
Stencil Lang

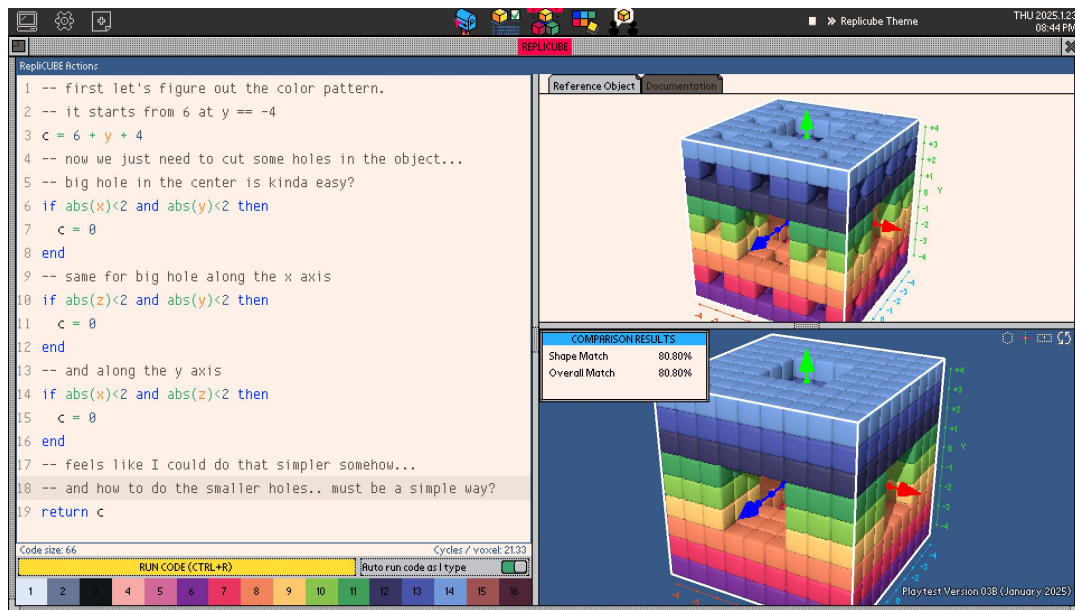
LogComp 25.1
Pedro Fracassi



Inspiração

Replicube é um jogo onde você deve escrever código em Lua para replicar um objeto composto por voxels coloridos. Seu código é executado em loop, uma vez para cada voxel.

A ideia da **Stencil** foi replicar algo parecido com o jogo, porém com uma linguagem própria e em 2D.



Stencils (loops)

Um **stencil** é um "modelo" que pode ser aplicado na tela de pintura posteriormente. O código de um stencil é executado uma vez para cada pixel na área em que ele for aplicado.

Stencils usam o comando **paint** para definir com qual cor o pixel atual será pintado, e passar para a próxima iteração do loop.

Dentro de um stencil, as variáveis **x** e **y** são automaticamente populadas com as coordenadas do pixel atual.

```
stencil circle {  
    var dx = x - 6;  
    var dy = y - 6;  
    var dist = dx * dx + dy * dy;  
    if (dist > 36) {  
        paint 0;  
    } else {  
        paint 1;  
    }  
}
```

Aplicando:


```
apply circle at [2, 2];
```

Tamanho e coordenada

É possível definir onde e com qual tamanho um stencil será aplicado, através das diretivas de tamanho e coordenada do comando apply:

```
apply rainbow at [5, 5] size 5;
```

```
insper-logcomp-aps-again on % main [!+]  
• > ./stencil-run
```



Cores ANSI

Como estamos pintando no terminal, temos as oito cores ANSI, representadas pelos seus números:



Representar as cores com números nos permite fazer coisas interessantes, como desenhar um "arco-íris" utilizando a coordenada atual como cor.

```
stencil rainbow {  
  paint abs(x+y);  
}  
  
apply rainbow at [0, 0];
```



Funções

Além dos stencils, também é possível definir **funções**, que podem ser utilizadas em stencils ou em outras funções para evitar a repetição de código

```
func abs(var x) {  
    if (x < 0) {  
        return -x;  
    } else {  
        return x;  
    }  
}
```

Outras features

Condicionais: É possível utilizar if / else para controlar o fluxo de execução do programa.

```
if (x < 0) {  
    return -x;  
} else {  
    return x;  
}
```

Variáveis: É possível definir variáveis e reutilizá-las posteriormente

```
var a = 1;
```