

Programação com a linguagem Céu

Pico-Céu

Anny Caroline

`annycarolinegnr@gmail.com`

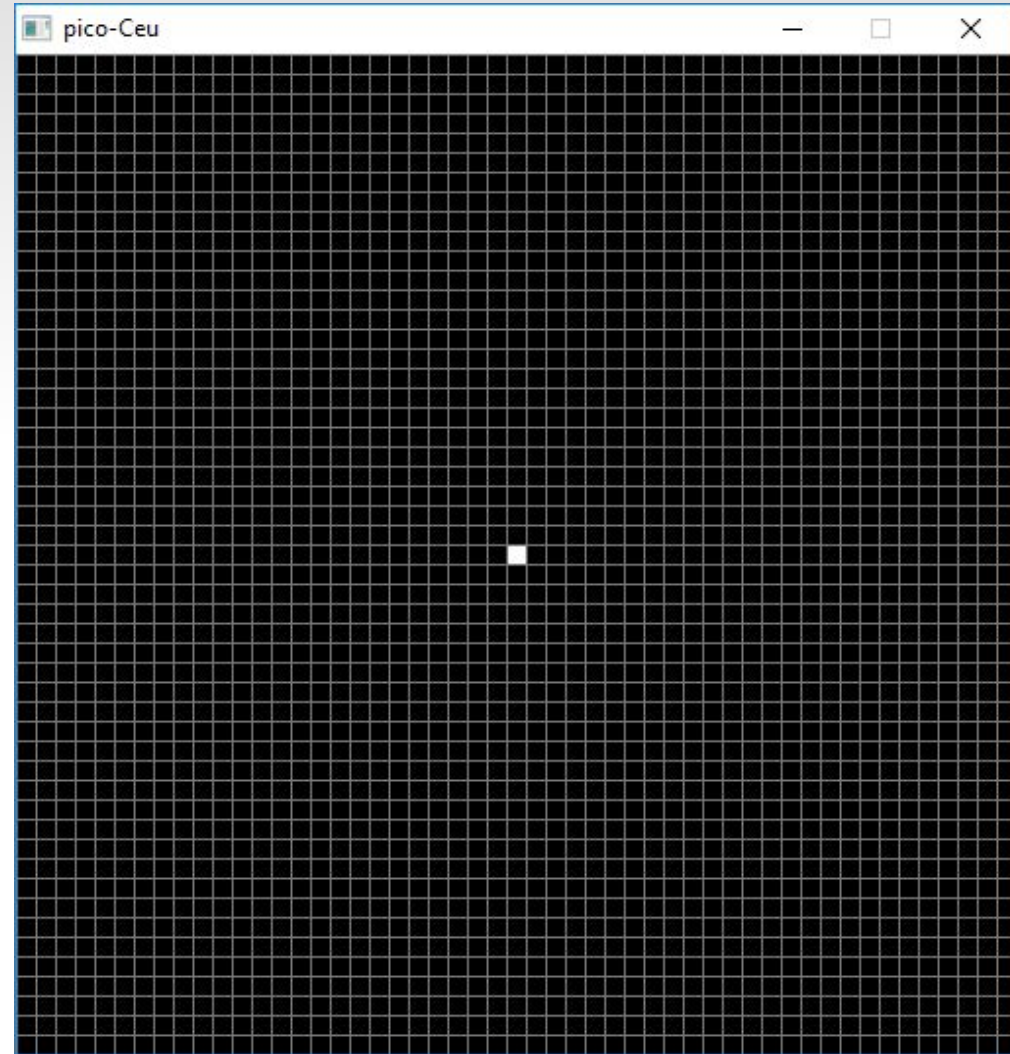
Francisco Sant'Anna

`francisco@ime.uerj.br`

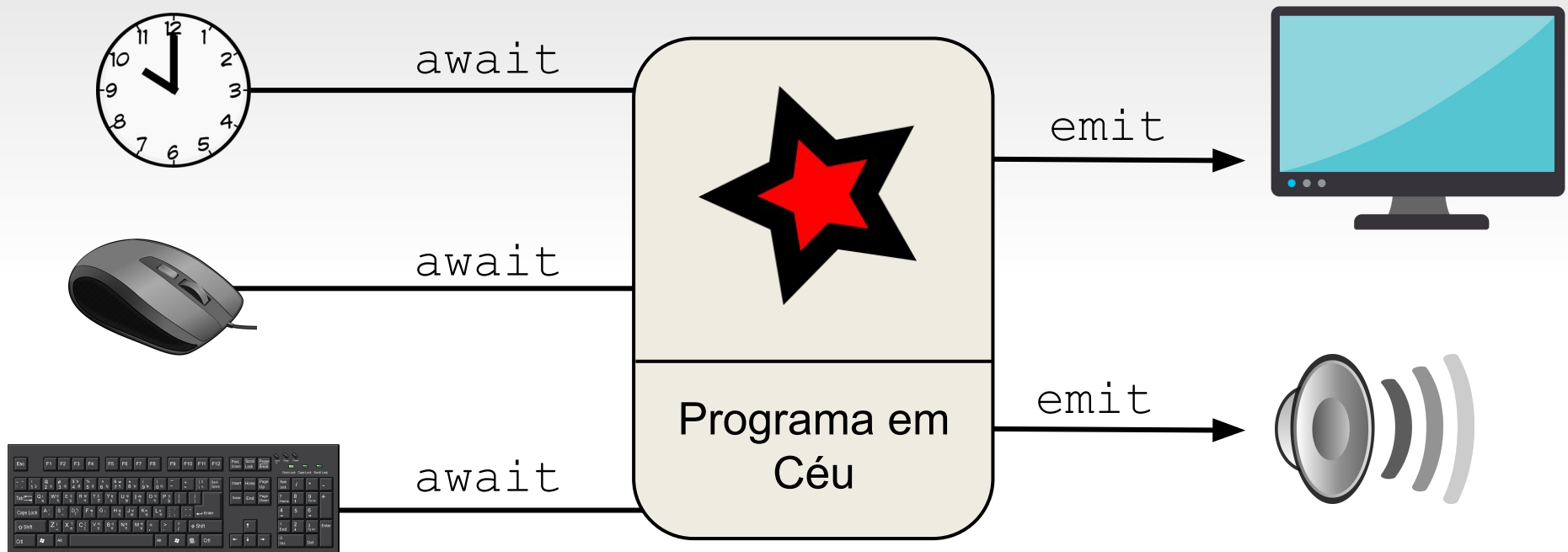
Desenhar um pixel na tela

```
emit GRAPHICS_DRAW_PIXEL(0,0);
```

1-desenharPixel.ceu



Eventos



Eventos

- Emitir um evento

```
emit NOME_DO_EVENTO(parametro1 , ... , parametroN);
```

- Aguardar um evento

```
await NOME_DO_EVENTO;
```

Exercício

- Crie um desenho utilizando o **GRAPHICS_DRAW_PIXEL**
- Tente utilizar o evento **GRAPHICS_SET_COLOR_NAME**

