# HTML5+JavaScript

## Δημιουργώντας παιχνίδια
## Ο εύκολος τρόπος


Παπαστεργίου Κωνσταντίνος

# HTML5+JavaSctipt
# Δημιουργώντας παιχνίδια – Ο εύκολος τρόπος

ISBN: 978-618-00-1734-2

Κωνσταντίνος Παπαστεργίου

2017

# ΠΕΡΙΕΧΟΜΕΝΑ

# ΕΙΣΑΓΩΓΗ

Από μικρός ήθελα να δημιουργώ παιχνίδια για υπολογιστές. Με την εμφάνιση των έξυπνων κινητών τηλεφώνων η ιδέα για δημιουργία παιχνιδιών και εφαρμογών έγινε ακόμα πιο γοητευτική.
Πολλές γλώσσες προγραμματισμού υποστηρίζουν την ιδέα αυτή, μα πάντα ήθελα να γράψω ένα βιβλίο για νέους προγραμματιστές με αναφορά μια απλή γλώσσα. Μια γλώσσα που θα ήταν εύκολο να μάθουν και να αγαπήσουν.
Ο προγραμματισμός όμως παιχνιδιών με τη χρήση κώδικα είναι γενικά σύνθετος και πολλοί νέοι υποψήφιοι προγραμματιστές αποθαρρύνονται. Έπρεπε να βρεθεί η κατάλληλη γλώσσα ώστε να μπορέσω να γράψω ένα βιβλίο για τη δημιουργία παιχνιδιών με κώδικα που θα ενθαρρύνει τους νέους προγραμματιστές.
Η ευκαιρία δόθηκε με την εμφάνιση της HTML5 σε συνδυασμό με την ύπαρξη της JavaScript.
Η HTML5 είναι μια γλώσσα προγραμματισμού για τον Παγκόσμιο Ιστό, η οποία έρχεται να δημιουργήσει μια νέα εποχή στο χώρο των παιχνιδιών και των εφαρμογών.
Η JavaScript είναι μια γνωστή και απλή γλώσσα προγραμματισμού η οποία θα χρησιμοποιηθεί για να ελέγχει την δημιουργία γραφικών με την χρήση της HTML5.
Προσπάθησα και νομίζω τα έχω καταφέρει να γράψω ένα βιβλίο που θα βάλει τον αναγνώστη στην λογική του προγραμματισμού απλά και φιλικά

**Με λίγα λόγια η HTML5 θα δημιουργεί και η JavaScript θα ελέγχει.**

## Τι θα χρειαστούμε

Έναν υπολογιστή με Microsoft Windows.
Έναν editor για προγραμματισμό. Προτείνω το notepad++.
Ένα φυλλομετρητή(web browser) τελευταία έκδοση, που να υποστηρίζει HTML5.
Google chrome, Internet explorer,Mozzila Firefox
Ένα πρόγραμμα επεξεργασίας εικόνας:photoplus, photoshop.
Το αρχείο script **requestAnimationFramePolyfill.js** που είναι απαραίτητο. Υπάρχει στη διεύθυνση http://www.edaskalos.gr/book.htmlκαι θα πρέπει να το κατεβάσετε.
Και σίγουρα θα πρέπει να κατεβάσουμε όλα τα προγράμματα που περιέχει το βιβλίο από τη διεύθυνση: http://www.edaskalos.gr/book.html
Θα είναι πολύ χρήσιμη βοήθεια.

Είναι επίσης χρήσιμο και μάλλον αναγκαίο ο αναγνώστης να γνωρίζει τις βασικές αρχές του δομημένου προγραμματισμού και τις βασικές εντολές.
Στο τελευταίο κεφάλαιο του βιβλίου υπάρχει σύντομη περιγραφή των εντολών της JavaScript

## ΞΕΚΙΝΩΝΤΑΣ

Θα χρησιμοποιήσουμε την HTML5 για να δημιουργήσουμε γραφικά και αντικείμενα σε ένα καμβά (canvas) τις διαστάσεις του οποίου εμείς θα ορίσουμε. Θα ορίσουμε πλάτος(width) και ύψος(height).
Για παράδειγμα :

```
<canvas width="480" height="320" ></canvas>
```

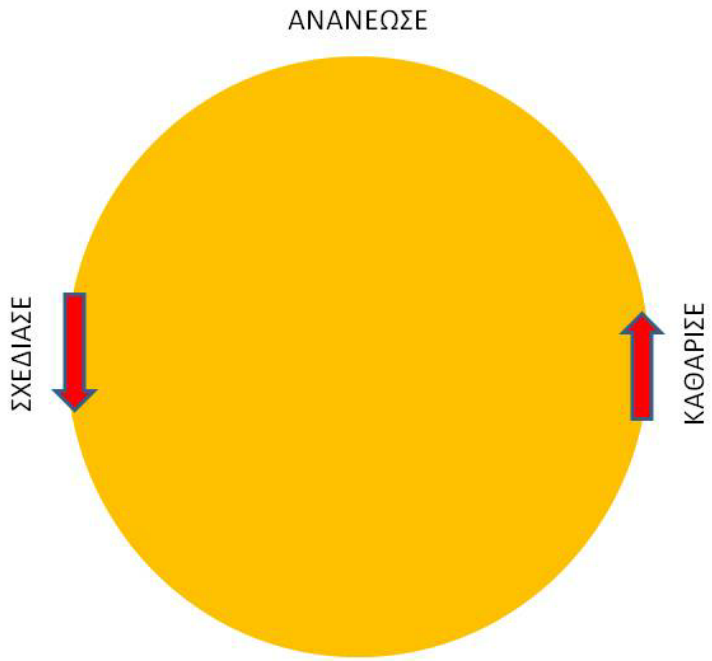
**ΚΑΜΒΑΣ**

ΕΔΩ ΜΕΣΑ ΘΑ ΕΚΤΕΛΕΣΤΕΙ Η ΕΦΑΡΜΟΓΗ

ΠΛΑΤΟΣ:480

ΥΨΟΣ:320

Η κεντρική ιδέα στην οποία θα κινηθεί ο τρόπος που θα προγραμματίσουμε θα είναι η εξής:




Δηλαδή τα γραφικά θα σχεδιάζονται μετά θα διαγράφονται και μετά θα ξανασχεδιάζονται ανανεωμένα σε ελάχιστο χρόνο ώστε να μην το αντιλαμβάνεται το ανθρώπινο μάτι και έτσι θα δημιουργείται η αίσθηση της συνέχειας ή της κίνησης.

Το έτοιμο αρχείο JavaScript **requestAnimationFramePolyfill.js** θα μας βοηθήσει στο να υπάρχει αυτή η αίσθηση της συνέχειας.

***Αν δεν θέλουμε την βοήθεια του έτοιμου script, θα δημιουργήσουμε συνθήκες επανάληψης και ανανέωσης με τη χρήση του παρακάτω κώδικα:***

```
play();
// Συνάρτηση επανάληψης
function play(){

setTimeout(play, 33); //Ανανέωση καμβά κάθε 0,33 δευτερόλεπτα
}
```

Για να έχουμε όμως πιο καλή και ομαλή κίνηση, προτιμότερο είναι να χρησιμοποιήσουμε το έτοιμο script.
Εδώ φαίνεται ο κώδικας του αρχείου **requestAnimationFramePolyfill.js** που εσείς θα χρησιμοποιείται έτοιμο ως αρχείο.

```
(function() {
var lastTime = 0;
var vendors = ['ms', 'moz', 'webkit', 'o'];
for(var x = 0; x < vendors.length && !window.requestAnimationFrame; ++x)
{
 window.requestAnimationFrame = window[vendors[x]+'RequestAnimationFrame'];
 window.cancelAnimationFrame = window[vendors[x]+'CancelAnimationFrame']
 window[vendors[x]+'CancelRequestAnimationFrame'];
}

if (!window.requestAnimationFrame)
 window.requestAnimationFrame = function(callback, element) {
 var currTime = new Date().getTime();
 var timeToCall = Math.max(0, 16 - (currTime - lastTime));
 var id = window.setTimeout(function() { callback(currTime + timeToCall); },
timeToCall);
 lastTime = currTime + timeToCall;
 return id;
};

if (!window.cancelAnimationFrame)
{
 window.cancelAnimationFrame = function(id)
 {
  clearTimeout(id);
 };
}

}());
```

Ο προκαθορισμένος κώδικας στον οποίο θα κινούμαστε για το σχεδιασμό παιχνιδιών θα είναι ο παρακάτω:

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
```

```
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

update();


function update() {
requestAnimationFrame(update, canvas);
ctx.fillStyle= "black";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ΕΝΤΟΛΕΣ

}

</script>
</html>
```

## ***ΑΝΑΛΥΤΙΚΟΤΕΡΑ***

**<!doctype html>**
Κεφαλίδα δημιουργίας HTML5 εγγράφου.
**<meta charset="utf-8" />**
Κωδικοποίηση γραμματοσειράς
**<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">**
Προσαρμόζει την εφαρμογή σε διαφορετικές οθόνες στα κινητά.
**<title>My game</title>**
Τίτλος εφαρμογής
** **
Κεντράρει την εφαρμογή
**<canvas width="480" height="320" ></canvas>**
Δημιουργία καμβά συγκεκριμένων διαστάσεων
**<script src="requestAnimationFramePolyfill.js"></script>**
Χρησιμοποιούμε ένα έτοιμο JavaScript που μας βοηθάει να ανανεώνει την οθόνη ομαλά.
**<script>**
Δημιουργούμε το δικό μας script

**var canvas = document.querySelector("canvas");**
**var ctx = canvas.getContext("2d");**
Δημιουργούμε την μεταβλητή ctx , η οποία θα χρησιμοποιείται για σχεδιασμό γραφικών στον καμβά. Αντί για ctx θα μπορούσαμε να χρησιμοποιήσουμε οποιαδήποτε ονομασία.

**update();**

Καλώ τη συνάρτηση η οποία θα ανανεώνει την οθόνη
**function update() {**
Δημιουργία συνάρτησης update
**requestAnimationFrame(update, canvas);**
Καλώ το έτοιμο JavaScript requestAnimationFrame να ανανεώσει τον καμβά.
**ctx.fillStyle= "black";**
Ο καμβάς μου έχει μαύρο χρώμα(Μπορεί να έχει ότι χρώμα θέλω)
**ctx.fillRect(0, 0, canvas.width, canvas.height);**
Σβήνω ότι υπάρχει στον καμβά

ΕΝΤΟΛΕΣ

**}**
Κλείνει η συνάρτηση
**</script>**
Κλείνει το script
**</html>**
Κλείνει το έγγραφο HTML5
*Το αρχείο που έχουμε δημιουργήσει θα το αποθηκεύσουμε με ως index.html ή με όποιο όνομα θέλουμε έχοντας πάντα την κατάληξη .html. Δίπλα στο αρχείο δεν ξεχνάμε να βρίσκεται το αρχείο requestAnimationFramePolyfill.js το οποίο είναι απαραίτητο.*
*Καλό είναι να βρίσκονται όλα αυτά μέσα σε ένα φάκελο.*

***Αν δεν θέλουμε να χρησιμοποιήσουμε το requestAnimationFramePolyfill.js θα έπρεπε να δημιουργήσουμε μόνοι μας συνθήκες επανάληψης και ανανέωσης του καμβά όπως παρακάτω****:*

```
<!doctype html>
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title> loop</title>
<center>
<canvas width="480" height="320" ></canvas>
<script>

var canvas = document.querySelector("canvas");
```

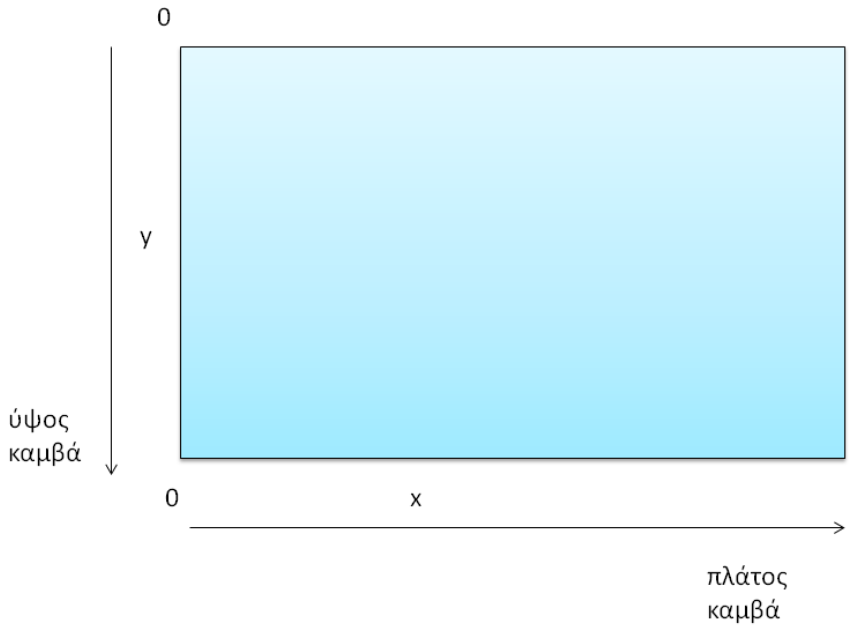
```
var ctx = canvas.getContext("2d");
play();

function play(){

ctx.fillStyle= "black";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ΕΝΤΟΛΕΣ

setTimeout(play, 33);
}

</script>
</html>
```

# 1.Δημιουργία Γραφικών

## *1.1 Δημιουργία παραλληλόγραμμου*

Για ναν κατασκευάσουμε ένα παραλληλόγραμμο ή τετράγωνο χρησιμοποιούμε τις παρακάτω εντολές:

```
ctx.fillStyle= "Χρώμα σχήματος";
ctx.fillRect(x,y,πλάτος,ύψος);
```



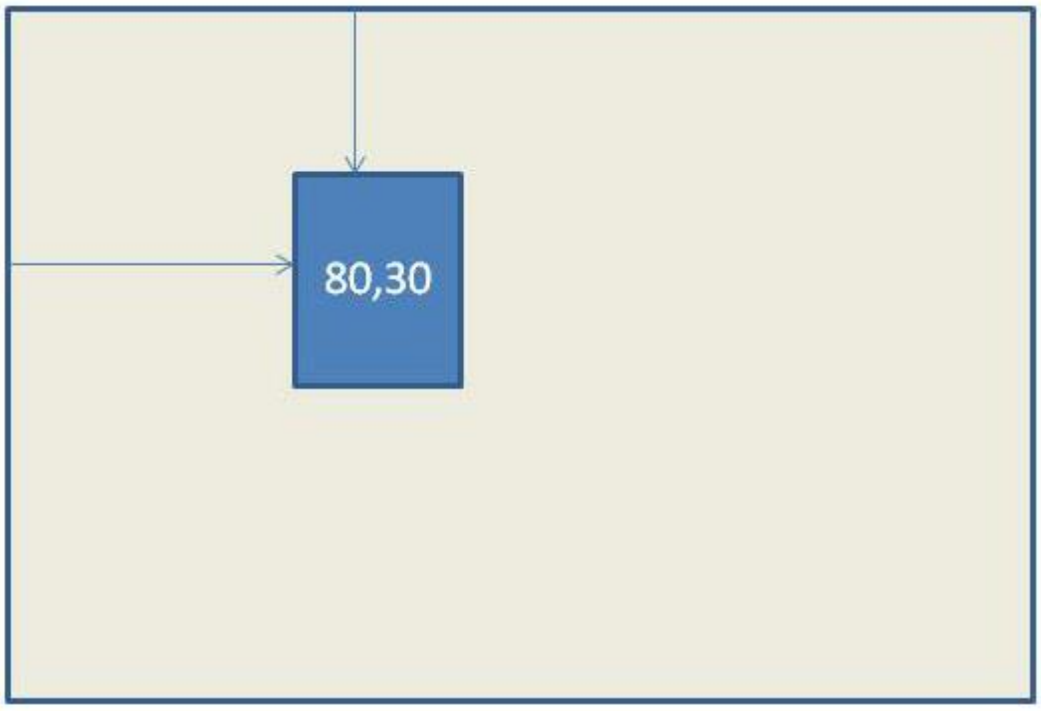

Στο x τοποθετούμε από ποιο σημείο του άξονα x στον καμβά θα ξεκινάει το παραλληλόγραμμο.
Στο y τοποθετούμε από ποιο σημείο του άξονα y στον καμβά θα ξεκινάει το παραλληλόγραμμο.

Όπου πλάτος και ύψος τοποθετούμε τις διαστάσεις pixels που θέλουμε να έχει το παραλληλόγραμμο

Για παράδειγμα:

```
ctx.fillStyle= "blue";
ctx.fillRect(20,30,50,20);
```


```
ctx.fillStyle= "blue";
ctx.fillRect(80,30,50,70);
```




Χρησιμοποίησα το χρώμα μπλε , αλλά η HTML έχει μεγάλη ποικιλία χρωμάτων που απεικονίζονται με την παρακάτω κωδικοποίηση.

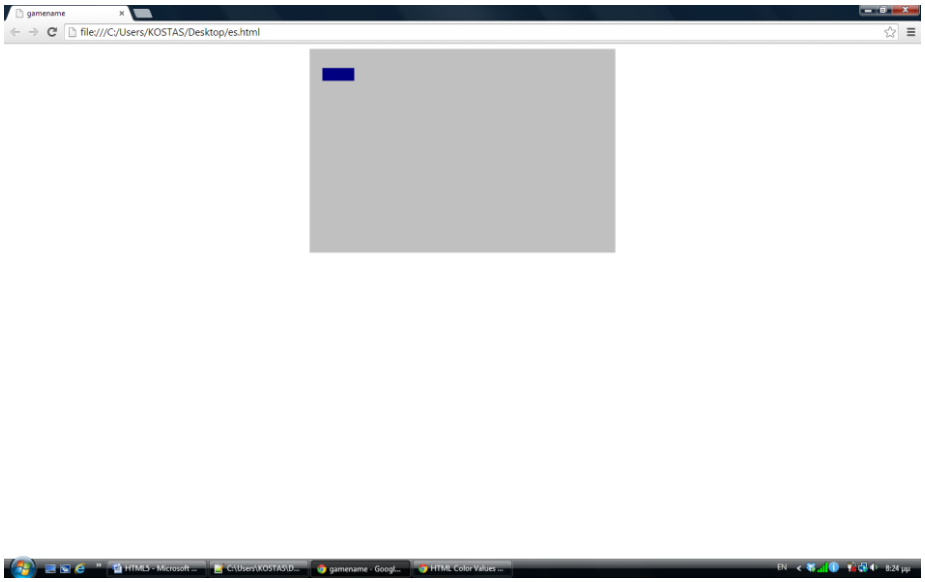
ctx.fillStyle= "#000080”;

Μπορείτε να δείτε κάποιες τιμές χρώματα στη παρακάτω σελίδα:
http://www.w3schools.com/html/html_colorvalues.asp

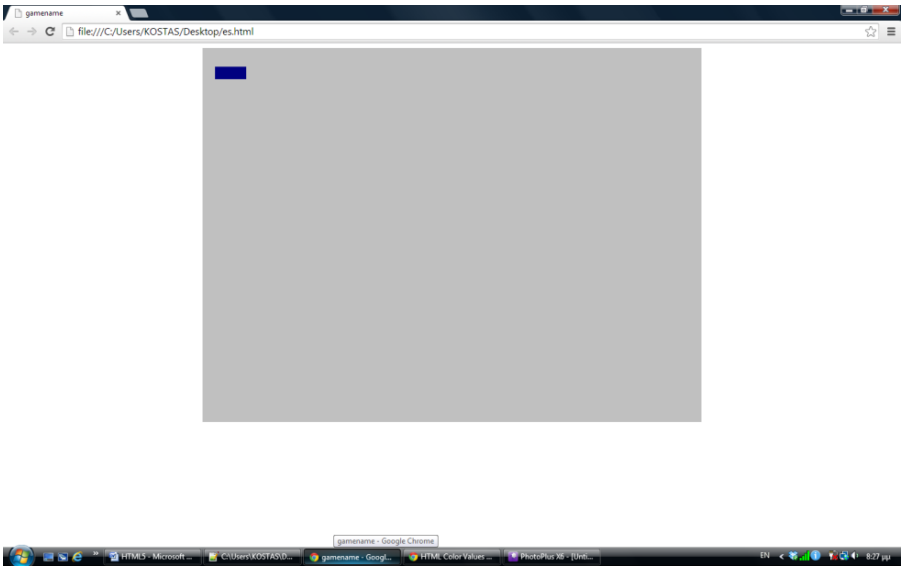
Ας δούμε τη πρώτη εφαρμογή για τη δημιουργία παραλληλόγραμμου:

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

update();


function update() {
requestAnimationFrame(update, canvas);
ctx.fillStyle= "#C0C0C0";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.fillStyle= "#000080";
ctx.fillRect(20,30,50,20);


}

</script>
</html>
```

