

Aula 6

Boas práticas, integração e impressão 3D

This work © 2024 by Lucas Seiki Oshiro is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

O que vamos ver hoje?

- Boas práticas de programação
- Integração
- Dicas para impressão 3D

Boas práticas

Boas práticas

Como o OpenSCAD é uma **linguagem de programação**, todas as boas práticas de computação **se aplicam aqui!**

Vamos ver algumas delas e algumas **específicas** do OpenSCAD!

Decomposição

- **Quebrar problemas em problemas menores**
- Um dos conceitos mais importantes da programação
- Como se aplica no OpenSCAD:
 - O modelo vai ser grande? Enxergue ele como **vários modelos pequenos**
 - Faça um por vez, definindo **módulos**
- Facilita a **modelagem**
- Facilita a **manutenção** posterior
 - Você vai querer isso
 - Frequentemente a gente só vê os problemas depois que imprime
- Módulos muito **grandes** pode ser indício de que poderia ser quebrado em **menores**

Evitar números mágicos

- Um **número mágico**:
 - é usado **sem ser explicado**
 - geralmente aparece **mais de uma vez**
 - pode ser substituído por uma **constante**
- Exemplo:
 - imagina se usássemos **3,14159** em vez de **pi**
 - ou pior, usar **algum múltiplo** do **pi**

Legibilidade

- “Qualquer tolo pode escrever código que um **computador** pode entender. Bons programadores escrevem código que os **humanos** podem entender.” - Martin Fowler
- Escreva o código para que **pessoas no futuro** consigam ler com facilidade
 - Inclusive você
- O código deve **falar por si só**
 - Nem sempre é possível, nesses casos, use **comentários**

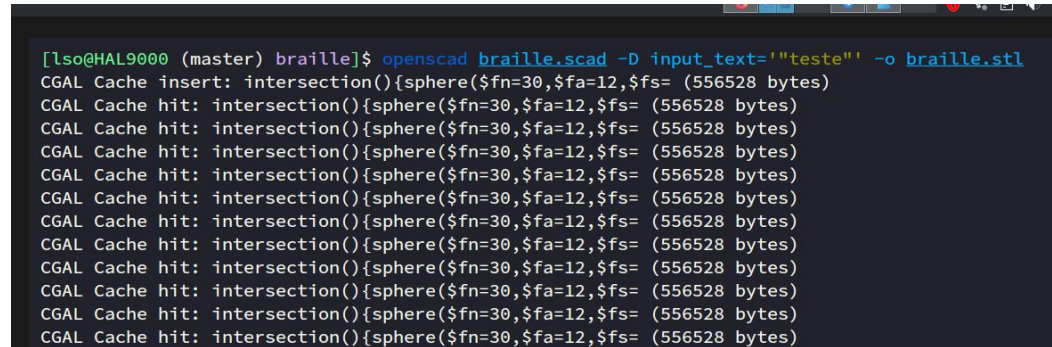
Por fim...

- Boas práticas de programação é um **assunto extenso**
 - Se o curso todo fosse sobre isso, seria muito pouco
 - Isso é tema para até um **doutorado** (inclusive tem alguns no IME)
- Mas existem vários **materiais** sobre isso!
- E boas práticas não são **leis**
 - Por favor, não façam desses livros uma religião!

Integração

Linha de comando

- O **OpenSCAD** pode ser usado sem interface gráfica, direto na **linha de comando!**
- Sintaxe:
 - `openscad entrada.scad -o saida.stl -D parametro=valor`

A terminal window showing the execution of the OpenSCAD command. The prompt is [lso@HAL9000 (master) braille]\$ and the command is openscad braille.scad -D input_text="teste" -o braille.stl. The output consists of 10 lines of CGAL Cache hit messages, each indicating an intersection of a sphere with parameters fn=30, fa=12, and fs=, resulting in 556528 bytes.

```
[lso@HAL9000 (master) braille]$ openscad braille.scad -D input_text="teste" -o braille.stl
CGAL Cache insert: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
CGAL Cache hit: intersection(){sphere($fn=30,$fa=12,$fs= (556528 bytes)
```