GRAPHICS_SET_COLOR_NAME

- Altera a cor de todas as operações de desenho subsequentes

- COLOR_WHITE

- COLOR_GREEN

- COLOR_GRAY

- COLOR_AQUA

- COLOR_BLACK

- COLOR_TEAL

- COLOR_RED

- COLOR_BLUE

- COLOR_MAROON

- COLOR_NAVY

- COLOR_YELLOW

- COLOR_FUCHSIA

- COLOR_OLIVE

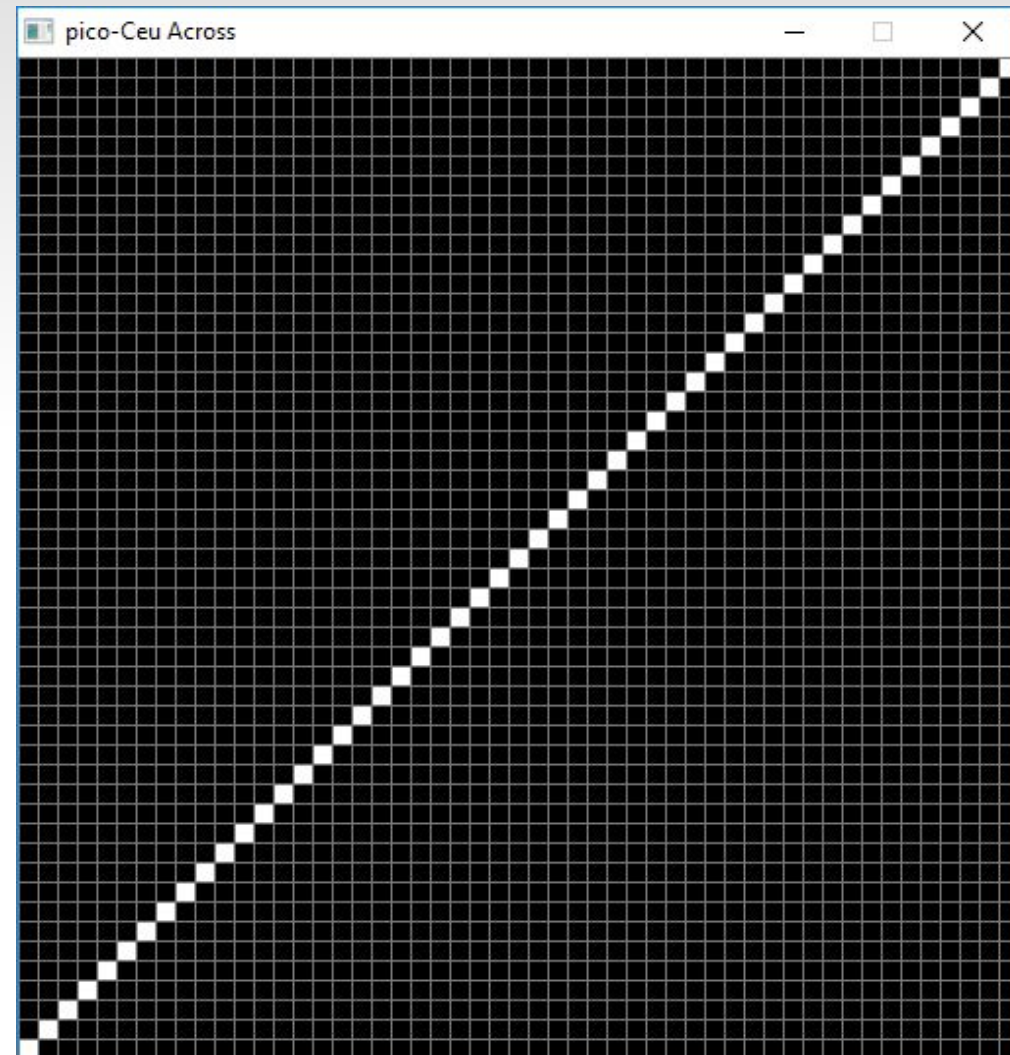
- COLOR_PURPLE

- COLOR_LIME

Desenhar uma linha

