Comp 125 - Visual Information Processing

Spring Semester 2019 - Week 11 - Friday

Dr Nick Hayward

HTML5 Canvas - basic canvas

- start by creating a basic HTML5 file with a <canvas> element
  - *canvas is our container for drawing...*
- need to add a link to the external JavaScript file for the drawing logic

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Drawing - Canvas - Basic</title>
    </head>
    <body>
        <header>
      <h3>Drawing with Canvas - Basic</h3>
    </header>
    <main>
          <!-- add canvas -->
          <canvas id="drawing" width="600" height="400"></canvas>
        </main>
        <!-- script files -->
        <script src="./assets/js/drawing.js"></script>
    </body>
</html>
```

HTML5 Canvas - basic drawing

- might begin by drawing some rectangles with JavaScript on the `canvas`
- the following JavaScript will add a rectangle

```
// define canvas
var canvas = document.getElementById('drawing');
// define context for drawing
var context = canvas.getContext('2d');

// 1. rectangle
context.fillRect(0, 0, 100, 50);
```

HTML5 Canvas - basic canvas example

- we might use the canvas to combine rectangles to create various basic shapes
- update HTML for canvas

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Drawing - Canvas - Basic</title>
    </head>
    <body>
        <header>
      <h3>Drawing with Canvas - Basic</h3>
    </header>
    <main>
      <section id="drawings">
                <header>
         <h5>rectangle & staircase</h5>
        </header>
            <!-- add canvas -->
            <canvas id="drawing" width="600" height="400"></canvas>
      </section>
        </main>
        <!-- script files -->
        <script src="./assets/js/drawing.js"></script>
    </body>
</html>
```

- Example - basic drawing - rectangle & staircase
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic/*

HTML5 Canvas - basic drawing

stepped pyramid

- modify our example to draw multiple shapes, thereby creating a pattern on the canvas
  - *e.g. a stepped pyramid,*

```
// 3. pattern with rectangles - stepped pyramid - x,y,width,height
for (i = 1; i < 7; i++) {
    var start = 100;
  var width = i * 30;
  var x = (start - (width / 2))
    context.fillRect(x, i * 20, width, 20);
}
```

- Example - basic drawing - stepped pyramid
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic2/*

HTML5 Canvas - modify colours

- as we draw various shapes, we may also vary the colour for the fill
- specify a `fillStyle` property and value on the context for the canvas
  - *e.g.*

```
context.fillStyle = "YellowGreen";
```

- CSS supports over a 100 named colours
  - *many more shades using HEX values*
- CSS Tricks - Named Colours

HTML5 Canvas - various colours and drawing

- use various colours to output a series of rectangles
  - *e.g. a set of pan pipes*

```
// define colours
var colours = ["YellowGreen", "DarkSeaGreen", "MediumSeaGreen", "LightSeaGreen", "Turquois
// 5. draw many shapes with different colours
for (i = 1; i < 6; i++) {
  var width = 30;
    var height = i * 25;
    var x = 30 * i;
    var y = 75;
    context.fillStyle = colours[i-1];
    context.fillRect(x, y, width, height);
}
```

- Example - various colours
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic3/*

HTML5 Canvas - rectangle outlines

- we may also draw the outline of a rectangle with no fill

```
// 6. draw rectangle outline with stroke/line - no fill
content.strokeRect(5, 5, 150, 50)
```

- we might modify the colour of the *stroke* for the rectangle
  - *set a custom width for the line*

```
// 7. draw rectangle outline with colour
context.strokeStyle = "DarkSeaGreen";
context.lineWidth = 3;
context.strokeRect(5, 75, 300, 50);
```

- Example - basic drawing - rectangle outlines
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic4/*

HTML Canvas - draw lines - part I

- we may also draw lines to the `canvas`
  - *may be rendered individually or combined to create other shapes*
- for a line, we may also define a value for the colour
  - *define using the `strokeStyle` property*
  - *add a width for the lines*

```
// 8. draw lines with paths
context.strokeStyle = 'LightSeaGreen';
context.lineWidth = 3;
```

