ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"Δημιουργία παιχνιδιού με χρήση Html, Css, Javascript και Canva εως 13 Kb"

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΩΝ: Μόρτος Ισίδωρος, ΑΜ:1479

Ζαχείλας Ιωάννης , ΑΜ:1686

ΕΠΙΒΛΕΠΩΝ: Σωτήρης Χριστοδούλου

ΑΝΤΙΡΡΙΟ 2017

Εγκρίθηκεαπότην τριμελήεξεταστικήεπιτροπή

Αντίρριο, 2017,

ΕΠΙΤΡΟΠΗΑΞΙΟΛΟΓΗΣΗΣ

1. ,Υπογραφή

2. ,Υπογραφή

3. ,Υπογραφή

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία μίας ιστοσελίδας που θα περιέχει ένα παιχνίδι μέχρι 13kb. Πιο συγκεκριμένα για την υλοποίηση του παιχνιδιού χρησιμοποιήθηκαν οι γλώσσες προγραμματισμού HTML, Javascript , CSS και η χρήση Canva. Υλοποίηση και ανάπτυξη μιας ιδέας για τη συμμετοχή σε έναν παγκόσμιο διαγωνισμό της σελίδας js13kgames.com με στόχο την δημιουργία παιχνιδιού δύο διαστάσεων με όριο ολικής χωρητικότητας τα 13kb όπως προαναφέραμε.

Πίνακας περιεχομένων

Κεφάλαιο 1 - Εισαγωγή

Υπάρχουν αρκετά προγράμματα που προσφέρουν ιδιαίτερα εύχρηστο και απλό περιβάλλον για την κατασκευή επαγγελματικών και κομψών ιστοσελίδων με πολλές δυνατότητες σε ελάχιστο χρόνο και χωρίς να απαιτούν από τον χρήστη ειδικές γνώσεις. Η γλώσσα html τείνει να εξαφανιστεί καθώς υπάρχουν αρκετές άλλες περισσότερο εξελιγμένες και με μεγαλύτερες δυνατότητες όπως η XML, η sHTML και άλλα διάφορα. Ωστόσο η άποψη αυτή είναι αβάσιμη καθώς αρκετές φορές θα χρειαστεί κάποιος να επέμβει στον κώδικα τόσο για αλλαγές ή ενέργειες τις οποίες το πρόγραμμα που χρησιμοποιούμε δεν μπόρεσε να εκτελέσει όσο και για διορθώσεις σφαλμάτων. Με λίγα λόγια η html μας διευκολύνει και να είναι η μόνη λύση για την κατασκευή μίας ιστοσελίδας. Ενδεικτικά μπορούμε να αναφέρουμε ότι η html μπορεί να δημιουργήσει εντυπωσιακές σελίδες με πλαίσια, φωτογραφίες, και άλλα διάφορα. Η JavaScript από την άλλη είναι γλώσσα συγγραφής σεναρίων (scripting language) που χρησιμοποιείται για να προσθέσει εφέ και διαλογικότητα στις ιστοσελίδες μας και είναι ανταγωνιστική της γλώσσας προγραμματισμού VBScript. Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί κιόλας να εκτελεσθεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JavaScript, αρκεί μόνο ο φυλλομετρητής (browser) να υποστηρίζει την JavaScript. Ο βασικός στόχος της παρούσας πτυχιακής είναι η δημιουργία ενός παιχνιδιού με μικρή χωρητικότητα αποθήκευσης όπου μέσα από αυτό θέλουμε να αναδείξουμε την ιδέα που δεν περιορίζεται. Επίσης θα βασίζεται σε μία ιστοσελίδα φτιαγμένη από html και εσωτερικό css.

1.2 Δομή Εργασίας

Η δομή της εργασίας μας αποτελείται από τα εξής κεφάλαια:

(α) **Κεφάλαιο 1^{ο}.**

(β) **Κεφάλαιο 2^{ο}.**

(γ) **Κεφάλαιο 3^{ο}.**

1.3 Τα πρώτα δικτυακά παιχνίδια

Πολύ πριν την εξάπλωση του Διαδικτύου, είχαν διαδοθεί δικτυακά παιχνίδια. Αυτά συνήθως προσφέρονταν ως υπηρεσία μέσω εταιρειών που παρείχαν δικτυακές υπηρεσίες τύπου Dialup bulletin. Οι υπηρεσίες αυτές ήταν πολύ διαδεδομένες στο εξωτερικό. Τα πρώτα τέτοια συστήματα ήταν διαθέσιμα από τα τέλη της δεκαετίας του ’70, ενώ γνώρισαν μεγάλη επιτυχία στις αρχές και μέσα του ’80. Έδιναν στον χρήστη ένα περιβάλλον επικοινωνίας που βασιζόταν σε κείμενο και πολύ γρήγορα εμφανίστηκαν τα παιχνίδια, που υποστήριζαν πολλούς παίκτες και μετέφεραν τους συνδρομητές σε φανταστικούς κόσμους γεμάτους περιπέτειας. Τα παιχνίδια αυτά ονομάστηκαν Multi user Dungeons και ήταν ο πρόδρομος των σημερινών Massively Multiuser Online Role Playing Games (MMORG), όπως το πολύ γνωστό World of Warcraft ή και το Eve Online. Ανάμεσα στους προδρόμους των δικτυακών παιχνιδιών πρέπει να αναφέρουμε το Maze War (1974), διαδικτυακό παιχνίδι, που λειτουργούσε σε ένα μεγάλο δίκτυο ερευνητικών υπολογιστών, καθώς και το Spasim, ένα τρισδιάστατο παιχνίδι προσομοίωσης διαστημοπλοίου, που έτρεχε κάτω από υπέρ-υπολογιστές της εποχής του.

Κεφάλαιο 2 – Γνωρίζοντας τους Κώδικες

2.1 JS13K και GitHub

Πολύ σημαντική ιστοσελίδα για αυτούς που θέλουν να πάρουν ιδέες σχετικά με την δημιουργία παιχνιδιών αποτελεί το JS13Kgames, όπου όποιος επιθυμεί μπορεί να πάρει μέρος και σε διαγωνισμούς ,όπως μία κατηγορία είναι να φτιάξεις παιχνίδια μέχρι 13kb. Πέρα όμως από ιδέες, το JS13K συνδυάζεται με την ιστοσελίδα GitHub που εκεί υπάρχουν διαθέσιμοι κώδικες με τους οποίους έχουν δημιουργηθεί τα games. Είναι αρκετά αποτελεσματικό γιατί σε βοηθάει να κατανοήσεις πως γίνεται η υλοποίηση στο παιχνίδι το οποίο παρακολουθείς. Επίσης υπάρχουν άλλες ιστοσελίδες όπου συμπεριλαμβάνουν παιχνίδια έτοιμα, δείχνουν τον κώδικα που είναι γραμμένος και είναι αρκετά βοηθητικά, αναλυτικά, φιλικά και κατανοητά.

