

作业4 学习GH工具

晏瑞然 87 PB19000196

实验目的

学习节点编程工具GH，并用其实现之前的某次作业，本次作业选择实现之前的曲线拟合任务。

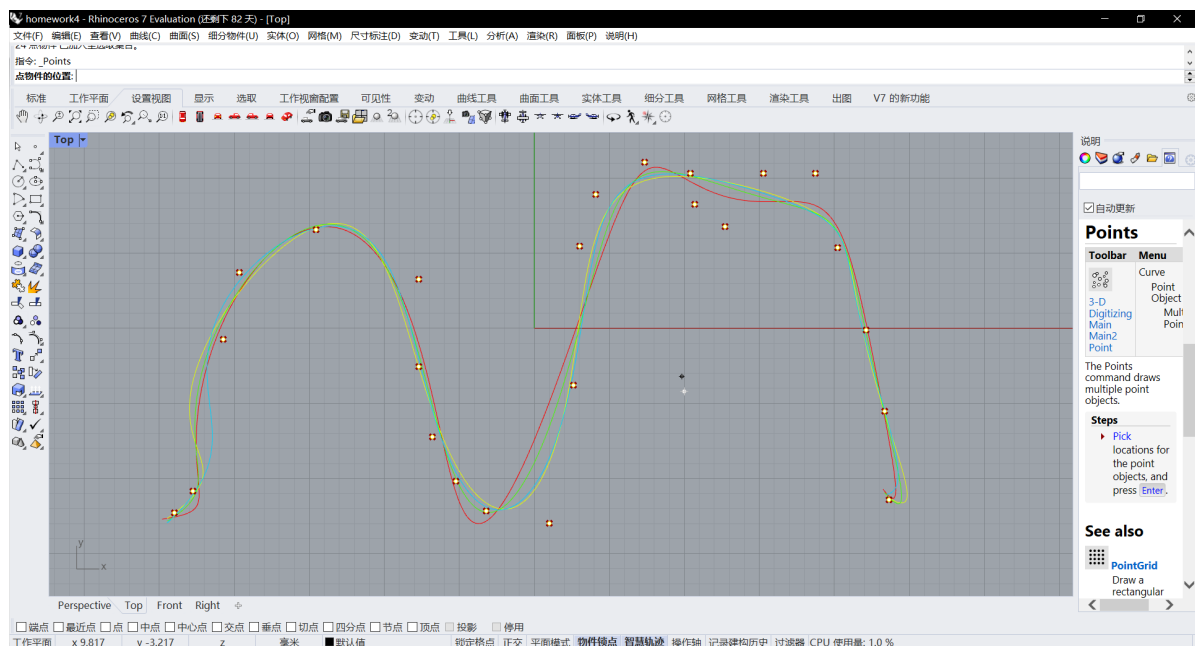
