

COMP4021  
Internet Computing

More on SVG

David Rossiter and Gibson Lam

# Grouping and Gradients

- You have learned the basics of SVG, i.e. the shapes (lines, rectangles, circles and so on) that you can use for drawing, in the previous discussion
- In this presentation, we will introduce grouping and how to use gradients to improve the images

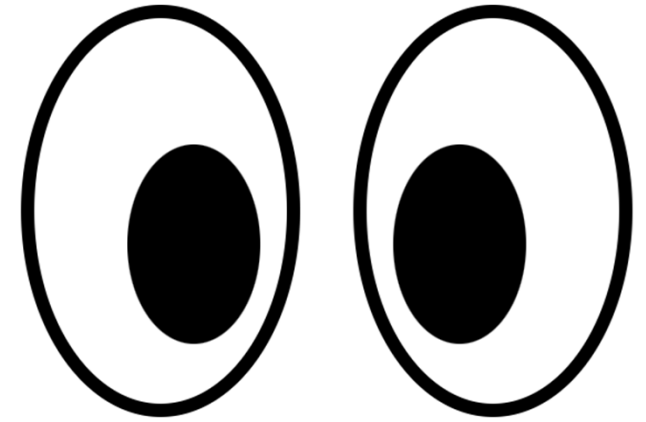
# Grouping Things Together

- At the moment, your SVG shapes are added one by one in the SVG file
- If you have many of them, it will be hard to manage
- You can group SVG things together using the `<g>` tag to have a better organization of the SVG content

```
<svg ...>
  <rect ...>
  <circle ...>
  <circle ...>
  <circle ...>
  <circle ...>
  <circle ...>
  <circle ...>
  <path ...>
  <path ...>
  <circle ...>
  .
  .   Many more
  .   shapes here
  .
</svg>
```

# Creating Groups

- Here is an example with two groups of ellipses:



```
<svg xmlns="http://www.w3.org/2000/svg"
      width="300" height="200">
```

```
  <g id="right-eye">
```

```
    <ellipse ... />
```

```
    <ellipse ... />
```

```
  </g>
```

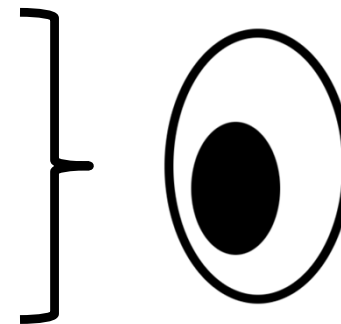
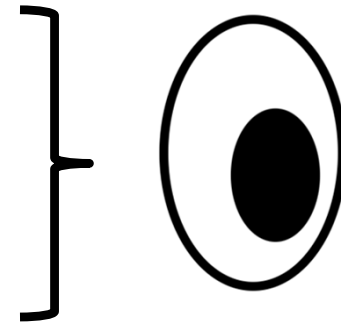
```
  <g id="left-eye">
```

```
    <ellipse ... />
```

```
    <ellipse ... />
```

```
  </g>
```

```
</svg>
```



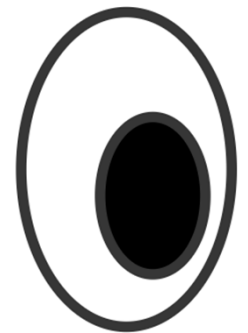
*Ellipses are similar to circles except that they have two radii rx and ry*

# Other Use of Groups

- If you put SVG things into groups, you can also do these easily:

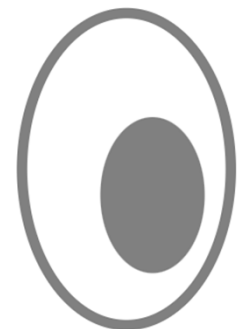
- Applying CSS properties to a group of SVG shapes, e.g.:

```
<g id="right-eye"  
  stroke="red"  
  stroke-width="4">  
  <ellipse ... />  
  <ellipse ... />  
</g>
```



- Writing JavaScript code for an SVG group, e.g.:


```
$("#right-eye").css("opacity", "0.5");
```



# Defining Things in SVG

- You can define things in a 'definitions' area of SVG
- Those things can then be used in the rest of the SVG (as many times as you want)
- Here are some useful things you can define:
  - Gradients
  - Patterns
  - Clipping paths
  - Filters

