argonavis.com.br

SVG 9

animação smil

Helder da Rocha (helder@argonavis.com.br)

Animação em SVG com SMIL

- Mais simples que scripting, porém menos suporte nos browsers
  - Chrome, Opera, FF 4 em diante), IE 9.0 (suporte parcial)
- Elementos e atributos para animação de gráficos e grupos
  - Baseados na especificação **SMIL – Synchronized Multimedia Integration Language** (*http://www.w3.org/TR/smil-animation/*)
  - Especificação SVG usa parte do SMIL e acrescenta extensões: *http://www.w3.org/TR/SVG/animate.html*
  - Suporta scripting, eventos, estilo e operações com SVG DOM
- Elementos
  - SMIL: **<animate>**, **<set>**, **<animateColor>**, **<animateMotion>**
  - Extensões: **<animateMotion>** com **<mpath>**, **<animateTransform>**
- O alvo da animação pode ser indicado de duas formas
  - via **xlink:href="#id"**
  - sendo o elemento pai de um bloco **<animate>**, **<set>**, etc.

Animação básica

- A forma mais simples de animação consiste em variar o valor de um **atributo** de um elemento, por determinada duração
  - Pode ser um atributo do CSS ou do XML
- Os valores em cada instante são calculados por **interpolação**
  - A informação mínima pode ser um valor final, par de valores, etc.
  - Valores intermediários são calculados por interpolação
  - Pode-se passar mais valores intermediários, para maior controle
- Exemplo: Ao clicar no retângulo vermelho, ele irá mover-se para a direita até ficar sob o retângulo azul, em 3 segundos

```
<rect x="50"  y="50" width="100" height="50" fill="red" >
   <animate    attributeName="x" to="200" dur="3s"
               begin="click" fill="freeze" />
</rect>
<rect x="200" y="50" width="100" height="50" fill="blue"
```

<animate>

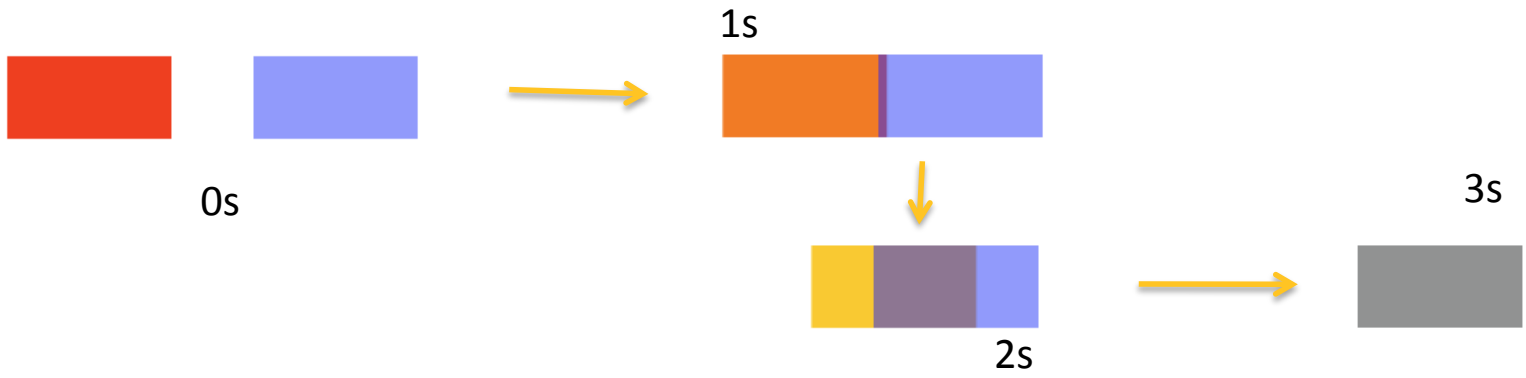
- Atributos essenciais
  - **attributeName**: nome do atributo que terá seu valor alterado (deve conter um valor interpolável*)
  - **from**: valor inicial (se não estiver presente, será considerado o valor atual do atributo)
  - **to**: valor que será atingido no fim da animação
  - **by**: alternativa a *to* – *by* é somado a *from* para obter valor final
  - **dur**: tempo que irá durar a animação (o valor pode ser expresso em vários formatos de tempo)
  - **begin**: ocorrências que causarão o início da animação – tipicamente um evento ou instante (se não houver **begin**, a animação começa quando o elemento for carregado)
  - **fill**: o que acontece quando a animação termina. Pode ser **freeze** (o último estado é preservado) ou **remove** (volta ao estado antes da animação iniciar)
- Os valores dos atributos podem ser diferentes e terem outros significados, em diferentes contextos

* O que é ou não 'interpolável' depende de outros atributos. Ex: números geralmente são; strings às vezes.

