

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΑΝΑΠΤΥΞΗ ΠΑΙΧΝΙΔΙΟΥ ΜΕ ΤΕΧΝΟΛΟΓΙΑ HTML5 CANVAS

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΤΗΣ
ΣΤΑΜΑΤΙΚΗΣ ΚΑΠΕΛΕΡΗ
ΑΜ 1700



Επιβλέπων Καθηγητής
ΧΡΙΣΤΟΔΟΥΛΟΥ ΣΩΤΗΡΙΟΣ

ΦΕΒΡΟΥΑΡΙΟΣ, 2017

ΑΝΤΙΡΙΟ

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ον/μο: Υπογραφή

2. Ον/μο: Υπογραφή

3. Ον/μο: Υπογραφή

Πνευματικά δικαιώματα

Copyright © Stamatiki Kapeleri, 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής του

Τεχνολογικού Εκπαιδευτικού Ιδρύματος Δυτικής Ελλάδας δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω ιδιαίτερα τους γονείς μου που παρ' όλες τις οικονομικές δυσκολίες φρόντισαν ανιδιοτελώς για τις σπουδές μου και τον καθηγητή μου Σωτήριο Χριστοδούλου για την στοχευμένη καθοδήγηση και την αξιέπαινη διδασκαλία καθ' όλη τη διάρκεια των σπουδών μου.

Πίνακας περιεχομένων

ΕΥΧΑΡΙΣΤΙΕΣ	3
1. ΠΡΟΛΟΓΟΣ.....	5
2. ΣΚΟΠΟΣ	5
3. ΠΑΡΟΥΣΙΑΣΗ ΠΑΙΧΝΙΔΙΟΥ	6
3.1 ΚΑΝΟΝΕΣ.....	6
4. ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΙΚΩΝ ΟΘΟΝΩΝ ΚΑΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕ ΤΟ ΧΡΗΣΤΗ	7
4.1 ΜΕΝΟΥ ΠΑΙΧΝΙΔΙΟΥ	7
4.2.1 ΕΠΙΠΕΔΑ	10
5. ΜΟΝΤΕΛΟ ΔΕΔΟΜΕΝΩΝ.....	13
6. ΥΛΟΠΟΙΗΣΗ	17
6.1 ΑΡΧΙΚΟ ΜΕΝΟΥ	17
6.2 ΟΘΟΝΗ ΠΑΙΧΝΙΔΙΟΥ	18
6.3 ΠΡΟΣΩΡΙΝΗ ΑΠΟΘΗΚΕΥΣΗ ΣΤΟΙΧΕΙΩΝ ΣΤΟ BROWSER	25
ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ	31
ΒΙΒΛΙΟΓΡΑΦΙΑ	32

1. Πρόλογος

Τη σημερινή εποχή οι υπολογιστές και το ίντερνετ έχουν εισχωρήσει για τα καλά στη ζωή των περισσότερων με πολλαπλές αυτοματοποιήσεις στην καθημερινότητα μας. Οι αυτοματοποιήσεις αυτές, όμως, μπορεί να έχουν αρνητικές επιπτώσεις στη συγκέντρωση προσοχής, στην απομνημόνευση, στην ανάκληση πληροφοριών αλλά και γενικότερα στην απόδοση της μνήμης σε κάθε άνθρωπο. Η συνεχής και υπερβολική χρήση των έξυπνων κινητών και υπολογιστών δεν μας επιτρέπει την αναγκαία εξάσκηση του μυαλού και κατ' επέκταση την θετική εξέλιξή του. Ένα παράδειγμα το οποίο σχετίζεται με τέτοιες επιπτώσεις είναι η νόσος Αλτσχάιμερ που εμφανίζεται όλο και συχνότερα. Φυσικά υπάρχουν διάφοροι τρόποι πρόληψης για τέτοιες περιπτώσεις αλλά ένας βασικός είναι και η ενασχόληση με παιχνίδια μνήμης και αντίληψης. Έτσι λοιπόν, στα πλαίσια της πτυχιακής μου εργασίας επέλεξα να ασχοληθώ με τη δημιουργία ενός τέτοιου παιχνιδιού για smartphones εμπνεόμενη από την αυξανόμενη χρήση τους και τις επιπτώσεις που έχουν στην μνήμη του ανθρώπου.

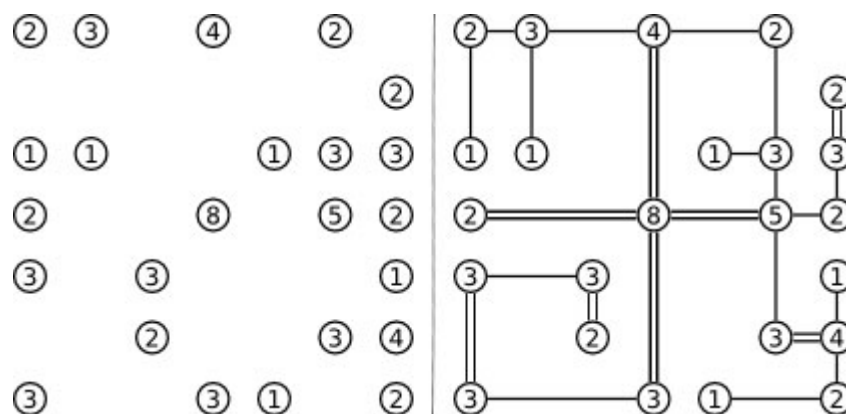
2. Σκοπός

Η εργασία αυτή στοχεύει στην αξιοποίηση των δυνατοτήτων της HTML5 με σκοπό την κατασκευή και το σχηματισμό παιχνιδιού με τεχνολογία *canvas*.

Επιπρόσθετα, για την ανάπτυξη και τη μορφοποίηση του παιχνιδιού έγινε χρήση Javascript και CSS3.

3. ΠΑΡΟΥΣΙΑΣΗ ΠΑΙΧΝΙΔΙΟΥ

Το εν λόγω παιχνίδι ονομάζεται στην ελληνική γλώσσα *ΓΕΦΥΡΕΣ* και ανήκει στην κατηγορία παιχνιδιών: σταυρόλεξα, κρυπτόλεξα, βρες τις διαφορές, sudoku, δηλαδή είναι κι αυτό ένα παιχνίδι αντίληψης. Συναντάται κυρίως σε περιοδικά με σταυρόλεξα αλλά και στο ίντερνετ. Γενικά, παρουσιάζεται ως ένα πλαίσιο στο οποίο εμπεριέχονται κύκλοι με αριθμούς και σκοπός του παίχτη είναι να συνδέσει όλους τους κύκλους σωστά με βάση τους κανόνες.



Εικόνα 1. Bridges example

3.1 ΚΑΝΟΝΕΣ

Κάθε κύκλος στο πλέγμα αντιπροσωπεύει ένα νησί. Οι γέφυρες μπορούν να συνδέουν ένα ζευγάρι νησιών μόνο οριζόντια ή κατακόρυφα έτσι ώστε όλα τα νησιά να σχηματίζουν μία συνδεδεμένη ενιαία ομάδα. Οι γέφυρες δεν μπορούν να διασταυρώνονται με τον εαυτό τους ή μεταξύ τους. Ο αριθμός μέσα σε κάθε κύκλο αντιπροσωπεύει τον αριθμό των γεφυρών που χρειάζεται να συνδεθούν με αυτόν. Το πολύ δύο γέφυρες μπορούν να συνδέουν ένα ζευγάρι νησιών.

4. ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΙΚΩΝ ΟΘΟΝΩΝ ΚΑΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕ ΤΟ ΧΡΗΣΤΗ

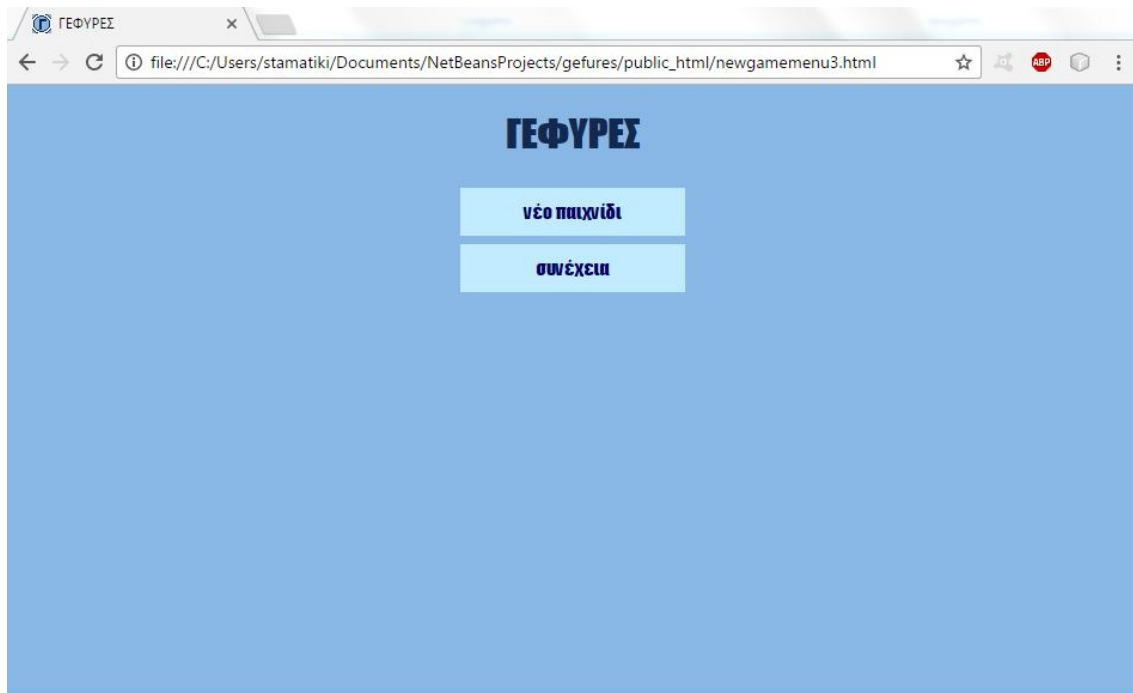
4.1 ΜΕΝΟΥ ΠΑΙΧΝΙΔΙΟΥ

Η πρώτη εικόνα που έχει ο χρήστης είναι ένα αρχικό μενού με τις εξής επιλογές: *Επίπεδο 1*, *Επίπεδο 2*, *Επίπεδο 3*, οι οποίες αφορούν το βαθμό δυσκολίας της κάθε ομάδας γύρων παιχνιδιού.