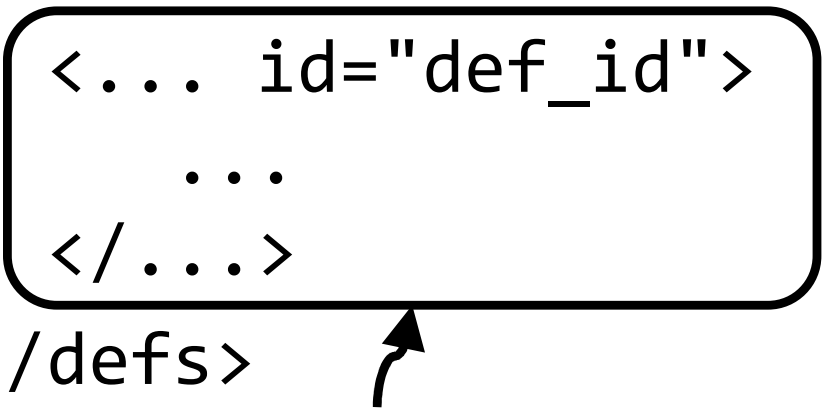
*We will briefly look  
at gradients in this  
discussion*



# The <defs> Area

- You first define something you want to use in the <defs> area, with a specific id
- An SVG element can then use the defined thing by referring to that id

```
<svg ...>
  <defs>
    <... id="def_id">
      ...
    </...>
  </defs>
  <... #def_id ... />
</svg>
```



# SVG Gradients

- The SVG `fill` attribute we have used so far is a solid flat colour, for example:

```
<rect width="700" height="100"  
      x="50" y="50" fill="blue" />
```

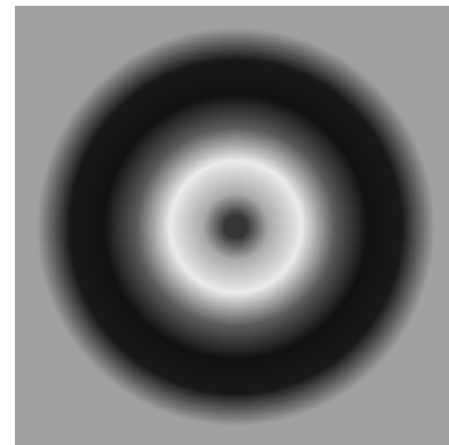
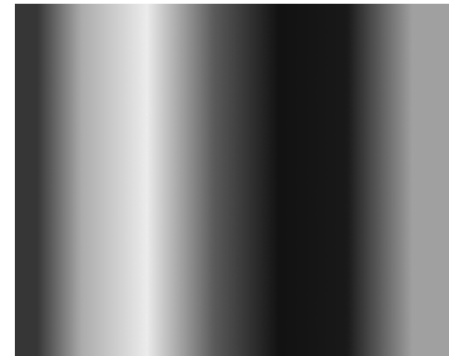


- Using SVG gradients you can create a smoothly changing gradient of colours in the colouring of SVG elements



# Two Types of Gradient

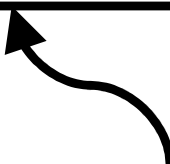
- There are two types of SVG gradient:
  - *Linear gradients*, that change colour linearly along a line
  - *Radial gradients*, that change colour from the centre of a shape and radiate outwards



# Using the Defs Area

- The gradients are defined in the defs area and then referred to using their ids

```
<svg ...>
  <defs>
    <linearGradient id="my_gradient">
      ...gradient colours...
    </linearGradient>
  </defs>
  ...
  <... fill="url(#my_gradient)" .../>
  ...
</svg>
```

A curved arrow points from the `url(#my_gradient)` attribute in the `<... fill="url(#my_gradient)" .../>` tag to the `</linearGradient>` tag within the `<defs>` block, illustrating how the gradient is referenced.

# Gradient Colours

- The content of a gradient is a list of colours defined in different positions
- Each colour is specified by the `stop` tag with a particular `offset` and `stop-color`
  - Offset is the positioning of the colour from 0 to 1
  - Stop colour is the colour used at that position
- For example, here is a red colour positioned at 50% of a gradient:

```
<stop offset="0.5" stop-color="red" />
```

# A Simple Linear Gradient

- A simple black and white two-colour gradient can be created like this:

```
<linearGradient id="gradient">  
  <stop offset="0" stop-color="white" />  
  <stop offset="1" stop-color="black" />  
</linearGradient>
```

- Applying the above gradient to a rectangle results in the fill colour shown on the right



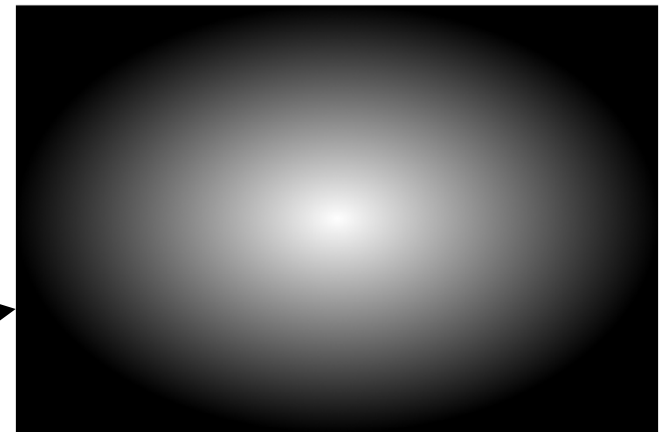
# A Simple Radial Gradient

- The previous example can be easily changed to a radial gradient like this:

```
<radialGradient id="gradient">  
  <stop offset="0" stop-color="white" />  
  <stop offset="1" stop-color="black" />  
</radialGradient>
```

