

LUMI SCRIPT - PARÂMETROS DO SCRIPT (Super Set Gerador) [UL603]

Luís Augusto Spranger

<http://www.lumikit.com.br>

LUMI SCRIPT - ANTES DE COMEÇAR

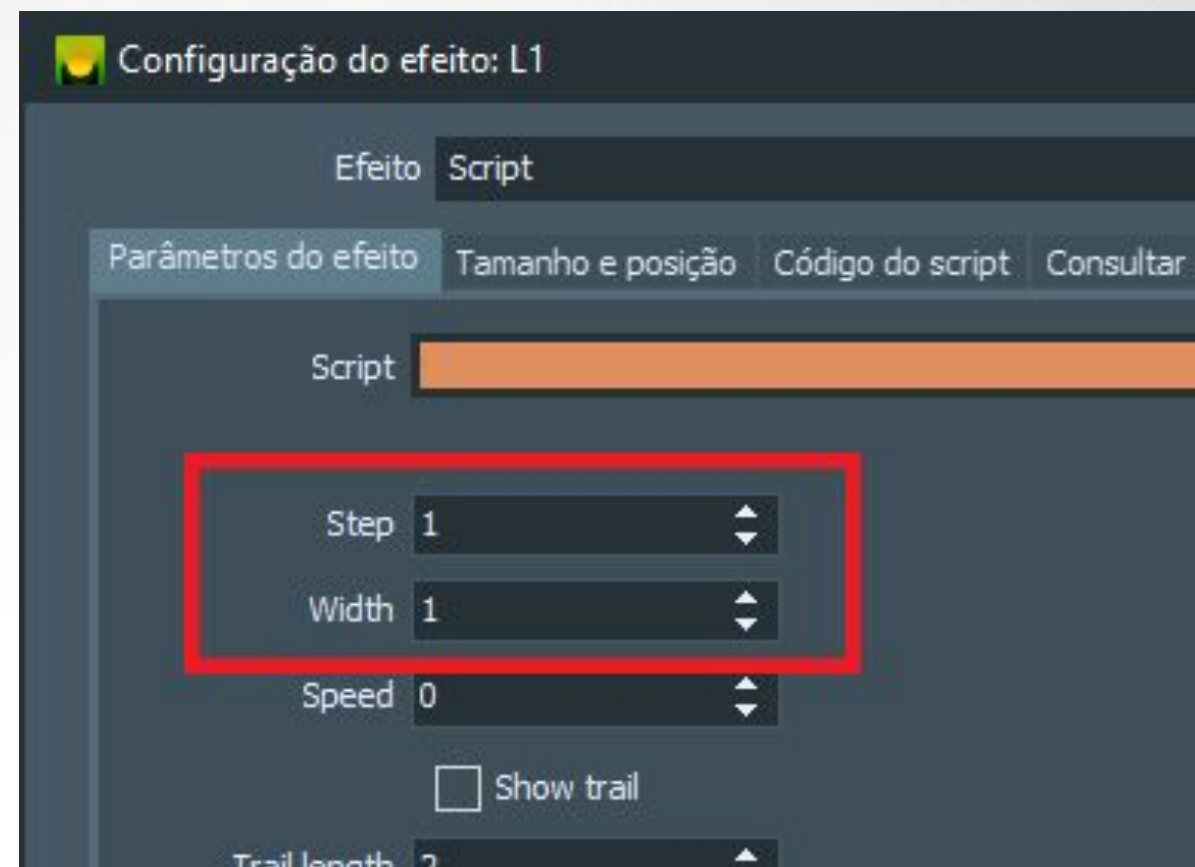
- No vídeo UL601 mostramos um pouco da linguagem do LUMI SCRIPT e alguns exemplos simples;
- No vídeo UL602 é mostrado o mesmo assunto desse vídeo para o Super Set DMX;