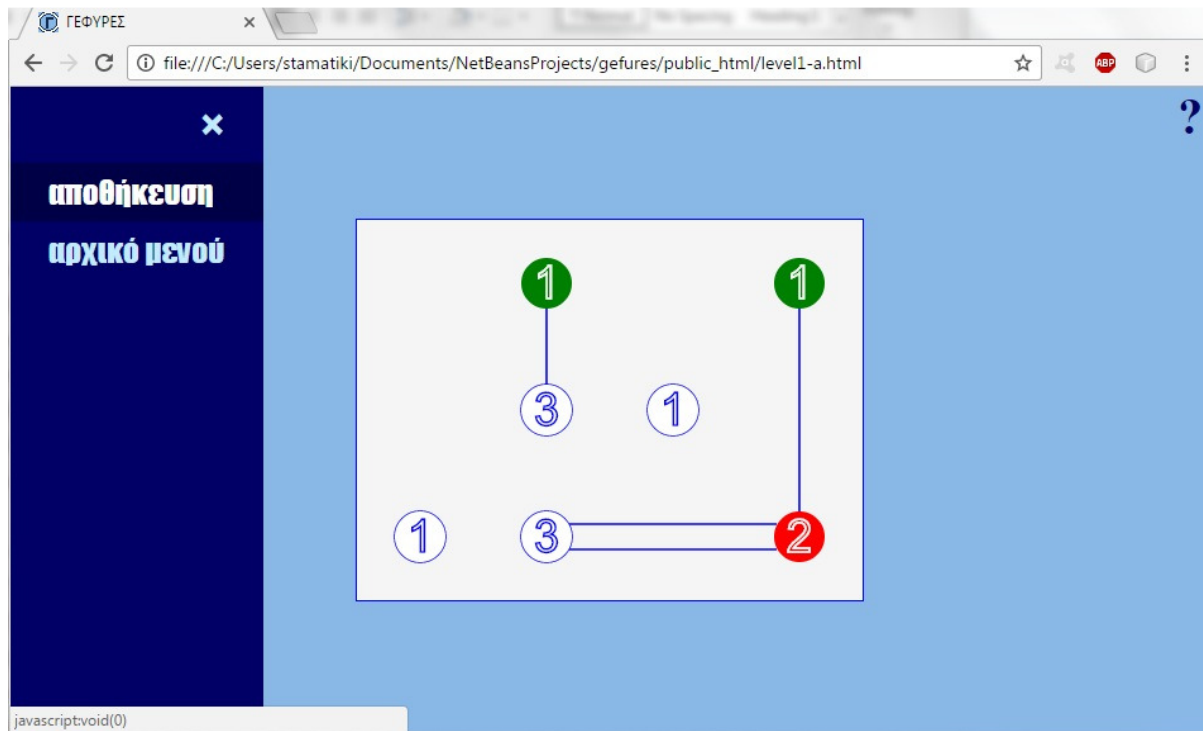


Εικόνα 2. Αρχικό μενού

Αφού γίνει η επιθυμητή επιλογή επιπέδου εμφανίζεται η οθόνη στην οποία καλείται ο χρήστης να ολοκληρώσει τον πρώτο γύρο του αντίστοιχου επιπέδου. Στην περίπτωση που ο παίχτης έχει ήδη ξεκινήσει κάποιο γύρο και έχει αποθηκεύσει τις μέχρι τώρα κινήσεις μετά το αρχικό μενού εμφανίζεται ένα δευτερεύον μενού επιλογών: *νέο παιχνίδι*, *συνέχεια* όπου μας ερωτάται αν θέλουμε να ξεκινήσουμε από την αρχή τον πρώτο γύρο ή να συνεχίσουμε το παιχνίδι στο οποίο έγινε αποθήκευση των κινήσεων.



Εικόνα 3. Δευτερεύον μενού

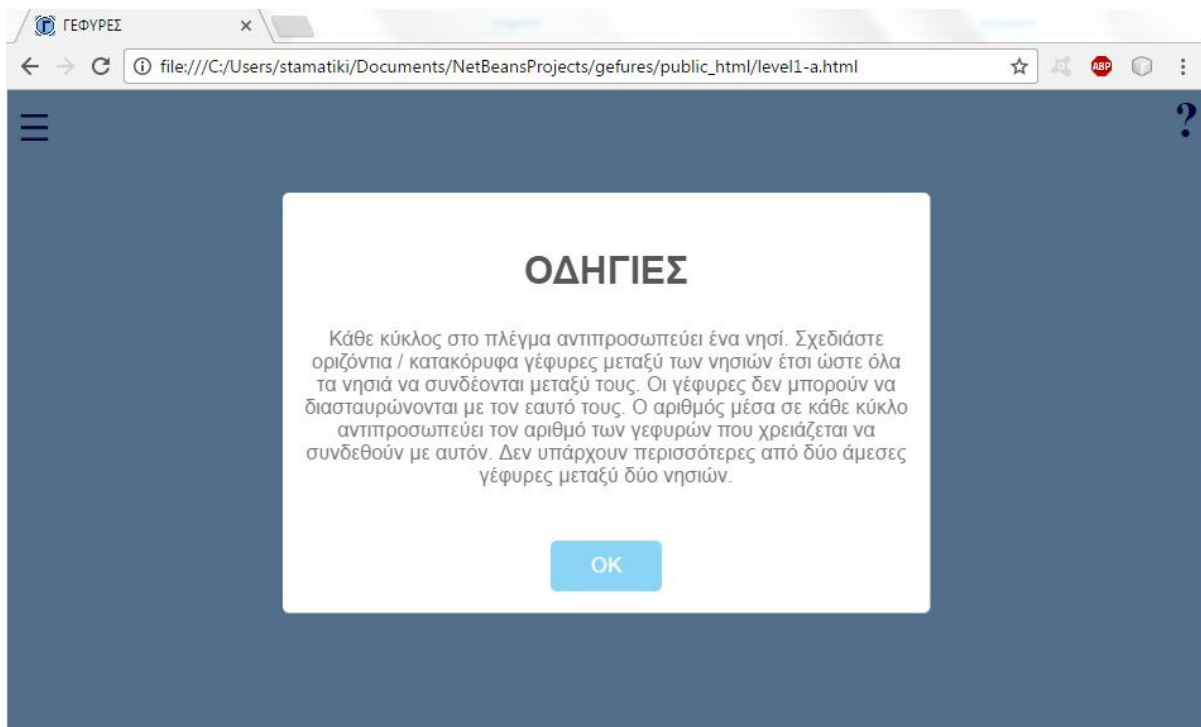


Εικόνα 4. Πλαινό μενού στα αριστερά της οθόνης

Η δυνατότητα της αποθήκευσης υπάρχει στο πλαινό μενού που βρίσκεται πάνω αριστερά στην οθόνη. Στο πλαινό αυτό μενού υπάρχει και η επιλογή *αρχικό μενού* που μας μεταφέρει

στο μενού επιλογής επιπέδου. Το μενού αυτό εμφανίζεται όταν γίνει κλικ στο εικονίδιο που βρίσκεται πάνω αριστερά στην οθόνη.

Επίσης, ένα εικονίδιο ερωτηματικού βρίσκεται πάνω δεξιά στην οθόνη στο οποίο όταν γίνει κλικ μας εμφανίζεται ένα μήνυμα κειμένου το οποίο μας εξηγεί τους κανόνες και μας δίνει τις οδηγίες του παιχνιδιού.

