WebGL

05. Ενσωματωση Κωδικα στην HTML

Παυλος Δ. Βασιλειαδης
MSc, PhD

HTML5 CANVAS

WebGL

- Για τη δημιουργία γραφικών εφαρμογών στο διαδίκτυο, η HTML-5 παρέχει ένα πλούσιο σύνολο δυνατοτήτων, όπως 2D Canvas, WebGL, SVG, 3D μετασχηματισμούς CSS και SMIL. Για να γράψουμε εφαρμογές WebGL, χρησιμοποιούμε το υπάρχον στοιχείο canvas της HTML-5.

- Η εντολή της HTML-5 **<canvas>** παρέχει μια εύκολη και ισχυρή επιλογή για τη σχεδίαση γραφικών με χρήση JavaScript. Μπορεί να χρησιμοποιηθεί για τον σχεδιασμό γραφημάτων, συνθέσεων φωτογραφιών ή τη δημιουργία απλών και πιο σύνθετων κινούμενων εικόνων.

- Ακολουθεί ένα απλό στοιχείο <canvas> που έχει μόνο δύο συγκεκριμένα χαρακτηριστικά width και height καθώς και όλα τα βασικά χαρακτηριστικά της HTML-5 όπως id, name και class.

HTML5 CANVAS

WebGL

- Ακολουθεί ένα απλό στοιχείο <canvas> που έχει μόνο δύο συγκεκριμένα χαρακτηριστικά width και height καθώς και όλα τα βασικά χαρακτηριστικά της HTML-5 όπως id, name και class.

- **Σύνταξη**

- Η σύνταξη της ετικέτας HTML canvas δίνεται παρακάτω. Πρέπει να αναφέρετε το όνομα του καμβά μέσα σε διπλά εισαγωγικά (" ").

```
<canvas id = "mycanvas" width = "100" height = "100"></canvas>
```

HTML5 CANVAS

WebGL

- **Χαρακτηριστικά της εντολής <canvas>**
  - Η ετικέτα canvas έχει τρία χαρακτηριστικά, δηλαδή, id, width και height.
    - **id** - Το id αντιπροσωπεύει το αναγνωριστικό του στοιχείου canvas στο Document Object Model (DOM).
    - **width** - Το width αντιπροσωπεύει το πλάτος του καμβά.
    - **height** - Το height αντιπροσωπεύει το ύψος του καμβά.
- Αυτά τα χαρακτηριστικά καθορίζουν το μέγεθος του καμβά. Εάν ένας προγραμματιστής δεν τα καθορίζει κάτω από την ετικέτα canvas, τότε προγράμματα περιήγησης όπως ο Firefox, ο Chrome και το Web Kit, από προεπιλογή, παρέχουν ένα στοιχείο canvas μεγέθους 300 × 150.

>> Για το **Document Object Model (DOM)**, βλ. https://www.w3.org/TR/WD-DOM/introduction.html .

HTML5 CANVAS

WebGL

- **Παράδειγμα 01 - Δημιουργία καμβά / 01_creating_canvas**

- Ο παρακάτω κώδικας δείχνει πώς μπορεί να δημιουργηθεί ένας καμβάς.
- Χρησιμοποιείται η CSS για να δώσουμε ένα χρωματιστό περίγραμμα στον καμβά.

```
<html>
  <head>
    <style>
      #mycanvas{border:1px solid red;}
    </style>
  </head>
  <body>
    <canvas id = "mycanvas" width = "100" height = "100"></canvas>
  </body>
</html>
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 02 - Δημιουργία καμβά & χρήση της getContext() / 02_get_context**

- Το στοιχείο <canvas>:
  - είναι αρχικά κενό. Για να εμφανίσουμε κάτι στο στοιχείο canvas, πρέπει να χρησιμοποιήσουμε μια γλώσσα σεναρίων (scripting language). Αυτή η γλώσσα σεναρίων θα πρέπει να έχει πρόσβαση στο rendering context και να σχεδιάζει σε αυτό.
  - διαθέτει μια μέθοδο DOM που ονομάζεται **getContext()**, η οποία χρησιμοποιείται για να ληφθεί το rendering context και οι λειτουργίες σχεδίασης. Αυτή η μέθοδος λαμβάνει μία παράμετρο, τον type of context **2d**.