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    await 100ms;  
end
```

3-linha.ceu



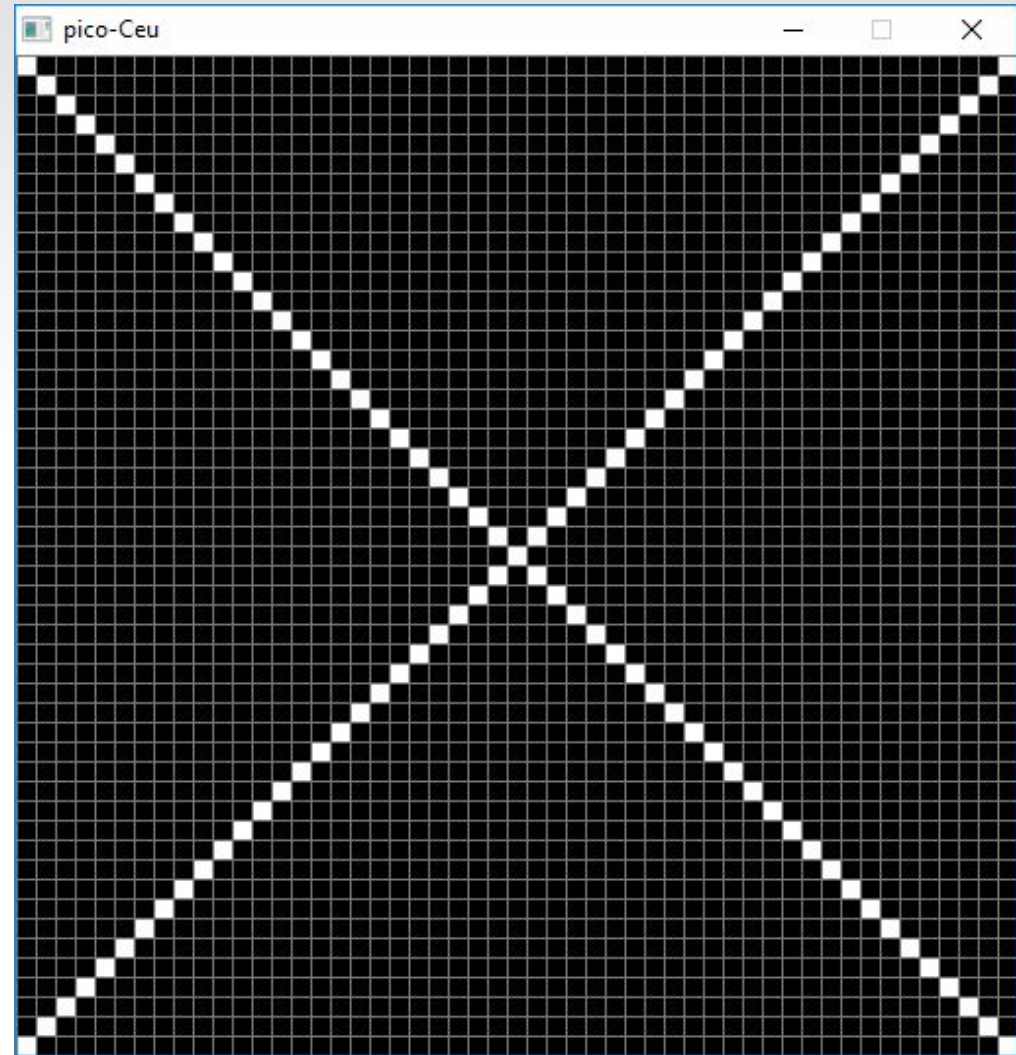
Exercício

- Modifique o exemplo anterior para desenhar um x
 - Utilize quantos loops achar necessário

Exercício - Solução 1

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    await 100ms;  
end  
  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(-i,i);  
    await 100ms;  
end
```