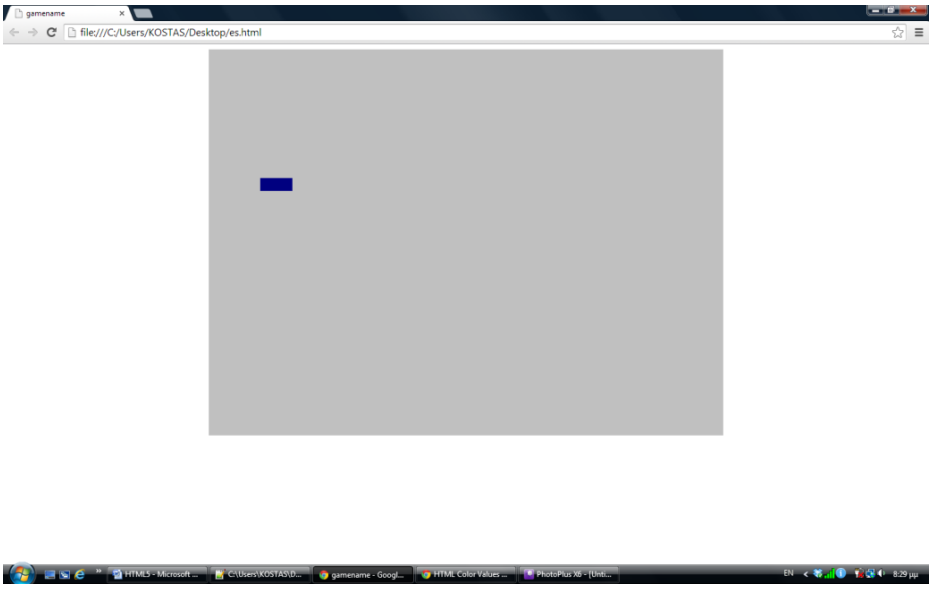
Αποθηκεύουμε ως index.html και
το αποτέλεσμα θα είναι το παρακάτω:
*!!Δεν ξεχνάμε το αρχείο requestAnimationFramePolyfill.js να είναι δίπλα στο αρχείο index.html*



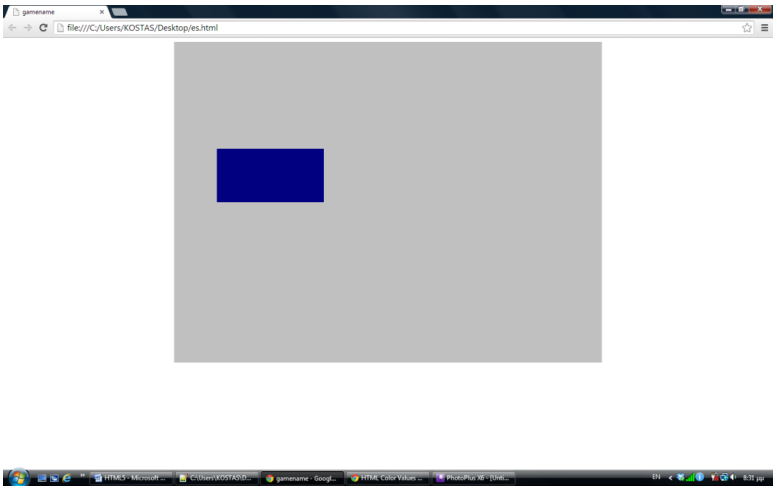
Αν αλλάξουμε το μέγεθος του καμβά
<canvas width="800" height="600" ></canvas>



Αν αλλάξουμε τη θέση του παραλληλογράμμου
ctx.fillRect(80,200,50,20);



Άν αλλάξουμε το μέγεθος του παραλληλογράμμου
ctx.fillRect(80,200,200,100);



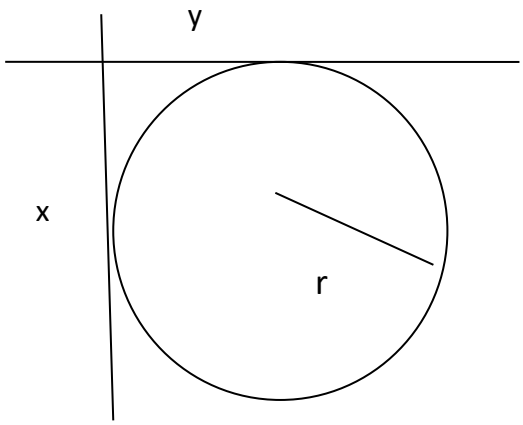
Δοκίμασε να αλλάξεις το μέγεθος του καμβά, τη θέση και το μέγεθος του παραλληλόγραμμου, καθώς και τα χρώματα του καμβά και του παραλληλόγραμμου.

**ΑΣΚΗΣΗ:** Δοκίμασε να δημιουργήσεις ένα κίτρινο τετράγωνο στη θέση x=50 και y=80.

## *1.2 Δημιουργία κύκλου*

Η δημιουργία κύκλου σε ένα καμβά είναι μια διαδικασία εύκολη αλλά λίγο πιο σύνθετη.

```
ctx.beginPath();
ctx.fillStyle= "χρώμα κύκλου";
ctx.arc(30,100,10,2*Math.PI,false);
ctx.fill();
ctx.closePath();

ctx.beginPath();
//  Ανοίγει η διαδικασία δημιουργίας κύκλου.
ctx.fillStyle= "χρώμα κύκλου";
//  Χρώμα κύκλου
ctx.arc(x,y,r,2*Math.PI,false);
//  συντεταγμένες κύκλου x ,y και ακτίνα κύκλου r
ctx.fill();
//  γέμισμα κύκλου με χρώμα
ctx.closePath();
// κλείσιμο διαδικασίας
```




Στη περίπτωση που θέλουμε ημικύκλιο :

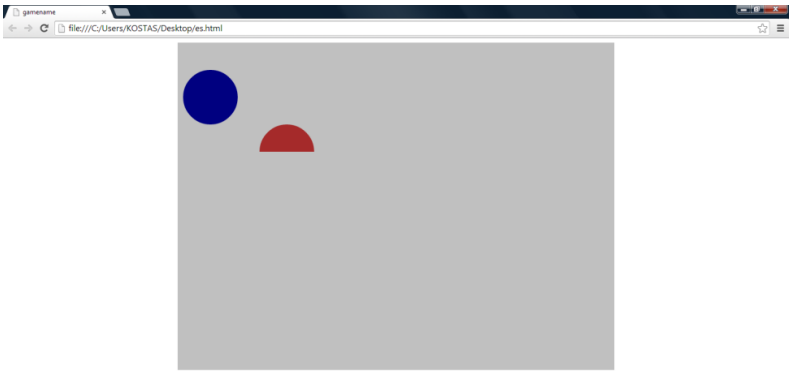
```
ctx.arc(x,y,r,Math.PI,false);
```

Ας δούμε μια εφαρμογή :

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
```

```
<title>gamename</title>
<center>
<canvas width="800" height="600" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

update();

function update() {
requestAnimationFrame(update, canvas);
ctx.fillStyle= "#C0C0C0";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.beginPath();
ctx.fillStyle= "#000080";
ctx.arc(60,100,50,2*Math.PI,false);
ctx.fill();
ctx.closePath();

ctx.beginPath();
ctx.fillStyle= "#A52A2A";
ctx.arc(200,200,50,Math.PI,false);
ctx.fill();
ctx.closePath();

}

</script>
</html>
```

Το αποτέλεσμα θα είναι το παρακάτω:



**ΑΣΚΗΣΗ:** Δοκίμασε να δημιουργήσεις έναν πράσινο κύκλο στη θέση x=150 και y=150 , με ακτίνα 60, και ένα κίτρινο ημικύκλιο στη θέση x=200 και y=200 με ακτίνα =30.

## *1.3 Δημιουργία γραμμής*

Η δημιουργία μιας γραμμής σε ένα καμβά είναι μια διαδικασία πολύ εύκολη.

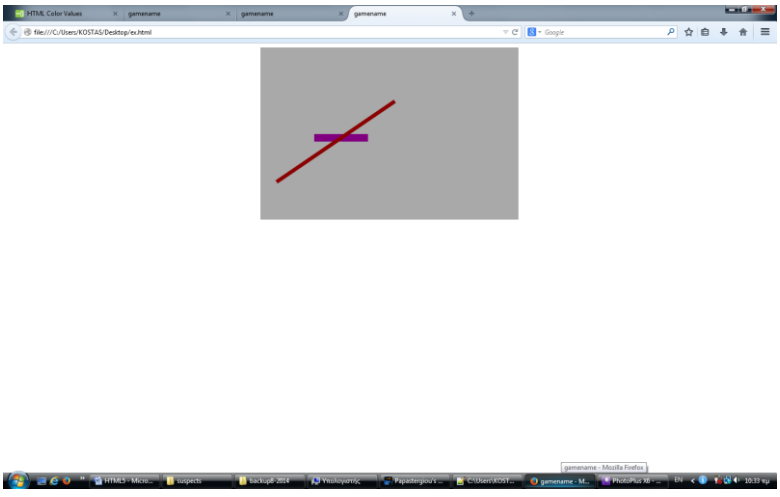
```
ctx.beginPath();
//Ανοίγει η διαδικασία δημιουργίας γραμμής
ctx.strokeStyle="#800080";
// Επιλογή χρώματος γραμμής
ctx.lineWidth=14;
// Επιλογή πάχους γραμμής
ctx.moveTo(100,168);
// Το σημείο  x,y  που ξεκινάει η γραμμή
ctx.lineTo(200,168);
// Το σημείο  x,y  που σταματάει η γραμμή
ctx.stroke();
// Χρματίζεται η γραμμή με το χρώμα που έχει επιλεγεί αραπάνω
ctx.closePath();
//Κλείνει η διαδικασία
```

Ας δούμε μια εφαρμογή :

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

update();
```

```
function update() {
requestAnimationFrame(update, canvas);
ctx.fillStyle= "#A9A9A9";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.beginPath();
ctx.strokeStyle="#800080";
ctx.lineWidth=14;
ctx.moveTo(100,168);
ctx.lineTo(200,168);
ctx.stroke();
ctx.closePath();

ctx.beginPath();
ctx.strokeStyle="#8B0000";
ctx.lineWidth=7;
ctx.moveTo(250,100);
ctx.lineTo(30,250);
ctx.stroke();
ctx.closePath();


}

</script>
</html>
```

Το αποτέλεσμα θα είναι το παρακάτω:



**ΑΣΚΗΣΗ:** Δοκίμασε να δημιουργήσεις μια κίτρινη γραμμή με πάχος 5 που να ξεκινάει από το σημείο (x=100,y=100) και να καταλήγει στο σημείο (x=250,y=250)

## *1.4 Δημιουργία – Εισαγωγή κειμένου*

Η εισαγωγή κειμένου σε έναν καμβά γίνεται με έναν σχετικά απλό τρόπο. Δηλώνουμε το μέγεθος και το είδος της γραμματοσειράς, το χρώμα της γραμματοσειράς και στο σημείο το καμβά που θα εμφανιστεί.

```
ctx.font = "24px Verdana";
// Μέγεθος και είδος γραμματοσειράς
ctx.fillStyle= "#FFFFCC";
// Χρώμα γραμματοσειράς
ctx.fillText("κείμενο”,200,220);
// Εμφάνιση του κειμένου στο σημείο του καμβά x,y
```

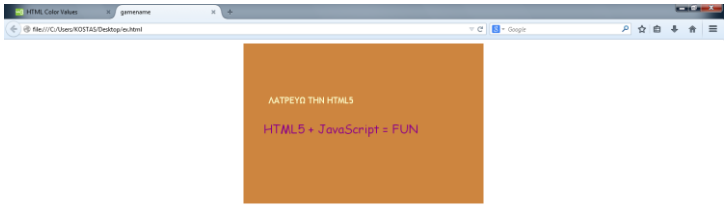
Παρακάτω βλέπουμε γνωστές γραμματοσειρές.

| | | |
|---|---|---|
| Andale Mono | Arial | Arial Bold |
| Arial Italic | Arial Bold Italic | Arial Black |
| Comic Sans MS | Comic Sans MS Bold | Courier New |
| Courier New Bold | Courier New Italic | Courier New Bold Italic |
| Georgia | Georgia Bold | Georgia Italic |
| Georgia Bold Italic | Impact | Lucida Console |
| Lucida Sans Unicode | Marlett | Minion Web |
| Symbol | Times New Roman | Times New Roman Bold |
| Times New Roman Italic | Times New Roman Bold Italic | Tahoma |
| Trebuchet MS | Trebuchet MS Bold | Trebuchet MS Italic |
| Trebuchet MS Bold Italic | Verdana | Verdana Bold |
| Verdana Italic | Verdana Bold Italic | Webdings |

Ας δούμε ένα παράδειγμα:

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

update();

function update() {
requestAnimationFrame(update, canvas);
ctx.fillStyle= "#CD853F";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.font = "18px Trebuchet MS";
ctx.fillStyle= "#FFFFCC";
ctx.fillText("ΛΑΤΡΕΥΩ ΤΗΝ HTML5" ,50,120);
ctx.font = "24px Comic Sans MS";
ctx.fillStyle= "#8B008B";
ctx.fillText("HTML5 + JavaScript = FUN" ,40,180);

}

</script>
</html>
```

Το αποτέλεσμα θα είναι το παρακάτω:




# *1.5 Εισαγωγή εικόνας*

Η HTML5 με την βοήθεια της JavaScript μπορεί πολύ εύκολα να αξιοποιήσει τις δυνατότητες μιας έτοιμης εικόνας. Καλό είναι η εικόνα να έχει την μορφή αρχείου png. Μπορείτε να μετατρέψετε οποιαδήποτε εικόνα σε μορφή png με τη χρήση προγράμματος επεξεργασίας εικόνων.
Ο φάκελος img με τις εικόνες που θα χρησιμοποιήσω πρέπει να βρίσκεται δίπλα στο βασικό αρχείο Index.html και στο αρχείο requestAnimationFramePolyfill.js.

Χρησιμοποιούμε απλές εντολές JavaScript για να δηλώσουμε την φωτογραφία:

```
var image1=new Image();
// Η μεταβλητή image1 είναι μια νέα φωτογραφία
Image1.src="img/pic.png";
// Η θέση της φωτογραφίας είναι στον φάκελο img και έχει ονομασία pic.png
```

Χρησιμοποιούμε εντολές HTML5 για να εμφανίσουμε την φωτογραφία στο μέγεθος που θέλουμε και στην θέση που θέλουμε.

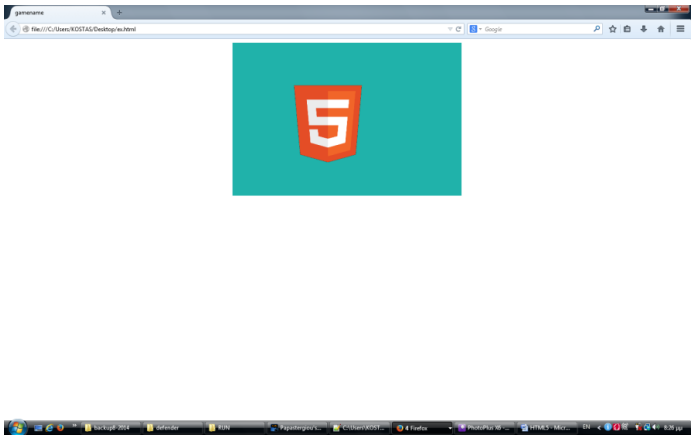
```
ctx.drawImage(image1,θέσηx,θέσηy,πλάτος,ύψος);
```

Ας δούμε ένα παράδειγμα:

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var image1=new Image();
image1.src="img/pic.png";

update();

function update() {
requestAnimationFrame(update, canvas);
ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.drawImage(image1,100,50,200,200);
```

```
}
</script>
</html>
```

Το αποτέλεσμα θα είναι:



# 2. Κίνηση Γραφικών

Η κίνησης γραφικών είναι μια διαδικασία που η HTML5 μαζί με τη JavaScript την κάνουν να φαίνεται απλή.
Θα δούμε αναλυτικά πως μπορούμε να δημιουργήσουμε παιχνίδια και εφαρμογές κίνησης , χρησιμοποιώντας κατάλληλες εντολές και δημιουργώντας απλούς αλγόριθμους.

Βασικό ρόλο στην κίνηση σχημάτων η εικόνων παίζουν τα αντικείμενα(Objects).
Πρέπει να δημιουργήσουμε ένα βασικό αντικείμενο, το οποίο θα το χρησιμοποιούμε για να παράγουμε άλλα αντικείμενα
Η γενική σύνταξη ενός αντικειμένου στην JavaScript είναι:

```
var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};
```

Δηλαδή δηλώνουμε το βασικό μας αντικείμενο spriteObject κατά τον παραπάνω τρόπο, δίνοντας του βασικές ιδιότητες θέσηx, θέσηy, πλάτος, ύψος.
Από το βασικό μας αντικείμενο θα δημιουργούμε άλλα αντικείμενα βάζοντας νέες ιδιότητες αν χρειάζεται.
Για παράδειγμα:

```
var shape = Object.create(spriteObject);
shape.x = 20;
shape.y = 20;
shape.width=150;
shape.height=60;
shape.vx=0;
```

Δηλαδή έχω δημιουργήσει ένα νέο αντικείμενο shape που είναι παράγωγο του αντικειμένου spriteObject.
Απλά βάζω νέες ιδιότητες, όπως την ταχύτητα vx.
Όσο σύνθετο φαίνεται στην αρχή τόσο απλό θα φανεί αργότερα κατά την χρήση του σε εφαρμογές.