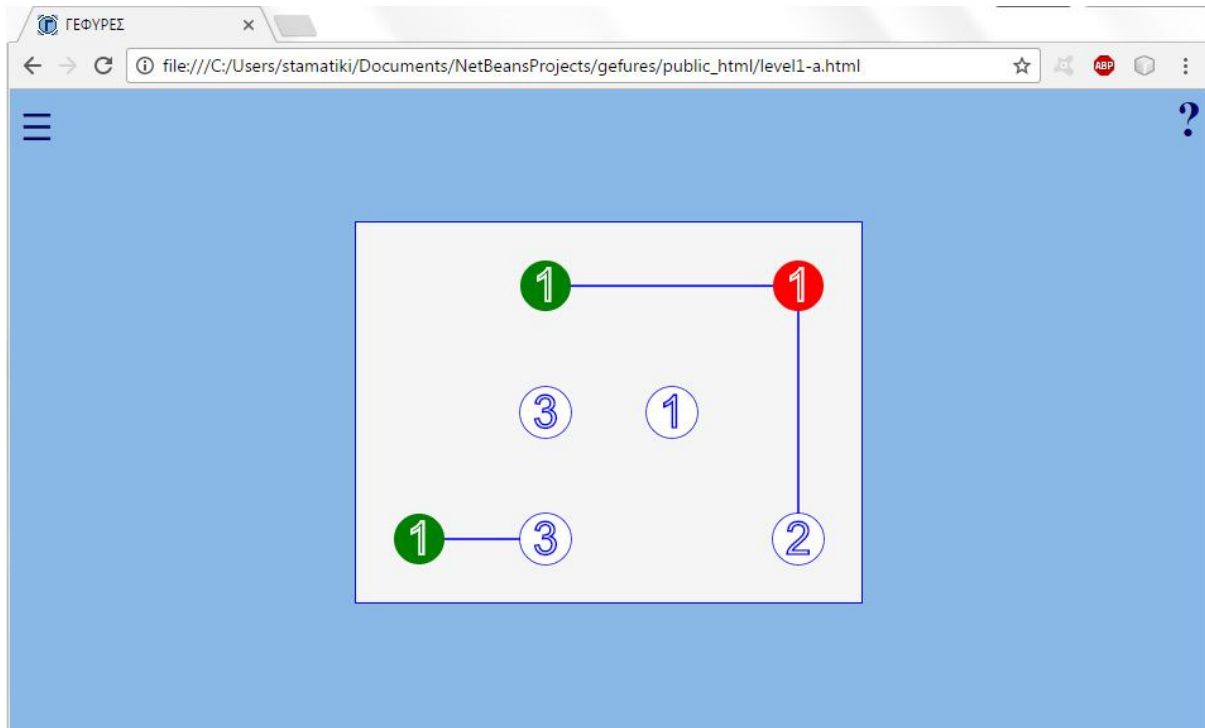


Εικόνα 5. Κανόνες παιχνιδιού

4.2.1 ΕΠΙΠΕΔΑ

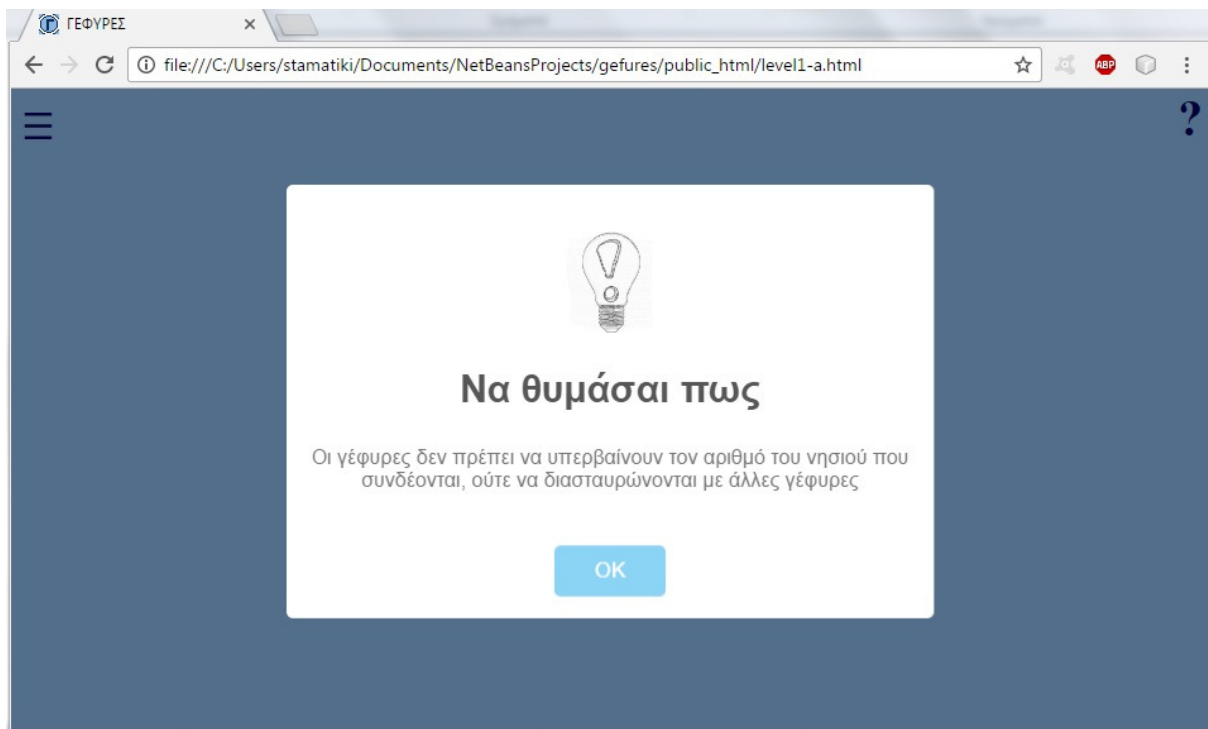
Υπάρχουν τρία (3) επίπεδα δυσκολίας.

Στο πρώτο επίπεδο το πλαίσιο του παιχνιδιού έχει διάσταση 3x4 και γίνονται κάποιες βοηθητικές ενέργειες υπέρ του παίκτη, έτσι ώστε να συνδέσει σωστά τα νησιά και να ολοκληρώσει το Επίπεδο 1 επιτυχώς.

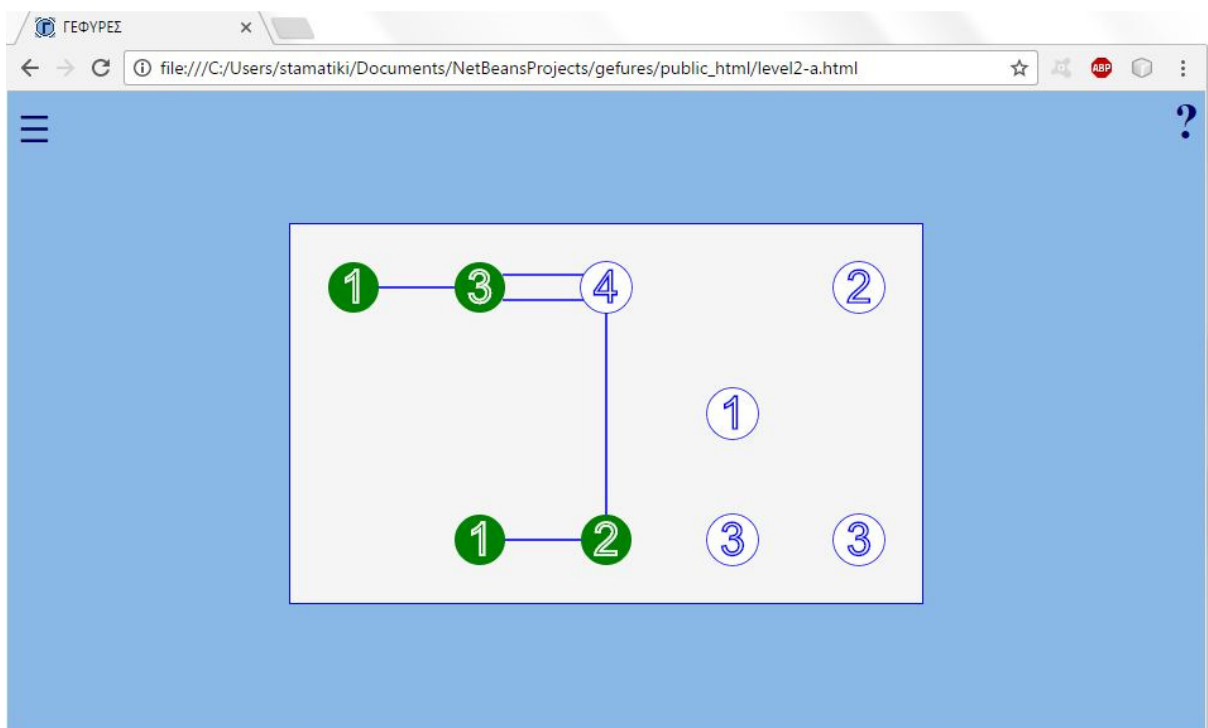


Εικόνα 6. Επίπεδο 1

Πιο αναλυτικά, οι βοηθητικές αυτές ενέργειες είναι δύο: χρωματισμός των νησιών και εμφάνιση υπενθύμισης κάποιων βασικών κανόνων με τη μορφή pop-up μηνύματος. Με το χρωματισμό των νησιών δείχνουμε στον παίκτη σε ποιο νησί έχουν συνδεθεί παραπάνω γέφυρες ή ίσες σε πλήθος με τον αριθμό του νησιού. Συγκεκριμένα, το νησί χρωματίζεται με πράσινο όταν ο αριθμός του αντιστοιχεί με το πλήθος των γεφυρών που συνδέονται σε αυτό και κόκκινο όταν το πλήθος των γεφυρών υπερβαίνει τον αριθμό.