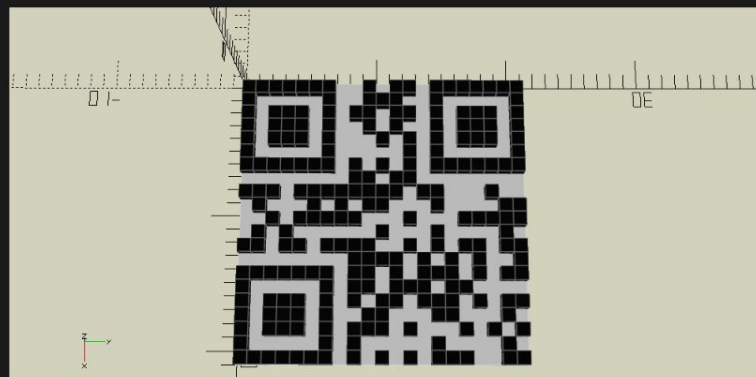
SolidPython

- Permite gerar código **OpenSCAD** em um código **Python**
- Junta o **poder** do Python com a modelagem do **OpenSCAD**

```

1 #!/usr/bin/env python3
2
3 import qrcode
4 from solid import *
5 import numpy as np
6 import matplotlib.pyplot as plt
7
8
9 qr = qrcode.QRCode(
10     version=1,
11     error_correction=qrcode.constants.ERROR_CORRECT_L,
12     box_size=1,
13     border=0,
14 )
15 qr.add_data(input())
16 qr.make(fit=True)
17
18 qr = np.array(qr.make_image(fill_color="black", back_color="white"))
19
20 scad = scale([1, 1, 1])(
21     color('white')(cube([*qr.shape, 1])) + |
22     color('black')(
23         union()(
24             translate([i, j, 1])(cube(1)) |
25             for i, row in enumerate(qr) |
26             for j, cell in enumerate(row) |
27             if not cell
28         )
29     )
30 )
31
32 |
33 print(scad_render(scad))
34

```



Console

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version.

Error-Log

Show All

Group Line

Viewport: translate = [8.57 10.73 1.42], rotate = [12.30 0.00 90.10], distance = 68.66, fov = 22.50 (1003x495) OpenSCAD 2021.01

```

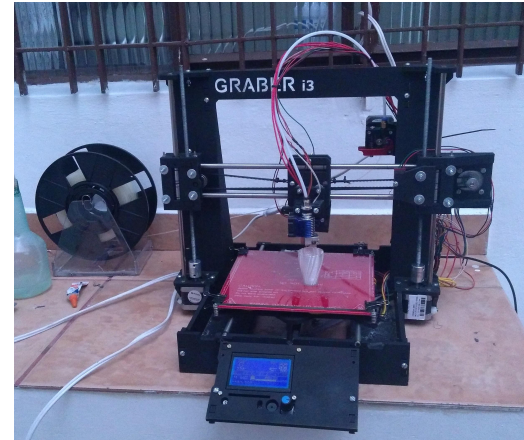
[lso@HAL9000 Desktop]$ ./qrcode_stl.py > out.scad
teste
[lso@HAL9000 Desktop]$ openscad out.scad
|

```

Impressão 3D

Impressão 3D

- Máquinas que transformam arquivos em **objetos físicos**
- Manufatura **aditiva**
- Tipos principais:
 - Filamento (FDM)
 - Resina (SLA)
- Máquinas parecidas:
 - Fresadoras
 - Cortadoras a laser



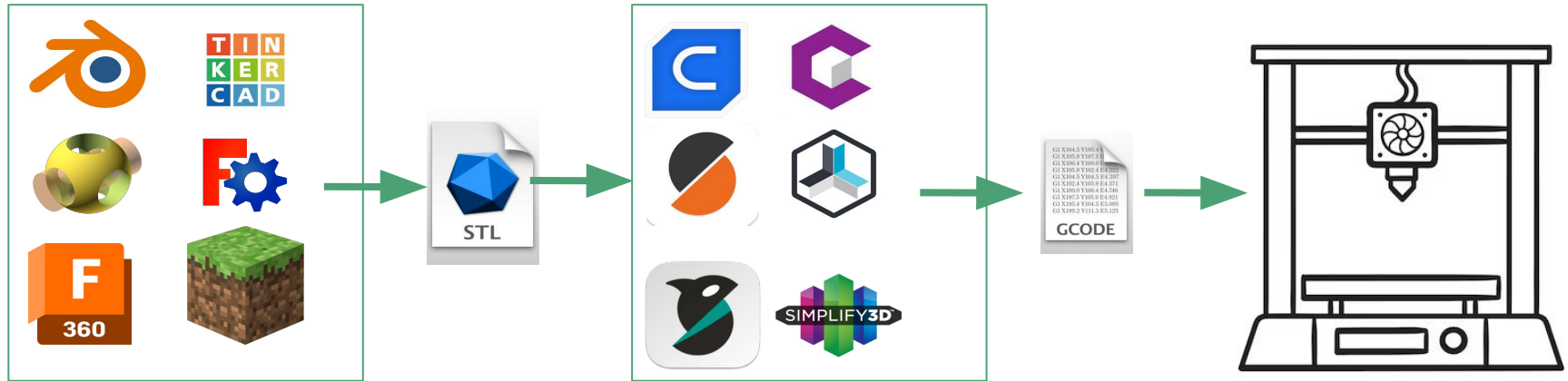
Materiais

- Resina
- Filamentos:
 - PLA
 - PET-G
 - ABS
 - TPU
 - HIPS

Processo de impressão



Processo de impressão



Dicas de modelagem para impressão 3D

- Prefira **bordas arredondadas**
- Prefira ângulos verticais de no máximo, **30 graus**
 - Caso contrário pode precisar de suporte
- Evite **pontes**
- Leve em conta a **folga** entre os encaixes
 - O OpenSCAD é bom pra isso, já que dá pra calcular os encaixes com base nas dimensões e nas folgas!
- **Evite detalhes** pequenos em relação ao tamanho da peça