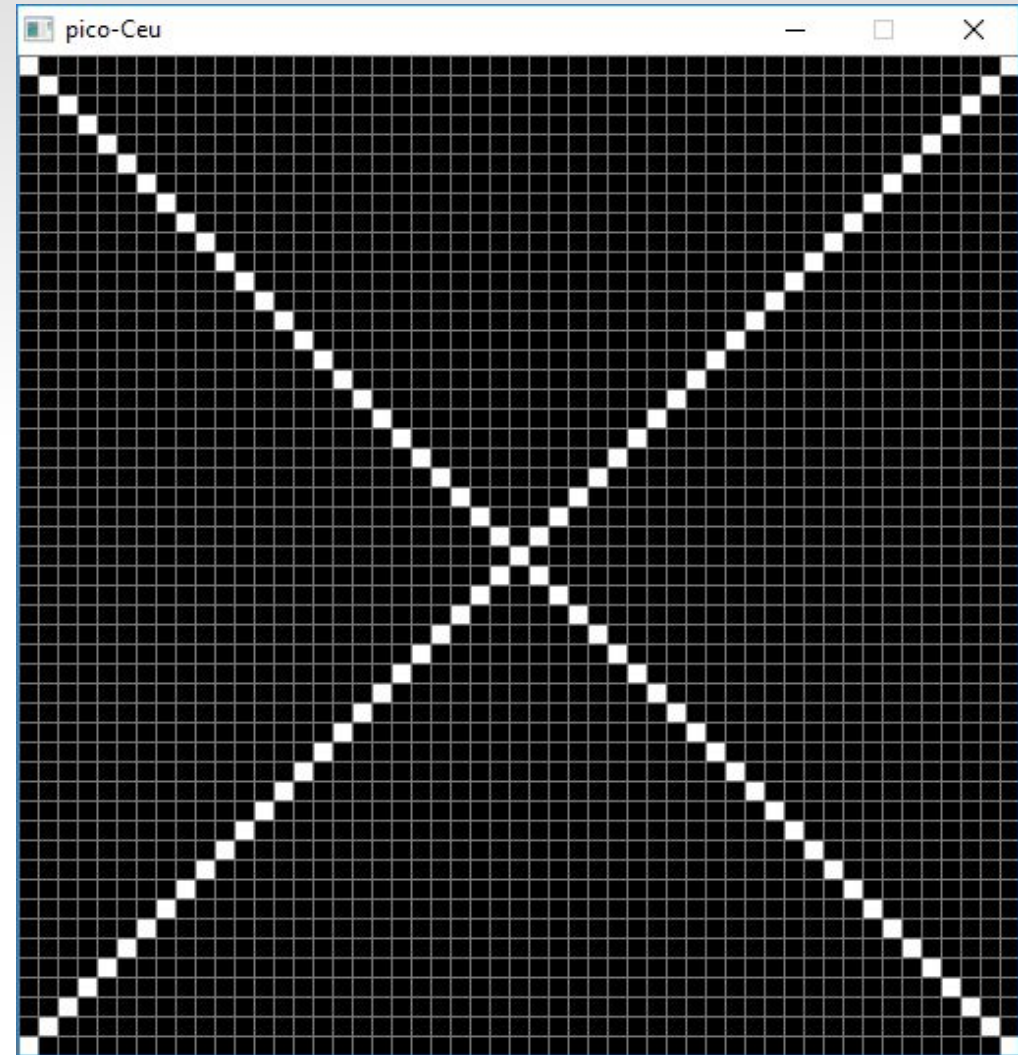
4-e1-x.ceu



Exercício - Solução 2

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    emit GRAPHICS_DRAW_PIXEL(-i,i);  
    await 100ms;  
end
```

4-e2-x.ceu



Em paralelo