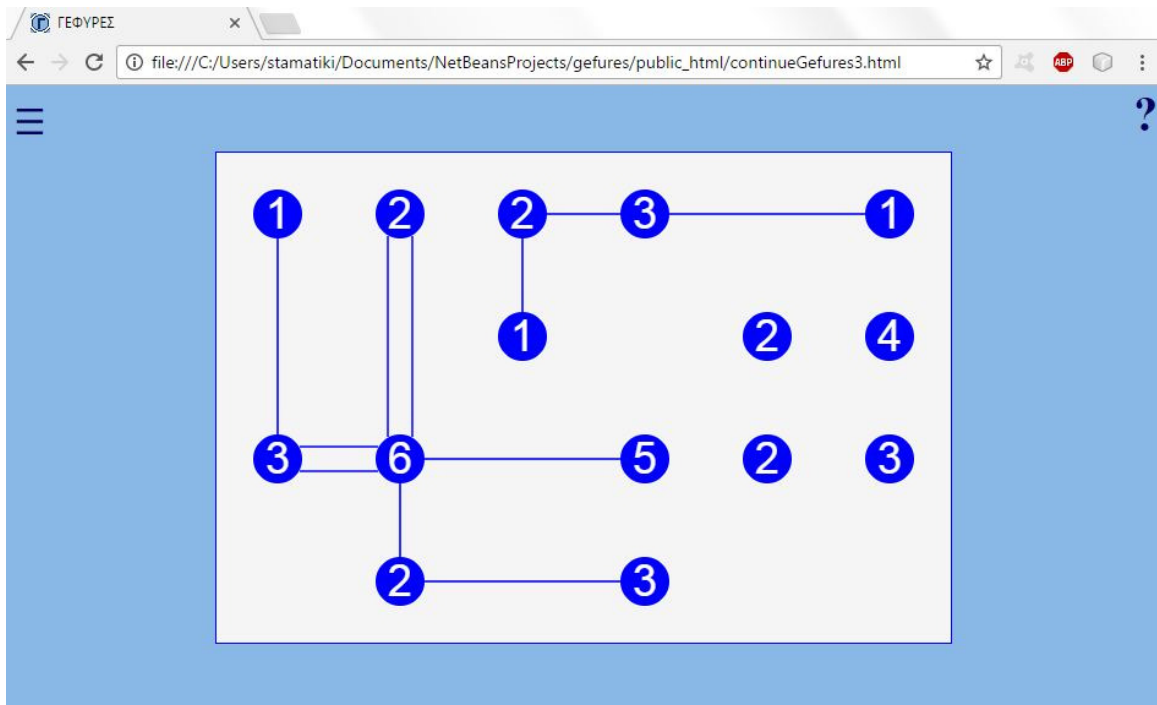


Εικόνα 7. Pop-up μήνυμα όταν οι γέφυρες υπερβαίνουν τον αριθμό κάποιου νησιού



Εικόνα 8. Επίπεδο 2

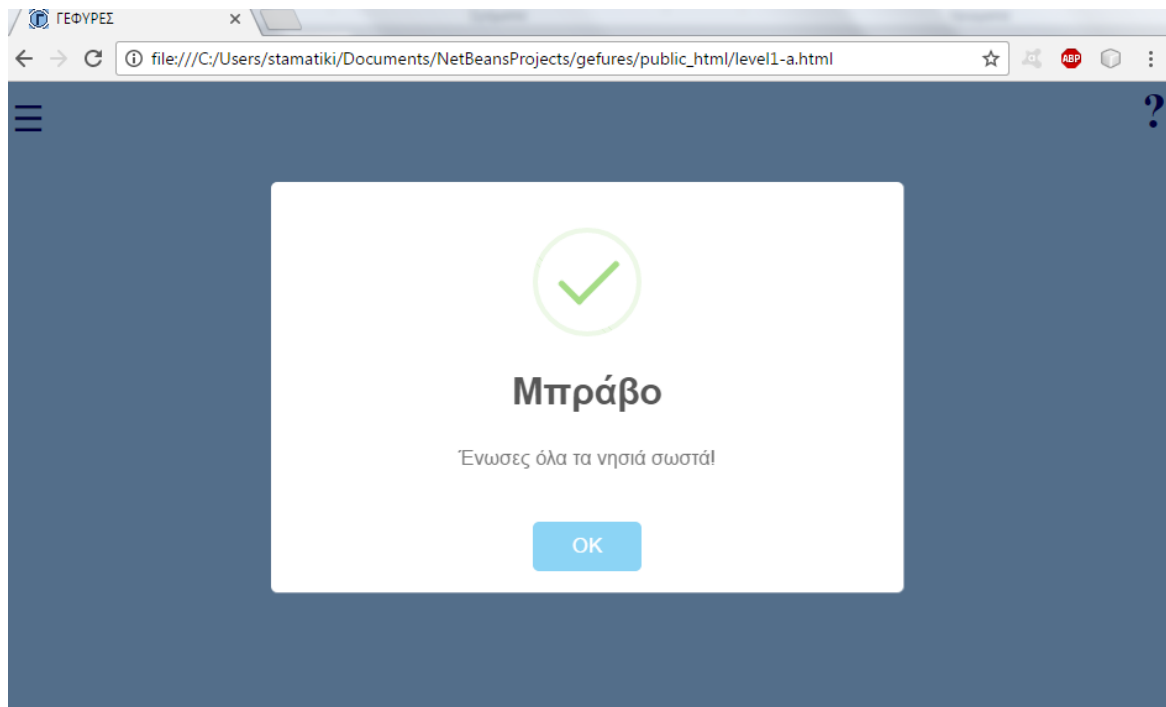
Στο δεύτερο επίπεδο το πλαίσιο του παιχνιδιού έχει διάσταση 3x5 και έχει αφαιρεθεί το pop-up μήνυμα των κύριων κανόνων λάθους, έχει όμως διατηρηθεί ο χρωματισμός των νησιών.



Εικόνα 9. Επίπεδο 3

Στο τρίτο επίπεδο το πλαίσιο του παιχνιδιού έχει διάσταση 4x6 και έχουν αφαιρεθεί όλες οι βοηθητικές ενέργειες.

Επιπρόσθετα, σε όλους τους γύρους και σε όλα τα επίπεδα, τη στιγμή που ο παίχτης συνδέσει σωστά όλα τα νησιά εμφανίζεται μήνυμα επιτυχίας το οποίο θα μας οδηγήσει στον επόμενο γύρο του επιπέδου.



Εικόνα 10. Μήνυμα επιτυχίας και ολοκλήρωσης γύρου

5. ΜΟΝΤΕΛΟ ΔΕΔΟΜΕΝΩΝ

Μελετώντας το θέμα πριν την υλοποίηση του αντιλαμβανόμαστε τις βασικές ανάγκες σχεδιασμού του μοντέλου. Γενικά, σαν μοντέλο ξέρουμε ότι θα είναι ένας καμβάς, ο οποίος θα έχει αριθμούς μέσα σε κύκλους σε διάφορα σημεία και θα υπάρχει ανάγκη για σχεδιασμό γεφυρών/ευθειών ανάμεσα στους κύκλους. Σίγουρα, χρειάζεται και ένα βασικό μενού επιπέδων δυσκολίας και ένα μενού για νέο παιχνίδι ή συνέχεια του προηγούμενου. Χρειαζόμαστε, επίσης, με κάποιον τρόπο να δείχνουμε στον χρήστη τους κανόνες του παιχνιδιού.

Έτσι λοιπόν, ξεκινώντας θεωρούμε τις διαστάσεις του canvas, πλάτος και ύψος. Έπειτα με βάση τις διαστάσεις αυτές δημιουργούμε έναν πίνακα ο οποίος είναι αυτός που θα έχει καταχωρημένους τους αριθμούς των κύκλων-νησιών στα διάφορα αλλά συγκεκριμένα σημεία.

Έστω λοιπόν canvas με 600x400 διάσταση και ο πίνακας *Island* με κελιά διάστασης 100.

	0	1	2	3	4	5
0	3	0	2	0	3	0
1	0	1	2	2	0	1

Ο πίνακας στο παράδειγμα που ακολουθεί αποτελείται από:

2	4	0	0	4	5	0
3	0	1	0	0	4	2

- 6 στήλες
- 4 γραμμές
- 13 κελιά με αριθμό αποδεκτών γεφυρών νησιά
- 11 κενά κελιά.

Για να μπορέσουμε να ορίσουμε τις περιοχές όπου θα είναι αποδεκτά τα κλικ και θα ζωγραφίζονται οι γέφυρες καθώς και να καταχωρείται το πλήθος των κλικ που έκανε ο παίχτης θα χρειαστούμε ακόμα έναν πίνακα.

