερματισμού και διακοπή (runSolveToEnd)

Η restartGame() επαναφέρει πλήρως την κατάσταση. Σταματά τα timers, μηδενίζει τα flags, δημιουργεί νέο λαβύρινθο, επιστρέφει το ρομπότ στην εκκίνηση, καθαρίζει την ουρά και επαναφέρει τα κουμπιά. Με αυτόν τον τρόπο κάθε νέα προσπάθεια είναι καινούρια και διαφορετική. Μόλις φορτώνεται η σελίδα, το παιχνίδι θέτει αρχική κατάσταση.

```

function restartGame(){
  if (runningTimer) { clearInterval(runningTimer); runningTimer = null; }

  gameOver = false;
  solving = false;
  reachedBySolution = false;
  winMsg.classList.remove('show');
  tryAgainMsg.classList.remove('show');
  errorMsg.classList.remove('show');
  hideExec();
  hideStopBtn();
  setActiveQueueIndex(null);
  generateMaze19();
  robotPos = { r: start.r, c: start.c };
  renderMaze();

  const controlsBar = document.querySelector('.controls');
  if (controlsBar) controlsBar.style.display = 'flex';

  queueBox.style.display = 'block';
  clearQueueToNone();
  runBtn.style.display = 'inline-block';
  clearBtn.style.display = 'inline-block';
  clearOneBtn.style.display = 'inline-block';
  stopSolveBtn.style.display = 'inline-block';
  runBtn.disabled = false;
  controlButtons.forEach(b=> b.disabled = false);
  hintBtn.style.display = 'inline-block';
  replayTopBtn.style.display = 'none';
}

replayTopBtn.addEventListener('click', restartGame);
clearQueueToNone();
generateMaze19();
robotPos = { r: start.r, c: start.c };
renderMaze();

```

88 Πλήρης επανεκκίνηση παιχνιδιού και αρχικοποίηση νέας παρτίδας (restartGame)

### 6.7.1 Συμπέρασμα Ρομπότ

Το παιχνίδι με το ρομπότ επιτυγχάνει ο μαθητής να μετατρέψει την κίνηση μέσα σε έναν λαβύρινθο σε διαδικασία προγραμματισμού. Αυτό έχει ως αποτέλεσμα να σκέφτεται, να σχεδιάζει και να δοκιμάζει ακολουθίες εντολών. Η επιλογή να εκτελούνται κινήσεις διαδοχικά βοηθά τον μαθητή να συνδέσει άμεσα κάθε εντολή με το αποτέλεσμα που προκαλεί, να εντοπίζει το σημείο στο οποίο γίνεται το λάθος και να κατανοεί την έννοια της βηματικής εκτέλεσης αλγορίθμου.

Παράλληλα δίνει σαφή ανατροφοδότηση και ενισχύει την λογική της επίλυσης ενός προβλήματος. Ο μαθητής δεν χάνει, αλλά προσαρμόζει την στρατηγική του τροποποιώντας την σειρά εντολών. Η δυνατότητα της λύσης λειτουργεί υποστηρικτικά καθώς δείχνει την σωστή διαδρομή χωρίς όμως να αφαιρεί την ανάγκη κατανόησης.

Έτσι το παιχνίδι ισορροπεί ανάμεσα στην διερεύνηση και στην καθοδήγηση ενώ η επανεκκίνηση με νέο λαβύρινθο διατηρεί το ενδιαφέρον και προσφέρει πολλαπλές ευκαιρίες εξάσκησης.

## 7. Συμπεράσματα

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε μία εκπαιδευτική εφαρμογή πρωτοβάθμιας εκπαίδευσης με τη μορφή διαδραστικών παιχνιδιών. Στόχος της εφαρμογής ήταν η ενίσχυση της μαθησιακής διαδικασίας μέσα από το παιχνίδι, αξιοποιώντας σύγχρονες τεχνολογίες διαδικτύου και φιλικές διεπαφές κατάλληλες για παιδιά μικρής ηλικίας.

Η εφαρμογή καλύπτει διαφορετικά γνωστικά αντικείμενα όπως Μαθηματικά, Γλώσσα, Μελέτη Περιβάλλοντος, Ιστορία, Γεωγραφία και Πληροφορική. Κάθε μάθημα υλοποιήθηκε με ξεχωριστά παιχνίδια, τα οποία ακολουθούν κοινή λογική πλοήγησης και αλληλεπίδρασης, ώστε οι μαθητές να εξοικειώνονται εύκολα με το περιβάλλον και να μπορούν να μεταβαίνουν από το ένα παιχνίδι στο άλλο χωρίς δυσκολία.

