Comp 125 - Visual Information Processing

Spring Semester 2019 - Week 12 - Monday

Dr Nick Hayward

Video - Design

Digital Prototyping

Rapid Prototyping 2 of 3: Digital Prototyping
Source: YouTube - Google

HTML Canvas - draw arcs and circles

- not restricted to simply drawing shapes with straight lines or rectangles
- we might also need to draw a circle, or a custom arc
- to draw a circle or arc, start by specifying
  - *the centre point for the circle*
  - *its radius*
  - *extent of the circumference*
- to draw an arc we provide a value
  - *for the starting angle and end angle*
  - *use to define the arc to draw*

HTML Canvas - draw arcs and circles

radians

- required start and end angles for drawing an arc are defined in **radians**
- to measure a circle using radians, we begin at 0
  - ***0*** *is equivalent to* ***3*** *on a clock*
- relative to a standard circle as a clock
  - *12pm = 270° or (π x 3 / 2 radians)*
  - *3pm = 0° (0 radians) & 360° (π x 2 radians)*
  - *6pm = 90° (π / 2 radians)*
  - *9pm = 180° (π radians)*

HTML Canvas - draw arcs and circles

`arc()` method

- expected parameters for the arc method is as follows

```
arc(x, y, radius, startAngle, endAngle, anticlockwise);
```

- where `anticlockwise` is set to `false` by default

HTML Canvas - draw arcs and circles

full circle - part 1

- using this pattern to draw a full circle
  - *start at 3pm and continue back round to 3pm*
- i.e. start at 0 radians and continue to `(π x 2 radians)`
- in JS, this may be represented as follows

```
// draw a full circle
context.beginPath();
context.arc(50, 100, 25, 0, Math.PI * 2, false);
context.stroke();
```

HTML Canvas - draw arcs and circles

full circle - part 2

```
// draw a full circle
context.beginPath();
context.arc(50, 100, 25, 0, Math.PI * 2, false);
context.stroke();
```

- call `arc()` method on the `context` object passing required arguments
  - *`50, 100` = the centre of the circle as x and y coordinates*
  - *`25` = radius of circle*
  - *`0` = 0 radians for the start position of the circle (0°)*
  - *`Math.PI * 2` = (π x 2 radians) for the end position for the end of the circle (360°)*

HTML Canvas - draw arcs and circles

arcs - part 1

- we can then create various arcs, including a semi-circle

```
// draw a semi-circle
context.beginPath();
context.arc(125, 100, 25, 0, Math.PI, false);
context.stroke();
```

- call `arc()` method on the `context` object passing required arguments
  - *`125, 100` = x & y centre of the circle*
  - *`25` = radius of circle*
  - *`0` = start position of arc (0°)*
  - *`Math.PI` = end position of arc (180°)*

HTML Canvas - draw arcs and circles

arcs - part 2

- we might also draw a quarter circle

```
// draw a quarter circle
context.beginPath();
context.arc(175, 100, 25, 0, Math.PI / 2, false);
context.stroke();
```

- **n.b.** `false` value in `arc()` method refers to anticlockwise parameter
  - *by default, an arc will follow a clockwise path*
- Example - arcs and circles
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic8/*

HTML Canvas - draw arcs and circles

Bézier curves

- we can also draw more fluid, or organic, shapes using bézier curves
- use a couple of default methods
- support for **cubic** or **quadratic** varieties of bézier curves

Bézier curves - Wikipedia

HTML Canvas - draw arcs and circles

quadratic - part 1

- we can draw a quadratic bézier curve from a defined start point
  - *i.e. current pen position on the canvas, using the following method*

```
quadraticCurveTo(cp1x, cp1y, x, y)
```

- `cp1x` & `cp1y` = controls points for curve
- `x` & `y` = standard `x` and `y` coordinates on the canvas
  - *defines end point from the current pen position*
- this type of curve has a defined start and end point with a single control point

HTML Canvas - draw arcs and circles

quadratic - part 2

- for example

```
// draw a quadratic bézier curve
context.beginPath();
context.moveTo(75, 25);
context.quadraticCurveTo(25, 25, 25, 62.5);
context.quadraticCurveTo(25, 100, 50, 100);
context.quadraticCurveTo(50, 120, 30, 125);
context.quadraticCurveTo(60, 120, 65, 100);
context.quadraticCurveTo(125, 100, 125, 62.5);
context.quadraticCurveTo(125, 25, 75, 25);
context.fill();
```

- Example - Bézier curves - quadratic
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic9-quadratic/*
  - *W3Schools - quadraticCurveTo()*

HTML Canvas - draw arcs and circles

cubic - part 1

- a cubic bézier curve, by contrast, has the following method and usage

```
bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)
```

- pattern is similar to a quadratic curve
  - *primary difference is use of two control points*
  - *potentially offers finer control over extent and nature of curve*

HTML Canvas - draw arcs and circles

cubic - part 2

- for example

```
// draw a cubic bézier curve
context.beginPath();
context.moveTo(75, 40);
context.bezierCurveTo(75, 37, 70, 25, 50, 25);
context.bezierCurveTo(20, 25, 20, 62.5, 20, 62.5);
context.bezierCurveTo(20, 80, 40, 102, 75, 120);
context.fill();
```

- Example - Bézier curves - cubic
  - *http://linode4.cs.luc.edu/teaching/cs/demos/125/drawing/basic9-cubic/*
  - *W3Schools - bezierCurveTo()*

HTML Canvas - draw arcs and circles

combine shapes - part 1

- 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 1

- *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/125/drawing/basic10-ankh/*

HTML Canvas - draw arcs and circles

cross bar of ankh shape - part 1

- 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/125/drawing/basic10-ankh/*

HTML Canvas - draw arcs and circles

stem of ankh shape - part 1

- 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/125/drawing/basic10-ankh/*