2,2 Περιοχή Canvas

Μια περιοχή canvas μπορεί να εμφανίσει διάφορα γραφικά σε μια σελίδα, όπως απλά διαγράμματα, εντυπωσιακά interfaces, κινούμενα γραφικά, παραστάσεις και εξωτερικές εικόνες. Η περιοχή canvas είναι μια ορθογώνια περιοχή. Η σχεδίαση γραφικών μέσα σε αυτήν γίνεται με την τεχνική pixel-based drawing API, δηλαδή η σχεδίαση των γραφικών γίνεται ορίζοντας σημεία μέσα στην περιοχή. Αν χρησιμοποιήσουμε σωστά και έξυπνα τη JavaScript καταφέρνουμε να δώσουμε «ζωή», στα σχεδιαστικά γραφικά στην περιοχή canvas. Με την ετικέτα <canvas> ορίζουμε μια περιοχή στην σελίδα μας η οποία δεν έχει από μόνη της σχεδιαστικές δυνατότητες. Η σχεδίαση γραφικών μέσα στην περιοχή γίνεται με την βοήθεια της γλώσσας JavaScript. Ο κώδικας JavaScript τοποθετείται μέσα σε μια function η οποία είναι γραμμένη έτσι ώστε να "τρέχει" κάθε φορά που φορτώνει η σελίδα. Σχεδιάζουμε γραφικά σε μια περιοχή χρησιμοποιώντας τις συντεταγμένες x και y.

Οι μέθοδοι του αντικειμένου context, μπορούν να σχεδιάσουν σχήματα, να μεταμορφώσουν ένα σχήμα (όπως να το περιστρέψουν), να δημιουργήσουν το εφέ της σκιάς σε ένα γραφικό περιβάλλον, να γεμίζουν με χρώμα ένα σχήμα, να βάλουν περίγραμμα. κτλ. Με τις μεθόδους του αντικειμένου μπορούμε να δημιουργήσουμε διαθέσιμα σχήματα όπως γραμμές, ορθογώνιο, τμήμα κύκλου ή αλλιώς τόξο, καμπύλες, σύνθετα σχήματα, επίσης μπορούμε φτιάξουμε εφέ όπως να βάλουμε σκιές περιγράμματα και μπορούμε να κάνουμε μεταμορφώσεις όπως κλιμάκωση και περιστροφή.

2.3 Παραδείγματα Φυσικών κινήσεων σε κώδικα

Όταν αρχίζεις να δημιουργείς ένα παιχνίδι θα πρέπει να έχεις μελετήσει για αυτό. Σε αυτό το σημείο θα δούμε πως μπορούμε να δημιουργήσουμε έναν χαρακτήρα αλλά και πως αυτός ο χαρακτήρας μπορεί να κινηθεί μέσα σε ένα γραφικό χώρο. Στο ίντερνετ υπάρχουν πάρα πολλά site και video τα οποία μας δείχνουν πως μπορούμε

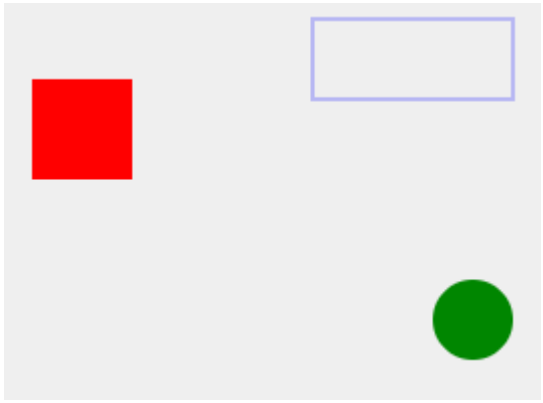
να επιτύχουμε κίνηση ενός χαρακτήρα είτε τον χειριζόμαστε είτε όχι. Η αρχή πάντα όμως γίνεται από το background. Εδώ βλέπουμε ποιος είναι ο κώδικας για να εισάγουμε canva. Επίσης στην συνέχεια υπάρχει εισαγωγή javascript που δείχνει πως γίνεται το γέμισμα με το χρώμα πράσινο μέσα στον ίδιο τον πίνακα.

```
<html>

<body>

<canvas id="myCanvas"

width="200" height="100"

style="border:5px solid black;">

</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

ctx.fillStyle = "green";

ctx.fillRect(0,0,200,200);

</script>

</body>

</html>
```

Δημιουργία και άλλων σχημάτων στον ίδιο canva.

```
<html>
<head>
<style>
canvas { background: #eee; }
</style>
```

```
</head>
<body>
<canvas id="myCanvas" width="480"
height="320"></canvas>
<script>
var ctx = canvas.getContext("2d");
ctx.beginPath();
ctx.rect(20, 40, 50, 50);
ctx.fillStyle = "#FF0000";
ctx.fill();
ctx.closePath();

ctx.beginPath();
ctx.arc(240, 160, 20, 0, Math.PI*2, false);
ctx.fillStyle = "green";
ctx.fill();
ctx.closePath();

ctx.beginPath();
ctx.rect(160, 10, 100, 40);
ctx.strokeStyle = "rgba(0, 0, 255, 0.5)";
ctx.stroke();
ctx.closePath();
</script>
</body>

</html>
```



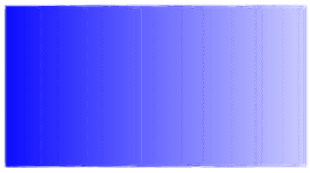
Με αυτόν εδώ τον κώδικα javascript μπορούμε να δούμε πως δημιουργείτε μία φυσική κίνηση στο canva που έχουμε προγραμματίσει.

```
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
var ballRadius = 10;
var x = canvas.width/2;
var y = canvas.height-30;
var dx = 2;
var dy = -2;

function drawBall() {
ctx.beginPath();
ctx.arc(x, y, ballRadius, 0, Math.PI*2);
ctx.fillStyle = "#0095DD";
ctx.fill();
ctx.closePath();
}
```

```
function draw() {
ctx.clearRect(0, 0, canvas.width, canvas.height);
drawBall();

if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) {
        dx = -dx;
    }
if(y + dy > canvas.height-ballRadius || y + dy < ballRadius) {
        dy = -dy;
    }

    x += dx;
    y += dy;
}


setInterval(draw, 10);
```

Η διαφορά με το παράδειγμα που προαναφέραμε είναι ότι στο παρακάτω σχήμα επιλέγουμε δύο χρώματα και του δημιουργούμε το εφέ της επισκίασης, ώστε να γίνεται διαφανές λίγο το χρώμα καθώς οδεύει προς την αλλαγή.

```
<html>

<body>



<canvas id="myCanvas" width="200"
height="100" style="border:1px solid
black;">

Your browser does not support the
HTML5 canvas tag.</canvas>



<script>

var c =
document.getElementById("myCanvas");
```



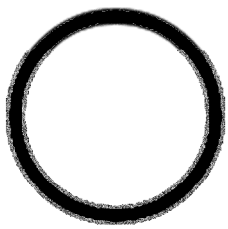
var ctx = c.getContext("2d");
 	
// Create gradient
 	
var grd = ctx.createLinearGradient(0,0,200,0);
 	
grd.addColorStop(0,"blue");
 	
grd.addColorStop(1,"white");
 	
// Fill with gradient
 	
ctx.fillStyle = grd;
 	
ctx.fillRect(10,10,150,80);
 	
</script>
 	
</body>
 	
</html>	

Σε αυτό το σημείο βλέπουμε ένα ακόμα παράδειγμα με το πώς μπορούμε να γράψουμε μέσα στο canva.