Múltiplas animações

- Pode-se ter múltiplos elementos <animate>, cada um lidando com uma animação específica

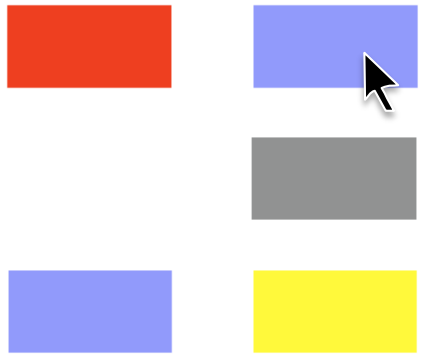
```
<rect x="50"  y="50" width="100"
      height="50" fill="red" >

   <animate attributeName="x" to="200" dur="3s"
            begin="click" fill="freeze" />
   <animate attributeName="fill" to="yellow"
            dur="3s" begin="click" fill="freeze" />

</rect>
```



<set>

- Aceita os mesmos atributos que <animate>, mas simplesmente muda o atributo (sem animação)
  - O atributo **dur** neste caso representa o tempo em que o atributo permanece alterado
  - No final do tempo, ele volta ao valor anterior (a menos que se defina **fill="freeze"**)

```
<rect x="50"  y="50" width="100" height="50" fill="red" >
   <set attributeName="x" to="200" begin="click" fill="freeze" />
   <set attributeName="fill" to="yellow" dur="3s" begin="click" fill="remove" />
</rect>
<rect x="200" y="50" width="100" height="50" fill="blue" opacity="0.5">
   <set attributeName="x" to="50" dur="3s" begin="click" fill="remove"/>
</rect>
```



Atributos de tempo

- Atributos de tempo (**max**, **min**, **end**, **dur, repeatDur**) aceitam tempo expresso nas seguintes sintaxes
  - 02:30:03 = 2 horas, 30 minutos e 3 segundos
  - 50:00:10.25 = 50 horas, 10 segundos e 250 ms
  - 02:33 = 2 minutos and 33 seconds
  - 00:10.5 = 10.5 segundos = 10 s e 500 ms
  - 3.2h = 3.2 horas = 3 horas e 12 minutos
  - 45min = 45 minutos
  - 30s = 30 segundos
  - 5ms = 5 milissegundos
  - 12.467 = 12 segundos e 467 milissegundos
- Ex: `<animate dur="10s" ... >`

Atributos de temporização

- **begin**=""
  - Lista de valores que especifica quando a animação inicia (pode conter um tempo, evento, referência a outra animação, **ou combinações** de tudo isto)
  - Veja sintaxe detalhada na spec SVG seção 19.2.6
  - Ex: begin="click + 3s"
- **end**=""
  - Aceita valores similares a begin
- **restart** = "always | whenNotActive | never"
  - Se a animação deve reiniciar, e como
- **repeatCount**="*número* | indefinite"
  - Número de vezes que a animação deve repetir
  - Se indefinite, animação repete até o documento terminar
- **repeatDur**="*tempo* | indefinite"
  - Duração total para as repetições
  - Se indefinite, animação repete até o documento terminar

