

Programação com a linguagem Céu

Frames - 2 players

Anny Caroline

`annycarolinegnr@gmail.com`

Francisco Sant'Anna

`francisco@ime.uerj.br`

FRAMES_REDRAW

- limpar a última posição do pixel
- `FRAMES_SET`
 - Habilita ou desabilita a geração periódica de eventos `FRAMES_UPDATE` e `FRAMES_REDRAW` para a aplicação
 - `emit FRAMES_SET(yes);`

FRAMES_REDRAW

- limpar a última posição do pixel
- `FRAMES_SET`
- Habilita ou desabilita a geração periódica de eventos `FRAMES_UPDATE` e `FRAMES_REDRAW` para a aplicação

```
every FRAMES_REDRAW do  
  //...  
end
```

```
emit FRAMES_SET(yes);

code/await Player(none) -> NEVER do
    var Point pt = val Point(0,0);

    par do
        var int key;
        every key in KEY_PRESS do
            //atualiza pt
        end
    with
        every FRAMES_REDRAW do
            emit GRAPHICS_SET_COLOR_NAME(COLOR_WHITE);
            emit GRAPHICS_DRAW_PIXEL(pt.x, pt.y);
        end
    end
end

end
```

```
emit FRAMES_SET(yes);
```

```
code/await Player(none) -> NEVER do
```

```
  var Point pt = val Point(0,0);
```

```
  par do
```

```
    var int key;
```

```
    every key in KEY_PRESS do
```

```
      //atualiza pt
```

```
    end
```

```
  with
```

```
    every FRAMES_REDRAW do
```

```
      emit GRAPHICS_SET_COLOR_NAME(COLOR_WHITE);
```

```
      emit GRAPHICS_DRAW_PIXEL(pt.x, pt.y);
```

```
    end
```

```
  end
```

```
end
```

```
  if key == KEY_LEFT then
```

```
    pt.x = pt.x - 1;
```

```
  else/if key == KEY_RIGHT then
```

```
    pt.x = pt.x + 1;
```

```
  else/if key == KEY_UP then
```

```
    pt.y = pt.y + 1;
```

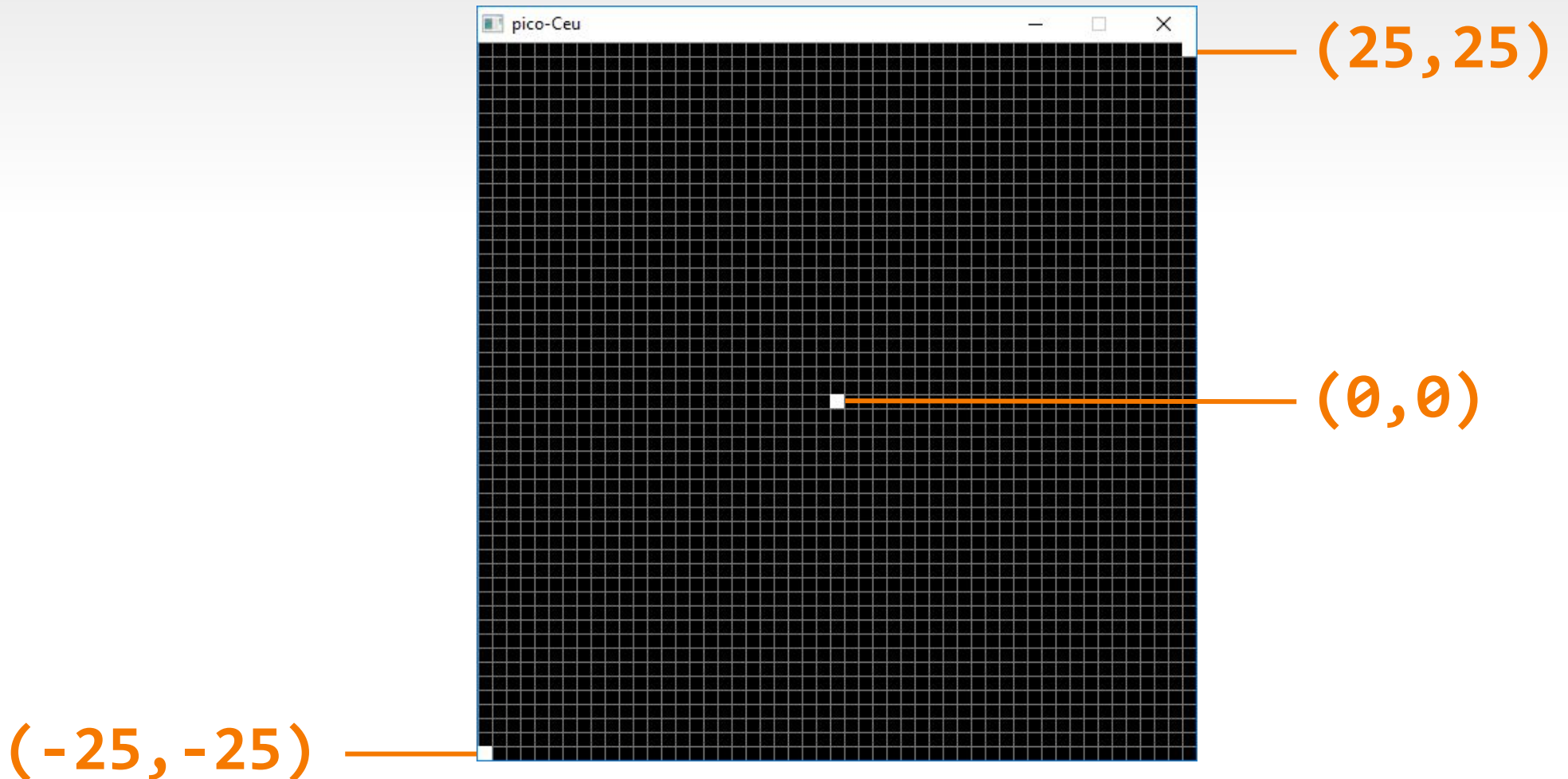
```
  else/if key == KEY_DOWN then
```

```
    pt.y = pt.y - 1;
```

```
  end
```