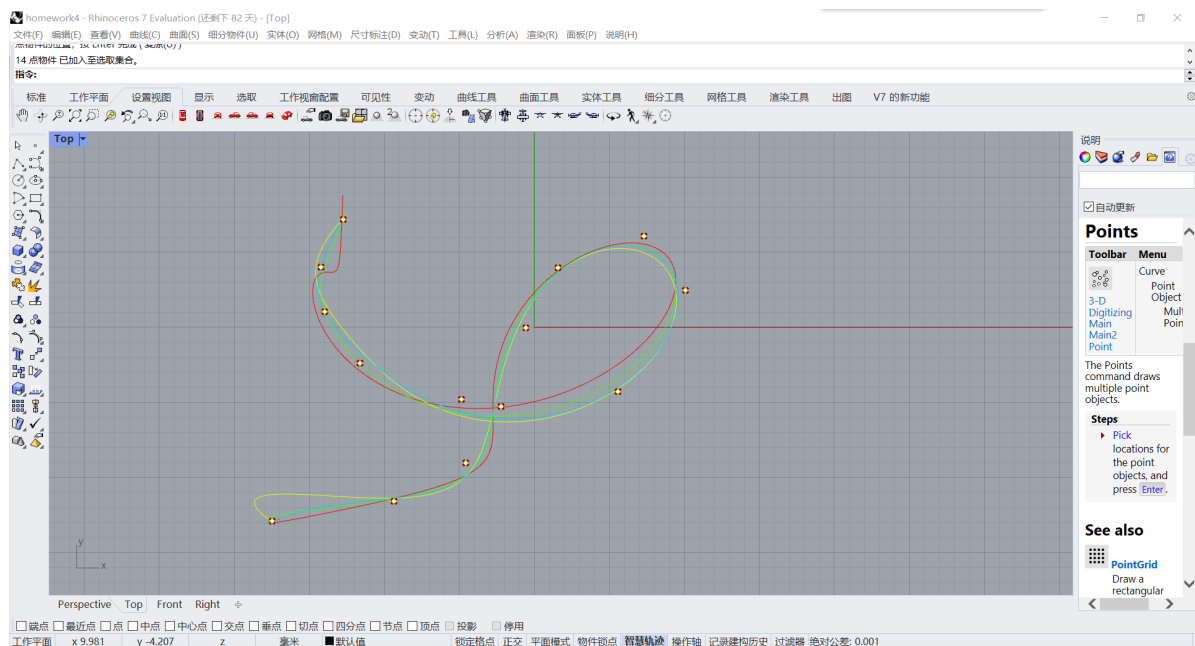
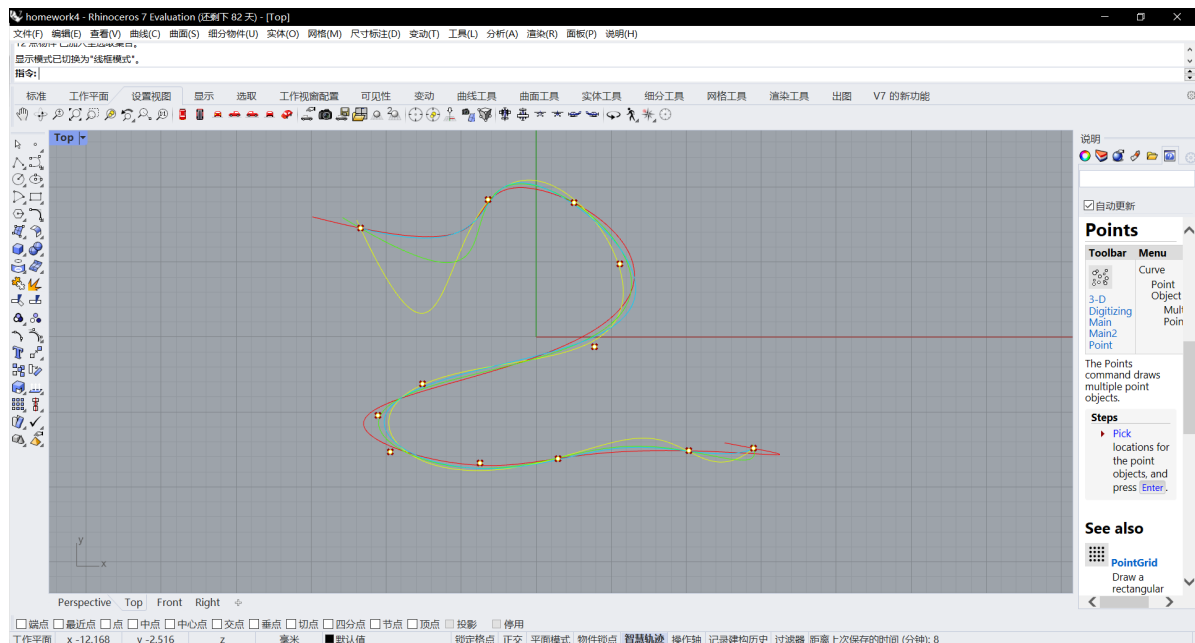
程序说明