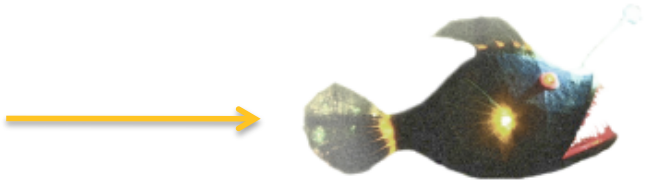
Animações aditivas e cumulativas

- Atributo **additive**="sum | replace"
  - Se for **sum**, valor do atributo é somado a valor anterior, pré-existente, ou valor de animações de menor prioridade (definidas antes)
  - Se for **replace** (default) valores prévios são ignorados
- Atributo **accumulate**="none | sum"
  - Considerado para animações que se repetem
  - Se for **sum**, cada repetição nova parte do valor final do atributo na animação anterior
  - Se for **none** (default) cada repetição é independente

values

- Valores do atributo a serem usados no curso da animação
  - Segmentação permite mais controle que **from-by** ou **from-to**
  - Valores devem ser separados por ponto-e-vírgula
  - Como o objeto se move no tempo depende do atributo **calcMode**
- Exemplo: o peixe leva o mesmo tempo para atravessar cada trecho

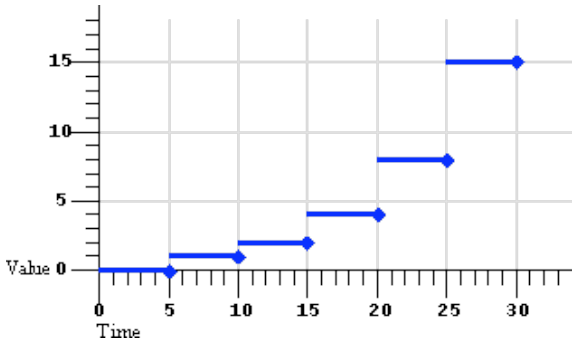
```
<svg xmlns="http://www.w3.org/2000/svg">
    <image id="abissal" xlink:href="TheFish.png"
           height="100" width="250" x="50" y="50" />
    <g stroke="blue" stroke-width="2">
        <line x1="200" x2="200" y1="175" y2="225"/> ... </g> ...
    <animate attributeName="x"
             values="50;150;200;500"
             dur="4s" begin="click"
             fill="freeze"
             xlink:href="#abissal" />
</svg>
```



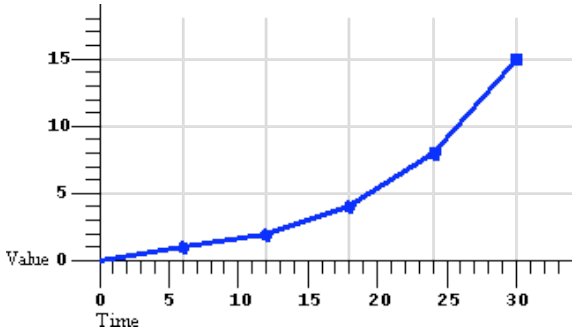
Atributo calcMode

- Modo como é calculada a animação
- calcMode="linear | discrete | paced | spline"
  - **linear**: interpolação linear simples calcula valores intermediários entre os pontos
  - **discrete**: durante o tempo da animação, salta de um valor para o outro (permanece um tempo em cada valor)
  - **paced**: média dos valores é usada para obter uma velocidade constante entre o ponto inicial e final
  - **spline**: calcula uma função de tempo não-linear, com dados fornecidos separadamente (atributo keySplines)
- Para valores numéricos, default é **linear**
- Para valores não interpoláveis, como strings, default (e único valor permitido) é **discrete**

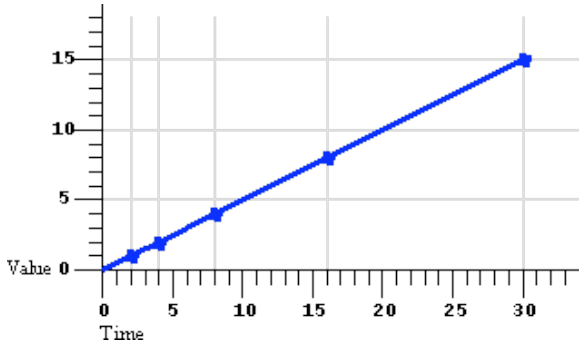
Modos de interpolação lineares




```
<animate dur="30s"
         values="0; 1; 2; 4; 8; 15"
         calcMode="discrete" />
```




```
<animate dur="30s"
         values="0; 1; 2; 4; 8; 15"
         calcMode="linear" />
```




```
<animate dur="30s"
         values="0; 1; 2; 4; 8; 15"
         calcMode="paced" />
```