```
<html>
   <body>
      <canvas id = "mycanvas" width = "600" height = "200"></canvas>
      <script>
         var canvas = document.getElementById('mycanvas');
         var context = canvas.getContext('2d');
         context.font = '20pt Calibri';
         context.fillStyle = 'green';
         context.fillText(`Hello world!', 70, 70);
      </script>
   </body>
</html>
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 03 - Δημιουργία καμβά & χρήση εντολών σχεδιασμού #1/3 / 03_canvas_drawing**

```
<!DOCTYPE HTML>

<html>
   <head>

      <style>
         #test {
            width: 100px;
            height:100px;
            margin: 0px auto;
         }
      </style>

      <script type = "text/javascript">
         function drawShape() {

            // χρήση του στοιχείου canvas με χρήση του DOM
            var canvas = document.getElementById('mycanvas');

            // επιβεβαίωση ότι δεν θα γίνει η εκτέλεση του κώδικα αν δεν υποστηρίζεται το canvas
            if (canvas.getContext) {
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 03 - Δημιουργία καμβά & χρήση εντολών σχεδιασμού #2/3 / 03_canvas_drawing**

```
                // χρήση της εντολής getContext για τον σχεδιασμό μέσω του canvas
                var ctx = canvas.getContext('2d');

                // σχεδιασμός σχημάτων
                ctx.beginPath();
                ctx.arc(75,75,50,0,Math.PI*2,true);  // εξωτερικός κύκλος

                ctx.moveTo(110,75);
                ctx.arc(75,75,35,0,Math.PI,false);   // στόμα

                ctx.moveTo(65,65);
                ctx.arc(60,65,5,0,Math.PI*2,true);  // αριστερό μάτι

                ctx.moveTo(95,65);
                ctx.arc(90,65,5,0,Math.PI*2,true);  // δεξί μάτι
                ctx.stroke();
            } else {
                alert('Δυστυχώς χρειάζεσαι Safari ή Firefox 1.5+ για να δεις το σχήμα.');
            }
        }
    </script>
</head>
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 03 - Δημιουργία καμβά & χρήση εντολών σχεδιασμού #3/3 / 03_canvas_drawing**

```
   <body id = "test" onload = "drawShape();">
      <canvas id = "mycanvas"></canvas>
   </body>

</html>
```

>> Για την εντολή **arc()**, βλ. https://www.w3schools.com/tags/canvas_arc.asp .

>> Για το attribute **onload**, βλ. https://www.javatpoint.com/javascript-onload .

HTML5 CANVAS

Μέθοδοι και Περιγραφή:

Μέθοδος / Περιγραφή

beginPath()
Αυτή η μέθοδος επαναφέρει την τρέχουσα διαδρομή.

moveTo(x, y)
Αυτή η μέθοδος δημιουργεί ένα νέο υπομονοπάτι (subpath) με το δεδομένο σημείο.

closePath()
Αυτή η μέθοδος χαρακτηρίζει το τρέχον υπομονοπάτι ως κλειστό και ξεκινά ένα νέο υπομονοπάτι με σημείο το ίδιο με την αρχή και το τέλος του πρόσφατα κλειστού υπομονοπατιού.

fill()
Αυτή η μέθοδος γεμίζει τα υπομονοπάτια με το τρέχον στυλ γεμίσματος.

stroke()
Αυτή η μέθοδος σημειώνει τις υποδιαδρομές με το τρέχον στυλ γραμμής.

arc(x, y, radius, startAngle, endAngle, anticlockwise)
Προσθέτει σημεία στο υπομονοπάτι έτσι ώστε να προστεθεί σε αυτό το τόξο που περιγράφεται από την περιφέρεια του κύκλου το οποίο περιγράφεται από τα ορίσματα, ξεκινώντας από τη δεδομένη γωνία έναρξης και τελειώνοντας στη δεδομένη γωνία τερματισμού, κινούμενο προς τη δεδομένη κατεύθυνση, και συνδεόμενο με το προηγούμενο σημείο με μια ευθεία γραμμή.