## *2.1 Αυτόματη κίνηση σχήματος.*

**Ας δούμε μια απλή εφαρμογή αυτόματης κίνησης ενός σχήματος.**

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");
// Βασικό αντικείμενο
var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};
// Δημιουργία αντικειμένου Shape
var shape = Object.create(spriteObject);
shape.x = 20;
shape.y = 20;
shape.width=150;
shape.height=60;
shape.vx=1;

update();

function update() {
requestAnimationFrame(update, canvas);
// Η τιμή του shape.x θα αυξάνει κατά 1 σε κάθε ανανέωση
shape.x=shape.x+shape.vx;
ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);
// Σχεδίαση παραλληλόγραμμου,, το οποίο θα φαίνεται ότι θα κινείται λόγω της
αλλαγής τιμής του shape.x

ctx.fillStyle= "#000080";
ctx.fillRect(shape.x,shape.y,shape.width,shape.height);
```

```
}

</script>
</html>
```

Έχουμε δημιουργήσει ένα αντικείμενο shape με ιδιότητες θέσηx , θέσηy , πλάτος , ύψος και ταχύτητα στον άξονα x, vx=1;

Έχουμε χρησιμοποιήσει τη σχέση shape.x=shape.x+shape.vx;
Επειδή κάθε φορά ανανεώνεται η εικόνα η τιμή shape.x θα αυξάνει κατά 1.
Έτσι κάθε φορά που θα ζωγραφίζεται στον καμβά το παραλληλόγραμμο θα φαίνεται ότι κινείται.

```
ctx.fillStyle= "#000080";
ctx.fillRect(shape.x,shape.y,shape.width,shape.height);
```

*Στον φάκελο LESSON2 υπάρχει το παραπάνω αρχείο με ονομασία lesson2.html το οποίο μπορείτε να δείτε τον κώδικα με κάποιο επεξεργαστή ή να το τρέξετε βλέποντας το αποτέλεσμα.*
*Το αποτέλεσμα θα είναι ένα παραλληλόγραμμο να κινείται προς την δεξιά κατεύθυνση και θα εξαφανιστεί απο τον καμβά όταν φτάσει στα όρια του καμβά.*

**Ας δούμε τώρα, πως το παραλληλόγραμμο όταν φεύγει από τα όρια του καμβά θα ξαναεμφανίζεται στην αρχή του καμβά.**
Οι εντολές της JavaScript που είναι υπεύθυνες για αυτό είναι οι παρακάτω:

```
if(shape.x>canvas.width){
shape.x=0-shape.width;
}
```

Στο φάκελο LESSON2 το αρχείο lesson2b.html εμφανίζει το αποτέλεσμα.

Ας δούμε τώρα κάτι πιο σύνθετο.
Πως το παραλληλόγραμμο θα αναπηδάει από το ένα όριο του καμβά στο άλλο όριο.
Οι εντολές της JavaScript που είναι υπεύθυνες για αυτό είναι οι παρακάτω:

```
if(shape.x+shape.width>canvas.width){
shape.vx=-shape.vx;
}
if(shape.x<0){
shape.vx=-shape.vx;
}
```

*Στο φάκελο LESSON2 το αρχείο lesson2c.html που εμφανίζει το αποτέλεσμα.*

**Ας δούμε τώρα το ίδιο παράδειγμα αλλά με κάθετη κίνηση στον άξονα y. Αυτά που αλλάζουν είναι τα παρακάτω:**

```
var shape = Object.create(spriteObject);
shape.x = 20;
shape.y = 20;
shape.width=150;
shape.height=60;
shape.vy=1;

shape.y=shape.y+shape.vy;

if(shape.y+shape.height>canvas.height){
shape.vy=-shape.vy;
}
if(shape.y<0){
shape.vy=-shape.vy;
}
```

*Στο φάκελο LESSON2 το αρχείο lesson2d.html που εμφανίζει το αποτέλεσμα.*

**Ας δούμε τώρα πως μπορεί να κινηθεί αυτόματα μια φωτογραφία. Είναι ακριβώς η ίδια διαδικασία με κάποιες αλλαγές.**

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};
```

```
var shape = Object.create(spriteObject);
shape.x = 150;
shape.y = 20;
shape.width=100;
shape.height=100;
shape.vy=2;

var image1=new Image();
image1.src="img/ball.png";

update();

function update() {
requestAnimationFrame(update, canvas);

shape.y=shape.y+shape.vy;

if(shape.y+shape.height>canvas.height){
shape.vy=-shape.vy;
}

if(shape.y<0){
shape.vy=-shape.vy;
}

ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.drawImage(image1,shape.x,shape.y,shape.width,shape.height);

}

</script>
</html>
```

*Στο φάκελο LESSON2 το αρχείο lesson2e.html εμφανίζει το αποτέλεσμα.*

# 3.Κίνηση σχήματος ή εικόνας με πληκτρολόγιο

Τώρα θα δούμε πως μπορούμε να κινήσουμε ένα σχήμα ή μία εικόνα με τα πλήκτρα του πληκτρολογίου.

Χρησιμοποιούμε τις 2 εντολές(listeners)
//Όταν πατήσω ένα πλήκτρο
window.addEventListener("keydown", function(event)
//Όταν αφήσω ένα πλήκτρο
window.addEventListener("keyup", function(event)

Ας δούμε ένα παράδειγμα κίνησης ενός σχήματος με τα βελάκια του πληκτρολογίου

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};

var shape = Object.create(spriteObject);
shape.x = 20;
shape.y = 20;
shape.width=50;
shape.height=50;
shape.vx=3
shape.vy=3;

//Κωδικοποίηση πλήκτρου βέλους πάνω
var UP = 38;
```

```
//Κωδικοποίηση πλήκτρου βέλους κάτω
var DOWN = 40;
//Κωδικοποίηση πλήκτρου βέλους δεξιά
var RIGHT = 39;
//Κωδικοποίηση πλήκτρου βέλους αριστερά
var LEFT = 37;

//Για κάθε κίνηση πάνω,κάτω,αριστερά,δεξιά υπάρχει μια μεταβλητή, η οποία
ξεκινάει με τιμή false
var moveUp = false;
var moveDown = false;
var moveLeft = false;
var moveRight = false;

// keyboard listeners
window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {
    case UP:  //Όταν πατηθεί το πλήκτρο UP
      moveUp = true;
      break;

    case DOWN:  //Όταν πατηθεί το πλήκτρο DOWN
      moveDown = true;
      break;

    case LEFT:  //Όταν πατηθεί το πλήκτρο LEFT
      moveLeft = true;
      break;

    case RIGHT:  //Όταν πατηθεί το πλήκτρο RIGHT
      moveRight = true;
       break;
 }
}, false);

window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {
     case UP:   //Όταν το πλήκτρο UP δεν είναι πατημένο
      moveUp = false;
       break;

    case DOWN:  ////Όταν το πλήκτρο DOWN δεν είναι πατημένο
```

```
      moveDown = false;
      break;

     case LEFT:  //Όταν το πλήκτρο LEFT δεν είναι πατημένο
       moveLeft = false;
       break;

     case RIGHT:  ////Όταν το πλήκτρο RIGHT δεν είναι πατημένο
       moveRight = false;
       break;

  }
}, false);

update(); // Τρέξε τη συνάρτηση update

//Συνάρτηση update
function update() {
//Ανανέωση καμβά
requestAnimationFrame(update, canvas);

// Αν πατηθεί το πλήκτρο πάνω και όχι το πλήκτρο κάτω
if(moveUp && !moveDown)
 {
   shape.vy = -3;
 }

// Αν πατηθεί το πλήκτρο κάτω και όχι το πλήκτρο πάνω
 if(moveDown && !moveUp)
 {
   shape.vy = 3;
 }

// Αν πατηθεί το πλήκτρο αριστερά και όχι το πλήκτρο δεξιά
 if(moveLeft && !moveRight)
 {
     shape.vx = -3;
 }

// Αν πατηθεί το πλήκτρο δεξιά και όχι το πλήκτρο αριστερά
 if(moveRight && !moveLeft)
 {
    shape.vx = 3;
 }

// Αν δεν πατηθεί το πλήκτρο πάνω και ούτε το πλήκτρο κάτω
```

```
if(!moveUp && !moveDown)
 {
  shape.vy = 0;
 }

// Αν δεν πατηθεί το πλήκτρο δεξιά και ούτε το πλήκτρο αριστερά
 if(!moveLeft && !moveRight)
 {
  shape.vx = 0;
 }


shape.x=shape.x+shape.vx;
shape.y=shape.y+shape.vy;

//Αν το σχήμα ξεπεράσει τα όρια του Καμβά
if(shape.x+shape.width>canvas.width){
shape.x=canvas.width-shape.width;
}
if(shape.x<0){
shape.x=0;
}
if(shape.y+shape.height>canvas.height){
shape.y=canvas.height-shape.height;
}
if(shape.y<0){
shape.y=0;
}

ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.fillStyle= "#000080";
ctx.fillRect(shape.x,shape.y,shape.width,shape.height);


}

</script>
</html>
```

*Στο φάκελο LESSON3 το αρχείο lesson3.html εμφανίζει το αποτέλεσμα.*
Το σχήμα κινείται μέσα στα όρια του καμβά με τα βελάκια του πληκτρολογίου.

Αν θέλουμε να χρησιμοποιήσουμε και άλλα πλήκτρα, για παράδειγμα SPACE για κάποια άλλη ενέργεια χρησιμοποιούμε την αντίστοιχη κωδικοποίηση σύμφωνα με τον πίνακα που ακουλουθεί.

Στον παρακάτω πίνακα βλέπουμε την κωδικοποίηση όλων των πλήκτρων του πληκτρολογίου, σε περίπτωση που χρειάζεται να χρησιμοποιηθούν σε κάποια εφαρμογή.

| Key | Code | Key | Code | Key | Code |
|---|---|---|---|---|---|
| backspace | 8 | e | 69 | numpad 8 | 104 |
| tab | 9 | f | 70 | numpad 9 | 105 |
| enter | 13 | g | 71 | multiply | 106 |
| shift | 16 | h | 72 | add | 107 |
| ctrl | 17 | i | 73 | subtract | 109 |
| alt | 18 | j | 74 | decimal point | 110 |
| pause/break | 19 | k | 75 | divide | 111 |
| caps lock | 20 | l | 76 | f1 | 112 |
| escape | 27 | m | 77 | f2 | 113 |
| page up | 33 | n | 78 | f3 | 114 |
| page down | 34 | o | 79 | f4 | 115 |
| end | 35 | p | 80 | f5 | 116 |
| home | 36 | q | 81 | f6 | 117 |
| left arrow | 37 | r | 82 | f7 | 118 |
| up arrow | 38 | s | 83 | f8 | 119 |
| right arrow | 39 | t | 84 | f9 | 120 |
| down arrow | 40 | u | 85 | f10 | 121 |
| insert | 45 | v | 86 | f11 | 122 |
| delete | 46 | w | 87 | f12 | 123 |
| 0 | 48 | x | 88 | num lock | 144 |
| 1 | 49 | y | 89 | scroll lock | 145 |
| 2 | 50 | z | 90 | semi-colon | 186 |
| 3 | 51 | left window key | 91 | equal sign | 187 |
| 4 | 52 | right window key | 92 | comma | 188 |
| 5 | 53 | select key | 93 | dash | 189 |
| 6 | 54 | numpad 0 | 96 | period | 190 |
| 7 | 55 | numpad 1 | 97 | forward slash | 191 |
| 8 | 56 | numpad 2 | 98 | grave accent | 192 |
| 9 | 57 | numpad 3 | 99 | open bracket | 219 |
| a | 65 | numpad 4 | 100 | back slash | 220 |
| b | 66 | numpad 5 | 101 | close braket | 221 |
| c | 67 | numpad 6 | 102 | single quote | 222 |
| d | 68 | numpad 7 | 103 | | |

*Στο φάκελο LESSON3 το αρχείο lesson3b.html περιγράφει την κίνηση μιας εικόνας με τα βελάκια, που δεν διαφέρει και ιδιαίτερα από το προηγούμενο παράδειγμα.*

Ας δούμε τώρα πως μπορούμε να κάνουμε ένα σχήμα ή εικόνα να πηδήξει πατώντας το πλήκτρο SPACE.

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};

var shape = Object.create(spriteObject);
shape.x = 20;
shape.y = 280;
shape.width=50;
shape.height=50;
shape.vx=3;
shape.vy=0;
shape.isOnGround=true;
//Ιδιότητα στο αντικείμενο όταν είναι στο έδαφος θα έχει τιμή αληθής
shape.jumpForce= -8;
//Ιδιότητα στο αντικείμενο όταν έχω άλμα ταχύτητα -8

var image1=new Image();
image1.src="img/ball.png";

var RIGHT = 39;
```

```
var LEFT = 37;
var SPACE=32;
//κωδικοποίηση πλήκτρου SPACE

var moveLeft = false;
var moveRight = false;
var jump=false;

window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {

    case LEFT:
     moveLeft = true;
     break;

    case RIGHT:
     moveRight = true;
      break;

    case SPACE:
      jump= true;
      break;
 }
}, false);

window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {

   case LEFT:
    moveLeft = false;
     break;

   case RIGHT:
    moveRight = false;
     break;

    case SPACE:
      jump= false;
     break;
 }
}, false);
```

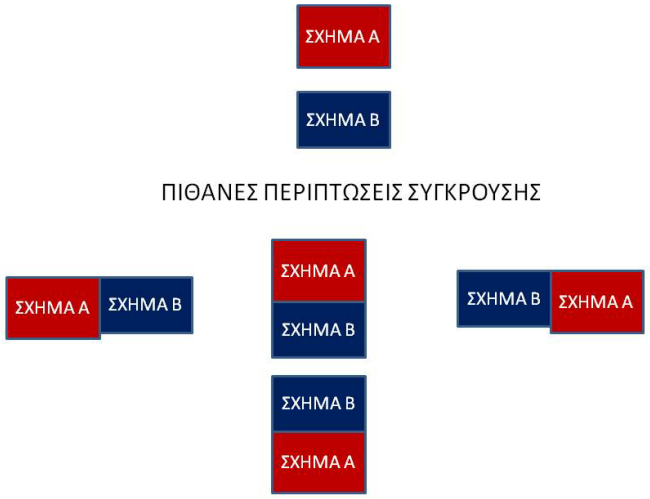
```
update();

function update() {
requestAnimationFrame(update, canvas);

 if(moveLeft && !moveRight)
 {
    shape.vx = -3;
 }

 if(moveRight && !moveLeft)
 {
   shape.vx = 3;
 }

 if(!moveLeft && !moveRight)
 {
  shape.vx = 0;
 shape.gravity = 0.3;
 }

//Όταν πατηθεί το SPACE και το αντικείμενο βρίσκεται στο έδαφος
 if(jump  && shape.isOnGround==true)
 {
  shape.vy += shape.jumpForce;
  shape.isOnGround = false;
  }

// Σε κάθε ανανέωση του καμβά η θέση shape.x και shape.y θα αλλάζει αν έχει
πατηθεί κάποιο πλήκτρο που επηρεάζει τη κίνηση
shape.x=shape.x+shape.vx;
shape.y=shape.y+shape.vy;

if(shape.isOnGround==false){
 shape.vy+=shape.gravity;}

if(shape.x+shape.width>canvas.width){
shape.x=canvas.width-shape.width;
}
if(shape.x<0){
shape.x=0;
}
if(shape.y+shape.height>canvas.height){
shape.y=canvas.height-shape.height;
shape.isOnGround = true;
}
```

```
if(shape.y<0){
shape.y=0;
}

ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.drawImage(image1,shape.x,shape.y,shape.width,shape.height);

}

</script>
</html>
```

*Στο φάκελο LESSON3 το αρχείο lesson3c.html εμφανίζει το αποτέλεσμα.*
Η μπάλα κινείται δεξιά αριστερά και πηδάει με το πλήκτρο SPACE.

# 4.Σύγκρουση μεταξύ αντικειμένων

Ας δούμε πως μπορούμε να ορίσουμε διάφορες καταστάσεις όταν ένα αντικείμενο ακουμπάει ένα άλλο.
Δηλαδή όταν υπάρχει σύγκρουση μεταξύ σχημάτων από οποιαδήποτε κατεύθυνση.




Για παράδειγμα όταν ένα σχήμα ακουμπάει ένα άλλο σχήμα αυτό να εξαφανίζεται.
Αυτό μπορεί να γίνει ως εξής.
Να ορίσουμε μια μεταβλητή λογική ή οποία θα ξεκινάει με τιμή αληθής.
var face=true;
και όταν ακουμπάει το ένα σχήμα το άλλο να γίνεται ψευδής.

```
if((shape1.x+shape1.width>shape2.x) && (shape1.x<=shape2.x+shape2.width) &&
(shape1.y+shape1.height>=shape2.y) && (shape1.y<=shape2.y+shape2.height)){
  face=false;}
```

Ας δούμε ένα παράδειγμα.

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
```

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

var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};

//Η μπάλα που θα κινεί ο χρήστης
var shape1 = Object.create(spriteObject);
shape1.x = 20;
shape1.y = 280;
shape1.width=50;
shape1.height=50;
shape1.vx=3;
shape1.vy=0;
shape1.isOnGround=true;
shape1.jumpForce= -12;

//Το αντικείμενο που όταν ακουμπήσει η μπάλα θα εξαφανιστεί
var shape2 = Object.create(spriteObject);
shape2.x = 250;
shape2.y = 190;
shape2.width=50;
shape2.height=130;

//Φωτογραφία μπάλας
var image1=new Image();
image1.src="img/ball.png";

//Όταν είναι αληθής θα εμφανίζεται η μπάλα
var face=true;

var RIGHT = 39;
var LEFT = 37;
var SPACE=32;

var moveLeft = false;
var moveRight = false;
var jump=false;

//Add keyboard listeners
window.addEventListener("keydown", function(event)
{
```

```
 switch(event.keyCode)
 {

    case LEFT:
     moveLeft = true;
     break;

    case RIGHT:
     moveRight = true;
     break;

    case SPACE:
     jump= true;
     break;
 }
}, false);

window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {

  case LEFT:
    moveLeft = false;
    break;

  case RIGHT:
    moveRight = false;
    break;

   case SPACE:
     jump= false;
     break;
 }
}, false);

update();

function update() {
requestAnimationFrame(update, canvas);

 if(moveLeft && !moveRight)
 {
    shape1.vx = -3;
 }
```

```
 if(moveRight && !moveLeft)
 {
   shape1.vx = 3;
 }

 if(!moveLeft && !moveRight)
 {
  shape1.vx = 0;
shape1.gravity = 0.3;
 }

 if(jump  && shape1.isOnGround==true)
 {
  shape1.vy += shape1.jumpForce;
  shape1.isOnGround = false;
  }

shape1.x=shape1.x+shape1.vx;
shape1.y=shape1.y+shape1.vy;

if(shape1.isOnGround==false){
 shape1.vy+=shape1.gravity;}

if(shape1.x+shape1.width>canvas.width){
shape1.x=canvas.width-shape1.width;
}
if(shape1.x<0){
shape1.x=0;
}
if(shape1.y+shape1.height>canvas.height){
shape1.y=canvas.height-shape1.height;
shape1.isOnGround = true;
}
if(shape1.y<0){
shape1.y=0;
}

//Χρώμα καμβά
ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

//Μόνο όταν είναι αληθής εμφανίζεται η μπάλα
if(face==true){
ctx.drawImage(image1,shape1.x,shape1.y,shape1.width,shape1.height);
}
ctx.fillStyle= "#000080";
```

```
ctx.fillRect(shape2.x,shape2.y,shape2.width,shape2.height);

//Αν η μπάλα ακουμπήσει το αντικείμενο θα εξαφανιστεί. Η μεταβλητή face θα γίνει ψευδής.
if((shape1.x+shape1.width>shape2.x) && (shape1.x<=shape2.x+shape2.width) &&
(shape1.y+shape1.height>=shape2.y) && (shape1.y<=shape2.y+shape2.height)){
  face=false;}

}

</script>
</html>
```

Η σύγκρουση δύο αντικειμένων είναι αρκετά πολύπλοκη κατάσταση και η παραπάνω εφαρμογή δείχνει την εκδοχή του σε μια απλή μορφή.