par/and do

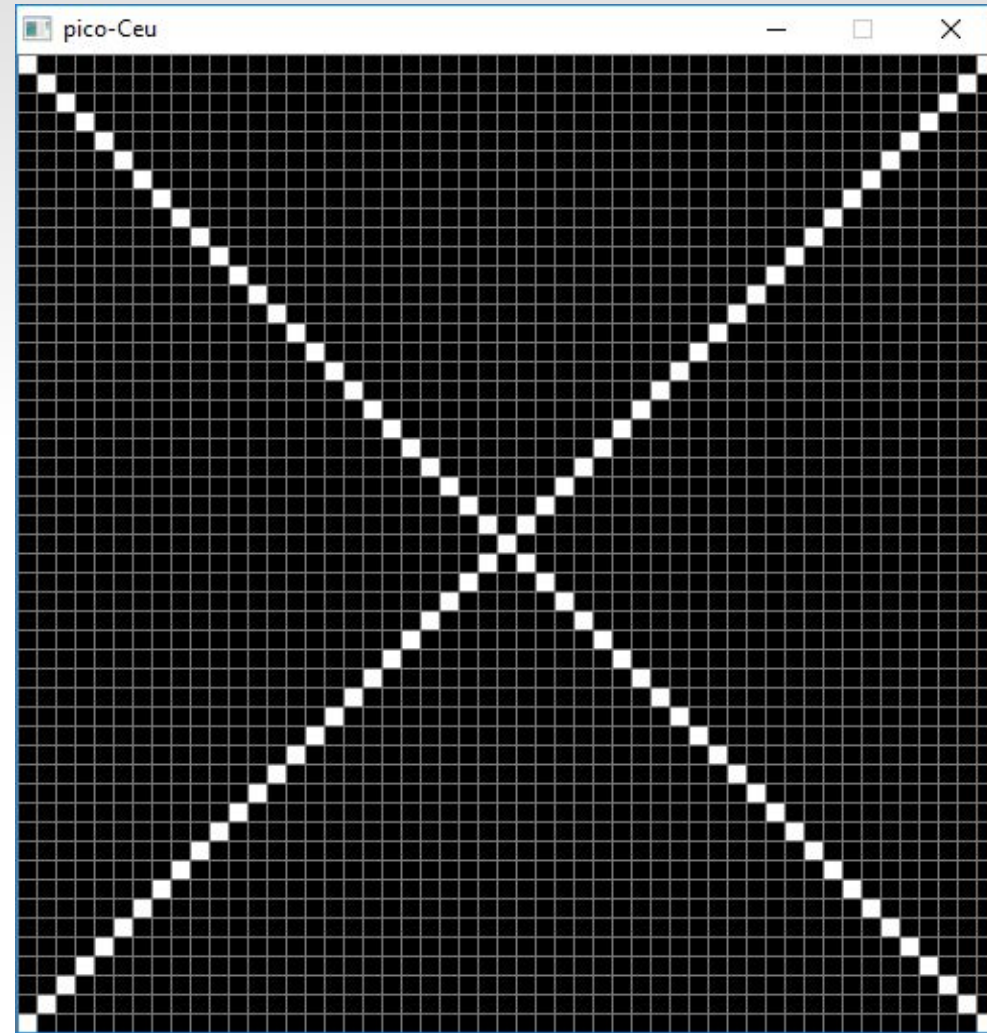
```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    await 100ms;  
end
```

with

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(-i,i);  
    await 100ms;  
end
```