HTML Canvas - draw lines - part 2

- to start recording the lines, and their locations
  - *need to call the `beginPath()` method*
- this starts recording defined calls to `moveTo()` and `lineTo()`

```
// start recording lines to draw...
context.beginPath();
```

- define where to start
  - *using the expected `x` and `y` coordinates*
- need to call the `stroke()` method to actually render the lines &c.

```
// move to starting position for line - x & y
context.moveTo(50, 10);
// define line - x & y
context.lineTo(100, 70);
// draw all lines
context.stroke();
```

HTML Canvas - draw lines - part 3

- we might draw a triangle, or pyramid, using the following basic logic

```
// 9. draw a pyramid
context.strokeStyle = 'GoldenRod';
context.lineWidth = 3;
// start recording lines to draw...
context.beginPath();
// move to starting position for line - x & y
context.moveTo(100, 100);
// define line - x & y
context.lineTo(50, 170);
// define line - x & y
context.lineTo(150, 170);
// define line - x & y
context.lineTo(100, 100);
// draw all lines
context.stroke();
```

- Example - draw lines - line & pyramid
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic5/*

HTML Canvas - draw a stickman - part 1

- combine drawing shapes to create a *stick man* drawing
  - *perhaps suitable for a Hangman game...*
- e.g. we might start by drawing the *head* with a rectangle outline

```
// HEAD - draw rectangle outline with stroke/line - no fill
context.strokeRect(80, 5, 40, 40);
```

- then add the *torso* for the stick man

```
// TORSO: draw lines with paths
// start recording lines to draw...
context.beginPath();
// move to starting position for line - x & y
context.moveTo(100, 45);
// define line - x & y
context.lineTo(100, 125);
```

HTML Canvas - draw a stickman - part 2

- then choose to add either the arms or the legs for the drawing of the stick man

```
// LEFT ARM:
context.moveTo(100, 75);
context.lineTo(65, 65);

// RIGHT ARM:
context.moveTo(100, 75);
context.lineTo(135, 65);

// LEFT LEG:
context.moveTo(100, 125);
context.lineTo(75, 185);

// RIGHT LEG:
context.moveTo(100, 125);
context.lineTo(125, 185);
```

HTML Canvas - draw a stickman - part 3

- render these lines to the canvas
- simply call the `stroke()` method on the `context` object

```
// draw all lines
context.stroke();
```

- Example - draw a stickman
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic6/*

HTML Canvas - fill paths - part 1

- as we use stroke/line to draw the outline of a shape
  - *we may also define a fill colour for complete shapes*
- e.g. if we again drew a pyramid
  - *set a colour for the shape's fill*

```
// define fill style
context.fillStyle = 'DarkSeaGreen';
// start recording lines to draw...
context.beginPath();
// move to starting position for line - x & y
context.moveTo(50, 50);
// define line - x & y
context.lineTo(75, 25);
context.lineTo(100, 50);
context.lineTo(50, 50);
// draw all lines and fill
context.fill();
```

HTML Canvas - fill paths - part 2

- we might take this a bit further
  - *create a diamond pattern with fill colour as well*

```
// define fill style
context.fillStyle = 'DarkSeaGreen';
// start recording lines to draw...
context.beginPath();
// move to starting position for line - x & y
context.moveTo(50, 50);
// define line - x & y
context.lineTo(75, 25);
context.lineTo(100, 50);
context.lineTo(125, 75);
context.lineTo(100, 100);
context.lineTo(75, 125);
context.lineTo(50, 100);
context.lineTo(25, 75);
// draw all lines and fill
context.fill();
```

HTML Canvas - fill paths - part 3

- we might also use **alpha transparency** with fill for shapes
  - *e.g. fill style with opacity set to 0.5*

```
...
// define a semi transparent blue colour
context.fillStyle = `rgba(0, 0, 200, 0.5)`;
...
```

- Example - fill paths
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic7/*

Resources

- W3Schools - HTML5
  - *media elements*
  - *canvas element*