*Στο φάκελο **LESSON4** το αρχείο **lesson4.html** εμφανίζει το αποτέλεσμα.*

# 5. Το ποντίκι στις εφαρμογές

Η συμμετοχή του ποντικιού και αργότερα της επιφάνειας επαφής στις εφαρμογές είναι πολύ σημαντική.
Θα δούμε πώς με το πάτημα του κουμπιού του ποντικιού μπορεί να καθοριστεί μια ενέργεια, καθώς και πως αλλάζει ο δείκτης του ποντικιού.

Για να καθορίσουμε ένα γεγονός χρήσης ποντικιού(EVENT) οι δύο βασικές εντολές που θα χρησιμοποιούμε είναι οι παρακάτω:

```
// Γεγονός για αλλαγή δείκτη
canvas.addEventListener("mousemove", mousemoveHandler, false);
//Γεγονός για το πάτημα αριστερού κομπιού του ποντικιού
canvas.addEventListener("mousedown", mousedownHandler, false);

function mousemoveHandler(event)
{
 //Εντοπίζει τη θέση x και τη θέση y του δείκτη του ποντικιού
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;

//Κώδικας

}


 function mousedownHandler(event)
{
 //Εντοπίζει τη θέση x και τη θέση y του δείκτη του ποντικιού
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;

//Κώδικας
}
```

Ας δούμε ένα απλό παράδειγμα που θα εφαρμόζει πιο αναλυτικά τα παραπάνω:

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
```

```
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};

var shape1 = Object.create(spriteObject);
shape1.x = 100;
shape1.y = 50;
shape1.width=80;
shape1.height=80;

var shape2 = Object.create(spriteObject);
shape2.x = 280;
shape2.y = 50;
shape2.width=80;
shape2.height=80;

//Δύο λογικές μεταβλητές με αρχική τιμή false
var flag1=false;
var flag2=false;

canvas.addEventListener("mousemove", mousemoveHandler, false);
canvas.addEventListener("mousedown", mousedownHandler, false);

function mousemoveHandler(event)
{
  //Βρίσκεται η θέση x και y του ποντικιού.
  var mouseX = event.pageX - canvas.offsetLeft;
  var mouseY = event.pageY - canvas.offsetTop;

  //Ο δείκτης του ποντικιού είναι ο προκαθορισμένος
  canvas.style.cursor = "default";
 //Αν ο κέρσορας βρίσκεται μετάξύ αυτών των τιμών, ο δείκτης του ποντικιού
αλλάζει σε pointer
  if (mouseX>100 && mouseX<180  && mouseY>50 && mouseY<130){
  canvas.style.cursor = "pointer";}
```

```
//Αν ο κέρσορας βρίσκεται μετάξύ αυτών των τιμών, ο δείκτης του ποντικιού
αλλάζει σε pointer
 if (mouseX>280 && mouseX<360  && mouseY>50 && mouseY<130){
 canvas.style.cursor = "pointer";}
 }

 function mousedownHandler(event)
{
 //Find the mouse's x and y position
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;

if (mouseX>100 && mouseX<180  && mouseY>50 && mouseY<130){
 //Αν κλικάρεις στο μπλε τετραγωνάκι το flag1 γίνεται αληθής
 flag1=true;}
 if (mouseX>280 && mouseX<360  && mouseY>50 && mouseY<130){
//Αν κλικάρεις στο κόκκινο τετραγωνάκι το flag2 γίνεται αληθής
 flag2=true;}
 }

update();

function update() {
requestAnimationFrame(update, canvas);

ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.fillStyle= "#0000CC";
ctx.fillRect(shape1.x,shape1.y,shape1.width,shape1.height);

ctx.fillStyle= "#CC0000";
ctx.fillRect(shape2.x,shape2.y,shape2.width,shape2.height);

//Αν το flag1 είναι αληθής
if(flag1==true){
ctx.font = "18px Trebuchet MS";
ctx.fillStyle= "#FFFFCC";
ctx.fillText("Κλίκαρες στο μπλε κουτάκι" ,100,220);
}
Αν το flag2 είναι αληθής
if(flag2==true){
ctx.font = "18px Trebuchet MS";
ctx.fillStyle= "#FFFFCC";
ctx.fillText("Κλίκαρες στο κόκκινο κουτάκι" ,100,250);
}
```

```
}
</script>
</html>
```

Στο παραπάνω παράδειγμα φαίνεται πως μπορούμε να συνδυάσουμε τη θέση x(mouseX) και τη θέση y(mouseY) του ποντικιού με διάφορες καταστάσεις. *Σημαντικό ρόλο παίζει και ο σωστός χειρισμό των μεταβλητών. Εδώ χρησιμοποιήσαμε 2 λογικές μεταβλητές flag1 Και flag2 που η αλλαγή των τιμών τους μας εμφανίζει το αποτέλεσμα που θέλουμε.*

*Στο φάκελο **LESSON5** το αρχείο **lesson5.html** εμφανίζει το αποτέλεσμα.*

Στον πίνακα που ακολουθεί βλέπουμε κάποιος από τους δείκτες που μπορούμε να χρησιμοποιούμε στα προγράμματα μας.

| | |
|---|---|
| `default` |  . |
| `none` | No cursor is rendered. |
| `help` |  |
| `pointer` |  |
| `progress` |  |
| `wait` |  |
| `cell` |  |
| `crosshair` |  |
| `text` |  |
| `vertical-text` |  |
| `alias` |  |
| `copy` |  |

move 

not-allowed 

col-resize 

row-resize 

n-resize 

e-resize 

s-resize 

w-resize 

ne-resize 

nw-resize 

se-resize 

sw-resize 

ew-resize 

ns-resize 

nesw-resize 

nwse-resize 

zoom-in 

zoom-out 

grab 

# 6. ΗΧΟΣ στις εφαρμογές

Ας δούμε πως μπορούμε να εισάγουμε ήχους στα προγράμματα και πως μπορούμε να διαχειριστούμε αυτούς τους ήχους.
Καλό είναι οι ήχοι ή τραγούδια που θα χρησιμοποιούμε να είναι σε μορφή .mp3

```
var myAudio= new Audio();
myAudio.src=”sounds/song.mp3";
```

Για να παίξει το αρχείο ήχου:

```
myAudio.play();
```

Για να παύσει ένα αρχείο:

```
myAudio.pause();
```

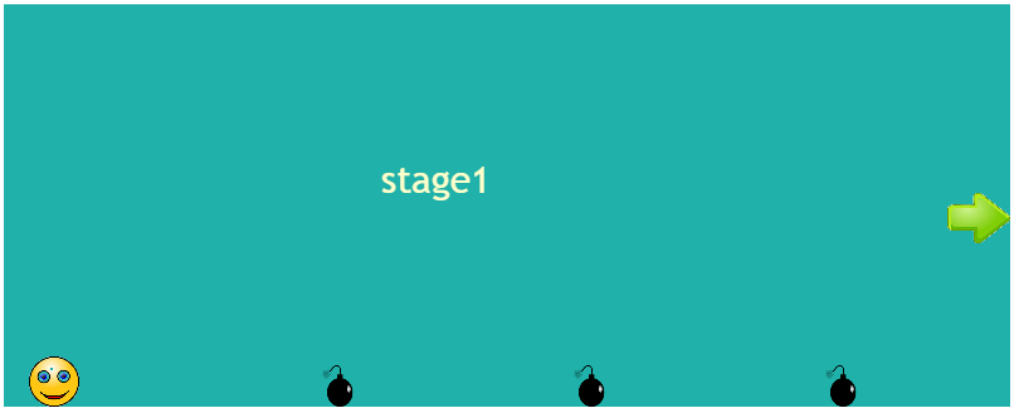
Σημαντικό ρόλο στην διαχείριση των ήχων παίζουν οι λογικές μεταβλητές όπως θα δούμε και παρακάτω.

# 7. Δημιουργία παιχνιδιών

Στο κεφάλαιο αυτό θα δημιουργήσουμε ορισμένα παιχνίδια , και θα αναλύσουμε αλγοριθμικά και τεχνικά τους τρόπους υλοποίησης τους.

## 7.1 Αποφεύγοντας τις βόμβες

Θα δημιουργήσουμε ένα πολύ απλό παιχνίδι που ο χρήστης θα πρέπει να προχωράει με τα βελάκια και πηδώντας να αποφεύγει τις βόμβες που θα συναντήσει για να πάει στο επόμενο επίπεδο.




Θα χρησιμοποιήσουμε για πρώτη φορά πίνακα αντικειμένων ώστε να εμφανίζονται οι βόμβες σε πολλές διαφορετικές θέσεις.

```
var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};
var bomb=new Array; // Δημιουργία πίνακα bomb.

// Στον πίνακα bomb θα αποθηκεύσω 6 αντικείμενα.
for (i=1;i<7;i++){
bomb[i] = Object.create(spriteObject);
bomb[i].x=50;
bomb[i].y=287;
```

```
bomb[i].width=30;
bomb[i].height=33;
}
```

Θα χρησιμοποιήσω για πρώτη φορά την μεταβλητή stage η οποία θα μας καθοδηγεί στις πίστες του παιχνιδιού.

Ας δούμε αναλυτικά τον κώδικα.

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title>gamename</title>
<center>
//Χρησιμοποιώ καμβά 800 X 320
<canvas width="800" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};

var shape1 = Object.create(spriteObject);
shape1.x = 20;
shape1.y = 290;
shape1.width=40;
shape1.height=40;
shape1.vx=3;
shape1.vy=0;
shape1.isOnGround=true;
shape1.jumpForce= -12;

var bomb=new Array; // Δημιουργία πίνακα bomb

for (i=1;i<7;i++){
bomb[i] = Object.create(spriteObject);
bomb[i].x=50;
```

```
bomb[i].y=287;
bomb[i].width=30;
bomb[i].height=33;
}

//Χρησιμοποιώ 3 εικόνες
var image1=new Image();
image1.src="img/hero.png";
var image2=new Image();
image2.src="img/bomb.png";
var image3=new Image();
image3.src="img/arrow.png";

// Χρησιμοποιώ τον ήχο expl.mp3
var myAudio= new Audio();
myAudio.src="sounds/expl.mp3"

var face=true;
var stage=1;

var RIGHT = 39;
var LEFT = 37;
var SPACE=32;

var moveLeft = false;
var moveRight = false;
var jump=false;

//Add keyboard listeners
window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {

    case LEFT:
      moveLeft = true;
      break;

    case RIGHT:
      moveRight = true;
      break;

    case SPACE:
      jump= true;
     break;
 }
```

```
}, false);

window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {

case LEFT:
 moveLeft = false;
 break;

case RIGHT:
 moveRight = false;
 break;

case SPACE:
  jump= false;
  break
 }
}, false);

update();

function update() {
requestAnimationFrame(update, canvas);

//Όταν face=true τότε υπάρχει μόνο κίνηση
   if(face==true){
 if(moveLeft && !moveRight)
 {
    shape1.vx = -3;
 }

 if(moveRight && !moveLeft)
 {
   shape1.vx = 3;
 }

 if(!moveLeft && !moveRight)
 {
  shape1.vx = 0;
  shape1.gravity = 0.3;
 }

if(jump  && shape1.isOnGround==true)
 {
```

```
   shape1.vy += shape1.jumpForce;
   shape1.isOnGround = false;
   }

shape1.x=shape1.x+shape1.vx;
shape1.y=shape1.y+shape1.vy;

if(shape1.isOnGround==false){
 shape1.vy+=shape1.gravity;}

if(shape1.x+shape1.width>canvas.width){
shape1.x=0;
stage++;
}
if(shape1.x<0){
shape1.x=0;
}
if(shape1.y+shape1.height>canvas.height){
shape1.y=canvas.height-shape1.height;
shape1.isOnGround = true;
}
if(shape1.y<0){
shape1.y=0;
}

}

ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

// Αν stage>0 και stage<5 το παιχνίδι εξελίσεται
if(stage>0 && stage<5){
ctx.drawImage(image3,750,150,50,40);

ctx.font = "30px Trebuchet MS";
ctx.fillStyle= "#FFFFCC";
ctx.fillText("stage"+stage,300,150);
}

// Αν stage=5 το παιχνίδι τελιώνει και νικάς
if(stage==5){
ctx.font = "30px Trebuchet MS";
ctx.fillStyle= "#FFFFCC";
ctx.fillText("Τα κατάφερες",300,150);
}
```

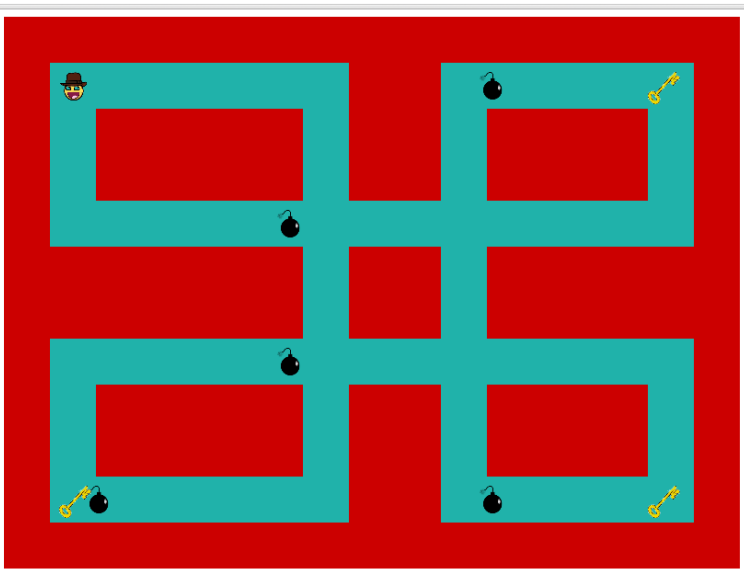
```
// Αν stage=0 χάνεις
if(stage==0){
ctx.font = "30px Trebuchet MS";
ctx.fillStyle= "#FFFFCC";
ctx.fillText("Έχασες. Πάτησε F5 για να ξαναπάιξεις",150,150);
}

// Αν face=true εμφανίζεται ο πρωταγωνιστής
if(face==true){
ctx.drawImage(image1,shape1.x,shape1.y,shape1.width,shape1.height);
}
if(stage==1){
//Εμφάνιση βομβών όταν stage=1
for (i=1;i<4;i++){
ctx.drawImage(image2,bomb[i].x+i*200,bomb[i].y,bomb[i].width,bomb[i].height);
// Σύγκρουση με κάποια βόμβα
if((shape1.x+shape1.width>bomb[i].x+i*200) &&
(shape1.x<=bomb[i].x+i*200+bomb[i].width) &&
(shape1.y+shape1.height>=bomb[i].y) && (shape1.y<=bomb[i].y+bomb[i].height)){
  face=false; // Ο πρωταγωνιστής δεν εμφανίζεται πια
  myAudio.play(); // ακούγεται ήχος της έκρηξης
  stage=0;// Το stage γίνεται 0
  }
}
}
if(stage==2){
//Εμφάνιση βομβών όταν stage=2
for (i=1;i<5;i++){
ctx.drawImage(image2,bomb[i].x+i*150,bomb[i].y,bomb[i].width,bomb[i].height);
// Σύγκρουση με κάποια βόμβα
if((shape1.x+shape1.width>bomb[i].x+i*150) &&
(shape1.x<=bomb[i].x+i*150+bomb[i].width) &&
(shape1.y+shape1.height>=bomb[i].y) && (shape1.y<=bomb[i].y+bomb[i].height)){
  face=false; //Ο πρωταγωνιστής δεν εμφανίζεται πια
  myAudio.play(); // ακούγεται ήχος της έκρηξης
  stage=0;  // Το stage γίνεται 0
}
}
}
if(stage==3){
//Εμφάνιση βομβών όταν stage=3
for (i=1;i<6;i++){
ctx.drawImage(image2,bomb[i].x+i*120,bomb[i].y,bomb[i].width,bomb[i].height);
// Σύγκρουση με κάποια βόμβα
```

```
if((shape1.x+shape1.width>bomb[i].x+i*120) &&
(shape1.x<=bomb[i].x+i*120+bomb[i].width) &&
(shape1.y+shape1.height>=bomb[i].y) && (shape1.y<=bomb[i].y+bomb[i].height)){
  face=false; //Ο πρωταγωνιστής δεν εμφανίζεται πια
  myAudio.play();// ακούγεται ήχος της έκρηξης
  stage=0; // Το stage γίνεται 0
}
}
}

if(stage==4){
//Εμφάνιση βομβών όταν stage=3
for (i=1;i<7;i++){
ctx.drawImage(image2,bomb[i].x+i*90,bomb[i].y,bomb[i].width,bomb[i].height);
// Σύγκρουση με κάποια βόμβα
if((shape1.x+shape1.width>bomb[i].x+i*90) &&
(shape1.x<=bomb[i].x+i*90+bomb[i].width) &&
(shape1.y+shape1.height>=bomb[i].y) && (shape1.y<=bomb[i].y+bomb[i].height)){
face=false; //Ο πρωταγωνιστής δεν εμφανίζεται πια
  myAudio.play();// ακούγεται ήχος της έκρηξης
  stage=0; // Το stage γίνεται 0
}
}

}
</script>
</html>
```

*Στο φάκελο **LESSON7** το αρχείο **lesson71.html** που εμφανίζει το αποτέλεσμα.*

# *7.2 Διαφυγή από τον λαβύρινθο*

Θα δημιουργήσουμε ένα πιο σύνθετο παιχνίδι όπου ο παίκτης θα κινείται σε έναν λαβύρινθο και θα προσπαθεί να αποφύγει τις βόμβες που τον ακολουθούνε. Σκοπός του παιχνιδιού είναι ο παίχτης να μαζέψει τα 3 κλειδιά που βρίσκονται μέσα στον λαβύρινθο.



Για την δημιουργία αυτού του παιχνιδιού θα χρησιμοποιήσουμε για πρώτη φορά την χρήση τυχαίου αριθμού(random), που είναι βασικός παράγοντας στην δημιουργία παιχνιδιού.

```
k=Math.floor(Math.random()*n);
```

Ο αριθμό k θα μπορεί να είναι οποιοσδήποτε τυχαίος αριθμός από το 0 μέχρι και το n-1. Ο υπολογιστής είναι αυτός που θα αποφασίσει τυχαία ποιον αριθμό θα επιστρέψει.

Ας δούμε τον κώδικα αναλυτικά και ας τον εξηγήσουμε.