LUMI SCRIPT - PARÂMETROS

- Em geral os scripts do gerador podem ser parametrizados, por exemplo é possível escolher a cor, se possui “rastro” e em alguns casos precisamos deixar mais parâmetros disponíveis, por exemplo:
 - Fazer um script que faça o efeito radar, no efeito radar temos a largura da linha e o “passo” (step) - de quantos em quantos pixels será o movimento;

LUMI SCRIPT - PARÂMETROS

- Em geral os scripts do gerador podem ser parametrizados, por exemplo é possível escolher a cor, se possui “rastros”, e em alguns casos precisamos deixar mais parâmetros disponíveis, por exemplo:
 - Fazer um script que faça o efeito radar, no efeito radar temos a largura da linha e o “passo” (step) - de quantos em quantos pixels será o movimento;



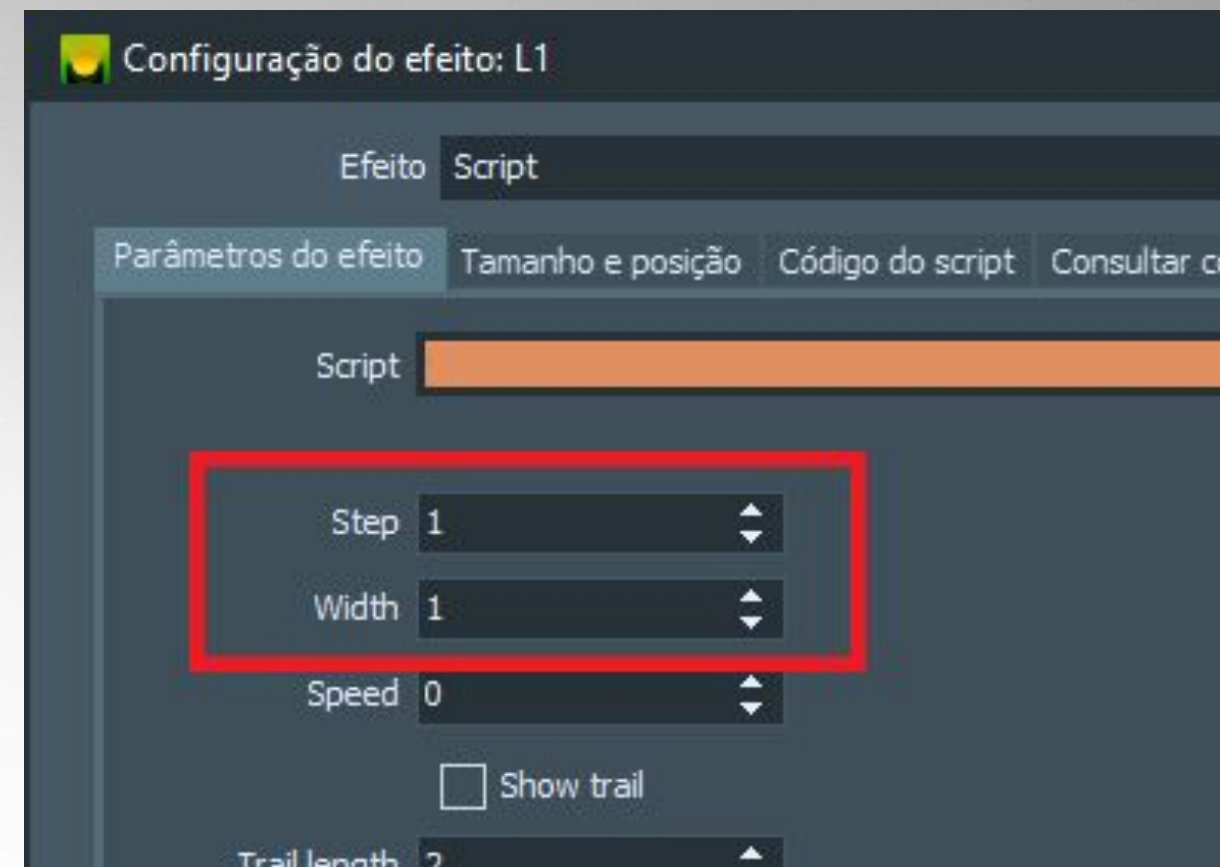
LUMI SCRIPT - PARÂMETROS