end

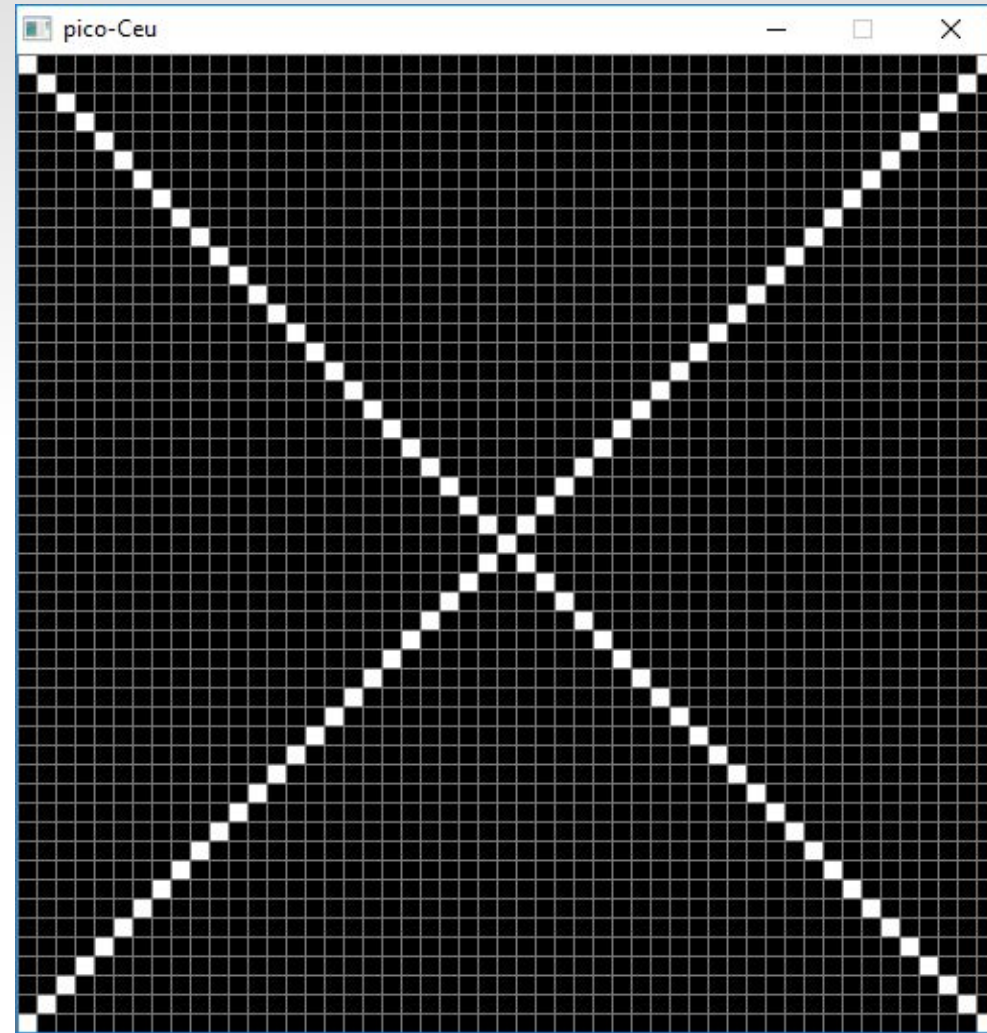
5-parAnd.ceu



Modificando os intervalos

```
par/and do
  var int i;
  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(i,i);
    await 150ms;
  end
with
  var int i;
  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(-i,i);
    await 100ms;
  end
end
```

5-parAnd-intervalo.ceu



Exercício

- Modifique o exemplo anterior para desenhar um asterisco (*)
- Utilize `par/and`

Exercício - Solução

```
par/and do
  var int i;

  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(i,i);