HTML5 CANVAS

WebGL

- **Παράδειγμα 04 - Δημιουργία καμβά & χρήση εντολών σχεδιασμού #1/3 / 04_canvas_drawing**

```
<!DOCTYPE HTML>

<html>
   <head>

      <style>
         #test {
            width: 100px;
            height:100px;
            margin: 0px auto;
         }
      </style>

      <script type = "text/javascript">
         function drawShape() {

            // χρήση του στοιχείου canvas με χρήση του DOM
            var canvas = document.getElementById('mycanvas');
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 04 - Δημιουργία καμβά & χρήση εντολών σχεδιασμού #2/3 / 04_canvas_drawing**

```
            // επιβεβαίωση ότι δεν θα γίνει η εκτέλεση του κώδικα αν δεν υποστηρίζεται το canvas
            if (canvas.getContext) {

               // χρήση της εντολής getContext για τον σχεδιασμό μέσω του canvas
               var ctx = canvas.getContext('2d');

               ctx.beginPath();
               ctx.moveTo(75,40);

               ctx.bezierCurveTo(75,37,70,25,50,25);
               ctx.bezierCurveTo(20,25,20,62.5,20,62.5);

               ctx.bezierCurveTo(20,80,40,102,75,120);
               ctx.bezierCurveTo(110,102,130,80,130,62.5);

               ctx.bezierCurveTo(130,62.5,130,25,100,25);
               ctx.bezierCurveTo(85,25,75,37,75,40);
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 04 - Δημιουργία καμβά & χρήση εντολών σχεδιασμού #3/3 / 04_canvas_drawing**

```
                ctx.fill();
            } else {
                alert('Δυστυχώς χρειάζεσαι Safari ή Firefox 1.5+ για να δεις το σχήμα.');
            }
        }
    </script>
</head>

<body id = "test" onload = "drawShape();">
    <canvas id = "mycanvas"></canvas>
</body>

</html>
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 05 - Δημιουργία καμβά & σχεδιασμός τετραγωνικών καμπυλών #1/3 / 05_quad_curves**

```
<!DOCTYPE HTML>

<html>
   <head>

      <style>
         #test {
            width: 100px;
            height:100px;
            margin: 0px auto;
         }
      </style>

      <script type = "text/javascript">
         function drawShape() {

            // χρήση του στοιχείου canvas με χρήση του DOM
            var canvas = document.getElementById('mycanvas');
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 05 - Δημιουργία καμβά & σχεδιασμός τετραγωνικών καμπυλών #2/3 / 05_quad_curves**

```
            // επιβεβαίωση ότι δεν θα γίνει η εκτέλεση του κώδικα αν δεν υποστηρίζεται το canvas
            if (canvas.getContext) {

                // χρήση της εντολής getContext για τον σχεδιασμό μέσω του canvas
                var ctx = canvas.getContext('2d');

                // σχεδιασμός σχημάτων
                ctx.beginPath();

                ctx.moveTo(75,25);
                ctx.quadraticCurveTo(25,25,25,62.5);

                ctx.quadraticCurveTo(25,100,50,100);
                ctx.quadraticCurveTo(50,120,30,125);

                ctx.quadraticCurveTo(60,120,65,100);
                ctx.quadraticCurveTo(125,100,125,62.5);
```

HTML5 CANVAS

WebGL

- **Παράδειγμα 05 - Δημιουργία καμβά & σχεδιασμός τετραγωνικών καμπυλών #3/3 / 05_quad_curves**

```
                ctx.quadraticCurveTo(125,25,75,25);
                ctx.stroke();
            } else {
                alert('Δυστυχώς χρειάζεσαι Safari ή Firefox 1.5+ για να δεις το σχήμα.');
            }
        }
    </script>
</head>

<body id = "test" onload = "drawShape();">
    <canvas id = "mycanvas"></canvas>
</body>

</html>
```