

Canvas Tutorial

2D Canvas

From Simple HTML to 2D Platform Game

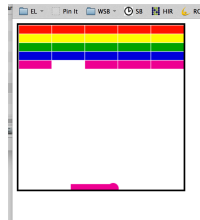
Goal: next 2-3 weeks

- Create a platform game (side scrolling game) leveraging
 - HTML
 - HTML5/Canvas
 - CSS, and
 - JavaScript
- Final skills (**after** 2-3 Weeks)
 - (before creating simpler games, pong, and breakout)
 - Jumping player entity
 - Scrolling background
 - Parallax
 - Gamification elements: keeping score, timer



Sub goals: Shorter Term

- We will start from the very beginning ...
 - Learn HTML/ Some JavaScript
 - Starting from a simple html page
 - Draw on a canvas
 - Animate
- Simple Shooter
- Breakout
 - Starting Tuesday next week



file:///Users/ingrid/Desktop/CLS/4070/2017-Spring-4070/WEB/tutorial1/11_bricks-really-pretty.html

The Basics “HTML” Pages (review for some)

- The language of the web (view source in a browser)
 - A browser languages, enables browser to display webpages according to specified formats.
 - Fonts, color, tables, paragraphs
 - Basic Document:
 - Heading
 - Paragraph
 - Another Paragraph
 - A markup language is a set of markup tags
 - The tags describes the document content
 - HTML documents contain HTML tags and plain text
 - HTML documents are simply called web pages
- A **Simple Example** of a BASIC HTML document ... next.

A simple and basic page

```
<html>
  <body>
    <h1>This a Heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

This a Heading

This is a paragraph.
This is another paragraph.

- Page full of ‘tags’
 - + <tagname> content </tagname>
 - + HTML tags normally come in pairs like <p> This is a paragraph </p>
 - + The first tag in a pair is the start tag, the second tag is the end tag
 - + The end tag is written like the start tag, but with a slash before the tag name

<0-mozilla-html-skeleton-no-canvas-00.html>

Anatomy of a web page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My Second Heading</h1>
    <p> My Third paragraph.</p>
  </body>
</html>
```



0-mozilla-html-skeleton-no-canvas-01.html

- The DOCTYPE declaration defines the document type to be HTML (instruction to the browser how to read the page).
- The text between <html> and </html> describes an HTML document
- The text between <head> and </head>
 - provides information about the document (preamble)
- The text between <title> and </title> provides a title for the document
 - Some browser put this text on the ‘title bar’
- The text between <body> and </body> describes the visible page content
 - The text between <h1> and </h1> describes a heading
 - The text between <p> and </p> describes paragraph

The <!DOCTYPE html> Declaration

- Denote which language you use
 - DOCTYPE html is for the browser.
 - Old school html, and the newer HTML5
 - <!DOCTYPE html>

http://www.w3schools.com/TAGs/tag_doctype.asp

Example Comment

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>A tiny document</title>
</head>
<body>
  <h1>Main heading in my document</h1>
  <!-- COMMENT -->
  <!-- Note that it is "h" + "1", not "h" + the letter "one" -->
  <p>Hey!!!! I am coding in (Hyper Text Markup Language)HTML.</p>
</body>
</html>
```

0-mozilla-html-skeleton-no-canvas-comment.html

HTML5

- As of October 2014 this is the new HTML standard:
 - Adds syntactic features to HTML:
 - new <video>, <audio> and the <canvas> elements
 - Handle Graphical and multimedia content without resorting to plug-ins, and new APIs
 - You should experiment with these
 - <canvas> is for graphics, and we use graphics for animation, and gaming.
 - It can draw graphics using scripting (usually javascript)
- It was a HTML5 before October 2014?
 - Yes, but now it is official, and now it is standard.

<http://en.wikipedia.org/wiki/HTML5>

Comments

- Comments in code, use a <!--
 - 2 dashes
 - tag to denote the beginning of a comment, a comment -->
 - concluded or 'closed' by a --> tag, see above:
- A comment:
 - <!-- A COMMENT -->
- Another comment:
 - <!--
Another COMMENT,
that spans multiple lines
-->

HTML more in-depth

- Some great tutorials are available, one of my favorites, that have nice WYSIWYG interfaces:
 - <http://www.w3schools.com/html/default.asp>
 - This is what we have used so far !
- Need a good editor:
 - Simple:
 - vi, notepad, text edit, emacs
 - Professional:
 - Dreamweaver (expensive), HTML debugging.
 - We will use simple ones – and I will use vi, and TextMate
 - Because we want understand the code exactly.
 - Webstorm's jetbrain? (you should be able to get this for free as students).

What is a <canvas>?

- A container for **hosting graphics**.
 - Can render Bitmap images (defined by JavaScript)
- A rectangular area on an HTML page.
- Canvas has several methods for drawing:
 - Lines, paths, boxes, circles, text, and graphic images.
 - Defined by JavaScript methods (APIs) for drawing the graphics (lines, paths, boxes, circles, shapes).
 - JavaScript API
- Also for text, animation, and interaction
- ... and of-course-games!
 - Animation + Interaction