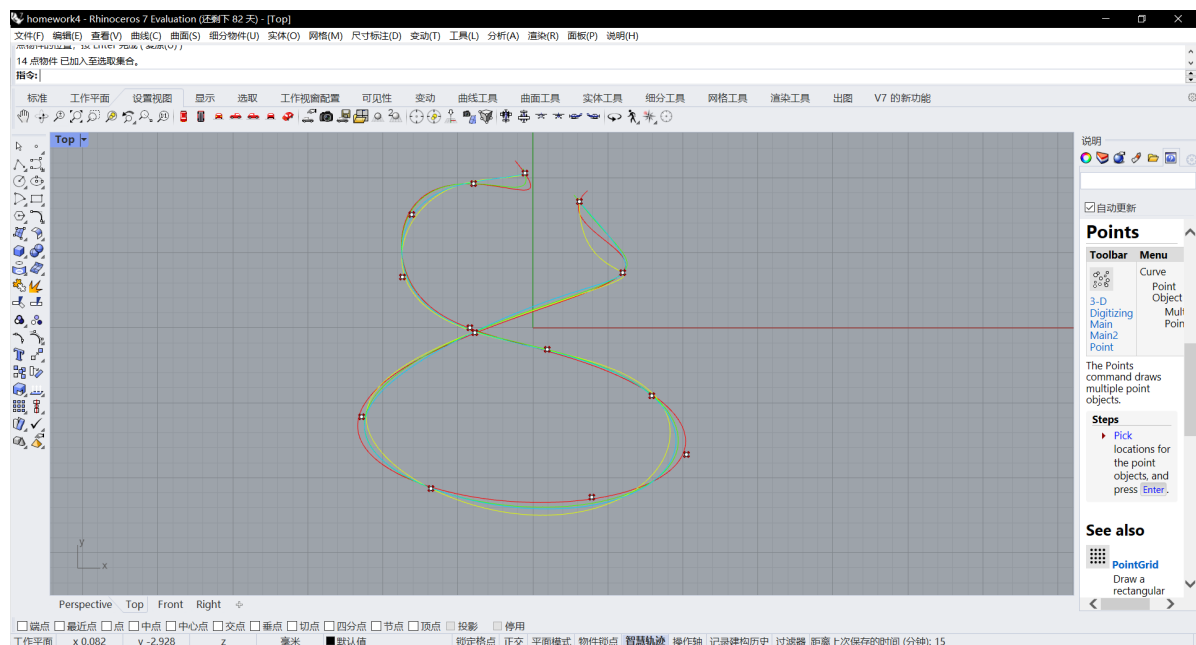
详细程序见附件homework4.gh文件。

下面对程序进行简单说明：

1. 取点：在Rhino面板中取点后加入到GH面板的points中，这些点作为生成拟合曲线的初始点。
2. 调节degree：程序中加入了一个slide可以进行回归次数degree的调节，其取值范围是1~10。
3. CPython电池：上述points电池与degree电池共同连接到一个CPython电池作为输入，通过不同参数化法回归返回三个列表xvals、yvals、zvals，得到过程是由points及degree输入及不同参数化法生成的参数进行最小二乘回归后得到多项式拟合曲线，在曲线上取1000个点每个点的xyz坐标。
4. lists2points：得到的三点坐标组经过一个lists2points电池转换为GH中的Points对象。
5. 插值拟合：为了将1000个点转换成GH中的曲线object，直接调用GH中Crv的电池Interpolate，其功能为对离散点进行插值。该电池缺陷为只能进行插值而不能进行拟合故当degree过低而points过多时返回的不是拟合结果。而本程序通过CPython电池进行拟合后按拟合曲线生成1000个点，用这些点进行插值就不会出现degree过低而points过多的情况，并且这种情况下插值与拟合的效果是一样的不会影响结果。
6. 修改颜色：通过preview与swatch电池对曲线object进行改色，不同参数化法得到的曲线颜色不一样，具体颜色如下：
 - 均匀参数化：红色
 - 弦长参数化：黄色
 - 中心参数化：绿色
 - Foley参数化：蓝色

结果





实验结果如上图所示。

总结

通过本次作业，了解了GH这种节点式编程方式，将程序编程一个个电池来可视化可以说是非常简单方便易于上手，并且各种电池库功能齐全而强大，唯一的缺点可能就是其没有linux版本让Linux用户十分难受TAT。助教录的视频和给的example也做的非常好，本次实验有许多地方都借鉴了助教的example的方式，可以说它为学习GH有很好的帮助。本次作业总用时大概四五小时左右(包括报告及看助教的视频)，非常开心能在短时间内学会这样一个如此简单、有趣而强大的工具。