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title>gamename</title>
<center>
<canvas width="800" height="600" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
```

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

var spriteObject =
{
  x: 0,
  y: 0,
  width: 20,
  height: 40
};

var shape = Object.create(spriteObject); //Δημιουργία αντικειμένου πρωταγωνιστή
shape.x = 60;
shape.y = 60;
shape.width=32;
shape.height=32;
shape.vx=3
shape.vy=3;

var s=new Array;//Δημιουργία πίνακα τοιχωμάτων λαβύρινθου
var s2=new Array;//Δημιουργία πίνακα βομβών
var key=new Array;//Δημιουργία πίνακα κλειδιών


for (i=1;i<14;i++){
s[i] = Object.create(spriteObject);//Δημιουργία 13 αντικειμένων τοιχωμάτων που
αποθηκεύονται σε πίνακα
}

for (i=1;i<6;i++){
s2[i] = Object.create(spriteObject);//Δημιουργία 5 αντικειμένων βομβών που
αποθηκεύονται σε πίνακα
s2[i].width=30;
s2[i].height=30;
s2[i].vx=-3;
s2[i].vy=0;
}

for (i=1;i<4;i++){
key[i] = Object.create(spriteObject);// Δημιουργία 3 αντικειμένων κλειδιών που
αποθηκεύονται σε πίνακα
key[i].width=35;
key[i].height=35;
key[i].face=true;
}
// Συντεταγμένες 4 κλειδιών
key[1].x=60;
```

```
key[1].y=510;
key[2].x=700;
key[2].y=510;
key[3].x=700;
key[3].y=60;

//Συντεταγμένες 5 βομβών
s2[1].x=700;
s2[1].y=60;
s2[2].x=700;
s2[2].y=510;
s2[3].x=480;
s2[3].y=210;
s2[4].x=480;
s2[4].y=360;
s2[5].x=200;
s2[5].y=510;

var image1=new Image();
image1.src="img/hero.png";// εικόνα πρωταγωνιστή
var image2=new Image();
image2.src="img/bomb.png";//εικόνα βόμβας
var image3=new Image();
image3.src="img/key.png";//εικόνα κλειδιού

var myAudio= new Audio();
myAudio.src="sounds/expl.mp3";//Ηχος έκρηξης βόμβας

var k;
var flaglife=true;// Όταν είναι αληθής θα εμφανίζεται ο πρωταγωνιστής
var stage=1;
var count=0;

var UP = 38;
var DOWN = 40;
var RIGHT = 39;
var LEFT = 37;


var moveUp = false;
var moveDown = false;
var moveLeft = false;
var moveRight = false;


//Add keyboard listeners
```

```
window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {
    case UP:
     moveUp = true;
     break;

    case DOWN:
     moveDown = true;
     break;

    case LEFT:
     moveLeft = true;
      break;

    case RIGHT:
     moveRight = true;
     break;

 }
}, false);

window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {
     case UP:
      moveUp = false;
      break;

     case DOWN:
       moveDown = false;
       break;


     case LEFT:
       moveLeft = false;
       break;

     case RIGHT:
       moveRight = false;
        break;

 }
}, false);
```

```
update();


function update() {
requestAnimationFrame(update, canvas);

//Συμβαίνουν όλα αν stage=1
if(stage==1){

//Συντεταγμένες 13 τοιχωμάτων
s[1].x = 0;
s[1].y = 0;
s[1].width=50;
s[1].height=600;

s[2].x = 750;
s[2].y = 0;
s[2].width=50;
s[2].height=600;

s[3].x = 0;
s[3].y = 0;
s[3].width=800;
s[3].height=50;

s[4].x = 0;
s[4].y = 550;
s[4].width=800;
s[4].height=50;

s[5].x = 100;
s[5].y = 100;
s[5].width=225;
s[5].height=100;

s[6].x = 0;
s[6].y = 250;
s[6].width=325;
s[6].height=100;

s[7].x = 100;
s[7].y = 400;
s[7].width=225;
s[7].height=100;
```

```
s[8].x = 375;
s[8].y = 0;
s[8].width=100;
s[8].height=200;

s[9].x = 375;
s[9].y = 400;
s[9].width=100;
s[9].height=200;

s[10].x = 525;
s[10].y = 100;
s[10].width=175;
s[10].height=100;

s[11].x = 525;
s[11].y = 400;
s[11].width=175;
s[11].height=100;

s[12].x = 525;
s[12].y = 250;
s[12].width=275;
s[12].height=100;

s[13].x = 375;
s[13].y = 250;
s[13].width=100;
s[13].height=100;

if(moveUp && !moveDown)
 {
  shape.vy = -3;
 }

 if(moveDown && !moveUp)
 {
  shape.vy = 3;
 }


 if(moveLeft && !moveRight)
 {
    shape.vx = -3;
 }
```

```
 if(moveRight && !moveLeft)
 {
   shape.vx = 3;
 }

if(!moveUp && !moveDown)
 {
  shape.vy = 0;
 }
 if(!moveLeft && !moveRight)
 {
  shape.vx = 0;
 }


shape.x=shape.x+shape.vx;
shape.y=shape.y+shape.vy;

//Κίνηση βομβών
for (i=1;i<6;i++){
s2[i].x=s2[i].x+s2[i].vx;
s2[i].y=s2[i].y+s2[i].vy;
}


if(shape.x+shape.width>canvas.width){
shape.x=canvas.width-shape.width;
}
if(shape.x<0){
shape.x=0;
}
if(shape.y+shape.height>canvas.height){
shape.y=canvas.height-shape.height;
}
if(shape.y<0){
shape.y=0;
}

ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

for (i=1;i<14;i++){
ctx.fillStyle= "#CC0000";
ctx.fillRect(s[i].x,s[i].y,s[i].width,s[i].height);//Σχεδίαση τοιχωμάτων
```

```
//Αν ο πρωταγωνιστής ακουμπήσει τα τοιχώματα σταματάει. Δεν τα περνάει
if((shape.x+shape.width>s[i].x) && (shape.x<=s[i].x+s[i].width) &&
(shape.y+shape.height>=s[i].y) && (shape.y<=s[i].y+s[i].height)){
shape.x=shape.x-shape.vx;
shape.y=shape.y-shape.vy;
 }
}

//Εμφάνιση πρωταγωνιστή αν μόνο η μεταβλητή είναι αληθής
if(flaglife==true){
ctx.drawImage(image1,shape.x,shape.y,shape.width,shape.height);
}

//Σχεδίαση βομβών
for (i=1;i<6;i++){
ctx.drawImage(image2,s2[i].x,s2[i].y,s2[i].width,s2[i].height);
}

//Κίνηση βομβών. Όταν οι βόμβες συναντούν τοιχώματα, η μεταβλητή k θα
αποφασίζει προς τα πού θα κινηθούν
for (i=1;i<6;i++){
for(j=1;j<14;j++){
if((s2[i].x+s2[i].width>s[j].x) && (s2[i].x<=s[j].x+s[j].width) &&
(s2[i].y+s2[i].height>=s[j].y) && (s2[i].y<=s[j].y+s[j].height)){
s2[i].x=s2[i].x-s2[i].vx;
s2[i].y=s2[i].y-s2[i].vy;
k=Math.floor(Math.random()*4);//Μεταβλητή k που παίρνει τυχαίες τιμές  0,1,2,3
if (k==0) {
 s2[i].vx=0;
 s2[i].vy=3;
}
if (k==1) {
 s2[i].vx=0;
 s2[i].vy=-3
}
if (k==2) {
 s2[i].vx=3;
 s2[i].vy=0;
}
if (k==3) {
 s2[i].vx=-3;
 s2[i].vy=0;
}

}
```

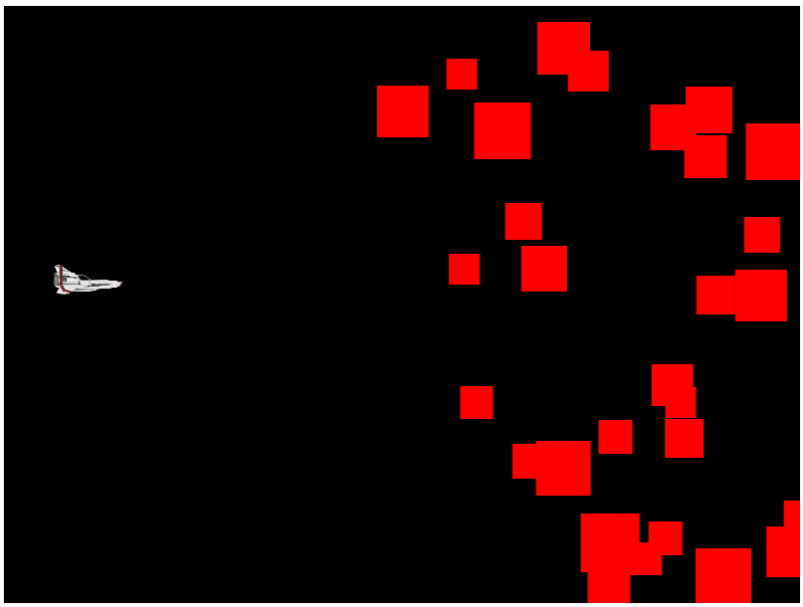
```
}
}


//Σχεδίαση κλειδιών.
for (i=1;i<4;i++){
if(key[i].face==true){
ctx.drawImage(image3,key[i].x,key[i].y,key[i].width,key[i].height);
}
}

//Αν ο πρωταγωνιστής ακουμπήσει βόμβες χάνει
for (i=1;i<6;i++){
if((shape.x+shape.width>s2[i].x) && (shape.x<=s2[i].x+s2[i].width) &&
(shape.y+shape.height>=s2[i].y) && (shape.y<=s2[i].y+s2[i].height)){
flaglife=false;//flaglife=ψευδής
 myAudio.play();//Ακούγεται έκρηξη
  stage=3;// stage=3
}
}

//Αν ο πρωταγωνιστής ακουμπήσει το κλειδί αυτό εξαφανίζεται
for (i=1;i<4;i++){
if((shape.x+shape.width>key[i].x) && (shape.x<=key[i].x+key[i].width) &&
(shape.y+shape.height>=key[i].y) && (shape.y<=key[i].y+key[i].height) &&
key[i].face==true){
key[i].face=false;
count++;//Η μεταβλητή count αυξάνει κατά 1
}
}
//Αν το count γίνει 3 ,που σημαίνει εξαφανίστηκαν και τα 3 κλειδιά
if(count==3){
stage=2;//stage=2
}
}

//Αν stage=2 εμφάνισε σκηνικό νίκησες
if(stage==2){
ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.font = "80px Verdana";
ctx.fillStyle= "green";
ctx.fillText("ΝΙΚΗΣΕΣ" ,200,300);
}
```

```
//Αν stage=3 εμφάνισε σκηνικό έχασες
if(stage==3){
ctx.fillStyle="#20B2AA";
ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.font = "80px Verdana";
ctx.fillStyle= "red";
ctx.fillText("ΕΧΑΣΕΣ" ,220,300);

}

}

</script>
</html>
```

*Στο φάκελο **LESSON7** το αρχείο **lesson72.html** που εμφανίζει το αποτέλεσμα.*

# *7.3 Παράξενη εισβολή*

Θα δούμε τώρα πως θα φτιάξουμε ένα παιχνίδι που πρωταγωνιστής θα είναι μια μορφή που πυροβολεί.
Συγκεκριμένα ένα διαστημόπλοιο το οποίο κινείται με τα βελάκια, και το οποίο πυροβολεί τα εχθρικά αντικείμενα.



Βασικό ρόλο στο παιχνίδι μας θα παίζει ο πίνακας αντικείμενο bullets, ο οποίος θα καθορίζει των αριθμό των πυρομαχικών που έχω.

```
var bullets=new Array;
for(i=0;i<150;i++){
bullets[i]= Object.create(spriteObject);
bullets[i].x = -5
bullets[i].y =-5
bullets[i].width=6;
bullets[i].height=3
bullets[i].vx=0;
bullets[i].face=1;
}
```

Ας δούμε αναλυτικά το παιχνίδι:

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-
scalable = no">
<title>gamename</title>
<center>
<canvas width="800" height="600" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

var spriteObject =
{
  x: 0,
  y: 0,
  width: 70,
  height: 30,
};

// Δημιουργία αντικειμένου πρωταγωνιστή
var hero = Object.create(spriteObject);
hero.x = 50;
hero.y =260;
hero.width=70;
hero.height=30;
hero.vx=0;
hero.vy=0;

// Δημιουργά αντικειμένου πίνακα για τις σφαίρες. Μέχρι 150 σφαίρες.
var bullets=new Array;
for(i=0;i<150;i++){
bullets[i]= Object.create(spriteObject);
bullets[i].x = -5
bullets[i].y =-5
bullets[i].width=6;
bullets[i].height=3
bullets[i].vx=0;
bullets[i].face=1;
}

// Δημιουργία πίνακα αντικειμένου με 70 εχθρικά σκάφη.
var a2=new Array;
for(i=1;i<70;i++){
a2[i]= Object.create(spriteObject);
a2[i].x = Math.floor(Math.random()*300)+900;
```

```
a2[i].y =Math.floor(Math.random()*580);
a2[i].width=Math.floor(Math.random()*30)+30;
a2[i].height=a2[i].width
a2[i].vx=-Math.random()*1-0.4;
a2[i].face=1;
}

//Εικόνα Διστημοπλοίου
image1=new Image();
image1.src="img/ship.png";
//μεταβητή που θα μετράει τις σφαίρες.
var k=0;
//Μεταβλητή που καθορίζει ότι δεν έχει χάσει ο πρωταγωνιστής.
var heroflag=true;

var UP = 38;
var DOWN = 40;
var RIGHT = 39;
var LEFT = 37;
var SPACE=32;

var moveUp = false;
var moveDown = false;
var moveLeft = false;
var moveRight = false;
var fire=false;

window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {
     case UP:
      moveUp = true;
      break;

     case DOWN:
      moveDown = true;
      break;

     case LEFT:
      moveLeft = true;
      break;

      case RIGHT:
       moveRight = true;
       break;
```

```
      case SPACE:
       fire= true;
       break;

 }
}, false);


update();

function update() {
requestAnimationFrame(update, canvas);


window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {
   case UP:
    moveUp = false;
    break;

   case DOWN:
    moveDown = false;
    break;


    case LEFT:
   moveLeft = false;
    break;


    case RIGHT:
     moveRight = false;
     break;

    case SPACE:
     fire= false;
     break;

 }
}, false);


//Κίνηση διαστημοπλοίου
```

```
if(moveRight==true){
hero.vx=4;}
if(moveLeft==true){
hero.vx=-4;}
if(moveUp==true){
hero.vy=-4;}
if(moveDown==true){
hero.vy=4;}

//Όταν πατάμε space θα φέυγει μια σφαίρα και το k θα αυξάνει κατά 1. Όταν το k γίνει 150 δεν θα υπάρχουν άλλες σφαίρες.
if(fire==true && k<150){
k++;
bullets[k].vx=8;
bullets[k].x=hero.x+20;
bullets[k].y=hero.y+20;
fire=false;}

if(moveUp==false && moveDown==false){
hero.vy=0;}
if(moveRight==false && moveLeft==false){
hero.vx=0;}


//Κίνηση διαστημοπλοίου.
hero.x=hero.x+hero.vx;
hero.y=hero.y+hero.vy;

if(hero.y+hero.height>canvas.height){
hero.y=canvas.height-hero.height;}

if(hero.y<0){
hero.y=0;}

if(hero.x<0){
hero.x=0;}

//Κίνηση σφαιρών.
for(i=1;i<150;i++){
bullets[i].x=bullets[i].x+bullets[i].vx;}

//Κίνηση εχθρικών σκαφών.
for(i=1;i<40;i++){
a2[i].x=a2[i].x+a2[i].vx;
}
```

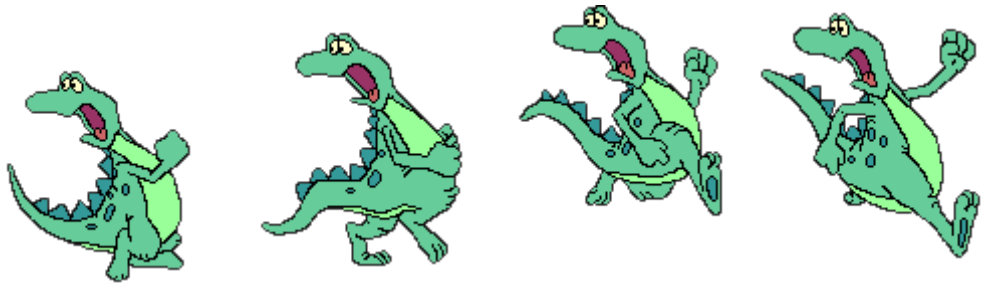
```
ctx.fillStyle="black";
ctx.fillRect(0, 0, canvas.width, canvas.height);

//Αν η μεταβλητή heroflag είναι αληθής τότε εμφανίζεται το διαστημόπλοιο.
if(heroflag==true){
ctx.drawImage
    (
     image1,
      hero.x,hero.y,
      hero.width, hero.height
    );
}

// Αν η μεταβλητή bullets[i].face==1 τότε εμφανίζεται η αντίστοιχη σφαίρα.
for(i=1;i<150;i++){
if(bullets[i].face==1){
ctx.fillStyle= "yellow";
ctx.fillRect(bullets[i].x,bullets[i].y,bullets[i].width,bullets[i].height);
}
//Αν η αντίστοιχη σφαίρα ακουμπήσει το εχθρικό σκάφος, αυτό εξαφανίζεται. Η
μεταβλητή bullets[i].face γίνεται 0 και η μεταβλητή a2[i].face γίνεται 0.
for(j=1;j<40;j++){
if((bullets[i].x+bullets[i].width>a2[j].x) && (bullets[i].x<=a2[j].x+a2[j].width) &&
(bullets[i].y+bullets[i].height>=a2[j].y) && (bullets[i].y<=a2[j].y+a2[j].height) &&
a2[j].face==1 && bullets[i].face==1){
a2[j].face=0;
bullets[i].face=0;}
}

}

//Αν η μεταβλητή a2[i].face==1 τότε εμφανίζεται το αντίστοιχο εχρικό σκάφος.
for(i=1;i<40;i++){
if(a2[i].face==1){
ctx.fillStyle= "red";
ctx.fillRect(a2[i].x,a2[i].y,a2[i].width,a2[i].height);
//Αν το εχρικό σκάφος ακουμπήσει το διαστημόπλοιο, αυτό εξαφανίζεται.
if((hero.x+hero.width>a2[i].x) && (hero.x<=a2[i].x+a2[i].width) &&
(hero.y+hero.height>=a2[i].y) && (hero.y<=a2[i].y+a2[i].height)){
heroflag=false;
}
}
}

}
```

```
</script>
</html>
```

*Στο φάκελο **LESSON7** το αρχείο **lesson73.html** που εμφανίζει το αποτέλεσμα.*

# *7.4 Scrolling*

Ας δούμε τώρα πω μπορούμε να δημιουργήσουμε ένα παιχνίδι με Scrolling. Δηλαδή με το να φαίνεται ότι ο πρωταγωνιστής του παιχνιδιού περιφέρεται ελεύθερα χωρίς περιορισμό σε μία επιφάνεια.
Είναι μια πολύ σημαντική παράμετρος για να δείχνουν αληθοφανή τα παιχνίδια.
Ταυτόχρονα με τη ανάλυση του Scrolling, θα δούμε και πως μία μορφή φαίνεται ότι τρέχει καρέ καρέ.



Θα δημιουργήσουμε κίνηση φυσική, με την χρήση έξυπνου κώδικα, χρησιμοποιώντας τα καρέ των αντίστοιχων φωτογραφιών

Ο τρόπος που επιλέχθηκε για να παρουσιαστεί ο κώδικας για ένα οριζόντιο scrolling είναι πολύ απλός.
Ας δούμε αναλυτικά:

```
<!doctype html>
<meta charset=”UTF-8”>
<meta name="viewport" content="width = device-width,initial-scale = 1.0, user-scalable = no">
<title>scrolling</title>
<center>
<canvas width="1000" height="600" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script src="main.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

//Δημιουργώ ένα αντικείμενο
var spriteObject =
{
  x: 0,
  y: 0,
  width: 100,
  height: 40
};

//Με βάση το βασικό αντικείμενο, δημιουργώ αντικείμενο bg(background)
var bg = Object.create(spriteObject);
bg.x = 0;
bg.y = 0;
bg.width=15000;
bg.height=600;
bg.vx=0;

//Δημιουργώ το αντικείμενο του πρωταγωνιστή
var hero = Object.create(spriteObject);
hero.x = 200;
hero.y =490;
hero.width=90;
hero.height=100;
hero.vx=0;

//Δηλώνω φωτογραφίες
var image2=new Image();
image2.src="pics/dyn1.png"
```

```
var image3=new Image();
image3.src="pics/dyn2.png"
var image4=new Image();
image4.src="pics/dyn3.png";
var image5=new Image();
image5.src="pics/dyn4.png";
var image6=new Image();
image6.src="pics/dyn5.png";
var image7=new Image();
image7.src="pics/dyn6.png";
var image8=new Image();
image8.src="pics/tree1.png";
var image9=new Image();
image9.src="pics/tree2.png";
var image10=new Image();
image10.src="pics/dyn1left.png"
var image11=new Image();
image11.src="pics/dyn2left.png"
var image12=new Image();
image12.src="pics/dyn3left.png";
var image13=new Image();
image13.src="pics/dyn4left.png";
var image14=new Image();
image14.src="pics/dyn5left.png";
var image15=new Image();
image15.src="pics/dyn6left.png";


//Μεταβλητές τις οποίες θα αναλύσουμε αργότερα
var mr=0;
var mb=0;
var flagl=true;
var flagr=true;
var st=1;

//Θα χρησιμοποιήσω μόνο τα κουμπιά αριστερό και δεξί βελάκι
var RIGHT = 39;
var LEFT = 37;

var moveLeft = false;
var moveRight = false;

window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {
```

```
        case LEFT:
           moveLeft = true;
            break;

       case RIGHT:
            moveRight = true;
            break;

  }
}, false);

window.addEventListener("keyup", function(event)
{
  switch(event.keyCode)
  {

        case LEFT:
            moveLeft = false;
            break;

      case RIGHT:
            moveRight = false;
            break;

  }
}, false);

update();
//Ανανεώνεται ο καμβάς σύμφωνα με το αρχείο requestAnimationFramePolyfill.js
function update() {
requestAnimationFrame(update, canvas);

//Χρησιμοποιώ την μεταβλητή mr για μετρητή χρόνου ώστε να αλλάζουν ομαλά τα
καρέ του πρωταγωνιστή όταν κινείται δεξιά
mr=mr+2.5;
if (mr>60) {
  mr=0;}
//Χρησιμοποιώ την μεταβλητή mb για μετρητή χρόνου ώστε να αλλάζουν ομαλά τα
καρέ του πρωταγωνιστή όταν κινείται αριστερά
mb=mb+2.5;
if (mb>60) {
  mb=0;}

//Όταν πατάμε αριστερό βέλος η μεταβλητή st=0
 if(!moveRight && moveLeft)
```

```
 {
    st=0;
//Όταν η μεταβλητή flagl=true,  ο πρωταγωνιστής είναι έτοιμος να κινηθεί
αριστερά.
   if (flagl==true) {
          hero.x=hero.x-5;
           flagl=false;
           flagr=true;
    }
//’όταν ο πρωταγωνιστής βρίσκεται στα παρακάτω όρια θα κινείται ο
πρωταγωνιστής και όχι το background
    if (hero.x>100 && hero.x<350) {
          hero.vx=-4;
         bg.vx=0;
//’όταν ο πρωταγωνιστής βρίσκεται στα παρακάτω όρια θα κινείται το background
και όχι ο πρωταγωνιστής
    }
    if (hero.x<=100 || hero.x>=350) {
          hero.vx=0;
          bg.vx=4;

    }
 }

//Όταν πατάμε δεξί βέλος η μεταβλητή st=1
if(moveRight && !moveLeft)
 {
  st=1;
//Όταν η μεταβλητή flagr=true,  ο πρωταγωνιστής είναι έτοιμος να κινηθεί δεξιά
    if (flagr==true) {
          hero.x=hero.x+5;
           flagr=false;
           flagl=true;
    }
  //’όταν ο πρωταγωνιστής βρίσκεται στα παρακάτω όρια θα κινείται ο
πρωταγωνιστής και όχι το background
    if (hero.x>100 && hero.x<350) {
          hero.vx=4;
          bg.vx=0;
    }
 //’όταν ο πρωταγωνιστής βρίσκεται στα παρακάτω όρια θα κινείται το background
και όχι ο πρωταγωνιστής
    }

    if (hero.x<=100 || hero.x>=350) {
          hero.vx=0;
```

```
            bg.vx=-4;
        }
    }

//Όταν δεν πατηθεί ούτε δεξιά θύτε αριστερά
if(!moveLeft && !moveRight)
 {
   mr=0;
  mb=0;
   bg.vx = 0;
   hero.vx=0;
   }

//Το bg.x αυξάνει κατά bg.vx. Κινείται το background
bg.x += bg.vx;
//Το hero.x αυξάνει κατά hero.vx. Κινείται ο πρωταγωνιστής
hero.x+=hero.vx;

//Παράμετροι για να μην βγαίνει ο πρωταγωνιστής έξω από το background.
if (hero.x<bg.x+100) {
 hero.x=bg.x+100
 bg.vx=0;
}

// Δημιουργία καμβά και background(bg)
ctx.fillStyle="#00FFCC";
ctx.fillRect(0, 0, canvas.width, canvas.height);
ctx.fillRect(bg.x+17300,0,400,600);

//Σχεδίαση δέντρων σε σχέση πάντα με το background(bg)
ctx.drawImage(image8, 700+bg.x,300, 110, 150 );
ctx.drawImage(image9, 1200+bg.x,300, 80, 150 );


//Αν st=1  ο πρωταγωνιστής κινείται δεξιά και αλλάζει καρέ σύμφωνα με τη
μεταβλητή χρόνου mr. Όταν η mr πάρει τιμή μεγαλύτερη του 60 αυτόματα
μηδενίζει- όπως έχω ορίσει παραπάνω- και τα καρέ αλλάζουν από την αρχή.
if (st==1){
 if(mr<=10){
   ctx.drawImage
     (
       image2,
        hero.x,hero.y,
        hero.width, hero.height
     );
              }
```

```
 if(mr>10 && mr<=20){
ctx.drawImage
  (
    image3,
     hero.x,hero.y,
     hero.width, hero.height
  );
          }
 if(mr>20 && mr<=30){
ctx.drawImage
  (
    image4,
     hero.x,hero.y,
     hero.width, hero.height
  );

 }

if(mr>30 && mr<=40){
ctx.drawImage
  (
    image5,
     hero.x,hero.y,
     hero.width, hero.height
  );

 }

if(mr>40 && mr<=50){
ctx.drawImage
  (
    image6,
     hero.x,hero.y,
     hero.width, hero.height
  );

 }

if(mr>50 && mr<=60){
ctx.drawImage
  (
    image7,
     hero.x,hero.y,
     hero.width, hero.height
  );
```

```
   }

 }

//Αν st=0  ο πρωταγωνιστής κινείται δεξιά και αλλάζει καρέ σύμφωνα με τη
μεταβλητή χρόνου mb. Όταν η mb πάρει τιμή μεγαλύτερη του 60 αυτόματα
μηδενίζει- όπως έχω ορίσει παραπάνω- και τα καρέ αλλάζουν από την αρχή.
if (st==0){
 if(mb<=10){
  ctx.drawImage
    (
      image10,
       hero.x,hero.y,
       hero.width, hero.height
    );
              }
   if(mb>10 && mb<=20){
  ctx.drawImage
    (
      image11,
       hero.x,hero.y,
       hero.width, hero.height
    );
              }
   if(mb>20 && mb<=30){
  ctx.drawImage
    (
      image12,
       hero.x,hero.y,
       hero.width, hero.height
    );

   }

  if(mb>30 && mb<=40){
  ctx.drawImage
    (
      image13,
       hero.x,hero.y,
       hero.width, hero.height
    );

   }

  if(mb>40 && mb<=50){
  ctx.drawImage
```

```
    (
      image14,
       hero.x,hero.y,
       hero.width, hero.height
    );

   }

  if(mb>50 && mb<=60){
  ctx.drawImage
    (
      image15,
       hero.x,hero.y,
       hero.width, hero.height
    );

   }

 }

}

</script>
</html>
```

*Στο φάκελο **LESSON7** το αρχείο **lesson74.html** που εμφανίζει το αποτέλεσμα.*

# *7.5 High score*

## Τοπικό High Score.

Η HTML5 είναι μία γλώσσα που μπορεί κανείς εύκολα να αποθηκεύσει σε τοπικές(local storage) μεταβλητές και έτσι είναι εύκολα σε ένα παιχνίδι να βάλεις κώδικά για τοπικό high score. Δηλαδή για το καλύτερο σκορ στη συσκευή που χρησιμοποιείς. Δηλαδή Υπολογιστή, Smartphone ή tablet.
Μια τοπική μεταβλητή είναι μια μεταβλητή η οποία υπάρχει αποθηκευμένη μέσα στη συσκευή που χρησιμοποιείς και την επεξεργάζεσαι όποτε χρειάζεσαι. Δηλαδή σαν ένα πιο εύχρηστο cookie.

Για να αποθηκεύσεις δεδομένα σε μια τοπική μεταβλητή η εντολή είναι η παρακάτω:

//Σε μία τοπική μεταβλητή με όνομα time αποθηκεύεις τη τιμή της μεταβλητής seconds. Η μεταβλητή seconds είναι μία μεταβλητή του κυρίως προγράμματος.
**localStorage.setItem("time",seconds);**

Για να σταλθούν δεδομένα από μια τοπική μεταβλητή η εντολή είναι η παρακάτω:

// Στη μεταβλητή t του προγράμματος αποθηκεύεται το περιεχόμενο της τοπικής μεταβλητής time.
**t=localStorage.getItem("time");**

Ας δούμε ένα παράδειγμα κώδικα για high score:

```
//Αποθηκεύουμε σε μία μεταβλητή του προγράμματος dt το περιεχόμενο της καινούργιας τοπικής μεταβλητής dreamtime που δημιουργήσαμε. Φυσικά η καινούργια τοπική μεταβλητή δεν θα έχει περιεχόμενο.
dt=localStorage.getItem("dreamtime");

//Αν η μεταβλητή dt δεν έχει περιεχόμενο τότε dt=3000. Βάζω μια δικιά μου αρχική τιμή. Μια αρχική τιμή high score που θα μπορεί κάποιος εύκολα να την ξεπεράσει.
if(dt==null){
dt=3000;}
//Αν ο χρόνος που έκανε ο παίκτης είναι μικρότερος από το περιεχόμενο της μεταβλητής dt τότε……
if(seconds<dt){
//Στην μεταβλητή name απθήκευσε το όνομα σου
name=prompt("New Record!!Give your name");
//Αποθήκευσε στη τοπική μεταβλητες dreamtime και dreamplayer τα περιεχόμενα των μεταβλητών seconds και name αντίστοιχα
localStorage.setItem("dreamtime",seconds);
localStorage.setItem("dreamplayer",name);
```

```
}
//Αποθήκευσε στις μεταβλητές dt,dn  αντίστοιχα το περιεχόμενο των τοπικών μεταβλητών dreamtime,dreamplayer.
dt=localStorage.getItem("dreamtime");
dn=localStorage.getItem("dreamplayer");

// Εμφάνισε το χρόνο και το όνομα αυτού που έχει το High score
ctx.font = "30px Verdana";
ctx.fillStyle= "black";
ctx.fillText("Ο χρόνος σου είναι: "+seconds+" seconds" ,260,200);
ctx.font = "45px Verdana";
ctx.fillStyle= "red";
ctx.fillText("Ο καλύτερο χρόνος είναι..." ,320,270);
ctx.font = "30px Verdana";
ctx.fillStyle= "black";
ctx.fillText(dn ,360,350);
ctx.fillText(dt+" seconds" ,390,390);
```

## On line High Score.

Για να έχουμε high score  που δεν θα είναι μόνο τοπικό, αλλά θα αφορά το καλύτερο σκορ όλων των χρηστών στο διαδίκτυο που παίζουν το παιχνίδι από τον υπολογιστή τους, υπάρχουν πολλοί τρόποι υλοποίησης. Ως πιο απλό θεωρώ αυτόν που υλοποιείται με την χρήση jQuery, PHP,SQL.
Χρησιμοποιείται η  jQuery ως αλυσίδα επικοινωνίας μεταξύ jscript και php.

```
if(flagst==true){
 name=prompt("Δώσε το όνομα σου");
$.post( "state.php", { scoreout:seconds , nameout:name})
  .done(function( data ) {
    res=data;
  });
flagst=false;
}
```

Για περισσότερες λεπτομέρειες:

http://api.jquery.com/jquery.post/

# *7.5 Δημιουργία ΜΕΝΟΥ*

Είναι λογικό σε ένα παιχνίδι να υπάρχει Μενού, ώστε ο χρήστης να πατάει επιλογή έναρξης και επιλογή για να ξαναπαίξει το παιχνίδι.
Βασικό ρόλο σε αυτό παίζουν τα παρακάτω:

1. Κατανόηση της χρήσης του ποντικιού(mouse)
2. Μεταβλητή stage η οποία θα μας δείχνει το δρόμο για το πως θα κινηθούμε.
3. Εντολή window.location.href = "index.html" , η οποία θα μας ξαναφορτώνει το αρχείο index.html ώστε να ξαναπαίξει ο χρήστης.



Ακολουθεί μια μικρή εφαρμογή όπου ο αναγνώστης θα κατανοήσει πρακτικά τα παραπάνω.

```
<!doctype html>
<meta charset=”UTF-8”>
<title>MENOY</title>
<CENTER>
<canvas width="800" height="500" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");
//Δήλωση αντικειμένου
var spriteObject =
{
  x: 0,
  y: 0,
  width: 20,
  height: 40
};
//Παράγωγο αντικειμένου hero
var hero = Object.create(spriteObject);
hero.x = 60;
hero.y = 20;
hero.width=40;
hero.height=40;
hero.vx=0
hero.vy=0;
//Παράγωγο αντικειμένου key
var key= Object.create(spriteObject);
key.x = 600;
key.y = 250;
key.width=120;
key.height=60;
//Αρχική τιμή μεταβλητής stage=0
var stage=0;
//Δήλωση φωτογραφιών
var image1=new Image();
image1.src="img/hat.png";;
var image2=new Image();
image2.src="img/key.png";

//Δήλωση μεταβλητών για κίνηση πληκρτολογίου
var UP = 38;
var DOWN = 40;
var RIGHT = 39;
var LEFT = 37;
var moveUp = false;
```

```
var moveDown = false;
var moveLeft = false;
var moveRight = false;

//Χρήση ποντικιού
canvas.addEventListener("mousemove", mousemoveHandler, false);
canvas.addEventListener("mousedown", mousedownHandler, false);

//Συνάρτηση δείκτη ποντικιού όταν κινείται
function mousemoveHandler(event)
{
 //Βρίσκει τη θέση του mouseX , mouseY
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;
     //Αν το stage=0 και κινηθείς στην αντίστοιχη περιοχή θα αλλάξει ο δείκτης
  if(stage==0){
  canvas.style.cursor = "default";
 if (mouseX>300 && mouseX<500  && mouseY>380 && mouseY<460){
   canvas.style.cursor = "pointer";}
           }
 //Αν το stage=1 και κινηθείς στην αντίστοιχη περιοχή θα αλλάξει ο δείκτης
  if(stage==1){
    canvas.style.cursor = "none";}


  if(stage==2){
  canvas.style.cursor = "default";
 if (mouseX>300 && mouseX<520  && mouseY>380 && mouseY<470){
   canvas.style.cursor = "pointer";}
           }
     }
 //Συνάρτηση δείκτη ποντικιού όταν κάνει κλικ.
 function mousedownHandler(event)
{
 // Βρίσκει τη θέση του mouseX , mouseY
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;
  //Αν το stage=0 και κάνεις κλικ στην αντίστοιχη περιοχή stage=1
 if(stage==0){
  if (mouseX>300 && mouseX<500  && mouseY>380 && mouseY<460){
     stage=1
     mouseY=0}
           }
    ////Αν το stage=1 και κάνεις κλικ στην αντίστοιχη περιοχή θα ανοίξει το
lesson75.html
    if(stage==2){
```

```
      if (mouseX>300 && mouseX<520  && mouseY>380 && mouseY<470){
        window.location.href = "lesson75.html";}
                                                                    }

                                                                          }
//Χρήση πληκτρολογίου
window.addEventListener("keydown", function(event)
{
 switch(event.keyCode)
 {
    case UP:
     moveUp = true;
     break;

    case DOWN:
     moveDown = true;
    break;

    case LEFT:
     moveLeft = true;
     break;

     case RIGHT:
      moveRight = true;
     break;

 }
}, false);

window.addEventListener("keyup", function(event)
{
 switch(event.keyCode)
 {
     case UP:
     moveUp = false;
     break;

     case DOWN:
     moveDown = false;
    break;

     case LEFT:
      moveLeft = false;
    break;

    case RIGHT:
```

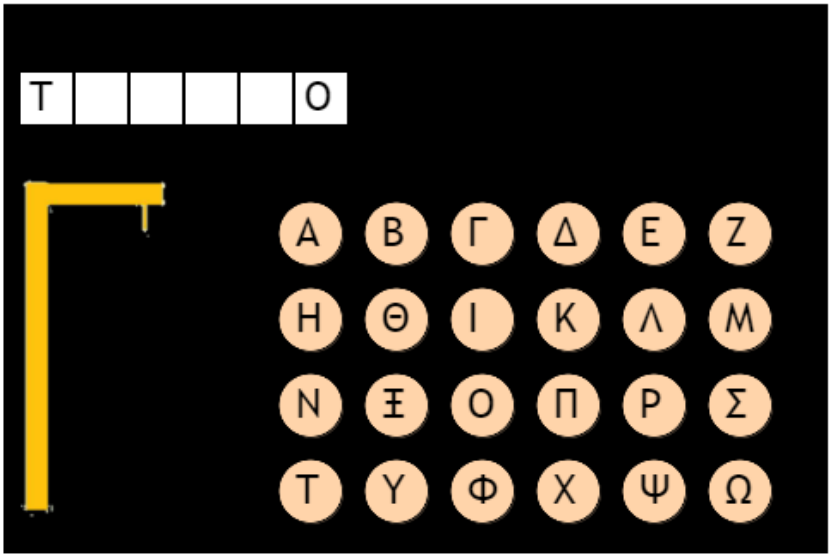
```
      moveRight = false;
     break;