- Para criar os controles que permitem informar os parâmetros para o script, deverão ser feitos 4 passos:
 - Definir variáveis para cada um dos parâmetros;
 - Chamar o método Register... para cada controle, dentro do método OnRegister;
 - No método OnRegister também devem ser definidos os parâmetros padrão:
 - Opções de "rastros";
 - Opções de cores padrão;
 - Criar um método OnSetParameterInteger e/ou OnSetParameterString, para verificar qual o parâmetro e lá definir o valor das variáveis criadas no primeiro passo;
 - Usar as variáveis com o valor dos parâmetros informados dentro do método OnExecute;

LUMI SCRIPT - PASSO 1

- Definir variáveis para cada um dos parâmetros:

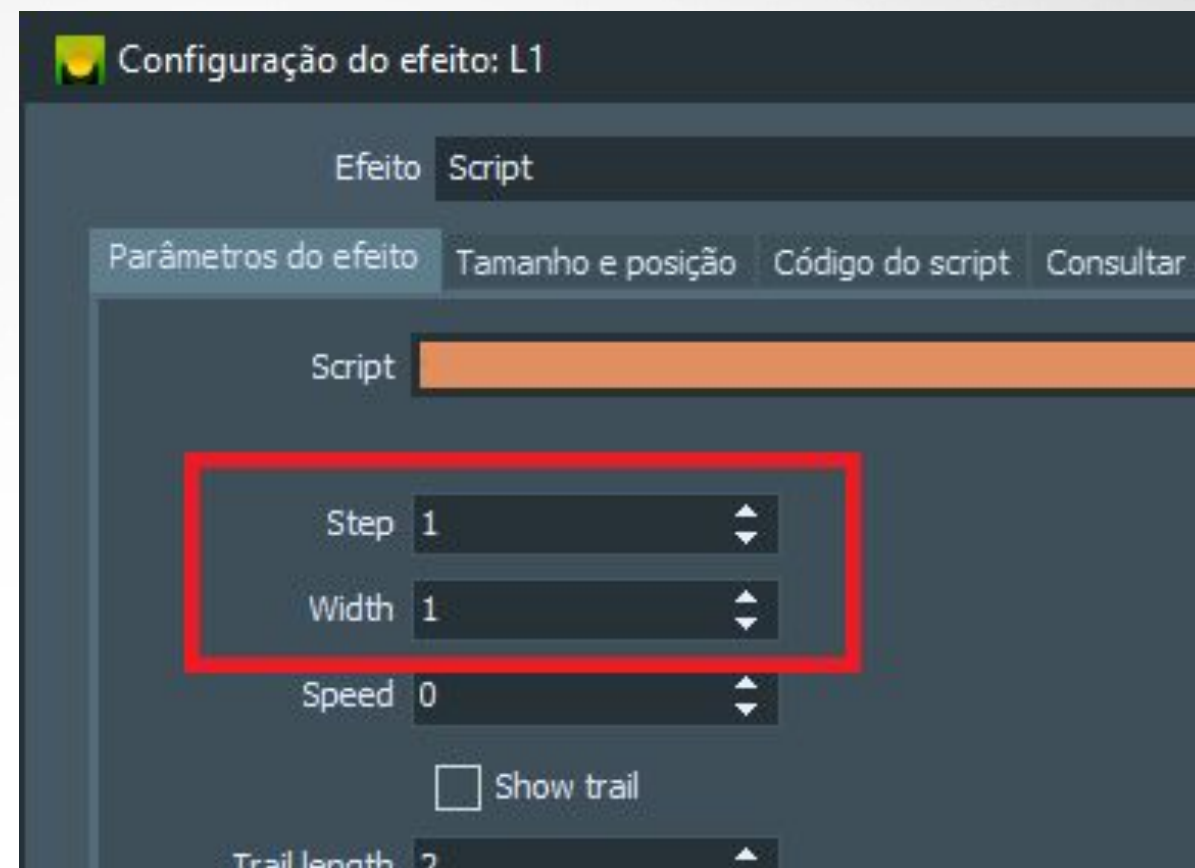
```
4  
5 var  
6   uiStep: integer;  
7   uiWidth: integer;  
8
```