    await 150ms;
  end
with
  var int i;

  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(-i,i);
    await 100ms;
  end
with
  //...
```

```
//...

  var int i;

  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(i,0);
    await 50ms;
  end
with
  var int i;

  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(0,i);
    await 50ms;
  end
end
```

par/or

par/or do

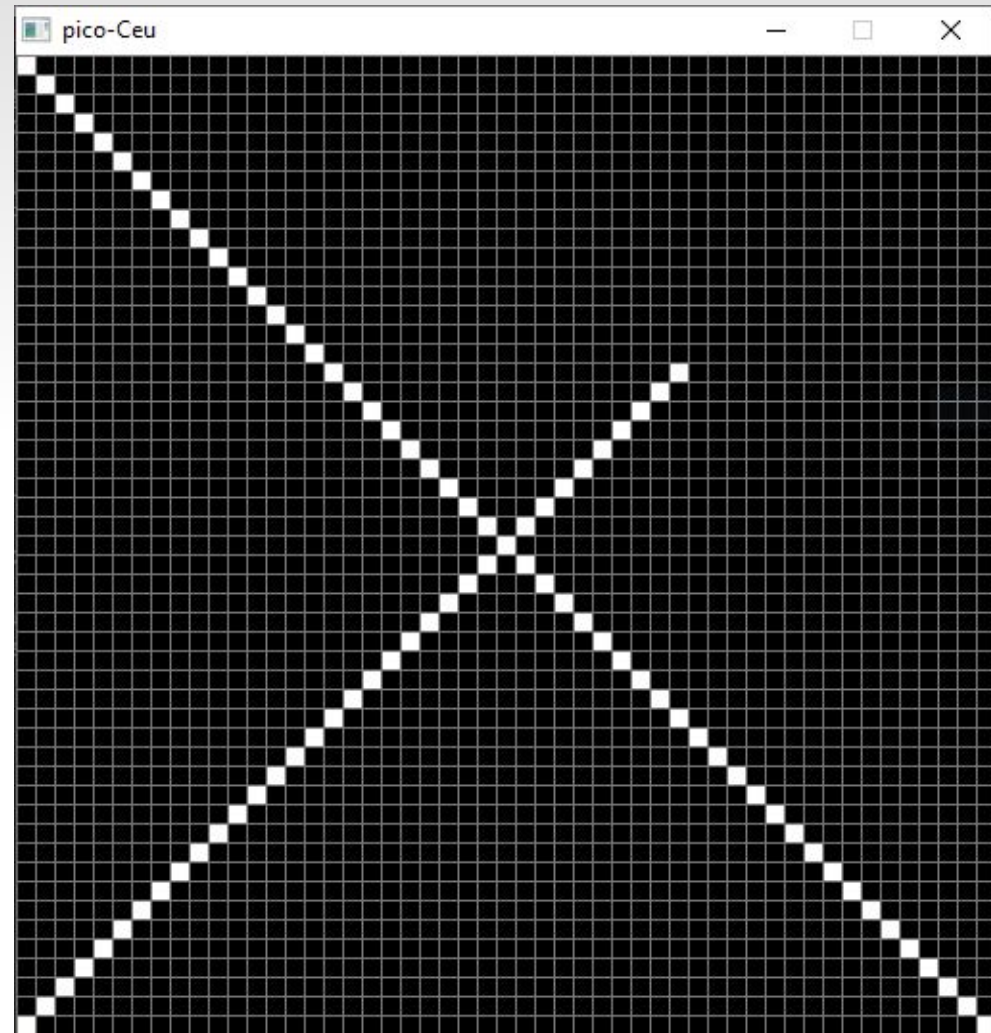
```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    await 150ms;  
end
```