Κατά την ανάπτυξη της εφαρμογής δόθηκε ιδιαίτερη έμφαση στη χρηστικότητα και στην παιδαγωγική προσέγγιση. Τα παιχνίδια παρέχουν άμεση ανατροφοδότηση σωστού ή λάθους, δυνατότητα βοήθειας μέσω του κουμπιού «Λύση» και επανάληψη χωρίς τιμωρητικό χαρακτήρα. Με αυτόν τον τρόπο, το λάθος αντιμετωπίζεται ως μέρος της μαθησιακής διαδικασίας και ενισχύεται η αυτοπεποίθηση των μαθητών.

Από τεχνικής πλευράς η εφαρμογή υλοποιήθηκε με χρήση HTML, CSS και JavaScript χωρίς την ανάγκη εγκατάστασης ή εξωτερικών βιβλιοθηκών. Ο κώδικας είναι οργανωμένος σε ξεχωριστά αρχεία ανά παιχνίδι, γεγονός που διευκολύνει την κατανόηση, τη συντήρηση και την πιθανή μελλοντική επέκταση της εφαρμογής. Η χρήση απλών δομών δεδομένων, σαφών κανόνων και καθαρής ροής ελέγχου συνέβαλε στη δημιουργία ενός λειτουργικού και σταθερού συστήματος.

Συνολικά, η ικανοποιεί επιτυχάνει τον σκοπό της προσφέροντας ένα ευχάριστο και αποτελεσματικό μαθησιακό περιβάλλον για μαθητές πρωτοβάθμιας εκπαίδευσης. Η εργασία αυτή δείχνει ότι η μάθηση μέσω παιχνιδιού μπορεί να αποτελέσει ένα ισχυρό εργαλείο υποστήριξης της εκπαίδευσης διαδικασίας, συνδυάζοντας τη γνώση με τη διασκέδαση και αξιοποιώντας τις δυνατότητες της σύγχρονης τεχνολογίας.

## 8. ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η παρούσα εκπαιδευτική εφαρμογή αποτελεί μία πλούσια βάση, η οποία όμως μπορεί να επεκταθεί και να βελτιωθεί στο μέλλον. Οι επεκτάσεις αυτές μπορούν να αφορούν τόσο το εκπαιδευτικό περιεχόμενο όσο και τις τεχνικές δυνατότητες της εφαρμογής.

Μία σημαντική μελλοντική επέκταση θα μπορούσε να είναι η προσθήκη νέων εκπαιδευτικών παιχνιδιών και δραστηριοτήτων. Η εφαρμογή μπορεί να εμπλουτιστεί με επιπλέον ασκήσεις για κάθε μάθημα, καθώς και με νέα γνωστικά αντικείμενα, όπως η Μουσική ή η Φυσική, ώστε να καλύπτει ακόμη περισσότερες ανάγκες της πρωτοβάθμιας εκπαίδευσης.

Επιπλέον, θα μπορούσε να προστεθεί σύστημα επιπέδων δυσκολίας. Με αυτόν τον τρόπο, το ίδιο παιχνίδι θα προσαρμόζεται στο επίπεδο του μαθητή, προσφέροντας πιο απλές ή πιο προχωρημένες δραστηριότητες. Η δυνατότητα αυτή θα ενίσχυε την εξατομικευμένη μάθηση και θα επέτρεπε την καλύτερη προσαρμογή στον ρυθμό κάθε παιδιού.

Μία ακόμη πιθανή επέκταση αφορά την αποθήκευση προόδου των μαθητών. Μέσω χρήσης βάσης δεδομένων ή τοπικής αποθήκευσης, θα μπορούσε να καταγράφεται η πορεία κάθε μαθητή, οι επιτυχίες του και οι δραστηριότητες που έχει ολοκληρώσει. Αυτό θα επέτρεπε τόσο στους εκπαιδευτικούς όσο και στους γονείς να παρακολουθούν την εξέλιξη της μάθησης.

Στον τομέα της διεπαφής, θα μπορούσε να προστεθεί υποστήριξη πολλαπλών γλωσσών. Η δυνατότητα αλλαγής γλώσσας θα καθιστούσε την εφαρμογή προσβάσιμη σε μεγαλύτερο αριθμό χρηστών και θα επέτρεπε τη χρήση της και σε μη ελληνόφωνο περιβάλλον.