LUMI SCRIPT - PASSO 2

- Chamar o método Register... para cada controle, dentro do método OnRegister:

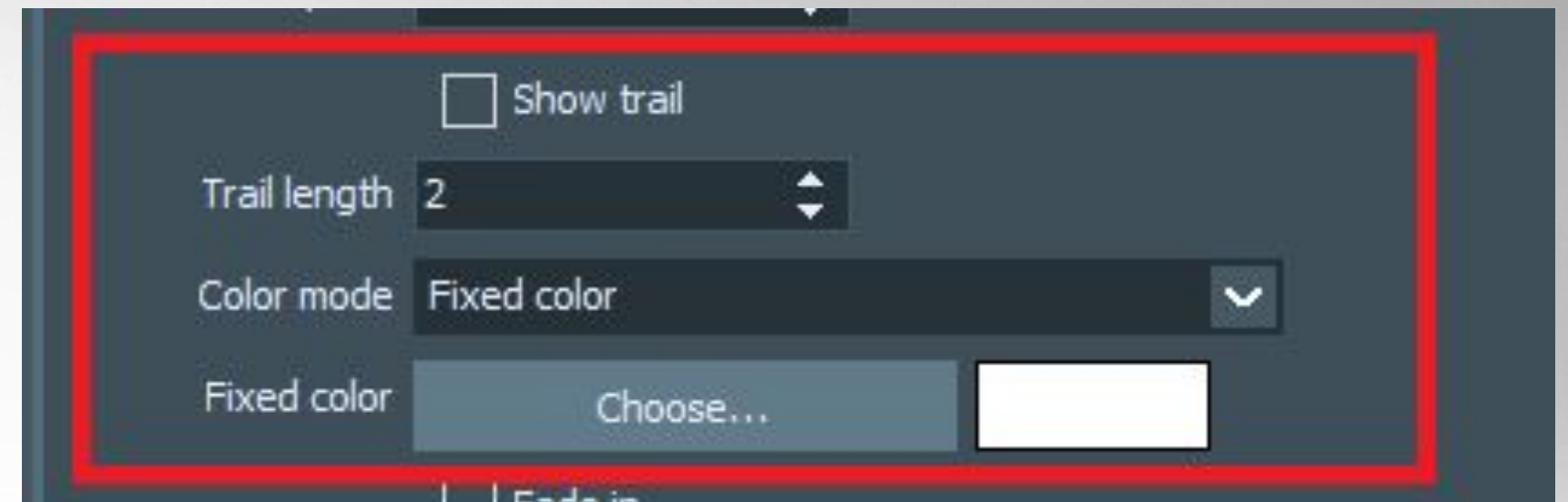
```
11  
12 { "OnRegister" will be executed after the script is compiled and here you must  
13 define the user interface (UI) options }  
14 procedure OnRegister;  
15 begin  
16   RegisterUIValue('Step', 1, 1, 20);  
17   RegisterUIValue('Width', 1, 1, 20);  
18
```



LUMI SCRIPT - PASSO 2

- Chamar o método `Register...` para cada controle, dentro do método `OnRegister`;
 - No método `OnRegister` também devem ser definidos os parâmetros padrão:
 - Opções de "rastros";
 - Opções de cores padrão;