Fonte: W3C SMIL spec

Intervalos irregulares: keyTimes

- Com **values** o tempo (dur) é dividido em partes iguais para cada segmento
  - Ex: havendo 4 valores, e dur="4s", cada segmento dura 1s
- Com o atributo **keyTimes**, é possível informar quanto tempo permanecer em cada segmento
  - Valores entre 0 e 1, representando um **instante**: valor acumulativo e proporcional à duração da animação
  - Cada valor sucessivo deve ser **maior ou igual** ao valor anterior
  - O **primeiro valor** deve ser sempre **0**
  - Para animação linear ou spline, **o último valor deve ser 1**
- Este atributo está relacionado com o atributo values
  - Cada valor em **keyTimes** está relacionado a um valor em **values**
  - É preciso ter mesmo número de valores em **keyTimes** e **values**

Exemplo com values e keyTimes

- Há 3 intervalos: 0.0 – 0.1 , 0.1 a 0.9 , 0.9 a 1.0
  - A velocidade no primeiro e último trecho é a mesma (o peixe passa 10% do tempo em cada um)
  - O peixe leva 80% do tempo para percorrer o trecho intermediário

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">
    <image id="abissal" xlink:href="TheFish.png" height="100" width="250"
x="50" y="50" />
    <g stroke="blue" stroke-width="2">...</g>
    <animate attributeName="x" values="50 ;200;350;500" dur="4s"
             begin="click"   keyTimes="0.0;0.1;0.9;1.0" fill="freeze"
             xlink:href="#abissal" />
</svg>
```

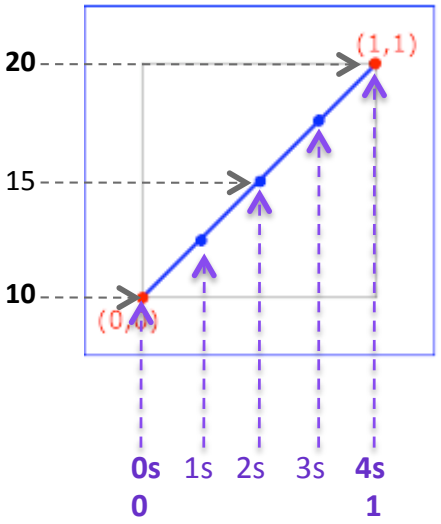
Movimento não-linear: keySplines

- Possibilita variação não-linear do movimento
- O atributo keySplines define um conjunto de **pontos de controle Bézier cúbicos** definem uma função que determina como varia o **tempo** de um segmento
  - Cada ponto de controle Bézier consiste de **quatro valores** x1,y1 x2,y2 (dois pontos de controle por segmento)
  - Valores na faixa de 0 a 1
  - **calcMode** deve ser **spline**
  - Os pontos são separados por ponto-e-vírgula
- Relacionamento entre os atributos **keySplines**, **keyTimes** e **values**:
  - Para cada **n** valores de **keyTimes** e **values**, deve haver **n-1** pontos de controle Bézier

Exemplos com keySplines (W3C)

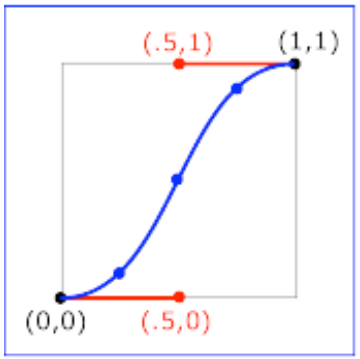
São gráficos de variação de TEMPO (não de espaço)!

(1)



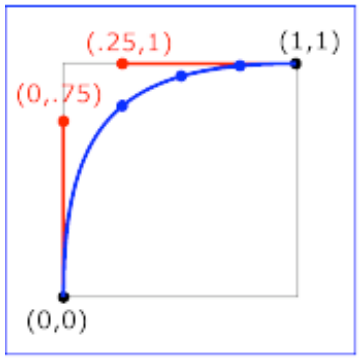

```
<animate dur="4s" values="10;20" keyTimes="0;1"
         calcMode="spline" keySplines="0,0 1,1">
```

(2)



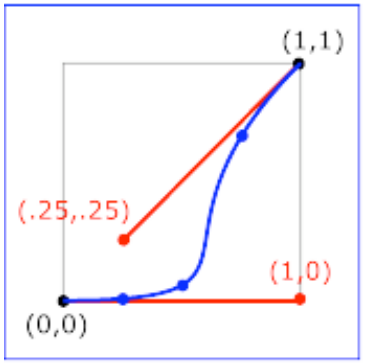

