

Comp 125 - Visual Information Processing

Spring Semester 2018 - week 13 - wednesday

Dr Nick Hayward

HTML Canvas - draw arcs and circles

combine shapes - part I

- we might combine various shapes to create a fun drawing
 - *such as an Ancient Egyptian **Ankh***
 - *Ankh - Wikipedia*
- we begin by defining the canvas element
 - *get element by `id` for drawing the shapes*
 - *then set a `context`*

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

HTML Canvas - draw arcs and circles

combine shapes - part 2

- we may define stroke style for our shapes
 - *define required line width to create outlined shapes*

```
// define stroke style and width
context.strokeStyle = 'SteelBlue';
context.lineWidth = 10;
```

- setup the canvas and the required drawing styles
 - *then we may start to draw our shapes*

```
// draw an egyptian ankh
context.beginPath();
// define start point for drawing
context.moveTo(150, 100);
```

HTML Canvas - draw arcs and circles

top of ankh shape - part I

- *n.b.* top part resembles a stylised head without eyes
- *n.b.* top part plus horizontal bar resembles a bishop piece in chess
- top of the ankh requires three quadratic bézier curves
- first curve forms the top of the shape, its head in effect...

```
// top of ankh symbol  
context.quadraticCurveTo(200, 50, 250, 100);
```

HTML Canvas - draw arcs and circles

top of ankh shape - part 2

- second and third curves form the sides
 - *curves complete the top of the Ankh's shape*

```
// right side of ankh symbol
context.quadraticCurveTo(300, 150, 200, 250);
// left side of ankh symbol
context.quadraticCurveTo(100, 150, 150, 100);
```

- Example - arcs and circles - combine shapes to create an *ankh*
 - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basicI0-ankh/>

HTML Canvas - draw arcs and circles

cross bar of ankh shape - part I

- to draw the cross bar of our ankh
 - *need to move the cursor on the canvas to a new start point*
 - *move cursor before drawing our shapes*

```
// define start point for horizontal bar  
context.moveTo(200, 260);
```

HTML Canvas - draw arcs and circles

cross bar of ankh shape - part 2

- then, we follow a pattern of
 - *left top, down, left bottom, right bottom, up*
 - *and finish with the right top line*

```
// draw left top line
context.lineTo(70, 255);
// draw left vertical line
context.lineTo(70, 285);
// draw left bottom line
context.lineTo(200, 280);
// draw right bottom line
context.lineTo(330, 285);
// draw right vertical line
context.lineTo(330, 255);
// draw right top line
context.lineTo(200, 260);
```

- **n.b.** we might also have started with the right side of our cross bar shape
 - *thereby using a clockwise path.*
- Example - arcs and circles - combine shapes to create an *ankh*
 - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basicI0-ankh/>

HTML Canvas - draw arcs and circles

stem of ankh shape - part I

- we may finish our ankh shape
 - *draw a stem at the bottom of the horizontal cross bar*
- move the cursor to the required starting position
 - *move underneath the cross bar and slightly offset to the right*

```
// define start point for vertical stem  
context.moveTo(210, 280)
```


HTML Canvas - draw arcs and circles

stem of ankh shape - part 2

- we can draw a vertical bar down for the right side of the stem
 - *then draw a horizontal bar at the bottom*
- then draw a matching bar on the left

```
// draw right side down - slight angle out
context.lineTo(215, 500);
// draw bottom of stem
context.lineTo(185, 500);
// draw left side up = slight angle in
context.lineTo(190, 280);
```

- Example - arcs and circles - combine shapes to create an *ankh*
 - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basicI0-ankh/>

HTML Canvas - draw with a function

- we may abstract drawing required shapes to a custom function
- a custom function may then be called to create a shape
 - e.g. *any size circle*

```
// define custom function to draw circle
function circle() {
  ...
}
```

HTML Canvas - draw with a function

custom drawn circles - part I

- create a function to draw a custom circle
 - *position, radius, and fill*
- function draws a standard circle of varying radius and fill
- e.g. we might start with the following initial function

```
// define custom function to draw circle
function circle(x, y, radius, fillCircle) {
}
```

- then call this function as follows

```
// outer circle for head
circle(100, 100, 50, false);
```

HTML Canvas - draw with a function

custom drawn circles - part 2

- fill out the logic for our working `circle` function as follows,

```
// define custom function to draw circle
function circle(x, y, radius, fillCircle) {
  // start recording
  context.beginPath();
  // define arc - x, y, radius, start posn, end posn, anticlockwise...
  context.arc(x, y, radius, 0, Math.PI * 2, false);
  // check fill or stroke
  if (fillCircle) {
    context.fill();
  } else {
    context.stroke();
  }
}
```

HTML Canvas - draw with a function

a certain well-known mouse - part I

- we might use this new custom circle function
 - *create a certain well-known mouse*
- start by defining the canvas element in our HTML

```
<!-- add canvas -->  
<canvas id="drawing" width="800" height="800"></canvas>
```

- then define the canvas and context in our JavaScript logic
 - *required to start drawing our shapes*

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

HTML Canvas - draw with a function

a certain well-known mouse - part 2

- add the `circle` function to our JavaScript
 - we may start *drawing the required shapes for our drawing*

```
// define custom function to draw circle
function circle(x, y, radius, fillCircle) {
    // start recording
    context.beginPath();
    // define arc - x, y, radius, start posn, end posn, anticlockwise...
    context.arc(x, y, radius, 0, Math.PI * 2, false);
    // check fill or stroke
    if (fillCircle) {
        context.fill();
    } else {
        context.stroke();
    }
}
```

HTML Canvas - draw with a function

a certain well-known mouse - part 3

- for this particular drawing
 - *add necessary specifics for colour of each circle's fill style*

e.g.

```
context.fillStyle = 'DarkRed';
```

HTML Canvas - draw with a function

a certain well-known mouse - part 4

- to draw the required shape for our well-known *mouse*
 - we can use *three circles*
- each circle will define
 - *position* - *x* and *y* coordinates
 - *a radius*
 - *and fill colour or not*
- then draw our well-known *mouse*
 - call the *circle* function three times

```
// 2. a certain well-known mouse
// left ear
circle(400, 100, 35, true);
// right ear
circle(500, 100, 35, true);
// head
circle(450, 160, 57, true);
```

- Example - circle function
 - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic/I1-function-circles/>

References

- [W3Schools - HTML5](#)
- [media elements](#)
- [canvas element](#)