<html> <body> <canvas id="myCanvas" width="230" height="100" style="border:1px solid #d3d3d3;"> Your browser does not support the HTML5 canvas tag.</canvas>	

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.strokeText("BOUNCE BALL",10,50);
</script>
</body>
</html>	

Σε αυτό το σημείο βλέπουμε τη δημιουργία ενός κυκλικού σχήματος.

<html>
<head>
<style>
body {
margin: 0px;
padding: 0px;
}
</style>
</head>
	

```
<body>

<canvas id="myCanvas" width="278"
height="200"></canvas>

<script>

var canvas =
document.getElementById('myCanvas');

var context = canvas.getContext('2d');

var centerX = canvas.width / 2;

var centerY = canvas.height / 2;

var radius = 50;



context.beginPath();

context.arc(centerX, centerY, radius, 0, 2 *
Math.PI, false);

context.fillStyle = 'white';

context.fill();

context.lineWidth = 8;

context.strokeStyle = 'black';

context.stroke();

</script>

</body>

</html>
```

Κεφάλαιο 3 – Σχεδιασμός του Παιχνιδιού

3.1 Σκοπός του παιχνιδιού

Σε αυτό το κεφάλαιο θα περιγράψουμε τον σκοπό του παιχνιδιού, παρακάτω θα δώσουμε μία μικρή περιγραφή περί της δημιουργίας των εικόνων μέσω προγράμματος αλλά και την εξήγηση σε σχέση με την διαμόρφωση τους.

Αρχικά υπάρχουν πέντε χαρακτήρες όπως ο βασικός παίχτης μα όπου είναι μία μπάλα, έπειτα δύο βόμβες, δύο καραμέλες και ένα σύννεφο. Επίσης υπάρχει εφέ με σύννεφα που μετακινούνται στην εικόνα, τα οποία επαναλαμβάνονται όταν βγουν έξω από τον χώρο του canva. Ο παίχτης αυτό που προσπαθεί να κάνει είναι να πιάσει όσες πιο πολλές καραμέλες μπορεί και να προσπαθήσει να παραμείνει στο παιχνίδι αποφεύγοντας τις κινούμενες βόμβες. Οι ζωές για τον παίχτη μας είναι τρεις και σε περίπτωση που τις χάσει το παιχνίδι τελειώνει σηματοδοτώντας το “Gameover”. Βέβαια το παιχνίδι δεν είναι τόσο απλό. Κάθε φορά που ο βασικός χαρακτήρας πιάνει μία καραμέλα ή τον χτυπάει μία βόμβα μετατοπίζονται όλα στην κίνηση κατά τυχαίο τρόπο μέσα στον χώρο με κίνδυνο να χάσει μία από τις διαθέσιμες ζωές του. Ο παίχτης έχει στη διάθεση του 60’ δευτερόλεπτα για να πάρει όσες καραμέλες μπορεί. Σε αυτό το σημείο όμως να εξηγήσουμε και τον τρίτο χαρακτήρα μας όπου είναι το σύννεφο. Το σύννεφο έχει το ρόλο να βοηθήσει τον παίχτη και να του χαρίσει μία έξτρα ζωή, ώστε να αποφύγει τυχόν πρόωρη λήξη του παιχνιδιού λόγο μηδενικής ζωής. Είναι στην ευχέρεια του παίχτη αν θα σπαταλήσει χρόνο ψάχνοντας το βοηθητικό σύννεφο, παραμερίζοντας ίσως το μεγαλύτερο δυνατό αποτέλεσμα συλλέγοντας καραμέλες. Ο παίχτης νικάει όταν τελειώσει ο χρόνος έχοντας διατηρήσει έστω μία ζωή.

3.2 Αναφορά δημιουργίας χαρακτήρων

Οι χαρακτήρες μας θυμίζουν πολύ ένα φανταστικό κόσμο. Η μπάλα βρίσκεται στον ουρανό για να θυμίσει ότι στην φαντασία δεν υπάρχουν όρια, ούτε βαρύτητα. Οι καραμέλες μας θυμίζουν τα παιδικά μας χρόνια και το τι άρεσε στους περισσότερους. Η βόμβα και η μπάλα έχουν τη μορφή του καλού και του κακού αντίστοιχα. Θυμίζοντας μας παιδικά video games όπου πάντα επικρατούσε η αυστηρή μορφή αντίπαλου και η γλυκιά αθώα μορφή του βασικού παίχτη. Το γραφικό περιβάλλον θυμίζει ουρανό με σύννεφα, καθαρή και γνώριμη εικόνα. Οι χαρακτήρες της βόμβας και της μπάλας δημιουργήθηκαν εξολοκλήρου από μας από το πρόγραμμα Photoshop το όποιο διατίθεται επί πληρωμή και από εκεί έγινε η επεξεργασία των αναλύσεων αλλά και η δημιουργία των χαρακτήρων μας.

3.3 Πρόγραμμα Επεξεργασίας Φωτογραφίας

Οι χαρακτήρες του παιχνιδιού είναι φωτογραφίες σε μορφή png. Όλες τις εικόνες τις δημιουργήσαμε εξολοκλήρου όπως προαναφέραμε στο κεφάλαιο 3.2. Γενικότερα όμως χρειάστηκε το πρόγραμμα επεξεργασίας γιατί χωρίς αυτό δεν θα μπορούσαμε να κάνουμε ποστεροποίηση τις εικόνες, ώστε να χαμηλώσουμε την ανάλυση, να τις μικρύνουμε σε μέγεθος και να τις κόψουμε. Αυτή η διαδικασία μας βοήθησε αρκετά, έτσι καταφέραμε να κάνουμε τις εικόνες μας bytes και να διατηρήσουμε το στόχο του παιχνιδιού στα 13kb.

3.4 Ανάλυση δημιουργίας του κώδικα

Ο κώδικας που δημιουργήσαμε το παιχνίδι ήταν βασισμένος σε ένα απλό παιχνίδι, στο οποίο υπήρχε ένας χρήστης που έπιανε απλά ένα τέρας σε μια περιοχή όσες φορές ήθελε. Εμείς εδώ δανειστήκαμε στοιχεία του κώδικα αυτού και δημιουργήσαμε αρκετούς χαρακτήρες με την επανάληψη του κώδικα ώστε να φτιάξουμε την ιδέα μας. Μετέπειτα βρήκαμε ένα άλλο παιχνίδι όπου είχε χρόνο 60 δευτερόλεπτα αντίστροφα και πήραμε τον κώδικα για να έχουμε χρονόμετρο. Από ένα άλλο παιχνίδι επίσης δανειστήκαμε τον κώδικα που δίνει κίνηση. Στη συνέχεια επεξεργαστήκαμε τον κώδικα που είχαμε εφόσον το κατανοήσαμε για να μπορέσουμε να αλλάξουμε την κίνηση της βόμβας και τέλος προσθέσαμε ήχο κατά την επαφή των μεταβλητών για πιο ρεαλιστικό αποτέλεσμα.