```
<animate dur="4s" values="10;20"
         keyTimes="0;1"
         calcMode="spline"
         keySplines="0.5,0 0.5,1">
```

(3)




```
<animate dur="4s" values="10;20"
         keyTimes="0;1"
         calcMode="spline"
         keySplines="0,0.75 0.25,1">
```

(4)



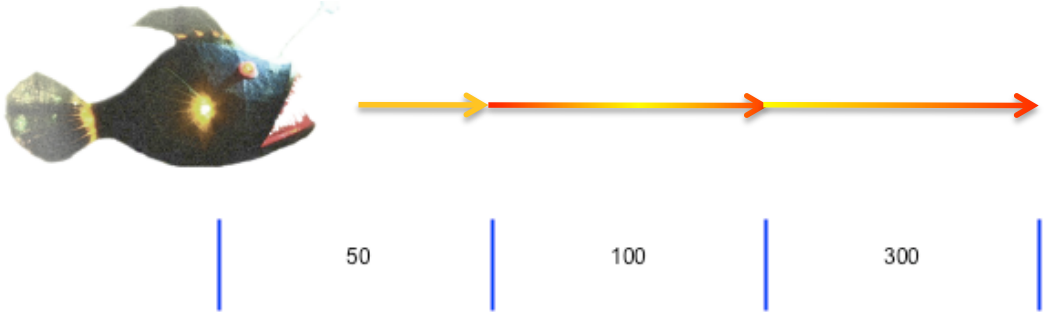

```
<animate dur="4s" values="10;20" keyTimes="0;1"
         calcMode="spline" keySplines="1,0 0.25,0.25">
```

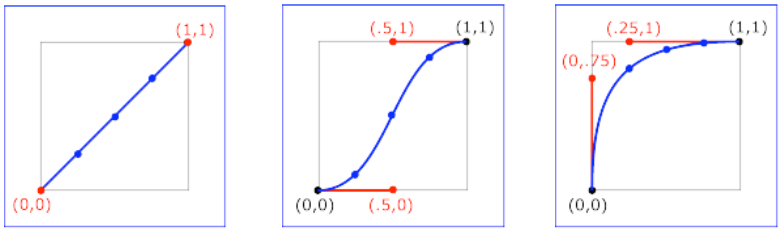
Fonte: W3C (spec SVG, 19.2.7)

Exemplo com keySplines

- No primeiro trecho, o movimento é **linear** (gráfico 1 – slide anterior)
- No segundo trecho, o peixe começa devagar, **acelera no meio** e depois **desacelera** (gráfico 2)
- No terceiro trecho, o peixe **dispara no início** e depois vai **gradualmente desacelerando até parar**(gráfico 3)