Μελετώντας τον πίνακα *Island* κατασκευάζεται ο πίνακας *Bridges* (9 στήλες, 16 γραμμές, συντεταγμένες κελιού γραμμής πίνακα *Island*, συντεταγμένες κελιού στήλης πίνακα *Island*, στήλη 4: αριθμός των click ανάμεσα στα αντίστοιχα νησιά 'default τιμή 0', στήλη 5: συντεταγμένες γραμμής 'x' αριστερού/άνω νησιού , στήλη 6: συντεταγμένες γραμμής 'y' αριστερού/άνω νησιού , στήλη 7: συντεταγμένες γραμμής 'x' δεξιού/κάτω νησιού , στήλη 8: συντεταγμένες γραμμής 'y' δεξιού/κάτω νησιού)

	0	1	2	3	4	5	6	7	8
0	0	0	0	2	0	70	40	230	60
1	0	2	0	4	0				
2	1	1	1	2	0				
3	1	2	1	3	0				
4	1	3	1	5	0				
5	2	0	2	3	0				
6	2	3	2	4	0				

7	3	1	3	4	0				
8	3	4	3	5	0				
9	0	0	2	0	0	40	70	60	230
10	1	1	3	1	0				
11	0	2	1	2	0				
12	1	3	2	3	0				
13	0	4	2	4	0				
14	2	4	3	4	0				
15	1	5	3	5	0				

Για να γεμίσει ο πίνακας αυτός πρέπει να γίνουν κάποιοι έλεγχοι.

1) Πρέπει να γίνεται ο έλεγχος αν το κελί είναι μηδενικό. $Island[i][j] \neq 0$

1. Αν το κελί είναι μηδενικό τότε δεν υπάρχει νησί και δεν καταχωρούνται στον πίνακα οι αντίστοιχες συντεταγμένες.
2. Αν το κελί περιέχει μη μηδενικό αριθμό τότε οι συντεταγμένες καταχωρούνται στον πίνακα, εφόσον:
 - a. Σε κάποιο από τα επόμενα κελιά της ίδιας γραμμής → αν ελέγχουμε οριζόντια ή της ίδιας στήλης → αν ελέγχουμε κάθετα* είναι μη μηδενικό αλλιώς προχωράμε στην επόμενη γραμμή/στήλη.
 - I. Πρέπει να γίνεται ο έλεγχος αν το τρέχον κελί είναι στο τελευταίο κελί γραμμής αν ελέγχουμε οριζόντια ή της ίδιας στήλης αν ελέγχουμε κάθετα*
 - i. Αν είναι στο τελευταίο κελί και είναι μηδενικό, τότε αλλάζει γραμμή αν ελέγχουμε οριζόντια ή στήλη αν ελέγχουμε κάθετα*

- ii. Αν είναι στο τελευταίο κελί και είναι μη μηδενικό, τότε καταχωρούνται στον πίνακα οι αντίστοιχες συντεταγμένες.
- iii. Αν δεν είναι στο τελευταίο κελί αλλά είναι μηδενικό, τότε προχωράει το δείκτη `tempj` μια θέση μετά.
- iv. Αν δεν είναι στο τελευταίο κελί και είναι μη μηδενικό, τότε καταχωρούνται στον πίνακα οι αντίστοιχες συντεταγμένες.

Ορίζουμε ακόμα 2 πίνακες:

IslandC διάστασης 4x6 αντίστοιχος του **Island** όμως με ήδη μηδενικές καταχωρημένες τιμές.

Αυτός ο πίνακας δημιουργήθηκε για να συγκρίνεται ο αριθμός των click με τον αριθμό του αντίστοιχου νησιού.

Solution διάστασης 1xcountB που εμπεριέχει τις τιμές της λύσης του παιχνιδιού, τιμές=αριθμός γεφυρών αντίστοιχων απέναντι νησιών με βάση τον πίνακα **Bridges**. Αυτός ο πίνακας δημιουργήθηκε για να ελέγχετε αν το πλήθος των σχηματισμένων γεφυρών στο κάθε σημείο είναι το σωστό για να ολοκληρωθεί ο γύρος.

Για να ζωγραφίσουμε τα νησιά και τις γέφυρες θα χρειαστούμε και ένα **function** (**drawIsland sAndBridges()**) κατά το οποίο θα χρησιμοποιούνται οι δυνατότητες του canvas σε ότι αφορά το σχεδιασμό των κύκλων και των ευθειών (χρώματα, στυλ, μέγεθος, σημείο σχεδιασμού).

Έτσι κάθε φορά που καλείται η συνάρτηση να σχηματίζονται όλα τα νησιά με αριθμούς και οι γέφυρες μεταξύ αυτών. Υποθέτουμε πως ίσως στο σημείο αυτό να γίνεται και ο έλεγχος αν το πλήθος των κλικ/γεφυρών είναι ο σωστός σε όλα τα σημεία.

Επίσης, για να πάρουμε τις συντεταγμένες του σημείου που έγινε το κλικ και πόσα κλικ έγιναν θα χρειαστούμε ακόμα ένα **function** (**startGame()**).

Επίσης, θα χρειαστεί δημιουργία **functions** και για την αποθήκευση/διαγραφή στοιχείων στο browser εάν θελήσουμε να αποθηκεύσουμε τη πρόοδο του παίχτη στον κάθε γύρο.

6. ΥΛΟΠΟΙΗΣΗ

Η υλοποίηση έγινε με Javascript και HTML5. Για τη μορφοποίηση και την απεικόνιση των αντικειμένων, των κειμένων και του canvas χρησιμοποιήθηκε CSS3.

6.1 ΑΡΧΙΚΟ ΜΕΝΟΥ

```
<!DOCTYPE html>
<html>
  <head>
    <title>ΓΕΦΥΡΕΣ</title> // Τίτλος στο tab του browser
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" type="text/css" href="menus.css" />
    <link rel="icon" href="favicon.ico" /> // εικονίδιο στο tab του browser
  </head>
  <body>
    <div>
      <h1>ΓΕΦΥΡΕΣ</h1> // Τίτλος στην οθόνη
      <nav> //αρχικό μενού επιλογής επιπέδου
        <ul>
          <li><a onclick="checksaveddata1()">Επίπεδο 1</a></li>
          <li><a onclick="checksaveddata2()">Επίπεδο 2</a> </li>
          <li><a onclick="checksaveddata3()">Επίπεδο 3</a> </li>
        </ul>
      </nav>
    </div>
    <script src="storecurrentstage.js"></script>
  </body>
</html>
```

Για τη μορφοποίηση:

```
body { background-color: #8AB8E6; } /*Χρώμα φόντου

p,h1{
  font-family: fantasy; /*Γραμματοσειρά κειμένου σε παράγραφο ή τίτλο h1*/
  color: #122a52; /*Χρώμα Γραμματοσειράς*/
  text-align: center; /*Στοιχισι Γραμματοσειράς στο κέντρο*/
}
nav {
  width: 200px; /*πλάτος μενού*/
  position: absolute; /* Θέση*/
  top: 5em; /*απόσταση πάνω*/
  bottom: 0; /*απόσταση κάτω*/
  left: 0; /*απόσταση αριστερά*/
  right: 0; /*απόσταση δεξιά*/
  margin: auto; /*περιθώριο εξωτερικά*/
}
li {
  background-color: #C2EBFF; /*χρώμα φόντου λίστας μενού*/
  border: 1px solid #8AB8E6; /*περίγραμμα, πάχος, στυλ και χρώμα*/
  list-style-type: none; /*στυλ λίστας*/
  padding: 10px; /*περιθώριο εσωτερικά*/
```

```

margin: 5px; /*περιθώριο εξωτερικά*/
text-align: center; /*Στοιχισή στο κέντρο*/
}

nav a{
text-decoration:none; /* στυλ*/
font-family:fantasy; /*Γραμματοσειρά*/
cursor: pointer; /*στυλ δείκτη ποντικού */
color: #000066; /*Χρώμα Γραμματοσειράς*/
padding-left: 30px; /*περιθώριο εσωτερικά αριστερά*/
padding-right: 30px; /*περιθώριο εσωτερικά δεξιά*/
}

nav ul {
margin: 0;
padding: 0;
}

a:hover{ color:#122a52;} /*Χρώμα Γραμματοσειράς όταν το ποντίκι βρίσκεται στο σημείο "a"*/

```

6.2 ΟΘΟΝΗ ΠΑΙΧΝΙΔΙΟΥ

```

<!DOCTYPE html>
<html>
  <head>
    <title>ΓΕΦΥΡΕΣ</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="sweetalert.css"> //Ετοιμη βιβλιοθήκη
για όμορφα alerts
    <link rel="stylesheet" type="text/css" href="canvasgaming.css" />
//μορφοποίηση σελίδας
    <link rel="icon" href="favicon.ico" />
  </head>
  <body >
    <div id="myNavmenu" class="leftsidenav"> //πλαινό μενού
      <a href="javascript:void(0)" class="closebtn"
onclick="closeNav()" ">&times;</a> //κλικ για κλείσιμο μενού
      <a href="javascript:void(0)"
onclick="savemode()" ">αποθήκευση</a> //επιλογή για αποθήκευση κινήσεων
      <a href="index.html">αρχικό μενού</a> //επιλογή για επιστροφή
στο αρχικό μενού
    </div>
    <span id="showmenu" onclick="openNav()" ">&#9776;</span> //κλικ για
άνοιγμα μενού
    <span id="instructions" onclick="openInstructions()" ">&quest;</span>
////κλικ για άνοιγμα οδηγιών-κανόνων παιχνιδιού
    <div style="padding-top: 10px;">
      <canvas id="myCanvas" width="600" height="400">Your browser
does not support the HTML5 canvas tag.</canvas> // tag canvas
    </div>