```
<canvas
  id="drawing"
  width= "200"
  height= "100"
  style= "border:1px solid black">
</canvas>
```

- http://www.w3schools.com/tags/ref_canvas.asp
- <http://www.pageresource.com/html5/canvas-2d-interface-reference/>

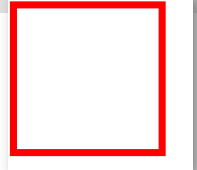
Simple Canvas

```
<!DOCTYPE html>
<html>
<body>

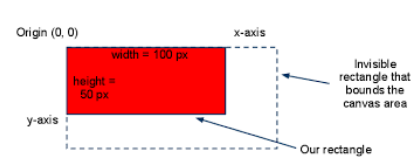
<canvas id="myCanvas" width="200" height="200" style="border:10px solid #FF0000;">
  Your browser does not support the HTML5 canvas tag.
</canvas>

</body>
</html>
```

- No JavaScript (yet)
- http://www.w3schools.com/html/html5_canvas.asp



- Anatomy of the canvas
 - X,Y and origin
- Drawing a rectangle on the canvas (not border) we will get to this soon.



Canvas “Images” or Drawings

Background:

- After drawing a ‘shape’ on canvas it is ‘gone’ canvas does not know of the element anymore (bitmap, raster images, paint with pixels)
- Fixed Sets of Dots
- This is in contrast to Scalable Vector Graphics (SVG), where you can manipulate the shapes after they are drawn (mathematical formulas describing the shape, resolution independent)

http://en.wikipedia.org/wiki/Scalable_Vector_Graphics

Strategy of Drawing Images on Canvas

Done by **JavaScript** in 3 steps:

- Obtain a **reference** to the **canvas** element.
- Obtain a **2D context** from the **canvas** element
- Draw graphics using the draw functions of 2D context.
 - (now the drawing is permanent)
- (not a 4th step)

Drawing on the Canvas with JavaScript

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="200" style="border:10px solid #2200c3;">
  Your browser does not support the canvas element.
</canvas>

<script>
  var canvas = document.getElementById("myCanvas"); // obtain the canvas element
  var ctx = canvas.getContext("2d"); // obtain 2D object ctx from
  // from canvas element
  ctx.fillStyle = "#FF00CC"; // now can draw
  ctx.fillRect(0,0,150,75); // using the methods of ctx
</script>

</body>
```

What does this look like?

1-mozilla-canvas-skeleton-1-js-.html

Drawing on the Canvas with JavaScript

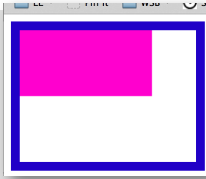
```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="200"
  style="border:10px solid #2200c3;">
  Your browser does not support the canvas element.
</canvas>

<script>
  var canvas = document.getElementById("myCanvas"); // finds the canvas element
  var ctx = canvas.getContext("2d"); // get 2D object ctx
  ctx.fillStyle = "#FF00CC"; // now we can draw
  ctx.fillRect(0,0,150,75); // using the methods of ctx
</script>

</body>
```

1-mozilla-canvas-skeleton-1-js-.html



- How about modularization?
 - Pull out the javascript and put it elsewhere?

Drawing on the Canvas with 'external' JavaScript

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="200"
  style="border:10px solid #2200c3;">
  Your browser does not support the canvas element.
</canvas>

<script src="js/drawRectangle.js"></script>

</body>
```

1-mozilla-canvas-skeleton-1-js-ex.html

- Standard practice to have a subdirectory "js/" for javascript.

CSS & canvas on-load Detour

```
<html>
<head>
  <title>Canvas tutorial</title>

  <script type="text/javascript">
    function draw()
    {
      var canvas = document.getElementById('tutorial');
      if (canvas.getContext){
        var ctx = canvas.getContext('2d');
      }
    }
  </script>

  <style type="text/css">
    canvas { border: 10px solid blue; }
  </style>
</head>

<body onload="draw();" >
  <canvas id="tutorial" width="150" height="150"></canvas>
</body>
</html>
```

1-mozilla-canvas-skeleton-2-css.html

Simple Graphics

- Examples
 - Drawing
 - Color
 - Opacity
 - Mouse
 - Keyboard

Good Exercise

- http://www.w3schools.com/graphics/canvas_clock.asp

- Ⓞ <http://www.html5-tutorials.org>
- Ⓞ <http://www.w3schools.com/html/default.asp>
- Ⓞ <http://en.wikipedia.org/wiki/HTML5>
- Ⓞ <http://tutorials.jenkov.com/html5-canvas/overview.html>

Ⓞ HTML, XML, XHTML, HTML5, and Canvas, CCS, JavaScript

Editors/IDE:

- Ⓞ Webstorm
- Ⓞ Textmate
- Ⓞ Vim