```
<animate  attributeName="x"  values="50; 200; 350; 500"
          dur="4s"           keyTimes="0;0.333; 0.667; 1"
          begin="click"      keySplines="0.0,0.0   1.0, 1.0;
                                         0.5,0.0   0.5, 1.0;
                                         0.0,0.75  0.25,1.0;"
                             calcMode="spline"
                             fill="freeze"
                             xlink:href="#abissal" />
```





<animateMotion>

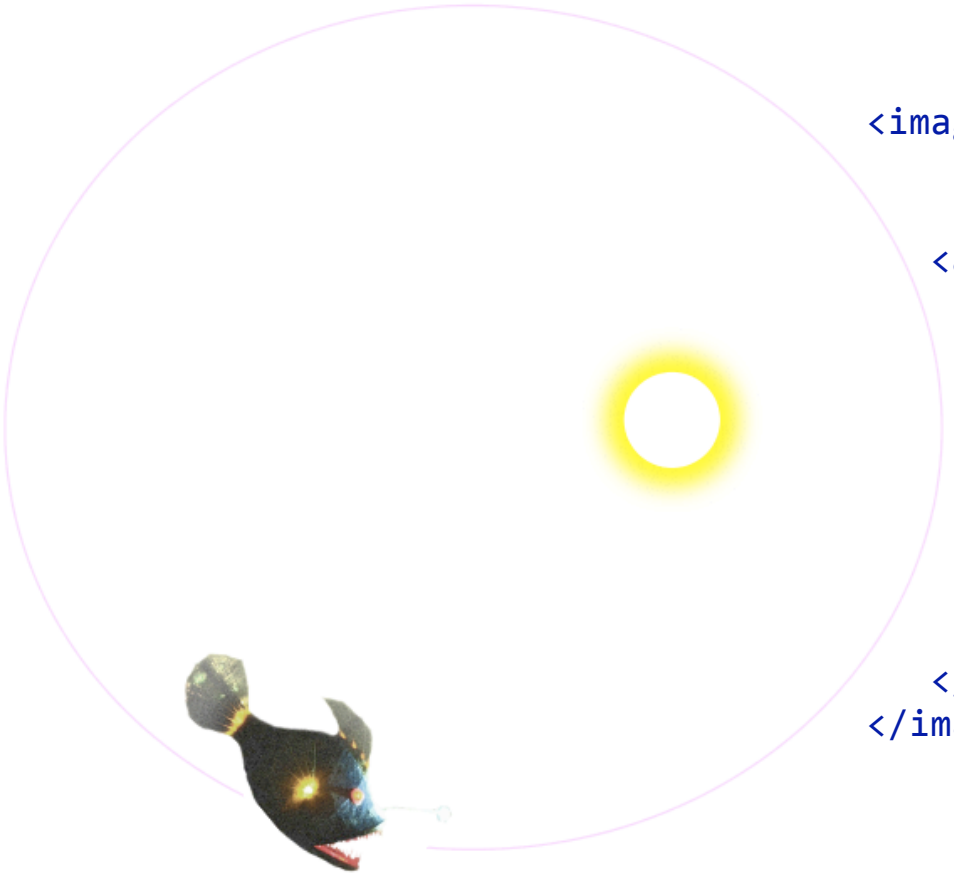
- Permite fazer que um objeto se mova por um caminho (path).
- O caminho pode ser referenciado de três formas
  - Atributo **`xlink:href="#path"`** (referência para um **<path>**)
  - Elemento filho **`<mpath xlink:href="#path" />`**
  - Atributo **`path`**: mesma sintaxe do atributo **d** do elemento **<path>**
- Atributo **keyPoints**
  - Cada valor **keyTimes** pode ser associado a um ponto do caminho
  - Valores entre **0** e **1** (proporção da distância percorrida)
- Atributo **rotate**="*número* | auto | auto-reverse"
  - Se rotate for **auto**, objeto ficará sempre alinhado com o vetor tangencial do movimento; **auto-reverse** = auto + 180 graus.
  - Se for um número, é o **ângulo fixo** de rotação do objeto **sobre o eixo x**.
  - Para variar o ângulo durante a execução pode-se usar JavaScript.

Sintaxe de atributos comuns

- from, by, to, values
  - Em **<animateMotion>** esses atributos recebem **um par de valores**, representando coordenadas x,y
  - Ex: `values="20,50; 50;100; 100;100"`
- keyPoints, keyTimes, keySplines
  - Se **keyPoints** for usado, **values** é ignorado.
  - Se **path** ou <**mpath>** for usado, **values** é ignorado.
  - Se **values** for usado, **from**, **by** e **to** são ignorados
  - O valor default de **calcMode** é **paced**, logo **keyTimes** será ignorado por default: para usar, mude o valor para **linear** ou **discrete**)
  - **keySplines** é suportado normalmente

Exemplo de <animateMotion>

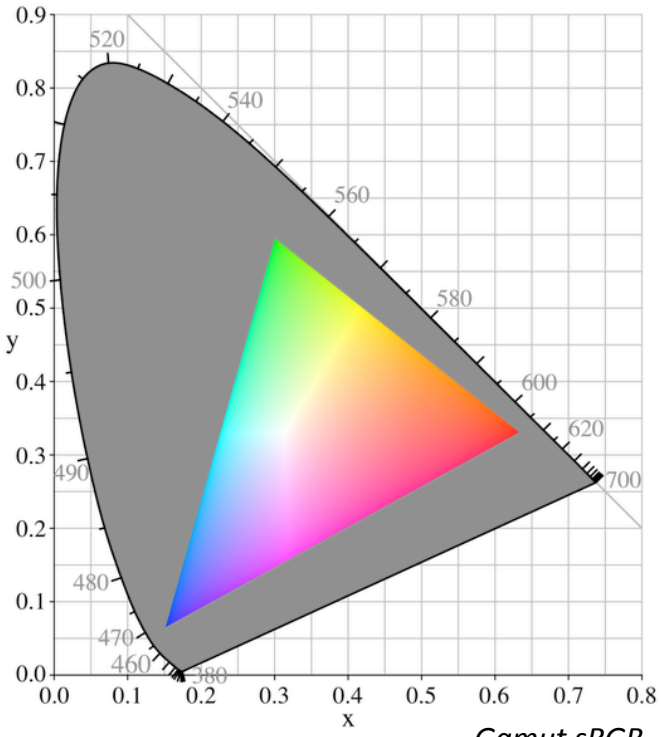
- O peixe girando em órbita elíptica
  - Ele acelera quando se aproxima do Sol