```

```

        <script src="level3-a.js"></script> //js file για το παιχνίδι στον
canva
        <script type="text/javascript" src="storecurrentstage.js"></script>
//js file για αποθήκευση στοιχείων στον browser
        <script src="leftsidenavmenu.js"></script> //js file για το
αριστερά μενού
        <script src="instructions.js"></script> //js file για το pop-up
μήνυμα με τις οδηγίες
        <script src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
//js file για το sweetalert
        <script src="sweetalert-dev.js"></script> //js file για το
sweetalert
    </body>
</html>

```

Για τη μορφοποίηση:

```

body {
    background-color: #8AB8E6;
}
/*o canvas*/
#myCanvas {
    position: absolute; /*θέση*/
    top:0;
    bottom: 0;
    left: 0;
    right: 0;
    margin:auto;
    background-color: whitesmoke;
    border:1px solid blue;
}

/* κρυφό μενού */
.leftsidenav {
    height: 100%; /* ύψος */
    width: 0; /* πλάτος*/
    position: fixed; /* θέση, μείνε εκεί */
    z-index: 1; /* προτεραιότητα εμφάνισης - layer */
    top: 0;
    left: 0;
    background-color: #000066;
    overflow-x: hidden; /* απενεργοποίηση κάθετου scroll */
    padding-top: 60px; /* */
    transition: 0.5s; /* 0.5 δευτερόλεπτα εφέ εμφάνισης */
}

/* links μενού */
.leftsidenav a {
    padding: 8px 30px 8px 30px;
    text-decoration: none;
    font-size: 25px;
    color: #C2EBFF;
    display: block;
    transition: 0.3s;
    font-family: fantasy;
    font-weight: lighter;
}

```

```

/* στο mouse over στα links, αλλαγή χρώματος */
.leftsidenav a: hover, .offcanvas a: focus{
    color: #ffffff;
    background-color: #000046;
}

/* Θέση και στυλ του κουμπιού κλεισίματος (εpanw δε3ia gwnia sto
leftsidenav) */
.leftsidenav .closeln{
    position: absolute;
    top: 0;
    font-size: 36px;
    margin-left: 120px;
}

/* στυλ περιεχόμενου οθόνης */
#main {
    transition: margin-left .5s;
    padding: 20px;
}

/* Se o8ones me upsos mikrotero apo 450px */
@media screen and (max-height: 450px) {
    .leftsidenav {padding-top: 15px;}
    .leftsidenav a {font-size: 18px;}
}

.questmark {
    text-align: right;
    font-weight: 700;
    font-size: 50px;
}

/* span (?) epanw δε3ia sti gwnia to oπoio emfanizei tis odigies*/
#instructions{
    position: fixed; /* θέση, μείνε εκεί */
    top: 0;
    right: 0;
    padding-right: 5px;
    font-weight: 700;
    font-size: 40px;
    color: #000066;
    text-decoration: none;
    cursor: help;
}

/* span epanw aristera sti gwnia to oπoio emfanizei to leftsidenav*/
#showmenu {
    font-family: fantasy;
    font-size: 30px;
    cursor: pointer;
    color: #000066;
}

/* στο mouse over, αλλαγή χρώματος */
#showmenu: hover , #instructions: hover{
    color: white;
}

```

JavaScript για την υλοποίηση (ισχύει αντίστοιχα για όλους τους γύρους και για όλα τα επίπεδα)

Ενδεικτικά παράδειγμα από το Επίπεδο 3, ο 1^{ος} γύρος:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var i, j, APOSTASI = 100; // APOSTASI anamesa sta kentra 2 kuklwn &
diastaseis keliwn n*n
var m = 4, n = 6; // diastaseis pinaka circles
var x = APOSTASI / 2, y = APOSTASI / 2;
var rad = APOSTASI / 5; //aktina kuklou
var Island = [[3, 0, 2, 0, 3, 0], [0, 1, 2, 2, 0, 1], [4, 0, 0, 4, 5, 0],
[0, 1, 0, 0, 4, 2]];
var countB = 0; //ari8mos twn eggegrammenwn grammwn tou pinaka bridges
var startb1 = x + rad; //arxiki timi suntetagmenwn tou kentrou tou
kuklou+rad (50+20)
var startb2 = y - rad / 2, tempi, tempj;
var w = rad; // platos tis clickable area
var numofnx; //deixnei ti 8esi tou n orizontia
var numofny = 0; //deixnei ti 8esi tou n ka8eta
var s; //metritis tw n pou einai kena
var countwrongmoves=0; //metritis gia to poses fores energopoih8ike to swal
var stage='b';

//array me apo8ikeumenes tis suntetagmenes tw n apenanti nisiwn
//+ mia stili me ton ari8mo tw n click(gefures) meta3y aytw n, default=0
var Bridges = new Array(80);
for (i = 0; i < 80; i++)
{
    Bridges[i] = new Array(9);
}
for (i = 0; i < m; i++)
{
    numofnx = 0; //ari8mos keliwn pou proigountai tou aristerou nisiou
    for (j = 0; j < n; j++)
    {
        if (Island[i][j] !== 0) //an uparxei kuklos sto keli
        {
            if (j < n - 1) //an oxi stin teleutaia stili tis grammis tote
            {
                tempj = j + 1; //apo8ikeuse stin metabliti tempj tin epomeni
                8esi stilis
                s = 0;
                if (Island[i][tempj] === 0) //an stin epomeni 8esi stilis
                den uparxei kuklos tote
                {
                    //au3ise tin tempj+1 mexri tempj
                    = teleutaia stili tis grammis
                    while (Island[i][tempj] === 0 && tempj < n - 1)
                    {
                        tempj++;
                        s++; //ari8mos midenikwn nisiwn anamesa sta tuxon
                        apenanti nisia
                    }
                }
            }
            if (Island[i][tempj] !== 0 && tempj < n)
            {
                //suntetagmenes aristerou kuklou
                Bridges[countB][0] = i;
                Bridges[countB][1] = j;
                //suntetagmenes de3iou kuklou
                Bridges[countB][2] = i;
                Bridges[countB][3] = tempj;
            }
        }
    }
}
```

```

//poses fores egine click stin perioxi auti
Bridges[countB][4] = 0;
if (i === 0 && j === 0)//an stin epanw aristeri gwnia
tou canva, dil stin arxi
{
    Bridges[countB][5] = startb1;
    Bridges[countB][6] = startb2;
}
else
{
    Bridges[countB][5] = startb1 + numofnx * APOSTASI;
    Bridges[countB][6] = startb2 + numofny * APOSTASI;
}
Bridges[countB][7] = startb1 + (numofnx + s) * APOSTASI
+ APOSTASI - 2 * rad;
Bridges[countB][8] = startb2 + numofny * APOSTASI + w;
countB++;
numofnx++;
    }
}
}
else
    numofnx++;
}
numofny++;
}
horizontal_lines = countB;
numofnx = 0;
for (j = 0; j < n; j++)
{
    numofny = 0;
    for (i = 0; i < m; i++)
    {
        if (Island[i][j] !== 0)
        {
            if (i < m - 1)//an oxi stin teleutaia grammi tis stilis tote
            {
                tempi = i + 1; //apo8ikeuse stin metabliti tempi tin
epomeni 8esi grammis
                s = 0;
                if (Island[tempi][j] === 0)//an stin epomeni 8esi grammis
den uparxei kuklos tote
                {
                    //au3ise tin tempi+1 mexri tempi
= teleutaia grammi tis stilis
                    while (Island[tempi][j] === 0 && tempi < m - 1)
                    {
                        tempi++;
                        s++;
                    }
                }
            }
            if (Island[tempi][j] !== 0 && tempi < m)
            { //suntetagmenes anw kuklou
                Bridges[countB][0] = i;
                Bridges[countB][1] = j;
                //suntetagmenes katw kuklou
                Bridges[countB][2] = tempi;
                Bridges[countB][3] = j;
                //poses fores egine click stin perioxi auti
                Bridges[countB][4] = 0;
                if (i === 0 && j === 0)
                {

```

```

        Bridges[countB][5] = startb2;
        Bridges[countB][6] = startb1;
    }
    else
    {
        Bridges[countB][5] = startb2 + numofnx * APOSTASI;
        Bridges[countB][6] = startb1 + numofny * APOSTASI;
    }
    Bridges[countB][7] = startb2 + numofnx * APOSTASI + w;
    Bridges[countB][8] = startb1 + (numofny + s) * APOSTASI
+ APOSTASI - 2 * rad;
    countB++;
    numofny++;
    }
    }
    }
    else
        numofny++;
    }
    numofnx++;
}

// dimiourgia kuklwn
function drawIslandsAndBridges ()
{
    ctx.clearRect(0, 0, c.width, c.height, "white");
    ctx.font = '36px sans-serif';
    ctx.lineWidth = 1.6;
    y = APOSTASI / 2;
    for (i = 0; i < m; i++)
    {
        x = APOSTASI / 2;
        for (j = 0; j < n; j++)
        {
            if (Island[i][j] !== 0)
            {
                ctx.beginPath();
                ctx.strokeStyle = 'white';
                ctx.fillStyle = 'blue';
                ctx.arc(x, y, rad, 0, 2 * Math.PI);
                ctx.stroke();
                ctx.fill();
                ctx.fillStyle = 'white';
                ctx.fillText(Island[i][j], x - rad / 2, 2 + y + rad / 2);
            }
            x = x + APOSTASI;
        }
        y = y + APOSTASI;
    }
    ctx.strokeStyle = 'blue';
    for (i = 0; i < horizontal_lines; i++)
    {
        if (Bridges[i][4] === 1)
        {
            ctx.lineCap = 'butt';
            ctx.beginPath();
            ctx.moveTo(Bridges[i][5], Bridges[i][6] + 10);
            ctx.lineTo(Bridges[i][7], Bridges[i][6] + 10);
            ctx.stroke();
        }
        if (Bridges[i][4] === 2)

```

