

Tutorial sNCL

Este tutorial tem o objetivo de apresentar a linguagem sNCL (Simpler Nested Context Language).

Parte I

sNCL é uma linguagem declarativa de domínio específico (DSL) que foi desenvolvida com o objetivo de facilitar a autoria de aplicações multimídia que compila para NCL.

A sintaxe é parecida com a linguagem imperativa Lua.

Mas o que são documentos multimídia?

São documentos que possuem mais de um tipo de mídia, por exemplo: imagem, audio, vídeo.

Para construir esses tipos de documentos, precisamos definir: o que desejamos tocar, onde (em que área da tela), quando.

Elementos de um documento sNCL

- Media
- Port
- Link
- Region
- Context

- **Media**

É a palavra reservada para representar uma mídia, que pode ser: foto, vídeo, audio.

```
1  media introducao
2      src = "media/video.mp4"
3      area segInicio
4          begin = "12s"
5      end
6      area segPhoto begin = "41s" end
7      width = "100%" height = "100%"
8  end
```

Figura 1

A figura 1 exemplifica a declaração de uma mídia em sNCL. Observe que a palavra reservada **media** é seguida da palavra **introdução**, que é o identificador (**id**) da mídia. Esse identificador deve ser único e serve para a mídia ser referenciada em outras partes do código.

A palavra reservada **src** é onde indicamos o local que a mídia se encontra.

A mídia pode conter atributos como que definem seu tamanho e posição como: **width, height, left, right, top, bottom**. Note que também há uma declaração de **área** no exemplo, essa área é uma âncora que é um ponto de entrada para os nós de mídia ou contextos. O objetivo de se utilizar âncoras é utilizar segmentos de um nó de mídia ou contexto, seja como origem ou destino de elos.

Em outras palavras a **area** representa um trecho no tempo ou espaço da mídia a que pertence. No exemplo mostrado podemos perceber que temos duas areas, uma com todas as informações em uma linha e a segunda é feita a declaração de maneira indentada e em linhas separadas.

Vale ressaltar que código de maneira indentada não é obrigatório, mas dessa maneira torna o código legível para entendimento.

- **Port**

Ao iniciar uma aplicação, o player precisa saber qual mídia será tocada no início, o port tem essa finalidade, inidicar a mídia para ser executada no início. O **port** pode ser declarado em qualquer parte do código, contanto que não seja filho de nenhum elemento. Deve ser declarado de forma independente, como mostra o exemplo a seguir:

```
1  port entry introducao
2
3  media animation
4      src = "media/video.mp4"
5      area segInicio
6          begin = "5s"
7      end
8      area segPhoto begin = "6s" end
9      width = "100%" height = "100%"
10     zIndex = "2"
11     explicitDur = "25s"
12 end
13 media audio1
14     src = "media/audio1.mp4"
15 end
16 media te
17     src = "media/dribble.mp4"
18     left = "5%" top = "6.7%"
19     width = "18.5%" height = "18.5%"
20     zIndex = "3"
21 end
22 media photo
23     src = "media/photo.png"
24     left = "5%" top = "6.7%"
25     width = "18.5%" height = "18.5%"
26     zIndex = "3"
27     explicitDur = "5s"
28 end
```

Figura 2

Observe que há um novo atributo nas medias **animation** e **photo**, o atributo é explicitDur, ele especifica a duração de cada mídia explicitamente.

- **Link**

Através dos links podemos sincronizar os eventos em um programa. Por exemplo, os links servem para iniciar a execução de uma mídia simultaneamente com outra, define também seu término.

As condições para a execução das mídias são realizadas pelos **links**, essas condições são blocos, que começam pela palavra definindo a ação, e a mídia que sofre a ação, e terminando com a palavra "end", dentro deles podendo ser declarados os parâmetros da ação. como mostra o exemplo a seguir:

```
1  port entry introducao
2
3  media animation
4      src = "media/video.mp4"
5      area segInicio
6          begin = "5s"
7      end
8      area segPhoto begin = "6s" end
9      width = "100%" height = "100%"
10     zIndex = "2"
11     explicitDur = "25s"
12 end
13 media audio1
14     src = "media/audio1.mp4"
15 end
16 media textos
17     src = "media/legenda.jpg"
18     left = "5%" top = "6.7%"
19     width = "18.5%" height = "18.5%"
20     zIndex = "3"
21 end
22 media photo
23     src = "media/photo.png"
24     left = "5%" top = "6.7%"
25     width = "18.5%" height = "18.5%"
26     zIndex = "3"
27     explicitDur = "5s"
28 end
29 onBegin animation do
30     start audio1
31         delay = "5s"
32     end
33 end
34 onBegin animation.segInicio do
35     start audio1 end
36 end
37 onBegin animation.segPhoto do
38     start photo end
39 end
40 onEnd animation do
41     stop audio1 end
42 end
```