Κεφάλαιο 4 - Λύσεις στην παρουσία προβλημάτων

4.1 Εξωτερικές εικόνες αντί canvas

Το παιχνίδι που φτιάξαμε δανείστηκε ιδέες από ένα mix παιχνιδιών.. Χρησιμοποιήσαμε τρόπου γραφής κώδικα των άλλων παιχνιδιών όπου μέσα από αυτό εφαρμόσαμε την δική μας ιδέα. Τις εικόνες προσπαθήσαμε να τις ενσωματώσουμε σε μία αλλά τελικά βρήκαμε άλλον τρόπο και δεν το εφαρμόσαμε. Οπότε δεν χρησιμοποιήσαμε css sprite. Αυτό γνωρίζαμε ότι μπορεί να μας έφερνε προβλήματα στον κώδικα μας γιατί το παιχνίδι έχει απαιτήσεις και περιορισμένο διαθέσιμο χώρο, όμως σκεφτήκαμε λύσεις όπως να τις επεξεργαστούμε για να χαμηλώσουμε την ανάλυση και τα kb.

4.2 Απλος κώδικας

Ο κώδικας που πήραμε είναι αρκετά εύκολος και κατανοητός. Αυτό όμως δεν σημαίνει ότι δεν μας παρουσιάστηκαν προβλήματα σε σχέση με τον χώρο. Χρησιμοποιήσαμε εύκολο κώδικα γιατί για μας ήταν άγνωστο το κομμάτι του προγραμματισμού διαδικτύου για παιχνίδι. Όσο αναφορά τη κίνηση της βόμβας και της καραμέλας χρησιμοποιήσαμε το ίδιο κώδικα αντίστοιχα για να μπορέσουμε να εξοικονομήσουμε χώρο. Για να κινούνται συνεχόμενα οι βόμβες μέσα στο πλαίσιο αφαιρέσαμε κώδικα γιατί προηγουμένως η κίνηση της βόμβας μας περιόριζε στην κίνηση.

4.3 Δημιουργία ζωής για τον Χρήστη

Σε αυτό το σημείο θα αναφέρουμε πόσο εύκολο και δύσκολο ήταν να δημιουργήσουμε τις ζωές για τον παίχτη μας. Αρχικά αυτό που κάναμε είναι να δηλώσουμε την βόμβα μας σαν μεταβλητή ίση με τρία. Στη συνέχεια βάλαμε κατά την επαφή με τον χρήστη να μειώνεται η μεταβλητή μας κατά ένα. Μετά για να μπορεί να τελειώσει το παιχνίδι η βόμβας μας πρέπει να είναι ίση με το μηδέν, έτσι εμφανίζεται το τέλος παιχνιδιού.

4.4 Εξωτερικός ήχος

Για τον εξωτερικό ήχο δανειστήκαμε στοιχεία από άλλα παιχνίδια. Ο ήχος ήταν το πιο δύσκολο κομμάτι για να μην ξεπεράσουμε τα 13kb. Στην αρχή είχαμε βάλει εξωτερικό ήχο αλλά βλέπαμε ότι είχαμε μεγάλο πρόβλημα και για αυτό είπαμε να χρησιμοποιήσουμε κώδικα γραπτό που βγάζει ήχο, νότες δηλαδή. Όμως επειδή δεν τα καταφέραμε ξανά γυρίσαμε στον εξωτερικό ήχο και καταφέραμε να μειώσουμε τα kb μέσω του προγράμματος Adobe Premiere Ellements κρατώντας μόνο ένα κομμάτι.

4.5 Πως τελειώνει το παιχνίδι

Για να τελειώσει το παιχνίδι αυτό που καταφέραμε να υλοποιήσουμε είναι να δημιουργήσουμε έναν κώδικα στη περίπτωση που ο παίχτης νικήσει ή χάσει ώστε να παγώνει η κίνηση των χαρακτήρων. Αυτόν τον κώδικα τον δανειστήκαμε από το "game over" του ήχου, όπου τον κάναμε παραμετροποίηση για τους δικού μας χαρακτήρες και τον φέραμε στα μέτρα και στις ανάγκες του παιχνιδιού μας.

ΚΕΦΑΛΑΙΟ 5ο – ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ

5.1 Σχέδιο Ιστοσελίδας και Κατασκευή Ιστοσελίδας.

Για την δημιουργία του παιχνιδιού αυτό που χρειάστηκε με την βοήθεια από το W3 school είναι μία εισαγωγή για να παρουσιαστεί το παιχνίδι μέσα από εκεί. Στην εισαγωγή του παιχνιδιού πατώντας το κουμπί που ονομάζεται startgame σε μεταφέρει στο παιχνίδι και έτσι ξεκινάει. Επειδή το παιχνίδι έχει τον πιο σημαντικό λόγο στην εισαγωγή αναδείχθηκε μόνο ένα μικρό κομμάτι γνώσης. Παρακάτω θα δούμε φωτογραφίες οι οποίες είναι απλές και κατανοητές. Αναδεικνύονται κώδικες που μας δείχνουν τι μπορούμε να κάνουμε.

5.2 Ανάλυση του παιχνιδιού

Το σχέδιο του παιχνιδιού δεν βασίζεται στον προγραμματισμό παρά στην ιδέα που θέλει ο χρήστης να υλοποιήσει. Η ιδέα είναι πιο δύσκολη να υλοποιηθεί όταν υπάρχει λιγότερος χώρος. Βέβαια όπως είδαμε στα παραπάνω κεφάλαια υπάρχουν αρκετοί τρόποι να αντλήσουμε ιδέες χρησιμοποιώντας όσο το δυνατότερο λιγότερο χώρο αποθήκευσης. Εισαγωγικά αναφέρουμε ότι η δική μας ιδέα ήταν να δημιουργήσουμε ένα παιχνίδι, το οποίο θα είναι εν μέρη γνώριμο στον κόσμο αλλά και λίγο διαφοροποιημένο από τα συνηθισμένα. Δημιουργήσαμε ένα παιχνίδι με ανταγωνισμό και με στόχο. Ο στόχος είναι να καταφέρει ο χρήστης να επιτύχει αυτό που ζητάει το παιχνίδι. Το παιχνίδι έχει ως σκοπό ο χρήστης να μαζέψει όσες πιο πολλές καραμέλες μπορεί. Κρατώντας το ενδιαφέρον του παίχτη σκεφτήκαμε να βάλουμε και μια σταγόνα έντασης. Προσθέσαμε λοιπόν ένα συννεφάκι το οποίο έχει θετικό ρόλο στο παιχνίδι, χαρίζοντας μια επιπλέον ζωή αν καταφέρει ο παίχτης να το πιάσει. Δεν είναι όμως εύκολα αντιληπτό διότι κινείται μαζί με τα υπόλοιπα σύννεφα, τα όποια μπορούν να τα αποκρύψουν. Αποφασίσαμε έτσι λοιπόν να προσθέσουμε και ένα χρονόμετρο. Τέλος για να βοηθήσουμε τον χρήστη του επιτρέψαμε να μην μπορεί να υπερβαίνει τα όρια του canva, ώστε να μπορεί να κινηθεί ελεύθερος σε όλο το γραφικό περιβάλλον. Αντίθετα παραδείγματος χάρη αν πάει να βγει έξω από τα όρια του canva είτε από τα δεξιά είτε από τα αριστερά θα επανέρχεται στο παιχνίδι από την αντίθετη πλευρά ώστε να μπορεί να συνεχίζει χωρίς να χάνει χρόνο. Επίσης έχει μπει ήχος κατά την επαφή του χρήστη με του άλλους χαρακτήρες για να δώσουμε μια ζωντάνια.