```

    {
        ctx.lineCap = 'butt';
        ctx.beginPath();
        ctx.moveTo(Bridges[i][5] - 2, Bridges[i][6]);
        ctx.lineTo(Bridges[i][7] + 2, Bridges[i][6]);
        ctx.moveTo(Bridges[i][5] - 2, Bridges[i][8]);
        ctx.lineTo(Bridges[i][7] + 2, Bridges[i][8]);
        ctx.stroke();
    }
}
for (i = horizontal_lines; i < countB; i++)
{
    if (Bridges[i][4] === 1)
    {
        ctx.lineCap = 'butt';
        ctx.beginPath();
        ctx.moveTo(Bridges[i][5] + 10, Bridges[i][6]);
        ctx.lineTo(Bridges[i][5] + 10, Bridges[i][8]);
        ctx.stroke();
    }
    if (Bridges[i][4] === 2)
    {
        ctx.lineCap = 'butt';
        ctx.beginPath();
        ctx.moveTo(Bridges[i][5], Bridges[i][6] - 2);
        ctx.lineTo(Bridges[i][5], Bridges[i][8] + 2);
        ctx.moveTo(Bridges[i][7], Bridges[i][6] - 2);
        ctx.lineTo(Bridges[i][7], Bridges[i][8] + 2);
        ctx.stroke();
    }
}
counter=0;
for (i = 0; i < countB; i++)
{
    if (Bridges[i][4]===Solution[i])counter++;
}
if (counter===countB)
{
    $(document).ready(function() {
        swal({
            title: "Μπράβο",
            text: "Ενώσεσ όλα τα νησιά σωστά!",
            type: "success",
            showConfirmButton: true
        },
        function(){
            window.location.href = 'level3-b.html';
        });
    });
}
};
//alert(Bridges[9][5]);

var IslandC = [[0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0]];// antistoixos tou pinaka Island
function startGame() {
    ctx.canvas.addEventListener('click', function (event)
    {
        var mouseX = event.clientX - ctx.canvas.offsetLeft;

```



```

    var mouseY = event.clientY - ctx.canvas.offsetTop;
    //alert(mouseX+" | "+mouseY);
    for (i = 0; i < countB; i++)
    {
        if (mouseX > Bridges[i][5] && mouseX < Bridges[i][7] && mouseY
> Bridges[i][6] && mouseY < Bridges[i][8])
        {
            Bridges[i][4]++;
            o = Bridges[i][0]; // ari8mos keliou grammis
            t = Bridges[i][1]; // ari8mos keliou stilis
            h = Bridges[i][2]; // ari8mos keliou grammis
            f = Bridges[i][3]; // ari8mos keliou stilis
            IslandC[o][t]++; // au3anw +1 ston neo pinaka wste na
sugkrinw me ti lusi
            IslandC[h][f]++;
            if (Bridges[i][4] === 3) //an ginoun 3 click midenise kai
to antistoixo keli
            {
                Bridges[i][4] = 0;
                IslandC[o][t] = IslandC[o][t] - 3; //den bazw apeu8eias
0 gt to ka8ena apo ayta apeu8unetai se ena nisi
                IslandC[h][f] = IslandC[h][f] - 3; //to oporto mporei
na dexetai apo 4 diaforetikes pleures gefures
            }
            drawIslandsAndBridges();
            break;
        }
    }
});
};

//solution array, specified clicks/bridges of solution
var Solution = [1, 1, 1, 1, 0, 2, 1, 1, 1, 2, 0, 0, 1, 2, 2, 1];

window.addEventListener('load', drawIslandsAndBridges);
window.addEventListener('load', startGame);

```

6.3 ΠΡΟΣΩΡΙΝΗ ΑΠΟΘΗΚΕΥΣΗ ΣΤΟΙΧΕΙΩΝ ΣΤΟ BROWSER

Δευτερεύον Μενού για συνέχιση παιχνιδιού:

```

<!DOCTYPE html>
<html>
  <head>
    <title>ΓΕΦΥΡΕΣ</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" type="text/css" href="menus.css" />
    <link rel="icon" href="favicon.ico" />
  </head>
  <body>
    <div>
      <h1>ΓΕΦΥΡΕΣ</h1>
      <nav>
        <ul>

```

```

        <li><a onclick="clearSavedData()" href="level1-
a.html">νέο παιχνίδι</a></li>
        <li><a href="continueGefures1.html">συνέχεια</a></li>
    </ul>
</nav>
</div>
<script src="storecurrentstage.js"></script>
<script src="continuegame1.js"></script>
</body>
</html>

```

JavaScript για την υλοποίηση της αποθήκευσης / διαγραφής στοιχείων στο browser (ισχύει για όλους τους γύρους και για όλα τα επίπεδα)

```

function savemode() {
localStorage.setItem("currentactions", JSON.stringify(Bridges));
localStorage.setItem("theseIslands", JSON.stringify(Island));
localStorage.setItem("thewidth", JSON.stringify(n));
localStorage.setItem("theheight", JSON.stringify(m));
localStorage.setItem("thesolution", JSON.stringify(Solution));
localStorage.setItem("theCisland", JSON.stringify(IslandC));
localStorage.setItem("thecountB", JSON.stringify(countB));
localStorage.setItem("thehorizontal_lines",
JSON.stringify(horizontal_lines));
localStorage.setItem("the-stage", stage);

swal({
  title: "Αποθήκευση...",
  timer: 500,
  showConfirmButton: false
});
}

//διαγραφή στοιχείων από το browser
function clearSavedData() {
  localStorage.removeItem("currentactions");
}

//an exei ginei apothikeusi emfanise menoy alliws emfanise lo guro toy
antistoixou epipedou
function checksaveddata1() {
  x=localStorage.getItem("currentactions");
  if (x===null)
    window.location.href = 'level1-a.html';
  else
    window.location.href = 'newgamemenu.html';
}
function checksaveddata2() {
  x=localStorage.getItem("currentactions");
  if (x===null)
    window.location.href = 'level2-a.html';
  else
    window.location.href = 'newgamemenu2.html';
}
function checksaveddata3() {
  x=localStorage.getItem("currentactions");

```

```

    if (x===null)
        window.location.href = 'level3-a.html';
    else
        window.location.href = 'newgamemenu3.html';
}

```

Ανάκληση / φόρτωση αποθηκευμένων στοιχείων για συνέχιση του παιχνιδιού

Ενδεικτικά από το Επίπεδο 1:

```

Bridges=localStorage.getItem("currentactions");
Bridges=JSON.parse(Bridges);

Island=localStorage.getItem("theseIslands");
Island=JSON.parse(Island);

n=localStorage.getItem("thewidth");
n=JSON.parse(n);

m=localStorage.getItem("theheight");
m=JSON.parse(m);

Solution=localStorage.getItem("thesolution");
Solution=JSON.parse(Solution);

IslandC=localStorage.getItem("theCisland");
IslandC=JSON.parse(IslandC);

countB=localStorage.getItem("thecountB");
countB=JSON.parse(countB);

horizontal_lines=localStorage.getItem("thehorizontal_lines");
horizontal_lines=JSON.parse(horizontal_lines);

stage=localStorage.getItem("the-stage");
var c = document.getElementById("myCanvas");
c.width = n*100;
c.height=m*100;

var ctx = c.getContext("2d");
var i, j, APOSTASI = 100; // APOSTASI anamesa sta kenra 2 kuklwn &
diastaseis keliwn n*n
var x = APOSTASI / 2, y = APOSTASI / 2;
var rad = APOSTASI / 5; //aktina kuklou
var startb1 = x + rad; //arxiki timi suntetagmenwn tou kentrou tou
kuklou+rad (50+20)
var startb2 = y - rad / 2, tempi, tempj;
var w = rad; // platos tis clickable area
var numofnx; //deixnei ti 8esi tou n orizontia
var numofny = 0; //deixnei ti 8esi tou n ka8eta
var s; //metritis tw n pou einai kena
var countwrongmoves=0; //metritis gia to poses fores energopoih8ike to swal

// dimiourgia kuklwn-nisiwn
function drawIslandsAndBridges()
{
    ctx.clearRect(0, 0, c.width, c.height, "white");

```