Exercício

- limitar a movimentação do pixel à janela



```
code/await Player(none) -> NEVER do
```

```
  var Point pt = val Point(0,0);
```

```
  par do
```

```
    var int key;
```

```
    every key in KEY_PRESS do
```

```
      //atualiza pt Só esta parte é modificada
```

```
    end
```

```
  with
```

```
    every FRAMES_REDRAW do
```

```
      emit GRAPHICS_SET_COLOR_NAME(COLOR_WHITE);
```

```
      emit GRAPHICS_DRAW_PIXEL(pt.x, pt.y);
```

```
    end
```

```
  end
```

```
end
```

Solução

```
if (key == KEY_LEFT) and (pt.x > -25) then
    pt.x = pt.x - 1;
else/if (key == KEY_RIGHT) and (pt.x < 25) then
    pt.x = pt.x + 1;
else/if (key == KEY_UP) and (pt.y < 25) then
    pt.y = pt.y + 1;
else/if (key == KEY_DOWN) and (pt.y > -25) then
    pt.y = pt.y - 1;
end
```


Exercício

- movimentar dois pixels ao mesmo tempo
- usar um conjunto de teclas diferentes
 - como se fossem dois jogadores

Solução

```
emit FRAMES_SET(yes);
```

```
code/await Player(var int up,  
                  var int right,  
                  var int down,  
                  var int left,  
                  var Color color) -> NEVER do
```

```
//...
```

```
end
```

```
spawn Player(KEY_UP, KEY_RIGHT, KEY_DOWN, KEY_LEFT, COLOR_BLUE);
```

```
spawn Player(KEY_w , KEY_d      , KEY_s      , KEY_a      , COLOR_RED);
```

```
await FOREVER;
```

```
code/await Player(var int up,  var int right, var int down,
                  var int left, var Color color) -> NEVER do
var Point pt = val Point(0,0);
par do
  var int key;
  every key in KEY_PRESS do
    if (key == left) and (pt.x > -25) then
      pt.x = pt.x - 1;
    else/if (key == right) and (pt.x < 25) then
      pt.x = pt.x + 1;
    else/if (key == up) and (pt.y < 25) then
      pt.y = pt.y + 1;
    else/if (key == down) and (pt.y > -25) then
      pt.y = pt.y - 1;
    end
  end
end
with
  //every FRAME_REDRAW
end
end
```

```
code/await Player(var int up,  var int right, var int down,
                  var int left, var Color color) -> NEVER do
var Point pt = val Point(0,0);
par do
    //every key in KEY_PRESS do
with
    every FRAMES_REDRAW do
        emit GRAPHICS_SET_COLOR_NAME(color);
        emit GRAPHICS_DRAW_PIXEL(pt.x, pt.y);
    end
end
end
end
```