5.3 Αναλύσεις κώδικα

Σε αυτό το σημείο θα αναλύσουμε τον κώδικα μας. Για αρχή θα δούμε πως φτιάχτηκε η εισαγωγή του παιχνιδιού. Στη συνέχεια ένα άλλο αρχείο html, όπου συνδέεται με ένα εξωτερικό script που είναι γραμμένος όλος ο κώδικας λειτουργίας του παιχνιδιού. Θα δούμε εικόνες για να κατατοπίζουμε με ακρίβεια την εξήγηση του κώδικα ώστε να γίνει όσο πιο κατανοητό.

Εικόνα 1.1

```
<html>// Ανοίγουμε HTML
<head>// Θα γραψουμε στο κεφάλι κάποια πράγματα
<style type="text/css">// Ανοίγουμε εσωτερικού Css
div{ // Το στοιχείο <div> χρησιμοποιείται μαζί με το CSS, για τη διαμόρφωση μιας ιστοσελίδας
width: 700px;// Διάσταση πλάτους
height: 100%;// Διάσταση μήκους
text-align:center;// να είναι τα γράμματα στην μέση στο χώρο που έχουμε
font: 24px Courier New bold;// γραμματοσειρα
```

```
background: linear-gradient(cyan, lightgreen);// δηλώση δυο χρωμάτων
background-repeat: no-repeat; // Να μην επαναλαμβάνεται
}
h2// δηλωση επικεφαλιδας

{
font: 50px Ravie;
color:MediumVioletRed;// Μαύρο χρώμα
position: absolute;// Απόλυτη θέση
left: 180px;// Δήλωση θέσης από την αριστερή μεριά

}

p {// δηλωση γραμμάτων για θέση και γραμματοσειρά
position: absolute;
font-family:maiandra GD;
}

a:link, a:visited {// Δήλωση δυνδέσμου και να διαβάζει κάποια πράγματα
background-color: MediumVioletRed;//Δήλωση χρώματος περιγράμματος
color: white;
padding: 14px 25px;// Το παραπέτασμα καθαρίζει μια περιοχή γύρω από το
περιεχόμενο (μέσα στο περίγραμμα) ενός στοιχείου.
text-align: center;// να είναι τα γράμματα στην μέση στο χώρο που έχουμε
text-decoration: none;
display: inline-block;
}
</style>
```

```
<body>// Ανοιγουμε το body
<div id="intro">
<span>
<h2>CandyBomb</h2>
<p style="left: 130px; top: -40px; color: black; font-size:300%;">&#9839;</p>//
Θεση των γραμματων , μεγεθος , χρωμα και συμβολο.
<p style="left: 265px; top: -230px; color: white; font-size:1200%;">&#9729;</p>
<p style="left: 260px; top: 200px; color: white; font-size:250%;">&#9899;</p>
<p style="left: 257px; top: 185px; color: red; font-size:300%;">&#9787;</p>
<p style="left: 300px; top: -5px; color: black; font-size:600%;">&#9732;</p>
```

```
<p style="left: 305px; top: 138px; color: black; font-size:300%;">&#9865;</p>
<p style="left: 210px; top: 504px; color: black; font-size:300%;">&#9758;</p>
<p style="left: 280px; top: 339px;"> Do you like Candies?</p>// Εδώγράφουμετηνεισαγωγή
<p style="left: 280px; top: 400px;"> Let's try in the 60' seconds!</p>
<p style="left: 280px; top: 470px;"> And catch them all! </p>
<p style="left: 287px; top: 580px; font-family:Curlz MT;"><a href="kinoymenesvomves.html">START GAME</a><p>// Εισαγωγήγραμμάτωνμεθεση, μεγεθοςκ.τ.λ. Έχουμεόμωςκαιεισαγωγηιστοσελίδαςγιαναμπορείνασυνδεθείστοπαιχνίδι.
</span>
</div>
</body>// Κλείνουμε το body
</html>// Κλείνουμε την html
```

Τώρα φτιάχνουμε ένα ακόμα αρχείο html το οποίο θα συνδέεται με τον κώδικα javascript όπου θα είναι ξεχωριστά μεταξύ τους.

```
<html>// Ανοίγουμε την html

<head>

<style>

a {

position: absolute;// Απόλυτηθέση

left: 70px;

top: 20px;

color: white;

font-size:130%;// Μέγεθος γραμμάτων

}

p {

position: absolute;
```

```
}

</style>

</head>
```

```
<body>

<audio id="soundEfx" src="Slurp+3.mp3" style="display: none;"></audio>//
Δηλώνουμετοήχοπουθαβάλουμε

<audio id="soundBfx" src="Cannon+3.mp3" style="display: none;"></audio>//
Δηλώνουμετοήχοπουθαβάλουμε

<audio id="soundClo" src="Bing-sound.mp3" style="display: none;"></audio>//
Δηλώνουμετοήχοπουθαβάλουμε

<a href="kinoymenesvomves.html">Restart</a>//
Βάζουμετηνιστοσελίδαπουθασυνδέεταιστο Restart

<p style="left: 162px;top: -33px;color: red;font-size:240%;">&#10084;</p>//
Γράφουμε τα σύμβολα που θέλουμε να βάλουμε στο παιχνίδι, τι χρώμα, τι
μέγεθοςγραμμάτων κ.τ.λ

<p style="left: 32px; top: -35px; color: limegreen; font-size:240%;">&#9851;</p>

<p style="left: 275px; top: -45px; color: salmon; font-size:290%;">&#9787;</p>

<script type="text/javascript" src="game.js" ></script>//Εισάγουμετονκώδικα
javascript

</body>
```

```
</html>
```

Εικόνα 1.2

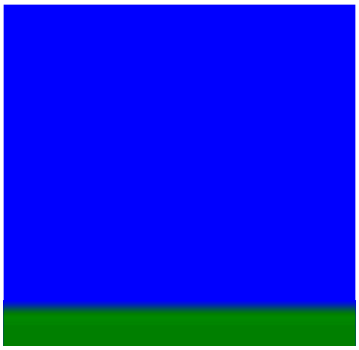
Βλέπουμε στην εικόνα 1.2 το παιχνίδι μας. Για να φτιαχτεί το παιχνίδι στην αρχή δημιουργήθηκε ένας canvas. Ο κώδικας καθώς βλέπουμε από εδώ και κάτω είναι αρχείο javascript το οποίο συνδέεται όπως αναφέραμε με το παραπάνω αρχείο μας που είναι html.

```
// Create the canvas

var canvas = document.createElement("canvas");

var ctx = canvas.getContext("2d");

canvas.width = 700;// Δηλώνουμε πλάτος
```

```
canvas.height = 530;// Δηλώνουμε ύψος

document.body.appendChild(canvas);
```

