

Python部分

Installation

```
pip install polyscope  
pip install tqdm  
pip install scipy
```

Run

```
python ./main.py [if_display] [repulsion_constraint] [center_constraint] []
```

if_display: 1:显示结果并处理json文件，0：只处理文件不显示结果

repulsion_constraint(optional): 数值越大不同channel之间距离越大，默认0.1

center_constraint(optional): 数值越大每个的channel的连接点越靠近整个球体的中心（即向中心聚拢 / 间距变小），默认0.5

force_udpate(optional): 1: 处理所有json文件，0: 如果json文件已经被处理过则会跳过处理

Example:

```
python ./main.py 1 0.8 0.08
```

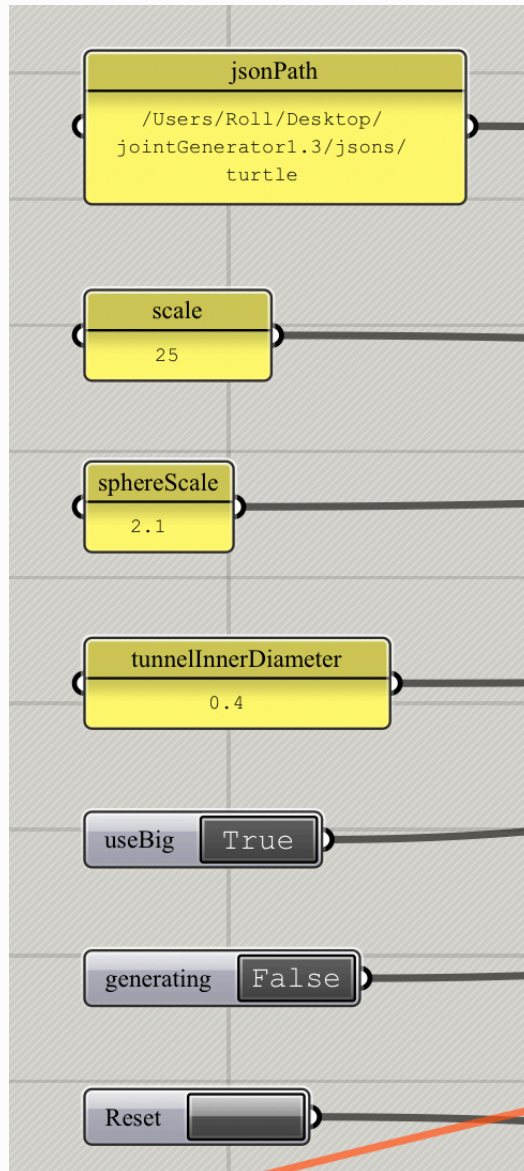
备注:

- 点可视化界面的关闭键会自动显示下一组气道的结构以此类推，如果想直接关掉可视化需要在命令行强退
- 最终你输入的json文件会被修改，然后就可以放进grasshopper里做下一步生成

Grasshopper 部分

新版的生成器会自动地进行布尔运算，依次生成连接件，运算过程比较慢，一分钟大约生成3~6个连接件。

使用方法： 设定完文件路径和参数后，点击"reset"，再双击"generating"，然后等待连接件一个一个生成。

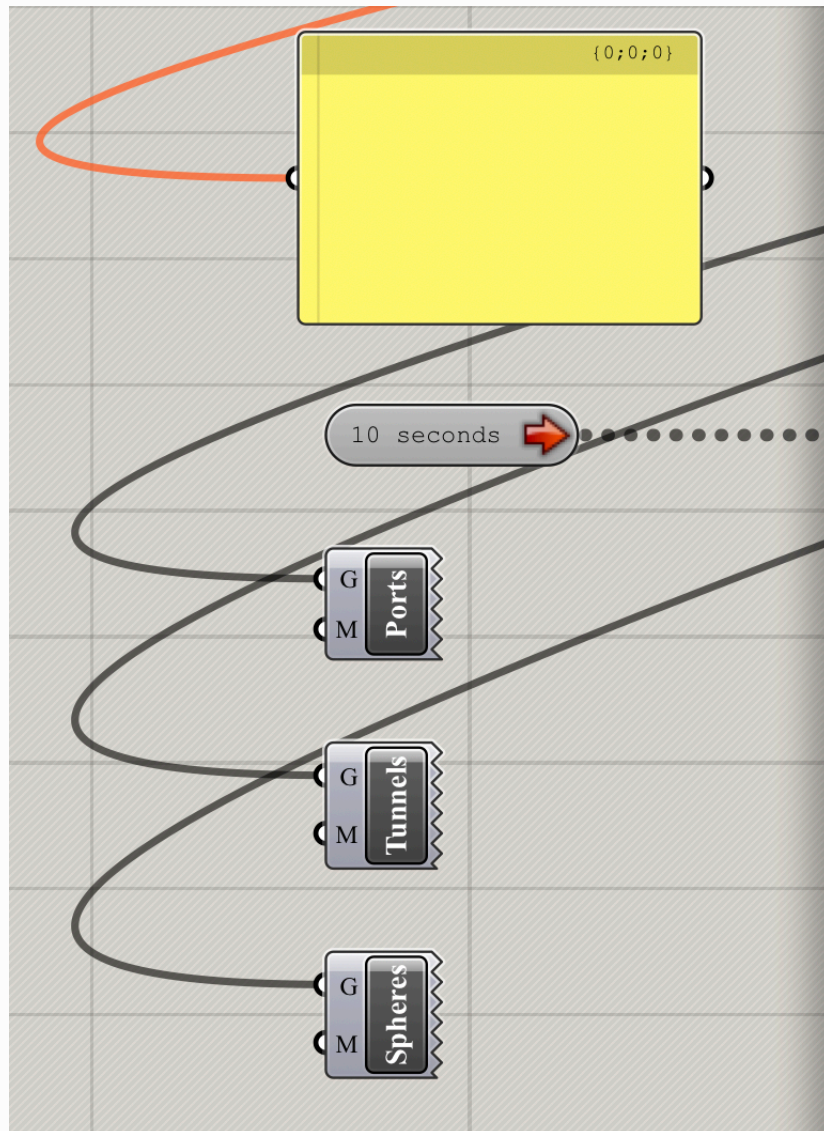


左侧的主要工具分别为:

1. jsonPath: python处理完的json文件路径，不用写".json"
2. scale: 放大倍率（只是joint位置的放大倍率，joint形状不变）
3. sphereScale: joint上的球的大小相对于管道直径的倍率
4. tunnelInnerRadius: 通道的相对内半径，球的内半径为1.0
5. useBig: 用大joint还是小joint
6. generating: False的状态即脚本暂停，True运行

7. Reset: 每次开始生成前最好点击一下Reset

左侧次要工具分别为：



1. 显示运行和报错信息，通常不用操作
2. 开始生成连接件的时间间隔，通常不用操作
3. Ports: 接口模型，只在出错后需要手动布尔时使用
4. Tunnels: 气道模型，只在出错后需要手动布尔时使用
5. Spheres: 球体模型，只在出错后需要手动布尔时使用