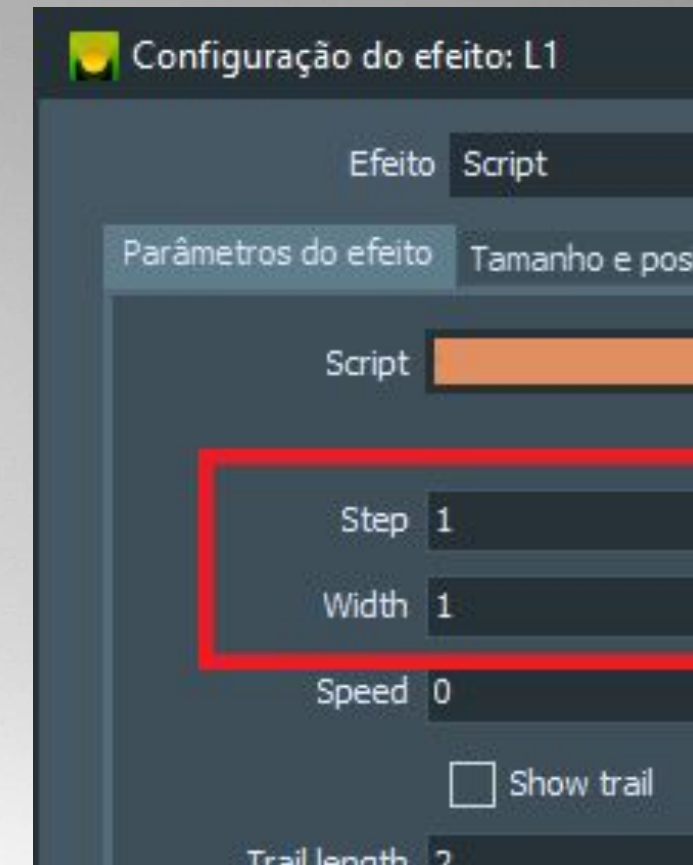
```
19 // Show "trail" option
20 RegisterDefaultTrailOption(true);
21 // Enable/disable "trail" option
22 RegisterDefaultTrailEnabled(false);
23
24
25 // Show default color options
26 RegisterDefaultColorOption(true);
27 // Define de default fixed color
28 RegisterDefaultColor(CL_WHITE);
29 // Show "random" option in color list
30 RegisterDefaultColorRandomOption(false);
```



LUMI SCRIPT - PASSO 3

- Criar um método OnSetParameterInteger e/ou OnSetParameterString, para verificar qual o parâmetro e lá definir o valor das variáveis criadas no primeiro passo:

```
40 end;
41
42 { "OnSetParameterInteger" is called every time
43 that a parameter registered as "Value", "CheckBox", "ComboBox" has changed:
44 param = name of the parameter
45 value = value set by the user }
46 procedure OnSetParameterInteger(const param: string; const value: integer);
47 begin
48   if param='Step' then uiStep := value;
49   if param='Width' then uiWidth := Value;
50 end;
51
52 { "OnExecute" runs every BASE TIME (40ms) to render the images:
```



LUMI SCRIPT - PASSO 4

- Usar as variáveis com o valor dos parâmetros informados dentro do método OnExecute:

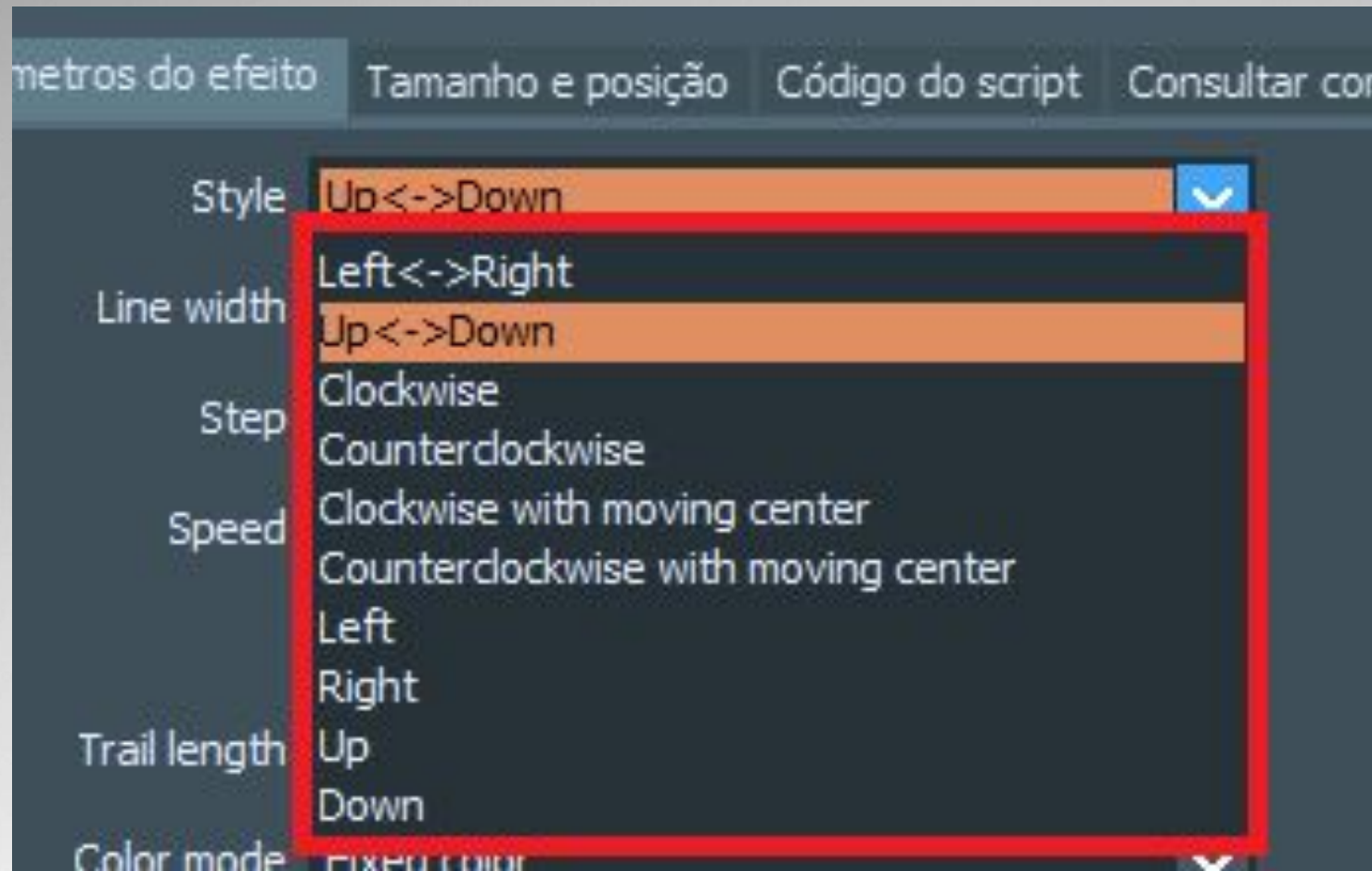
```
53 width = image width in pixels
54 height = image height in pixels }
55 procedure OnExecute(const width, height: integer);
56 var
57   i, halfWidth: integer;
58 begin
59   halfWidth := (uiWidth div 2);
60
61   for i:=-halfWidth to (-halfWidth+uiWidth-1) do
62     begin
63       DoLine(0, curPosition+i, width, curPosition+i);
64     end;
65
66   if direction then
67     begin
68       if curPosition>(height+halfWidth) then
69         direction := false
70       else
71         curPosition := curPosition + uiStep;
72     end
73   else
74     begin
75       if curPosition<-halfWidth then
76         direction := true
77       else
78         curPosition := curPosition - uiStep;
79     end;
80
```

LUMI SCRIPT - PRÁTICA 4

- Novo script que vai fazer o efeito radar (na horizontal):
 - Criar os controles na tela para que o usuário possa informar a largura da linha (Width) e o passo (Step);
 - Bônus;

LUMI SCRIPT - DESAFIO

- Adicionar a esse script feito nesse vídeo, todos os outros NOVE movimentos (Styles) do efeito “Radar” dentro do Lumikit SHOW:



Colocar o código fonte no <https://pastebin.com> e o link para o código aqui nos comentários do vídeo!

Obrigado!

Dúvidas:

suporte@lumikit.com.br