Στη συνέχεια τοποθετήθηκαν οι φωτογραφίες που έχουμε επιλέξει. Μπορούμε να προσθέσουμε και εικόνες με την χρήση canva και να αποφύγουμε τον σχεδιασμό χαρακτήρων. Αυτό έχουμε κάνει εμείς. Όπως βλέπουμε εξάλλου με αυτό τον τρόπο δημιουργήσαμε τις εικόνες του background μας και με τον ίδιο την μπάλα, την βόμβα, την καραμέλας και τους υπόλοιπους χαρακτήρες.

```
// Background image

var bgReady = false;

var bgImage = new Image();

bgImage.onload = function () {

bgReady = true;

};

bgImage.src = "images/background.png";
```

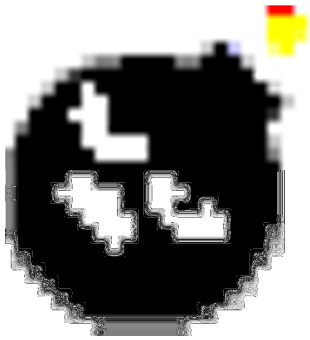


```
//surprise image

var cloudReady = false;

var cloudImage = new Image();

cloudImage.onload = function () {

cloudReady = true;

};

cloudImage.src = "images/clouds.png";
```

Εδώ επειδή η εικόνα δεν φαίνεται γιατί τασύννεφα είναι άσπρα και η εικόνα είναι σε μορφή png είναι τα σύννεφα που μετακινούνται μπροστά από το background.

```
// sweet image
var sweetReady = false;
var sweetImage = new Image();
sweetImage.onload = function () {
sweetReady = true;
};
sweetImage.src = "images/logo.png"
//sweet image
var karamelReady = false;
var karamelImage = new Image();
karamelImage.onload = function () {
karamelReady = true;
};
karamelImage.src = "images/logo.png";

//Hero image
var heroReady = false;
var heroImage = new Image();
heroImage.onload = function () {
```

```
heroReady = true;

};

heroImage.src = "images/hero.png";


// bomb image

var bombsReady = false;

var bombsImage = new Image();

bombsImage.onload = function () {

bombsReady = true;

};

bombsImage.src = "images/suprise.png";


// bomb image

var boomsReady = false;

var boomsImage = new Image();

boomsImage.onload = function () {

boomsReady = true;

};

boomsImage.src = "images/suprise.png";


//surprise image

var surpriseReady = false;
```



```
var surpriseImage = new Image();

surpriseImage.onload = function () {

surpriseReady = true;

};

surpriseImage.src = "images/cloud.png";
```

Έπειτα τοποθετήθηκαν οι μεταβλητές όπου περιέχουν την ταχύτητα του παίχτη και την αφετηρία του. Έγινε η εισαγωγή του ήχου, δημιουργήσαμε ταχύτητες επίσης και στους υπόλοιπους χαρακτήρες και δηλώσαμε σε δύο μεταβλητές ίσες με έναν αριθμό εκκίνησης.

```
// Αντικείμενα του παιχνιδιού

var hero = { //Δήλωση χαρακτήρα

speed: 256 // ταχύτητατουπαίχτη

};



var soundEfx; // Sound Efx

var soundLoad = "Slurp+3.wav"; // Δηλώνουμετονήχο



var soundBfx;

var soundLoad = "Cannon+3.wav";
```

```
var soundClo; // Sound Efx
var soundLoad = "Bing-sound.wav";


var booms = { /Δήλωση χαρακτήρα
speed: 210}; // Ταχύτητα της βόμβας
var boomCaught = 0;


var bombs = {
speed: 190};
var bombsCaught = 3; // Δηλώνουμε μεταβλητή τρία για να ξεκινάμε με ζωή=3


hero.x = canvas.width / 2 ; // Να ξεκινάει στην αρχή από αυτό το σημείο
hero.y = canvas.height / 2; // Να ξεκινάει στην αρχή από αυτό το σημείο


var sweet = {};
var sweetsCaught = 0;


var karamel= {};
var karamelsCaught = 0;


var surprise= {
speed: 80};
```

```
var surprisesCaught = 0;



var cloud= {

speed: 80};
```

Από εδώ και πέρα αρχίζουμε να προγραμματίζουμε την λειτουργία του παιχνιδιού.

```
// Χειρισμός των χειριστηρίων πληκτρολογίου

var keysDown = {};



addEventListener("keydown", function (e) {

keysDown[e.keyCode] = true;

}, false);



addEventListener("keyup", function (e) {

delete keysDown[e.keyCode];

}, false);
```

Όταν ο παίχτης πιάνει μία καραμέλα αμέσως μετά αλλάζει θέση όπως και οι υπόλοιπες καραμέλες φυσικά. Αυτό γίνεται με την var reset = function () όπου σε αυτό το σημείο επαναλαμβάνεται το παιχνίδι όταν ο χρήστης πιάσει μία καραμέλα. Μέσα στην function γράφεται ο κώδικας που θα κάνει τις καραμέλες να εμφανίζονται σε διαφορετικό σημείο μέσα στο canva.

```
// Επαναφέρετε το παιχνίδι όταν η συσκευή πιάνει τα γλυκά
```

```
var reset = function () {// Σε αυτό το σημείο θέλουμε να κάνουμε επαναφορά λειτουργίας



// Ρίξτε τα χρώματα κάπου στην οθόνη τυχαία

sweet.x = 32 + (Math.random() * (canvas.width - 84));

sweet.y = 45 + (Math.random() * (canvas.height - 35));

karamel.x = 32 + (Math.random() * (canvas.width - 84));

karamel.y = 45 + (Math.random() * (canvas.height - 35));



surprise.x = 20 + (Math.random() * (canvas.width + 14));

surprise.y = 45 + (Math.random() * (canvas.height - 255));

// Ρίξτε τα χρώματα κάπου στην οθόνη τυχαία

// Βάζουμε σε αυτό το σημείο και τις τρεις βόμβες για να κάνουμε πιο ενδιαφέρον το παιχνίδι. Δεν τοποθετούμε περιορισμούς ώστε όταν πιάνουμε την καραμέλα να μην αλλάζουν και οι βόμβες.



bombs.x = 32 + (Math.random() * (canvas.width – 10));

bombs.y = 45 + (Math.random() * (canvas.height - 255));



booms.x = 32 + (Math.random() * (canvas.width - 10));

booms.y = 45 + (Math.random() * (canvas.height - 135));
```