  }
}, false);

update();
//Ανανέωση καμβά
function update() {
requestAnimationFrame(update, canvas);
//Αν Η μεταβλητή stage είναι 1
if(stage==1){

if(moveDown)
 {
   hero.vy=3
  hero.vx = 0;
 }
 if(moveUp)
 {
   hero.vy=-3
   hero.vx = 0;
 }

if(moveRight)
 {
hero.vx = 3
hero.vy = 0;
 }
  if(moveLeft)
 {
hero.vx=-3
hero.vy = 0;
 }

hero.x=hero.x+hero.vx;
hero.y=hero.y+hero.vy;

ctx.fillStyle="#00FFCC";
ctx.fillRect(0, 0, canvas.width, canvas.height);

//Εμφάνιση της εικόνας του αντικειμένου hero
ctx.drawImage(image1,hero.x,hero.y,hero.width,hero.height);
////Εμφάνιση της εικόνας του αντικειμένου key
ctx.drawImage(image2,key.x,key.y,key.width,key.height);
```

```
//Αν το αντικείμενο hero αγγίξει το αντικέιμενο key, η μεταβλητή stage θα παρειι
τιμή 2
if((hero.x+hero.width>key.x) && (hero.x<=key.x+key.width) &&
(hero.y+hero.height>=key.y) && (hero.y<=key.y+key.height)){
stage=2;}
}

//Αν η μεταβλητής stage=0
if(stage==0){
ctx.fillStyle="#00BFFF";
ctx.fillRect(0, 0, canvas.width, canvas.height);
ctx.font = "50px Comic Sans MS";
ctx.fillStyle= "#FFDAB9";
ctx.fillText( "ΜΕΝΟΥ",290,80);
//Δημιουργία επιλογής ΠΑΙΞΕ
ctx.drawImage(image1,310,150,150,160);
ctx.fillStyle= "#FFDAB9";
ctx.fillRect(300,380,200,80)
ctx.font = "35px Comic Sans MS";
ctx.fillStyle= "#9400D3";
ctx.fillText( "ΠΑΙΞΕ",330,430)

}

//Αν η μεταβλητής stage=2
if(stage==2){
ctx.fillStyle="#00BFFF";
ctx.fillRect(0, 0, canvas.width, canvas.height);
ctx.drawImage(image1,320,140,150,160);
//Δημιουργία επιλογής ΗΑΝΑΠΑΙΞΕ
ctx.fillStyle= "#FFDAB9";
ctx.fillRect(300,380,220,90)
ctx.font = "30px Comic Sans MS";
ctx.fillStyle= "#9400D3";
ctx.fillText( "ΞΑΝΑΠΑΙΞΕ",310,435)

}

}

</script>
</html>
```

*Στο φάκελο **LESSON7** το αρχείο **lesson75.html** που εμφανίζει το αποτέλεσμα.*

# 8.1 Κρεμάλα

Θα δούμε τώρα πως μπορούμε να δημιουργήσουμε το κλασσικό παιχνίδι της κρεμάλας.




Θα χρειαστούμε πρώτα από όλα ένα αρχείο στο οποίο θα εισάγουμε τις λέξεις τις οποίες θα πρέπει να βρει ο χρήστης στο παιχνίδι.
Θα δημιουργήσουμε ένα αρχείο .js , το οποίο θα έχει τη παρακάτω μορφή:

```
var w = [ "ΑΥΤΟΚΙΝΗΤΟ" , "ΑΕΡΟΠΛΑΝΟ" , "ΕΛΙΚΟΠΤΕΡΟ" , "ΒΑΡΚΑ"];
```

Θα αποθηκεύσουμε το αρχείο ως words.js

*Στο φάκελο **LESSON8** υπάρχει το αρχείο **words**.js με πολλές λέξεις .*

Ας δούμε τώρα τον συγκεντρωτικό κώδικα για την δημιουργία του παιχνιδιού.

```
<!doctype html>
 <meta charset="utf-8" />
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title>gamename</title>
<center>
<canvas width="480" height="320" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script src="words.js"></script>
<script>
var canvas = document.querySelector("canvas");
```

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

var table=new Array;
var face=new Array;
var q=new Array;

for(i=0;i<24;i++){
face[i]=true;}

for(i=1;i<20;i++){
q[i]="0";}

var l;
var r;
var temp;
var word;
var letters="ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ";
var step=1;
var k=0;
var stage=0;
var win=0;

var image1=new Image();
image1.src="img/circle.png";
var image2=new Image();
image2.src="img/kr1.png";
var image3=new Image();
image3.src="img/kr2.png";
var image4=new Image();
image4.src="img/kr3.png";
var image5=new Image();
image5.src="img/kr4.png";
var image6=new Image();
image6.src="img/kr5.png";
var image7=new Image();
image7.src="img/kr6.png";
var image8=new Image();
image8.src="img/kr7.png";

// Στην μεταβλητη l αποθηκεύω το μέγεθος του πίνακα w, από το script word
l=w.length;

//Στον πίνακα table αποθηκεύω τον πίνακα w
for(i=0;i<l;i++){
table[i]=w[i];}
```

```
//Ανακατεύω τον πίνακα table ώστε οι λέξεις να μην έχουν πάντα την ίδια σειρά
for(var i=0; i<l;i++){
r=Math.floor(Math.random()*l);
temp=table[i];
table[i]=table[r];
table[r]=temp;}

canvas.addEventListener("mousemove", mousemoveHandler, false);
canvas.addEventListener("mousedown", mousedownHandler, false);

function mousemoveHandler(event)
{
 //Find the mouse's x and y position
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;

if(stage==1){
 canvas.style.cursor = "default";
 for(i=0;i<6;i++){
 if (mouseX>160+i*50 && mouseX<198+i*50  && mouseY>115 && mouseY<153){
 canvas.style.cursor = "pointer";}
 }
 for(i=6;i<12;i++){
 if (mouseX>160+(i-6)*50 && mouseX<198+(i-6)*50  && mouseY>165 &&
mouseY<198){
 canvas.style.cursor = "pointer";}
 }
 for(i=12;i<18;i++){
 if (mouseX>160+(i-12)*50 && mouseX<198+(i-12)*50  && mouseY>215 &&
mouseY<248){
 canvas.style.cursor = "pointer";}
 }
 for(i=18;i<24;i++){
 if (mouseX>160+(i-18)*50 && mouseX<198+(i-18)*50  && mouseY>265 &&
mouseY<303){
 canvas.style.cursor = "pointer";}
 }

 }
}

function mousedownHandler(event)
{
 //Find the mouse's x and y position
 var mouseX = event.pageX - canvas.offsetLeft;
 var mouseY = event.pageY - canvas.offsetTop;
```

```
if(stage==1){
  //Διαδικασία επιλογής γράμματος από τον χρήστη για τα πρώτα 6 γράμματα(A-Z)
 for(i=0;i<6;i++){
//Αν face[i]==true το αντίστοιχο γράμμα μπορεί να επιλεχθεί
  if(face[i]==true){
    count=0;
   if (mouseX>160+i*50 && mouseX<198+i*50  && mouseY>115 && mouseY<153){
    key=letters[i];
   //Για κάθε θέση της λέξης ψάχνει να βρει αν υπάρχει το γράμμα
    for(j=1;j<len-1;j++){
            // Αν το γράμμα πο επέλεξες υπάρχει στην λέξη
            if(key==word[j]){
            q[j]=key;
             face[i]=false;}
           // Αν το γράμμα που επέλεξες δεν υπάρχει στην θέση j της λέξης
             if(key!=word[j]){
             count++;}
               }
           //Αν τελικά το γράμμα δεν υπάρχει σε καμμιά θέση της λέξης
            if(count==len-2){
            face[i]=false;
            step++}
              }

                        }
                            }
  //Διαδικασία επιλογής γράμματος από τον χρήστη για τα άλλα 6 γράμματα(H-M)
for(i=6;i<12;i++){
  if(face[i]==true){
    count=0;
   if (mouseX>160+(i-6)*50 && mouseX<198+(i-6)*50  && mouseY>165 &&
mouseY<198){
    key=letters[i];
    for(j=1;j<len-1;j++){
            if(key==word[j]){
             q[j]=key;
             face[i]=false;}
             if(key!=word[j]){
             count++;}
                }
           if(count==len-2){
               face[i]=false;
            step++}
             }

               }
```

```
                }
 //Διαδικασία επιλογής γράμματος από τον χρήστη για τα άλλα 6 γράμματα(Ν-Σ)
for(i=12;i<18;i++){
 if(face[i]==true){
  count=0;
  if (mouseX>160+(i-12)*50 && mouseX<198+(i-12)*50  && mouseY>215 &&
mouseY<253){
   key=letters[i];
   for(j=1;j<len-1;j++){
           if(key==word[j]){
            q[j]=key;
            face[i]=false;}
            if(key!=word[j]){
            count++;}
             }
          if(count==len-2){
            face[i]=false;
           step++}
          }

             }
               }
 //Διαδικασία επιλογής γράμματος από τον χρήστη για τα άλλα 6 γράμματα(Τ-  Ω)
for(i=18;i<24;i++){
 if(face[i]==true){
  count=0;
  if (mouseX>160+(i-18)*50 && mouseX<198+(i-18)*50  && mouseY>265 &&
mouseY<303){
   key=letters[i];
   for(j=1;j<len-1;j++){
           if(key==word[j]){
            q[j]=key;
            face[i]=false;}
            if(key!=word[j]){
            count++;}
              }
          if(count==len-2){
          face[i]=false;
           step++}
             }

                }
                     }

                     }
```

```
                        }

update();

function update() {
requestAnimationFrame(update, canvas);

if(stage==0){

//Το k=0 , οπότε η πρώτη λέξη που θα επιλεχθεί είναι η table[0] που αποθηκεύεται στην μεταβλητή word
word=table[k];
//Στην μεταβλητή len αποθηκεύεται το μέγεθος της λέξης word
len=word.length;

// Στη θέση q[0] αποθηκεύεται το πρώτο γράμμα της λέξης word
q[0]=word[0];
//Στη θέση q[len-1] , αποθηκεύται το τελευταίο γράμμα της λέξης word
q[len-1]=word[len-1];
stage=1;}

if(stage==1){

ctx.fillStyle= "black";
ctx.fillRect(0, 0, canvas.width, canvas.height);

//Εμφάνιση των γραμμάτων της λέξης. Στην αρχή εμφανίζει μόνο το πρώτο και το τελευταίο γράμμα, και στην πορεία εμφανίζει τα γράμματα που βρίσκει  ο χρήστης
for(i=0;i<len;i++)
{ctx.fillStyle= "white";
ctx.fillRect(10+i*32,40,30,30);
}
for(i=0;i<len;i++){
 if(q[i]!="0"){
ctx.font = "24px Trebuchet MS";
ctx.fillStyle= "black";
ctx.fillText(q[i] ,15+i*32,62);
}
}
//Εμφανίζει όλα τα 24 γράμματα του αλφάβητου μέσα σε κυκλάκια-εικόνες- για να μπορεί να επιλέξει γράμμα ο χρήστης. Εμφανίζει τα γράμματα μόνο αν το αντίστοιχο face[i]==true , που σημαίνει ότι το γράμμα αυτό δεν έχει ξαναεπιλεγεί
for(i=0;i<6;i++){
ctx.drawImage(image1,160+i*50,115,38,38);
if(face[i]==true){
ctx.font = "24px Trebuchet MS";
```

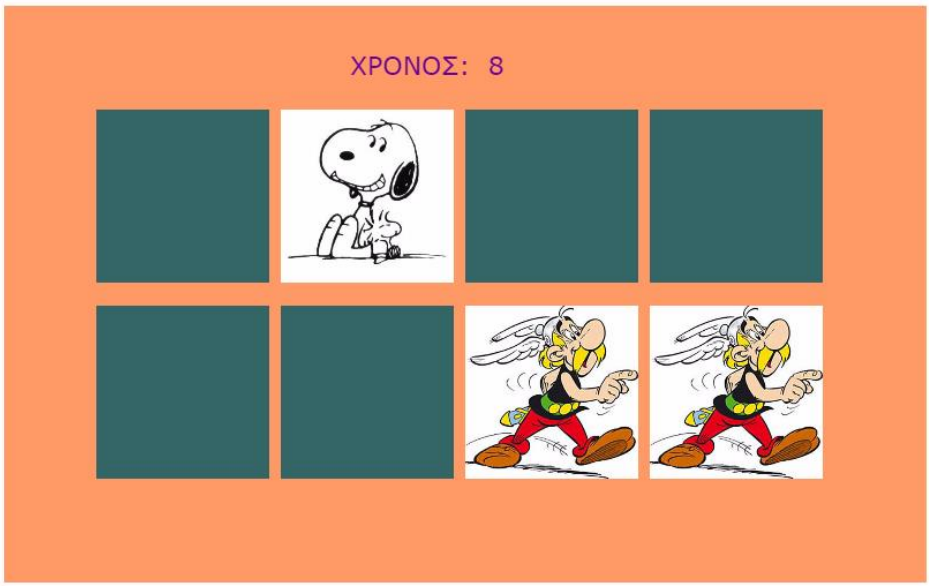
```
ctx.fillStyle= "black";
ctx.fillText(letters[i],170+i*50,141);}
}
for(i=6;i<12;i++){
ctx.drawImage(image1,160+(i-6)*50,165,38,38);
if(face[i]==true){
ctx.font = "24px Trebuchet MS";
ctx.fillStyle= "black";
ctx.fillText(letters[i],170+(i-6)*50,191);}
}
for(i=12;i<18;i++){
ctx.drawImage(image1,160+(i-12)*50,215,38,38);
if(face[i]==true){
ctx.font = "24px Trebuchet MS";
ctx.fillStyle= "black";
ctx.fillText(letters[i],170+(i-12)*50,241);}
}
for(i=18;i<24;i++){
ctx.drawImage(image1,160+(i-18)*50,265,38,38);
if(face[i]==true){
ctx.font = "24px Trebuchet MS";
ctx.fillStyle= "black";
ctx.fillText(letters[i],170+(i-18)*50,291);}
}

//Η μεταβλητή step δηλώνει τα στάδια που βρίσκεται το παιχνίδι ώστε να
εμφανίσει την αντίστοιχη φωτογραφία της κρεμάλας
if(step==1){
ctx.drawImage(image2,10,100,160,200);
}
if(step==2){
ctx.drawImage(image3,10,100,160,200);
}
if(step==3){
ctx.drawImage(image4,10,100,160,200);
}
if(step==4){
ctx.drawImage(image5,10,100,160,200);
}
if(step==5){
ctx.drawImage(image6,10,100,160,200);
}
if(step==6){
ctx.drawImage(image7,10,100,160,200);
}
if(step==7){
```

```
ctx.drawImage(image8,10,100,160,200);
}

//Αν step==3 έχασες
if(step==7){
stage=3;}

win=0;
//Κάθε φορά που επιλέγει ο χρήστης σωστό γράμμα η μεταβλητή win αυξάνει κατά
1
for(i=0;i<len-1;i++){
if(q[i]==word[i]){
win++;}
}
//Αν win==με το μήκος της λέξης κέρδισες
if(win==len-1){
stage=2;
}
}
if(stage==2){
ctx.fillStyle= "black";
ctx.fillRect(0, 0, canvas.width, canvas.height);
ctx.font = "30px Trebuchet MS";
ctx.fillStyle= "white";
ctx.fillText("ΝΙΚΗΣΕΣ" ,140,150);

}

if(stage==3){
ctx.fillStyle= "black";
ctx.fillRect(0, 0, canvas.width, canvas.height);
ctx.font = "30px Trebuchet MS";
ctx.fillStyle= "white";
ctx.fillText("ΕΧΑΣΕΣ" ,130,150);
ctx.font = "24px Trebuchet MS";
ctx.fillText("Η λέξη που έψαχνες είναι:" ,120,190);
ctx.fillText(word ,120,230);
}

}

</script>
</html>
```

*Στο φάκελο **LESSON8** το αρχείο **lesson81.html** που εμφανίζει το αποτέλεσμα.*