```

ctx.font = '36px sans-serif';
ctx.lineWidth = 1.6;
y = APOSTASI / 2;
for (i = 0; i < m; i++)
{
    x = APOSTASI / 2;
    for (j = 0; j < n; j++)
    {
        if (Island[i][j] !== 0)
        {
            ctx.beginPath();
            ctx.strokeStyle = 'blue';
            if (Island[i][j]===IslandC[i][j]){
                ctx.fillStyle = 'green';
                ctx.strokeStyle='white';}
            else if (Island[i][j]>IslandC[i][j])
                ctx.fillStyle = 'white';
            else
            {
                ctx.fillStyle = 'red';
                ctx.strokeStyle='white';
                if(countwrongmoves<1)
                {
                    swal({
                        title: "Να θυμάσαι πως",
                        text: "Οι γέφυρες δεν πρέπει να υπερβαίνουν
τον αριθμό του νησιού που συνδέονται, ούτε να διασταυρώνονται με άλλες
γέφυρες",
                        imageUrl: "message.jpg", timer:
3000,allowOutsideClick:true,
                        showConfirmButton: true
                    });
                    countwrongmoves=countwrongmoves+1;
                }
                else if(countwrongmoves<3)
                {
                    countwrongmoves=countwrongmoves+1;
                }
                else
                {
                    countwrongmoves=0;
                }
            }

            ctx.arc(x, y, rad, 0, 2 * Math.PI);
            ctx.stroke();
            ctx.fill();
            ctx.strokeText(Island[i][j], x - rad / 2, 2 + y + rad / 2);
        }
        x = x + APOSTASI;
    }
    y = y + APOSTASI;
}
ctx.strokeStyle = 'blue';
for (i = 0; i < horizontal_lines; i++)
{
    if (Bridges[i][4] === 1)
    {
        ctx.lineCap = 'butt';
        ctx.beginPath();

```

```

        ctx.moveTo(Bridges[i][5], Bridges[i][6] + 10);
        ctx.lineTo(Bridges[i][7], Bridges[i][6] + 10);
        ctx.stroke();
    }
    if (Bridges[i][4] === 2)
    {
        ctx.lineCap = 'butt';
        ctx.beginPath();
        ctx.moveTo(Bridges[i][5] - 2, Bridges[i][6]);
        ctx.lineTo(Bridges[i][7] + 2, Bridges[i][6]);
        ctx.moveTo(Bridges[i][5] - 2, Bridges[i][8]);
        ctx.lineTo(Bridges[i][7] + 2, Bridges[i][8]);
        ctx.stroke();
    }
}
for (i = horizontal_lines; i < countB; i++)
{
    if (Bridges[i][4] === 1)
    {
        ctx.lineCap = 'butt';
        ctx.beginPath();
        ctx.moveTo(Bridges[i][5] + 10, Bridges[i][6]);
        ctx.lineTo(Bridges[i][5] + 10, Bridges[i][8]);
        ctx.stroke();
    }
    if (Bridges[i][4] === 2)
    {
        ctx.lineCap = 'butt';
        ctx.beginPath();
        ctx.moveTo(Bridges[i][5], Bridges[i][6] - 2);
        ctx.lineTo(Bridges[i][5], Bridges[i][8] + 2);
        ctx.moveTo(Bridges[i][7], Bridges[i][6] - 2);
        ctx.lineTo(Bridges[i][7], Bridges[i][8] + 2);
        ctx.stroke();
    }
}
counter=0;
for (i = 0; i < countB; i++)
{
    if (Bridges[i][4]===Solution[i]) counter++;
}
if (counter===countB)
{
    if(stage==='a'){
        $(document).ready(function() {
            swal({
                title: "Μπράβο",
                text: "Ένωσες όλα τα νησιά σωστά και ολοκλήρωσες το Επίπεδο 1
!",
                type: "success",
                showConfirmButton: true
            },
            function() {
                window.location.href = 'level2-a.html';
            });
        });
    }
    else
    {

```

```

$(document).ready(function() {
    swal({
        title: "Μπράβο",
        text: "Ενώσεξ όλα τα νησιά σωσιά!",
        type: "success",
        showConfirmButton: true
    },
    function(){
        clearSavedData();
        window.location.href = 'level1-'+stage+'.html';
        //localStorage.clear();
    });
});
}
}
};
//alert(Bridges[9][5]);

function startGame() {
    ctx.canvas.addEventListener('click', function (event)
    {
        var mouseX = event.clientX - ctx.canvas.offsetLeft;
        var mouseY = event.clientY - ctx.canvas.offsetTop;
        //alert(mouseX+" | "+mouseY);
        for (i = 0; i < countB; i++)
        {
            if (mouseX > Bridges[i][5] && mouseX < Bridges[i][7] && mouseY
> Bridges[i][6] && mouseY < Bridges[i][8])
            {
                Bridges[i][4]++;
                o = Bridges[i][0]; // ari8mos keliou grammis
                t = Bridges[i][1]; // ari8mos keliou stilis
                h = Bridges[i][2]; // ari8mos keliou grammis
                f = Bridges[i][3]; // ari8mos keliou stilis
                IslandC[o][t]++; // au3anw +1 ston neo pinaka wste na
sugkrinw me ti lusi
                IslandC[h][f]++;
                if (Bridges[i][4] === 3) //an ginoun 3 click midenise kai
to antistoixo keli
                {
                    Bridges[i][4] = 0;
                    IslandC[o][t] = IslandC[o][t] - 3; //den bazw apeu8eias
0 gt to ka8ena apo ayta apeu8unetai se ena nisi
                    IslandC[h][f] = IslandC[h][f] - 3; //to oporto mporei
na dexetai apo 4 diaforetikes pleures gefures
                }
                drawIslandsAndBridges();
                break;
            }
        }
    });
};

window.addEventListener('load', drawIslandsAndBridges);
window.addEventListener('load', startGame);

```

ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Κατά την περίοδο αυτή, της δημιουργίας της πτυχιακής, απέκτησα πολύ καλύτερη αντίληψη πάνω στον προγραμματισμό, βελτίωσα τον τρόπο σκέψης μου, έμαθα να κάνω πιο στοχευμένη αναζήτηση στο ίντερνετ και να αντιλαμβάνομαι ταχύτερα που είναι το λάθος ή τι μπορεί να φταίει εάν κάτι δεν τρέχει όπως θα έπρεπε.

Μαθαίνοντας, λοιπόν, και αποκτώντας εμπειρία σκοπεύω να συνεχίσω τη βελτιστοποίηση του κώδικα. Υπάρχει φαντασία, διάθεση και ιδέες για τον σκοπό αυτό. Μία, ίσως, φιλόδοξη από πλευράς πολυπλοκότητας ιδέα είναι ενδεικτικά και η εξής:

Έστω $X_0, X_1, X_2, X_3, X_4, X_5$ το πλήθος του κάθε αριθμού που υπάρχει στο ταμπλό. (πλήθος/επανάληψη κάθε αριθμού που βρίσκεται μέσα σε κάθε κύκλο).

Πιο αναλυτικά:

X_0 : πλήθος των μηδενικών νησιών, δηλ. των κενών κελιών

X_1 : πλήθος των νησιών με τον αριθμό 1

X_2 : πλήθος των νησιών με τον αριθμό 2

X_3 : πλήθος των νησιών με τον αριθμό 3

X_4 : πλήθος των νησιών με τον αριθμό 4

X_5 : πλήθος των νησιών με τον αριθμό 5

Έστω y_0, y_1, y_2 το πλήθος των σημείων χωρίς γέφυρα, με μονή γέφυρα και με διπλές γέφυρες αντίστοιχα που ενώνουν τα νησιά.

Έχουμε την εξής εξίσωση:

$$0 \cdot X_0 + 1 \cdot X_1 + 2 \cdot X_2 + 3 \cdot X_3 + 4 \cdot X_4 + 5 \cdot X_5 = (0 \cdot y_0 + 1 \cdot y_1 + 2 \cdot y_2) \cdot 2$$

Το β' μέρος πολ/εται με 2, διότι το πλήθος των γεφυρών 'αφαιρείται' από τον αριθμό δύο νησιών.

Γενικά η εξίσωση μας επιβεβαιώνει πως το άθροισμα από το περιεχόμενο των νησιών ισούται με το άθροισμα όλων των γεφυρών επί δύο.

Σε απλοποιημένη μορφή θα είναι:

$$X_1 + 2 \cdot X_2 + 3 \cdot X_3 + 4 \cdot X_4 + 5 \cdot X_5 = 2 \cdot y_1 + 4 \cdot y_2$$

ΠΕΔΙΑ ΟΡΙΣΜΟΥ ΜΕΤΑΒΛΗΤΩΝ

$X_{0-4} = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]$

$X5 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]$

*Το π.ο. είναι μέχρι το 20 διότι δεν μπορεί να μπει σε γωνία το '5' διότι μπορεί να ενωθεί μόνο με 4 συνολικά γέφυρες οι οποίες δεν συμπληρώνουν το άθροισμα 5.

$y0, y1, y2 = [0 - 38]$

Επίσης πρέπει να ισχύει:

$$X0 + X1 + X2 + X3 + X4 + X5 = 24$$

και

$$y0 + y1 + y2 = 38$$

Πιο απλά, στόχος είναι η παραγωγή νέων γύρων παιχνιδιού αυτόματα χωρίς την προκατάληψη των τιμών σε πίνακες. Η αναζήτηση και η έρευνα στο πως θα κατασκευαστεί ένας αλγόριθμος ή μία συνάρτηση που θα βρίσκει λύσεις για την παραπάνω εξίσωση με πολλαπλούς αγνώστους έπεται συνέχεια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βικιπαίδεια

<http://stackoverflow.com>

<https://www.w3schools.com/>

<http://www.favicon.cc/>

<http://t4t5.github.io/sweetalert/>

<http://diveintohtml5.info/storage.html>

<https://jsfiddle.net/>

<https://www.tutorialspoint.com>

Χρησιμοποιήθηκαν τα προγράμματα netbeans, notepad++