Με αυτή την λειτουργία δηλώνουμε σε ποιο σημείο του άξονα x και y θέλουμε να κινούνται τα σύννεφα.

```
function init() {// Ρίξτε μονάδα λειτουργίας

cloud.x = (canvas.width + 100);

cloud.y = (canvas.height - 410);

}
```

```
// Ενημέρωση αντικειμένων παιχνιδιού



var update = function (modifier) {

if (38 in keysDown) { // Παίκτης που κρατά ψηλά

hero.y -= hero.speed * modifier;



}

if (40 in keysDown) { // Παίκτης που κρατά χαμηλά

hero.y += hero.speed * modifier;



}

if (37 in keysDown) {// Παίκτης που κρατά αριστερά

hero.x -= hero.speed * modifier;



}

if (39 in keysDown) { // Παίκτης που κρατά δεξιά

hero.x += hero.speed * modifier;
```

```
}
```

```
//Εδώ δηλώνουμε και τις βόμβες σε πιο άξονα θα κινούνται ώστε να μπορούν να κάνουν τις κινήσεις τους στον χώρο

bombs.x += (bombs.speed * modifier);

bombs.y += (bombs.speed * modifier);

booms.x += (bombs.speed * modifier);

surprise.x += (surprise.speed* modifier);

cloud.x += (cloud.speed* modifier);
```

Σε αυτό το σημείο θα φτιάξουμε τις επαφές που θα γίνονται κατά την επαφή της μπάλας και της καραμέλας όπως επίσης μεταξύ των βομβών και της μπάλας. Ο αριθμός που θα δούμε παρακάτω είναι ο τέλειος αριθμός που χρησιμοποιείται για την επαφή αυτή γιατί μοιάζει αρκετά ρεαλιστικό. Είναι σαν να κάνουν πραγματική επαφή οι χαρακτήρες μας. Ύστερα αυτό που ακολουθεί είναι να ξανά αλλάζουν θέση.

```
// Are they touching?

if (

hero.x <= (bombs.x + 32) // Επαφήστοσημείο x του hero

&& bombs.x <= (hero.x + 32)// Επαφή στο σημείο x

&& hero.y <= (bombs.y + 32)// Επαφή στο σημείο y του hero

&& bombs.y <= (hero.y + 32)// Επαφή στο σημείο y

) {

--bombsCaught;// Η επαφή με την βόμβα μειώνει τις ζωές
```

```
soundBfx.play();// Μπαίνει ο ήχος

reset();// Επαναλαμβάνεται

}

//Παρομοίως ισχύει και για τα παρακάτω και όπως βλέπουμε ότι ο κώδικα είναι ο ίδιος απλά το χρησιμοποιούμαι για άλλον χαρακτήρα.

if (

hero.x <= (booms.x + 32)

&& booms.x <= (hero.x + 32)

&&hero.y<= (booms.y + 32)

&&booms.y<= (hero.y + 32)

) {

--bombsCaught;

soundBfx.play();

reset();

}

if(

hero.x <= (sweet.x + 32)

&& sweet.x <= (hero.x + 32)

&&hero.y<= (sweet.y + 32)

&&sweet.y<= (hero.y + 32)

) {

++sweetsCaught;

soundEfx.play();
```

```
reset();
}
if (
hero.x <= (karamel.x + 32)
&& karamel.x <= (hero.x + 32)
&&hero.y<= (karamel.y + 32)
&&karamel.y<= (hero.y + 32)
)
{
++sweetsCaught;
soundEfx.play();
reset();
}
if (
hero.x <= (surprise.x + 32)
&& surprise.x <= (hero.x + 32)
&&hero.y<= (surprise.y + 32)
&&surprise.y<= (hero.y + 32)
)
{
++bombsCaught;
soundClo.play();
```

```
reset();

}
```

Τέλος αυτό που ακόμα θα τοποθετήσουμε μέσα στη function είναι για τον παίχτη μας μία κίνηση που όταν θα βγαίνει έξω από τη μία πλευρά του canva να ξανά μπαίνει μέσα στο παιχνίδι από την άλλη.

```
// Αν ο ήρωας τελειώσει από τον καμβά, ας το βάλουμε στην άλλη πλευρά του

if (hero.x < 0) // Αν είναι στον άξονα x μικρότερο από 0

hero.x = canvas.width; // Τότε ο ήρωας θα μπορεί να κινείται κατά πλάτος

if (hero.x > canvas.width) // Αν είναι στον άξονα x μεγαλύτερο από το πλάτος

hero.x = 0; //Τότε θα είναι ίσον με 0

if (hero.y < 45) // Αν είναι στον άξονα y μικρότερο από 45

hero.y = canvas.height; // Τότε ο ήρωας θα μπορεί να κινείται κατά ύψος

if (hero.y > canvas.height) // Αν είναι στον άξονα y μεγαλύτερο από το μήκος

hero.y = 45; //Τότε θα είναι ίσον με 45

if (surprise.x < 0)

surprise.x = canvas.width;

if (surprise.x >canvas.width)

surprise.x = 0;

if (surprise.y< 45)

surprise.y = canvas.height;

if (surprise.y> canvas.height)

surprise.y = 45;
```

```
if (bombs.x < 0)
bombs.x = canvas.width;
if (bombs.x >canvas.width)
bombs.x = 0;
if (bombs.y< 45)
bombs.y = canvas.height;
if (bombs.y> canvas.height)
bombs.y = 45;



if (booms.x < 0)
booms.x = canvas.width;
if (booms.x >canvas.width)
booms.x = 0;
if (booms.y< 45)
booms.y = canvas.height;
if (booms.y> canvas.height)
booms.y = 45;


if (cloud.x < -490)
```

```
cloud.x = canvas.width;

if (cloud.x >canvas.width)

cloud.x = -490;

if (cloud.y< 45)

cloud.y = canvas.height;

if (cloud.y> canvas.height)

cloud.y = 45;



};
```

Το παιχνίδι όμως για να λειτουργήσει σωστά θα πρέπει να ζωγραφίσουμε όλα τα γραφικά στοιχεία ώστε να κινούνται μέσα στον χώρο οι μεταβλητές μας. Αυτό επιτυγχάνεται με τον παρακάτω κώδικα.

```
// Σχεδιάστεταπάντα

var render = function () {

if (bgReady) {

ctx.drawImage(bgImage,0,0, 700, 550);

}

// Επειδή η εικόνα μας είναι μικρή τις δίνουμε εμείς διαστάσεις

if (heroReady) {

ctx.drawImage(heroImage, hero.x, hero.y,59,59);

}

// Επειδή η εικόνα μας είναι μικρή τις δίνουμε εμείς διαστάσεις
```

```
if (sweetReady) {
ctx.drawImage(sweetImage, sweet.x, sweet.y, 42,42);
}
// Επειδή η εικόνα μας είναι μικρή τις δίνουμε εμείς διαστάσεις
if (karamelReady) {
ctx.drawImage( karamelImage, karamel.x, karamel.y, 42,42);
}
// Επειδή η εικόνα μας είναι μικρή τις δίνουμε εμείς διαστάσεις
if (bombsReady) {
ctx.drawImage(bombsImage, bombs.x, bombs.y);
}

if (boomsReady) {
ctx.drawImage(boomsImage, booms.x, booms.y);
}
if (surpriseReady) {
ctx.drawImage( surpriseImage, surprise.x, surprise.y, 30,30);
}
// Επειδή η εικόνα μας είναι μικρή τις δίνουμε εμείς διαστάσεις
if (cloudReady) {
ctx.drawImage( cloudImage, cloud.x, cloud.y, 430,330);
}// Επειδή η εικόνα μας είναι μικρή τις δίνουμε εμείς διαστάσεις
```

```
soundEfx = document.getElementById("soundEfx");

soundBfx = document.getElementById("soundBfx");

soundClo = document.getElementById("soundClo");
```