Τέλος, μία μελλοντική βελτίωση θα μπορούσε να είναι η μετατροπή της εφαρμογής σε διαδικτυακή πλατφόρμα ή κινητές συσκευές. Με αυτόν τον τρόπο, οι μαθητές θα μπορούσαν να έχουν πρόσβαση στα εκπαιδευτικά παιχνίδια από οποιαδήποτε συσκευή, ενισχύοντας ακόμη περισσότερο τη χρήση της εφαρμογής στην καθημερινή εκπαιδευτική πρακτική.

Η εφαρμογή διαθέτει ισχυρές βάσεις και προσφέρει πολλές δυνατότητες εξέλιξης. Οι μελλοντικές επεκτάσεις μπορούν να συμβάλουν στη δημιουργία ενός ακόμη πιο ολοκληρωμένου και ευέλικτου εκπαιδευτικού εργαλείου, προσαρμοσμένου στις ανάγκες της σύγχρονης εκπαίδευσης.

## Βιβλιογραφία

- Nielsen, J. (1994). Usability engineering. *Academic Press*.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: Strategies for effective human-computer interaction. *Pearson Education*.
- Hourcade, J. P. (2008). Interaction design and children. *Foundations and Trends in Human-Computer Interaction*, 1(4), 277–392.
- Plowman, L., McPake, J., & Stephen, C. (2010). The technologisation of childhood? Young children and technology in the home. *Children & Society*, 24(1), 63–74.
- Ke, F. (2008). A case study of computer gaming for math: Engaged learning from gameplay? *Computers & Education*, 51(4), 1609–1620.
- Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & Gaming*, 33(4), 441–467.
- Li, Q., Ma, X., & Zheng, C. (2016). The impact of game-based learning on mathematics achievement and motivation: A literature review. *International Journal of Learning, Teaching and Educational Research*, 15(3), 1–15.
- Panoutsopoulos, H., & Sampson, D. G. (2012). A study on exploiting digital games into school mathematics education. *Educational Technology & Society*, 15(1), 15–27.
- Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting education in “educational” apps: Lessons from the science of learning. *Psychological Science in the Public Interest*, 16(1), 3–34.
- Neumann, M. M. (2018). Using tablets and apps to enhance emergent literacy skills in young children. *Early Childhood Research Quarterly*, 42, 239–246.
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2018). Educational apps from the Android Google Play for Greek preschoolers: A systematic review. *Computers & Education*, 116, 139–160.
- Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), 20–20.
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2017). Using educational games in environmental education: A case study with primary school students. *Education and Information Technologies*, 22(5), 2185–2201.
- Hadzigeorgiou, Y., & Skoumios, M. (2013). The development of environmental awareness through school activities: A case study. *International Journal of Environmental & Science Education*, 8(3), 405–425.
- Wu, Y. T., & Huang, Y. M. (2015). The effectiveness of digital game-based learning in environmental education. *Computers & Education*, 87, 239–251.
- Cheng, Y. M., Lou, S. J., Kuo, S. H., & Shih, R. C. (2013). Investigating elementary school students’ technology acceptance by applying digital game-based learning to environmental education. *Australasian Journal of Educational Technology*, 29(1), 96–110.
- Squire, K., & Jenkins, H. (2003). Harnessing the power of games in education. *Insight*, 3(1), 5–33.

- McCall, J. (2016). Teaching history with digital historical games: An introduction to the field and best practices. *Simulation & Gaming, 47*(4), 517–542.
- Fokides, E. (2017). Using digital educational games in history education: Results from a classroom intervention. *Journal of Educational Technology & Society, 20*(1), 263–274.
- Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE Review, 41*(2), 16–30.
- Hattie, J., & Yates, G. (2014). Visible learning and the science of how we learn. *Educational Psychology Review, 26*(2), 137–161.
- Tsekleves, E., Cosmas, J., & Aggoun, A. (2016). Benefits, barriers and guideline recommendations for the implementation of serious games in education for stakeholders and policymakers. *British Journal of Educational Technology, 47*(1), 164–183.
- Hong, J. C., Cheng, C. L., Hwang, M. Y., Lee, C. K., & Chang, H. Y. (2009). Assessing the educational values of digital games. *Computers & Education, 52*(2), 423–437.
- Sinton, D. S., Bednarz, S. W., Gersmehl, P. J., Kolvoord, R. A., & Uttal, D. H. (2013). The people, the tools, and the strategies: GIS in education. *Journal of Geography, 112*(2), 76–85.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas. Basic Books.*
- Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom. Teachers College Press.*
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67.
- Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). Learning programming at the computational thinking level via digital games. *Procedia Computer Science, 9*, 522–531.