with

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(-i,i);  
    await 100ms;  
end
```

end

8-parOr.ceu



par/or

- O x consegue ser desenhado completamente?
- Por que?

Exercício

- Modifique o exemplo anterior para que a tela seja apagada com o clique do mouse
- Vamos usar dois novos eventos:
 - **WINDOW_CLEAR**
 - **MOUSE_CLICK**

Exercício - Solução

par/or do

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    await 150ms;  
end
```

with

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(-i,i);  
    await 100ms;  
end
```

with

```
await MOUSE_CLICK;
```

end

```
emit WINDOW_CLEAR;
```

Exercício - Perguntas

- O que acontece quando a primeira linha termina de ser desenhada?
- O programa chega a desenhlar a segunda linha completa?

Exercício

- Utilize os eventos `WINDOW_SET_CLEAR_COLOR_NAME` ou `WINDOW_SET_CLEAR_COLOR_RGB` para definir a cor com que a tela deverá ser limpa

Exercício - Solução

```
emit WINDOW_SET_CLEAR_COLOR_RGB(255,0,0);
emit WINDOW_CLEAR;

par/or do
  var int i;
  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(i,i);
    await 150ms;
  end
with
  var int i;
  loop i in [-25 -> 25] do
    emit GRAPHICS_DRAW_PIXEL(-i,i);
    await 100ms;
  end
with
  await MOUSE_CLICK;
end

emit WINDOW_SET_CLEAR_COLOR_NAME(COLOR_YELLOW);
emit WINDOW_CLEAR;
```

E com par/and?

par/and do

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(i,i);  
    await 150ms;  
end
```

with

```
var int i;  
loop i in [-25 -> 25] do  
    emit GRAPHICS_DRAW_PIXEL(-i,i);  
    await 100ms;  
end
```

with

```
await MOUSE_CLICK;
```

end

```
emit WINDOW_CLEAR;
```

E com par/and?

- Quais são as mudanças no comportamento do programa com o uso do **par/and** ao invés do **par/or**?
- O que acontece quando a primeira linha termina de ser desenhada?
- O programa chega a desenhlar a segunda linha completa?
- Enquanto as linhas estão sendo desenhadas, um clique do mouse é capaz de apagar a tela imediatamente?
- Quando a tela é limpa?

Limpar a tela - versão 2

- Desenhe um x com velocidades diferentes
- Apagar a tela ao clicar no mouse
- O x deve ser desenhado completamente

Limpar a tela - versão 2

par/or do

par/and do

var int i;

loop i in [-25 -> 25] do

emit GRAPHICS_DRAW_PIXEL(i,i);

await 150ms;

end

with

var int i;

loop i in [-25 -> 25] do

emit GRAPHICS_DRAW_PIXEL(-i,i);

await 100ms;

end

end

with

await MOUSE_CLICK;

end

emit WINDOW_CLEAR;