Figura 3

- **Region**

O elemento **region** é importante, pois ele define uma região na tela, e é bastante utilizado, já que várias mídias podem referenciar uma mesma região.

O region possui atributos assim como o elemento media, entre seus atributos estão: width, height, zIndex, rg.

O atributo zIndex, especifica como fica a sobreposição de regiões. Uma região de maior valor para zIndex se sobrepõe àquela de menor valor.

O atributo rg é utilizado na media para referenciar uma **region**.

```
1  port entry introducao
2  region screenReg
3    width = "100%" height = "100%"
4    zIndex = "2"
5    region frameReg
6      left = "5%" top = "6.7%"
7      width = "18.5%" height = "18.5%"
8      zIndex = "3"
9    end
10 end
11 media animation
12   src = "media/video.mp4"
13   area segInicio
14     begin = "5s"
15   end
16   area segPhoto begin = "6s" end
17   width = "100%" height = "100%"
18   zIndex = "2"
19   explicitDur = "25s"
20 end
21 media audio1
22   src = "media/audio1.mp4"
23 end
24 media textos
25   src = "media/legenda.jpg" rg= frameReg
26   left = "5%" top = "6.7%"
27   width = "18.5%" height = "18.5%"
28   zIndex = "3"
29 end
30 media photo
31   src = "media/photo.png" rg = frameReg
32   left = "5%" top = "6.7%"
33   width = "18.5%" height = "18.5%"
34   zIndex = "3"
35   explicitDur = "5s"
36 end
37 onBegin animation do
38   start audio1
39     delay = "5s"
40   end
41 end
42 onBegin animation.segInicio do
43   start audio1 end
44 end
45 onBegin animation.segPhoto do
46   start photo end
47 end
48 onEnd animation do
49   stop audio1 end
50 end
```

Figura 4

- **Context**

O elemento **context** pode ser usado para estruturar uma aplicação. No exemplo a seguir será exemplificado o uso de contextos, agrupando todos os elementos da animação. Isso possibilitará, além de maior estruturação do programa, o reúso de toda a estrutura.

```

1  port entry introducao
2  region screenReg
3    width = "100%" height = "100%"
4    zIndex = "2"
5    region frameReg
6      left = "5%" top = "6.7%"
7      width = "18.5%" height = "18.5%"
8      zIndex = "3"
9    end
10 end
11 media animation
12   src = "media/video.mp4"
13   area segInicio
14     begin = "5s"
15   end
16   area segPhoto begin = "6s" end
17   width = "100%" height = "100%"
18   zIndex = "2"
19   explicitDur = "25s"
20 end
21 media audio1
22   src = "media/audio1.mp4"
23 end
24 media textos
25   src = "media/legenda.jpg" rg= frameReg
26   left = "5%" top = "6.7%"
27   width = "18.5%" height = "18.5%"
28   zIndex = "3"
29 end

30 media photo
31   src = "media/photo.png" rg = frameReg
32   left = "5%" top = "6.7%"
33   width = "18.5%" height = "18.5%"
34   zIndex = "3"
35   explicitDur = "5s"
36 end
37 context advert
38   port pIcon icon
39   port pShoes shoes
40   media icon
41     src = "media/icon.png"
42     rg = iconReg
43   end
44   media shoes
45     src = "media/shoes.mp4"
46     rg = shoesReg
47     explicitDur = "6s"
48   end
49   onSelection.RED icon do
50     start shoes end
51     stop icon end
52   end
53 end

55 onBegin animation do
56   start audio1
57   delay = "5s"
58   end
59 end
60 onBegin animation.segInicio do
61   start audio1 end
62 end
63 onBegin animation.segPhoto do
64   start photo end
65 end
66 onEnd animation do
67   stop audio1 end
68 end

```