Επίσης βλέπουμε το score αλλά και τον χρόνο του παιχνιδιού για να εμφανίζονται κατά την διάρκεια που παίζει ο χρήστης. Επίσης βλέπουμε τις διαστάσεις, τα χρώματα και τη γραμματοσειρά που επιλέξαμε.

```
// Εμφάνιση του παιχνιδιού μέσω μηνύματος όταν τελειώσει ο χρονομετρητής

if(finished==false){

ctx.font = "28px Curlz MT";

ctx.fillStyle = "rgb(255, 104, 0)";

ctx.fillText("Candies: " + sweetsCaught,312, 32);

ctx.font = "25px maiandra GD";

ctx.fillStyle = "rgb(255, 284, 0)";

ctx.fillText("life: " + bombsCaught, 192, 32);

ctx.font = "25px maiandra GD";

ctx.fillStyle = "rgb(150, 150, 250)";

ctx.fillText("Chronometer: " + count, 480, 32);

}



if(finished==true){

ctx.fillStyle = "rgb(255, 0, 0)";
```

```
ctx.font = "60px stencil";

ctx.fillText("GAME OVER", 190, 200);

bombsImage.src = onload;

bombsImage = play();

boomsImage.src = onload;

boomsImage =play();

heroImage.src = onload;

heroImage = play();

sweetImage.src = onload;

sweetImage =play();

karamelImage.src = onload;

karamelImage =play();

}

if(count==false){

ctx.font = "60px stencil";

ctx.fillStyle = "rgb(255, 0, 0)";

ctx.fillText("You WIN", 190, 200);

ctx.font = "24px maiandra GD";

ctx.fillText("You found: "+ sweetsCaught, 400, 390);

bombsImage.src = onload;

bombsImage = play();

boomsImage.src = onload;
```

```
boomsImage = play();

heroImage.src = onload;

heroImage = play();

sweetImage.src = onload;

sweetImage = play();

karamelImage.src = onload;

karamelImage =play();
```

Εδώ βλέπουμε τον χρόνο που βάλαμε στη μεταβλητή count, την μεταβλητή finished, και μέσα στην function που βάλαμε τις εικόνες μας για να εξαφανίζονται κατά τη λήξη του παιχνιδιού, ώστε να σταματήσει ο χρήστης να παίζει.

```
var count =60; // Πόσα δευτερόλεπτα διαρκεί το παιχνίδι - προεπιλογή 30

var finished = false;

var counter =function(){

count=count-1;// Αντίστροφη μέτρηση κατά -1 δευτερόλεπτο

if (count <= 0) // Όταν ο χρόνος θα είναι 0

{

// σταματάει το χρονόμετρο

clearInterval(counter);

// Το παιχνίδι έχει τελειώσει

finished = true;// Ισχυει

count=0; // Οποτεχρονος 0

// Κρυμμένοι οι χαρακτήρες
```

```
karamelReady=false; // Η εικόνα γίνεται αόρατη γιατί τελείωσε το παιχνίδι
heroReady=false;
sweetReady=false;
bombsReady=false;
boomsReady=false;
surpriseReady=false;
cloudReady=false;
finished = false;
count=0;
}
if (bombsCaught <= 0) // Αν φτάσει η βόμβα στο 0 τότε τελειώνει το παιχνίδι.
{
/ Σταματήστε το χρονοδιακόπτη
clearInterval(counter);
// Ορίστε το παιχνίδι όταν τελειώνει
finished = true;
bombsCaught=0;
count=0;
// Κρυμμένοιοιχαρακτήρες
karamelReady=false;
heroReady=false;
sweetReady=false;
```

```
bombsReady=false;

boomsReady=false;

surpriseReady=false;

cloudReady=false;

finished = true;

count=true;

}



}
```

Τέλος για να ολοκληρωθεί το παιχνίδι δώσαμε το σωστό χρόνο ώστε να κυλάνε τα δευτερόλεπτα ομαλά, παραδείγματός χάρη να μην από το 58 έχει να πάει στο 40 το χρονόμετρο.

Επίσης στο τέλος κλείσαμε τις λειτουργίες της reset και της main για να ξεκινήσει η περιπέτεια μας.

Το παιχνίδι προσπαθήσαμε να είναι ισορροπημένο και σωστά δομημένο ώστε να λειτουργεί ομαλά για να μπορεί ο χρήστης να παίζει με αυτό ευχάριστα.

```
// Το χρονόμετρο είναι κάθε δευτερόλεπτο(1000ms)

setInterval(counter, 1000);

setInterval(counter, 1000);

// Οκύριοςβρόχοςπαιχνιδιών

var main = function () {

// Εκτελέστε τη λειτουργία ενημέρωση

update(0.02);
```

```
// Εκτελέστε τη λειτουργία render

render();

// Ζητήστε να το κάνετε ξανά ASAP

requestAnimationFrame(main);

};

reset();

init();

main();
```

5.4 Συμπεράσματα

Προσπαθήσαμε να κατανοήσουμε τον κώδικα σε όλη του την μορφή, το οποίο για μας ήταν δύσβατο. Με αρκετή προσπάθεια και επιμονή καταφέραμε να υλοποιήσουμε κάτι δημιουργικό και έξυπνο, μάθαμε βασικά πράγματα όπως το να προγραμματίζεις συνειδητά, να διαχειρίζεσαι τον όγκο του συνολικού αρχείου και να μην γράφουμε κώδικα ανεξέλεγκτα. Με αυτή τη πτυχιακή εργασία αντλήσαμε δύναμη ώστε να αναλάβουμε ακόμα μεγαλύτερα projects αλλά και αγάπη για το προγραμματισμό διαδικτύου, που για μας είναι κάτι σημαντικό. Μέσα από μια μικρή ιδέα λοιπόν καταφέραμε κάτι σημαντικό και μεγάλο.

5.5 Ηλεκτρονικές πηγές και Βιβλιογραφία

Για το παιχνίδι χρειάστηκε:

- http://kata.coderdojo.com/wiki/Simple_Canvas_Game
- http://www.uiupdates.com/create-a-moving-background-for-games-in-canvas/
- https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Drawing_shapes
- https://lamosty.com/2013/02/04/create-simple-html5-canvas-game-using-pure-javascript/
- https://github.com/Techbot/template008
- http://www.wavsource.com/sfx/sfx3.htm
- http://codemahal.com/javascript-and-html5-canvas-game-tutorial-code/
- https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D_Breakout_game_pure_JavaScript
- http://www.w3schools.com
- http://silveiraneto.net/2011/06/02/simple-html5-animation-clouds-over-background/

- http://www.onlywebpro.com/2012/03/10/html5-game-development-adding-sound-effects/