```
<path id="orbita" stroke-width="1" ...
   d="M 20,175
      A 244,220 0 1,0 493,175
      A 244,220 0 0,0 20,175"/>

<image id="abissal" x="0" y="-100"
       xlink:href="TheFish.png"
       height="80" width="200" >
   <animateMotion dur="10s"
                  repeatCount="indefinite"
                  rotate="auto"
                  calcMode="spline"
                  keyTimes="0;1"
                  keyPoints="0;1"
                  keySplines="0.75,0.25
                              0.25, 0.75">
        <mpath xlink:href="#orbita"/>
   </animateMotion>
</image>
```

<animateColor>

- **<animateColor>** permite variar a cor de preenchimento ou de contorno ao longo do tempo
  - Os atributos **from**, **by** e **to** ou **values** recebem valores de cor (ex: #fff, red, rgb(100%,0,0), etc.)
  - A interpolação é feita percorrendo valores entre cores de acordo com o **espaço sRGB** (padrão)
  - É o mesmo algoritmo usado para computar gradientes



Gamut sRGB

```
<animateColor values="red, pink, yellow, blue"  ... />
<animateColor from="rgb(0,0,0)" to="rgb(255,255,255)" ... />
```

<animateTransform>

- Permite que a animação varie o deslocamento, rotação, escala ou cisalhamento de um objeto
- Atributo **type** = "translate | scale | rotate | skewX | skewY"
  - Determina o tipo de transformação
- Atributos **from**, **by** e **to**
  - Recebem valor compatível com o tipo de transformação
- Atributo **values**
  - Recebe lista de valores separados por ponto-e-vírgula
- O efeito de animações **to** é indefinido em SVG.
  - Recomenda-se usar **from-to**, **from-by** ou **values** (e não apenas **to** – devido à forma como opera, valor inicial é indefinido)
  - A transformação é multiplicada ao efeito das animações anteriores (quando atributo **additive**="sum")*

** Veja mais na spec SVG, seção 19.2.14*

Exemplo de <animateTransform>

- Várias animações simultâneas e aditivas

```
<image xlink:href="TheFish.png" height="80" width="200" x="50" y="50">
    <animateTransform type="rotate" from="0" to="15" attributeName="transform"
        dur="2s" begin="click" fill="freeze" />
    <animateTransform type="translate" to="300,0" fill="freeze"
        dur="2s" begin="click" calcMode="spline" attributeName="transform"
        keyTimes="0;1" keySplines="0.0,0.5  0.5, 1.0" additive="sum"/>
    <animateTransform type="rotate" to="-30" attributeName="transform"
        dur="4s" begin="click+1.5s" additive="sum"  fill="freeze"/>
    <animateTransform type="translate" to="200,0" attributeName="transform"
        dur="4s" begin="click+5s" calcMode="spline"  fill="freeze"
        keyTimes="0;1" keySplines="0.25,0.75  0.25, 1.0"  additive="sum"/>
    <animateTransform type="rotate" to="15" attributeName="transform"
        dur="6s" begin="click+10s" additive="sum"
        keyTimes="0;1" keySplines="0.25,0.75  0.25, 1.0" fill="freeze"/>
</image>
```