- Applying the gradient to the same rectangle makes the result on the right

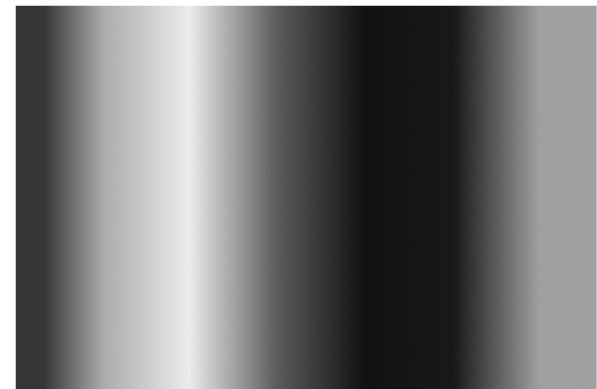
*The radial gradient has an oval shape if the SVG element does not span a square area*



# A Rainbow Gradient

- With more colours you can create a highly varied gradient such as the rainbow colour shown below:

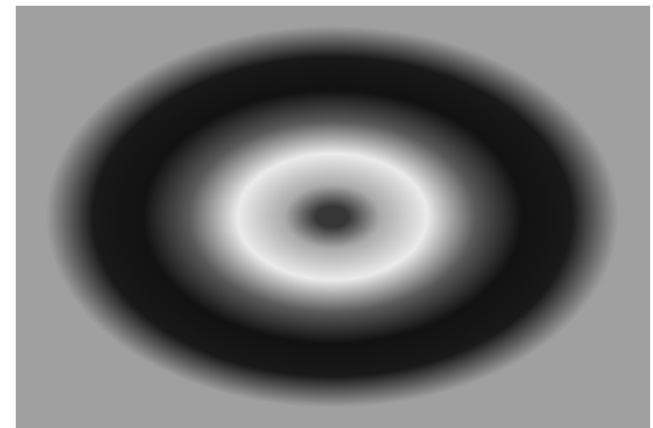
```
<linearGradient id="gradient">
  <stop offset="0.05" stop-color="red" />
  <stop offset="0.15" stop-color="orange" />
  <stop offset="0.3" stop-color="yellow" />
  <stop offset="0.45" stop-color="green" />
  <stop offset="0.60" stop-color="blue" />
  <stop offset="0.75"
        stop-color="indigo" />
  <stop offset="0.9"
        stop-color="violet" />
</linearGradient>
```



# Rainbow Radial Gradient

- Similarly, the rainbow can become a radial gradient like this:

```
<radialGradient id="gradient">  
  <stop offset="0.05" stop-color="red" />  
  <stop offset="0.15" stop-color="orange" />  
  <stop offset="0.3" stop-color="yellow" />  
  <stop offset="0.45" stop-color="green" />  
  <stop offset="0.60" stop-color="blue" />  
  <stop offset="0.75"  
    stop-color="indigo" />  
  <stop offset="0.9"  
    stop-color="violet" />  
</radialGradient>
```



# The Line in a Linear Gradient

- A linear gradient, by default, changes its colour from left to right horizontally
- You can change the orientation of the line using a set of attributes:  $x1$ ,  $y1$ ,  $x2$ ,  $y2$
- These attributes specify a line within the gradient area and the change of colour then starts from  $(x1, y1)$  to  $(x2, y2)$
- An example is shown in the next slide



# Changing the Line

- In this example, the line inside the gradient starts from the top-left hand corner to the bottom-right hand corner of the area

```
<linearGradient id="gradient"  
  x1="0%" y1="0%" x2="100%" y2="100%">  
  <stop offset="0"    stop-color="red" />  
  <stop offset="0.5"  stop-color="black" />  
  <stop offset="1"    stop-color="violet" />  
</linearGradient>
```



# Changing a Radial Gradient

- You can similarly change the way a radial gradient works, as shown below:

```
<radialGradient id="gradient"
  fx="60%" fy="60%"
  cx="20%" cy="30%" r="80%" >
  <stop offset="0"
    stop-color="black" />
  <stop offset="1"
    stop-color="white" />
</radialGradient>
```

*The starting point of the gradient*

*The target circular area*