# 8.2 Παιχνίδι Memory(Μνήμη)

Θα δούμε τώρα πως θα δημιουργήσουμε ένα παιχνίδι Μνήμης Ταυτόχρονα θα δείξουμε και ένα παράδειγμα αποθήκευσης σκορ με τοπική αποθήκευση(Local Storage).




```
<!doctype html>
<meta charset=”UTF-8”>
<title>memory</title>
<CENTER>
<canvas width="800" height="500" ></canvas>
<script src="requestAnimationFramePolyfill.js"></script>
<script src="main.js"></script>
<script>
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

//Δημιουργία αντικειμένου
var spriteObject =
{
  x: 0,
  y: 0,
  width: 150,
  height: 150
};

//Δημιουργία 2 πινάκων
var table=new Array;
var flag=new Array;
```

```
//Τα 8 πρώτα στοιχεία του πίνακα flag έχουν τιμή false
for(var i=1; i<9;i++){
flag[i]=false;}

//Δημιουργία 1ης κάρτας μνήμης
var card1= Object.create(spriteObject);
card1.info=1;
card1.image = new Image();
card1.image.src = "img/asterix.jpg";
table[1]=card1;
table[2]=card1;

//Δημιουργία 2ης κάρτας μνήμης
var card2= Object.create(spriteObject);
card2.info=2;
card2.image = new Image();
card2.image.src = "img/mafalda.png";
table[3]=card2;
table[4]=card2;

//Δημιουργία 3ης κάρτας μνήμης
var card3= Object.create(spriteObject);
card3.info=3;
card3.image = new Image();
card3.image.src = "img/rantaplan.jpg";
table[5]=card3;
table[6]=card3;

//Δημιουργία 4ης κάρτας μνήμης
var card4= Object.create(spriteObject);
card4.info=4;
card4.image = new Image();
card4.image.src = "img/snoopy.jpg";
table[7]=card4;
table[8]=card4;

var flagr=true;
var flagplay=true
var r;
var temp;
var count=0;
var choise1;
var choise2;
var r1;
var r2;
```

```
var t=0;
var seconds;
var start = new Date();
var starttime = Number(start.getTime());
var now;
var nt;
var score=0;
var dt;

canvas.addEventListener("mousemove", mousemoveHandler, false);
canvas.addEventListener("mousedown", mousedownHandler, false);

function mousemoveHandler(event)
{
  //Find the mouse's x and y position
  var mouseX = event.pageX - canvas.offsetLeft;
  var mouseY = event.pageY - canvas.offsetTop;

  //ο κέρσορας είναι default και γινεται σε μορφή χεράκι στις παρακάτω
περιτπώσεις.
 canvas.style.cursor = "default";
//Οι τέσσερις πρώτες κάρτες
for(var i=1;i<5;i++){
  if(mouseX>160*i-80 && mouseX<160*i+70 && mouseY>90 && mouseY<240){
  canvas.style.cursor = "pointer";}
 }
//Οι υπόλοιπες τέσσερις κάρτες
for(var i=5;i<9;i++){
  if(mouseX>160*(i-4)-80 && mouseX<160*(i-4)+70 && mouseY>260 &&
mouseY<410){
  canvas.style.cursor = "pointer";}
 }

 }

function mousedownHandler(event)
{
  //Find the mouse's x and y position
  var mouseX = event.pageX - canvas.offsetLeft;
  var mouseY = event.pageY - canvas.offsetTop;

for(var i=1;i<5;i++){
//Όταν το ποντίκι είναι στη περιοχή, είναι κλειστήη κάρτα(flag[i]=false) και η
μεταβλητή count είναι μικρότερη του 2
  if(mouseX>160*i-80 && mouseX<160*i+70 && mouseY>90 && mouseY<240 &&
flag[i]==false && count<2){
```

```
  if(count==1){
   count=count+1;
   r2=i;
   choise2=table[i].info}
 if(count==0){
   count=count+1;
   r1=i;
   choise1=table[i].info}
 flag[i]=true;}
 }
for(var i=5;i<9;i++){
 if(mouseX>160*(i-4)-80 && mouseX<160*(i-4)+70 && mouseY>260 &&
mouseY<410 && flag[i]==false && count<2){
  if(count==1){
   count=count+1;
   r2=i;
   choise2=table[i].info}
 if(count==0){
   count=count+1;
   r1=i;
   choise1=table[i].info}
 flag[i]=true;}
 }

 }

update();

function update() {
requestAnimationFrame(update, canvas);


//Ανακάτεμα των καρτών
if(flagr==true){
for(var i=1; i<9;i++){
r=Math.floor(Math.random()*8)+1;
temp=table[i];
table[i]=table[r];
table[r]=temp;
}
}
flagr=false;

//Έναρξη χρόνου
 now = new Date();
 nt = Number(now.getTime());
```

```
 seconds = Math.floor(.5+(nt-starttime)/1000);
//’όταν ο χρήστης αποτύχει να επιλέξει 2 ίδιες κάρτες
if(choise1!=choise2 && count==2){
t=t+1;
if(t>30){
flag[r1]=false;
flag[r2]=false;
count=0;
t=0;
}
}
//Όταν ο χρήστης πετυχαίνει να επιλέξει 2 ίδιες κάρτες
if(choise1==choise2 && count==2){
flag[r1]=true;
flag[r2]=true;
score++;
count=0;}

ctx.fillStyle="#FF9966";
ctx.fillRect(0, 0, canvas.width, canvas.height);

// Αν για κάθε κάρτα το αντίστοιχο flag[i]==false, τότε η κάρτα εμφανίζεται κλειστή.
Αν flag[i]==true η αντίστοιχη κάρτα είναι ανοιχτή
for(var i=1;i<5;i++){
if(flag[i]==false){
ctx.fillStyle= "#336666";
ctx.fillRect(160*i-80,90,150,150);
}
if(flag[i]==true){
ctx.drawImage
    (table[i].image,(160*i-80),90,150, 150);
}

}
for(var i=5;i<9;i++){
if(flag[i]==false){
ctx.fillStyle= "#336666";
ctx.fillRect(160*(i-4)-80,260,150,150);}
if(flag[i]==true){
ctx.drawImage
    (table[i].image,(160*(i-4)-80),260,150, 150);}

}

ctx.font = "22px Verdana";
ctx.fillStyle= "#660099";
```

```
//Ο χρήστης χάνει
if(score<4){
ctx.fillText("ΧΡΟΝΟΣ:  "+seconds ,300,60);}
//'Οχρήστης κερδίζει
if(score==4){
 if(flagplay==true){
  time=seconds;
  flagplay=false;}
ctx.fillText("Μπραβο! Ο χρόνος σου είναι:  "+time+"  δευτερόλεπτα" ,100,60);}

//Διαιδικασία για τοπικό high score
dt=localStorage.getItem("ttime");
if(dt==null){
dt=3000;}
if(time<dt){
localStorage.setItem("ttime",time);
dt=localStorage.getItem("ttime");}

ctx.fillStyle= "#8B0000";
ctx.fillText("Ο καλύτερος χρόνος είναι:  "+dt+"  δευτερόλεπτα" ,100,470);

}
</script>
</html>
```

*Στο φάκελο **LESSON8** το αρχείο **lesson81.html** που εμφανίζει το αποτέλεσμα.*

# 9.1 Touchscreen

Στα παραπάνω κεφάλαια είδαμε πως μπορούμε να χρησιμοποιούμε το ποντίκι σε ένα παιχνίδι ή εφαρμογή.
Τώρα θα δούμε ποιες είναι οι εντολές και τι αλλάζει, όταν χρησιμοποιούμε οθόνη αφής. Δηλαδή να διαχειριζόμαστε μια εφαρμογή με το άγγιγμα της οθόνης. Η λογική είναι σχεδόν η ίδια αλλά υπάρχουν κάποιες μικρές διαφοροποιήσεις.

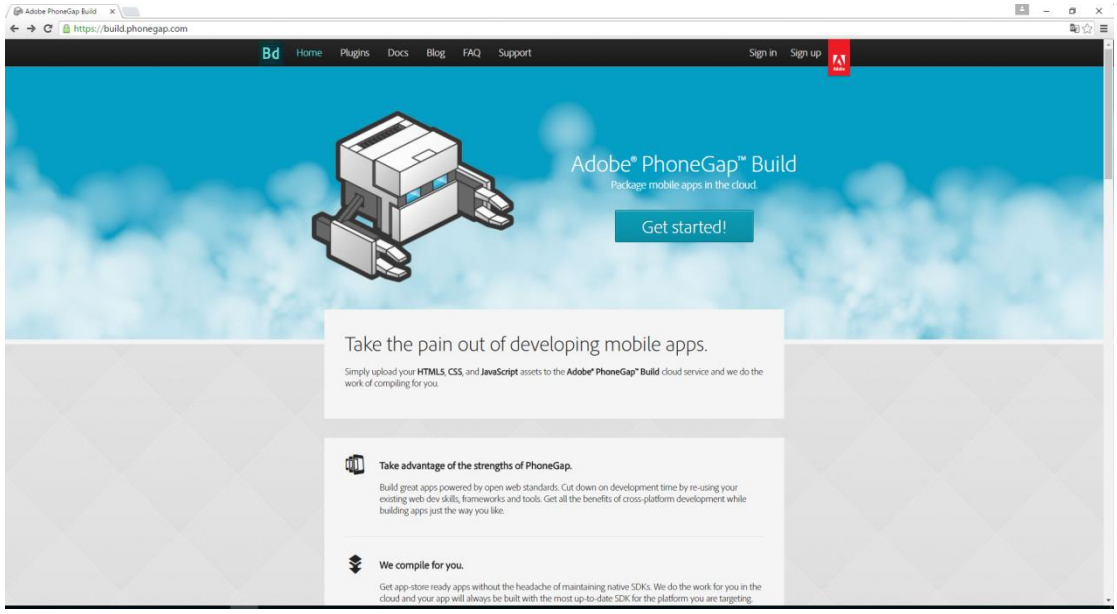
```
//Όταν ακουμπάς το δάχτυλο σου στην οθόνη
canvas.addEventListener("touchstart", touchstartHandler, false);
//Όταν παίρνεις το δάχτυλο σου από την οθόνη
canvas.addEventListener("touchend", touchstartHandler, false);
//Όταν κουνάς το δάχτυλο σου στην οθόνη
canvas.addEventListener("touchmove", touchstartHandler, false);

function touchstartHandler(event)
{
 //Βρες τις θέσης x και y του ποντικιού.
 //Subtract the canvas's top and left offset
 touchX = event.targetTouches[0].pageX - canvas.offsetLeft;
 touchY = event.targetTouches[0].pageY - canvas.offsetTop;

        event.preventDefault();

ΚΩΔΙΚΑΣ
}

function touchendHandler(event)
{
 //Find the touch point's x and y position.
 touchX = event.targetTouches[0].pageX - canvas.offsetLeft;
 touchY = event.targetTouches[0].pageY - canvas.offsetTop;

        event.preventDefault();

ΚΩΔΙΚΑΣ
}

function touchmoveHandler(event)
{
 //Find the touch point's x and y position.
 touchX = event.targetTouches[0].pageX - canvas.offsetLeft;
 touchY = event.targetTouches[0].pageY - canvas.offsetTop;

        event.preventDefault();
```

ΚΩΔΙΚΑΣ
}

Η λογική είναι ίδια με την λογική του mouse, μόνο που αντί για mouseX βάζουμε touchX και αντί για mouseY βάζουμε touchY.

Με αφή πατάμε στην οθόνη στη περιοχή του καμβά και αναλόγως σε ποια περιοχή έχουμε πατήσει θα στηθεί αλγοριθμικά μια διαδικασία, όπως στις παραπάνω εφαρμογές.

# 9.2 Phonegap build




Για να δημιουργήσουμε ένα application για android και ios θα χρησιμοποιήσουμε το Phonegap build. Είναι μια online εφαρμογή που μετατρέπει κώδικα HTML5 + JavaScript σε apps. Υπάρχουν και άλλοι τρόποι αλλά το phonegap build είναι ο πιο απλός.

Στην σελίδα https://build.phonegap.com/ κάνετε εγγραφή και ακλουθώντας τις απλές οδηγίες φτάνετε στο επιθυμητό αποτέλεσμα.

Μπορείτε δωρεάν να έχετε ένα application ενώ με μια συνδρομή μπορείτε να έχετε περισσότερες.
Για αρχή μια δωρεάν εφαρμογή είναι αρκετή.
Ας δούμε τώρα μερικά στοιχεία που θα σας κάνουν τη ζωή πιο εύκολη.
Χρησιμοποιείστε ανάλυση 480 X 320.

Αφού τελειώσετε με το παιχνίδι(ΠΡΟΣΟΧΗ!! Με TouchX και TouchY.), δημιουργείστε ένα φάκελο με όποιο όνομα θέλετε και βάλτε μέσα τα παρακάτω:

1, Το αρχείο index.html , που έχει τον κώδικα του παιχνιδιού.

2. Φάκελο με ονομασία img , όπου θα υπάρχουν όλες οι εικόνες που χρησιμοποιείτε.

3. Φάκελο με ονομασία icons, που θα περιέχει τις εικόνες(εικονίδια) του παιχνιδιού που θα φαίνεται στο PlayStore, σε 5 διαφορετικές αναλύσεις.

36X36 , 48X48 , 72X72 , 96X96 , 144X144

4. Φάκελο με ονομασία splash(Προαιρεικά) , που θα περίεχει την εικόνα του λογότυπου σας σε 4 αναλύσεις.

356X200 , 569X320 , 853X480 , 1280X768

5. Αρχείο με την ονομασία icon.png που θα περιέχει το εικονίδιο που θα φαίνεται στο PlayStore σε ανάλυση 512X512

6. Το αρχείο requestAnimationFramePolyfill.js και όποια άλλα αρχεία js θα χρειαστεί επιπλέον η εφαρμογή μας.

7. Ένα αρχείο config.xml το οποίο θα είναι υπέυθυνο για την δημιουργία της mobile εφαρμογής, το οποίο περιγράφεται παρακάτω.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <widget xmlns   = "http://www.w3.org/ns/widgets"
    xmlns:gap   = "http://phonegap.com/ns/1.0"
    xmlns:android   = "http://schemas.android.com/apk/res/android"
    id   = "mygame01"
    versionCode = "10"
    version    = "1.0.0" >

<platform name="android" />

//Κάθε φορά πρεέπι να μπαίνει η τελευταία έκδοση
<preference name="phonegap-version" value="cli-5.2.0" />

<preference name="orientation" value="landscape" />
<preference name="fullscreen" value="true" />

<preference name="android-minSdkVersion" value="7" />
<preference name="android-maxSdkVersion" value="24" /> //Με την έκδοση
νεώτερων εκδόσεων android, αυτό αλλάζει.
<preference name="android-targetSdkVersion" value="17" />
<preference name="target-device" value="universal" />
<preference name="prerendered-icon" value="true" />
<preference name="deployment-target" value="7.0" />

<plugin name="cordova-plugin-device" />

<supports-screens android:anyDensity="true" android:resizeable="true"
  android:smallScreens="true"
  android:normalScreens="true"
  android:largeScreens="true"
  android:xlargeScreens="true"
  />
```

```
 <gap:splash src="splash/android/ldpi.png" gap:platform="android"
gap:density="ldpi" />
<gap:splash src="splash/android/mdpi.png" gap:platform="android"
gap:density="mdpi" />
<gap:splash src="splash/android/hdpi.png" gap:platform="android"
gap:density="hdpi" />
<gap:splash src="splash/android/xhdpi.png" gap:platform="android"
gap:density="xhdpi" />


  <icon src="icon.png" />
 <icon src="icons/android/ldpi.png" gap:platform="android" gap:density="ldpi" />
  <icon src="icons/android/mdpi.png" gap:platform="android" gap:density="mdpi"
/>
  <icon src="icons/android/hdpi.png" gap:platform="android" gap:density="hdpi" />
  <icon src="icons/android/xhdpi.png" gap:platform="android" gap:density="xhdpi"
/>
  <icon src="icons/android/xxhdpi.png" gap:platform="android"
gap:density="xxhdpi" />

<name>My game </name>
   <author href="your site" email="your email">
     your name
   </author>

</widget>
```

Μετατρέψτε τον φάκελο σε συμπιεσμένο φάκελο zip, και είστε έτοιμοι.

Στον φάκελο PG υπάρχει μια απλή εφαρμογή έτοιμη για compile(μετατροπή σε app), με όλα αυτά που περιγράφονται παραπάνω, που θα σας φανεί χρήσιμη στην πρώτα σας βήματα.

Για περισσότερες πληροφορίες μπορείτε να μπείτε στη σελίδα του PhoneGap builder, όπου υπάρχουν αναλυτικά πληροφορίες. https://build.phonegap.com/

# 10. Η JavaScript ΣΥΝΟΠΤΙΚΑ

ΜΕΤΑΒΛΗΤΕΣ:
var x=12 //ΑΚΕΡΑΙΑ
var y="abc" //ΧΑΡΑΚΤΗΡΑΣ
var c=true; //ΛΟΓΙΚΗ
var table=new Array; //ΠΙΝΑΚΑΣ

ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕΣΤΕΣ:
+ , - , * , / , =

ΣΥΓΚΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ:

| | |
|---|---|
| ΜΕΓΑΛΥΤΕΡΟ | > |
| ΜΙΚΡΟΤΕΡΟ | < |
| ΜΕΓΑΛΥΤΕΡΟ ΙΣΟ | >= |
| ΜΙΚΡΟΤΕΡΟ ΙΣΟ | <= |
| ΙΣΟ ΣΕ ΣΥΝΘΗΚΗ | == |
| ΔΙΑΦΟΡΟ | != |

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ:

| | |
|---|---|
| ΚΑΙ | && |
| Ή | \|\| |
| ΟΧΙ | ! |

x++ // Το x αυξάνει κατά 1
x-- //Το x μειώνεται κατά 1

mod: %

## ΕΝΤΟΛΗ If(Αν)

```
if(ΙΧΥΕΙ ΣΥΝΘΗΚΗ)
{
  ΕΝΤΟΛΗ
}
```

ΠΑΡΑΔΕΙΓΜΑ

```
If(x==5){
y=4;}
```

## ΕΝΤΟΛΗ For(Για)

for(i=τιμή1;i<τιμή2;i++){
ΕΝΤΟΛΕΣ}

ΠΑΡΑΔΕΙΓΜΑ

```
for (i=1;i<7;i++){
window.alert("hello")
}
```

## ΕΝΤΟΛΗ While(Όσο)

While(ΙΣΧΥΕΙ ΣΥΝΘΗΚΗ){
ΕΝΤΟΛΕΣ}

ΠΑΡΑΔΕΙΓΜΑ

```
while(x<10){
flag1=false;}
```

Ο αναγνώστης για περισσότερες πληροφορίες σχετικά με τη JavaScript καλό είναι να ανατρέξει στη διεύθυνση:

http://www.w3schools.com/js/

# ***ΕΠΙΛΟΓΟΣ***

Είναι σημαντικό ο αναγώστης να κατεβάσει τα έτοιμα αρχεία του βιβλίου από τη διεύθυνση: http://www.edaskalos.gr/book.html . Τα αρχεία αυτά περιέχουν τον κώδικα των εφαρμογών του βιβλίου έτσι ώστε ο αναγνώσης να διευκολυνθεί στη μελέτη του βιβλίου.
Ένα από τα μεγαλύτερα προβλήματα του προγραμματισμού είναι τα συντακτικά λάθη λόγω απροσεξίας. Ο Προγραμματιστής πρέπει να προσέχει για να αποφεύγει τέτοια λάθη.
Τα έτοιμα αρχεία του βιβλίου λύνουν το πρόβλημα του αναγνώστη αφού θα έχει τον κώδικα των εφαρμογών χωρίς συντακτικά λάθη, κάτι που θα του κάνει τη ζωή πιο εύκολη.

Η πληροφορική και ειδικότερα ο προγραμματισμός είναι μια δυναμική επιστήμη που όλα αλλάζουν και βετιώνονται γρήγορα. Ο αναγνώστης πρέπει να ψάχνει στο διαδίκτυο συνεχώς για νέες τεχνικές που θα τον διευκολύνουν στον προγραμματισμό..
Η HTML5 και η JavaScript είναι γλώσσες που θα αποσχολήσουν τον κόσμο της πληροφορικής για πολλά ακόμα χρόνια οπότε ο αναγνώστης θα έχει την ευκαιρία να ασχοληθεί εκτενέστερα.

Στο βιβλίο περιγράψαμε μια λογική προγραμματισμού η οποία ελπίζουμε ότι θα δώσει στον αναγνώστη την ώθηση που χρειάζεται για να ξεκινήσει το ταξίδι του στον μαγικό κόσμο του game developing.
Δεν πρέπει να ξεχνάμε ότι στην αλγοριθμική και γεννικά στον προγραμματισμό υπάρχουν συνήθως πολλοί τρόποι για να φτάσει κανείς σε ένα επιθυμητο αποτέλεσμα. Δεν χρειάζεται να αντιγράφεις την τεχνική οποιουδήποτε, αλλά να αποκτήσεις μια δική σου τεχνική με την οποία θα φτάνεις στο ζητούμενο εύκολα, με την δική σου λογική.
Το ζητούμενο στην δημιουργία παιχνιδιών, πέρα πό τις γνωστικές δεξιότητες, είναι η φαντασία του προγραμματιστή και η επιθυμία που πρέπει να έχει, ώστε να δημιουργήσει από το μηδεν μια εφαρμογή. Μια εφαρμογή που θα τον αναγκάσει να χρησιμοποιήσει όλη του την αλγοριθμική δημιουργικότητα.

# ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΡΟΤΕΙΝΟΜΕΝΑ ΒΙΒΛΙΑ

Rex van der Spuy – Foundation GAME DESIGN with HTML5 & JavaScript , Apress 2012

Rob Hawkes – HTML5 Canvas for Games and Entertainment , Apress 2011

Billy Lamberta, Keith Paters – Foundation HTML5 Animation with javaScript , Apress 2011

David Geary – Core HTML5 Canvas , Prentice Hall 2012

Eric Rowell – HTML5 Canvas Cookbook , Packt Publishing 2011

Jeanine Meyer – Games to Learn HTML5 and JavaScript , Apress 2010

Juriy Bura - Pro Android Web Game Apps , Apress 2012

Marijn Haverbeke – Eloquent JavaScript , O Reilly 2011

David Flanagan – JavaScript The Definitive Guide , O Reilly 2011

ISBN: 978-618-00-1734-2

Κωνσταντίνος Παπαστεργίου

2017