

基于 Geogebra 的脚本编程技术研究

蔡 渊

武昌工学院, 中国 · 湖北 武汉 430065

【摘要】 Geogebra (以下简称 GGB) 是一款著名的数学、物理建模软件, 功能十分强大。会编程的用户能利用 GGB 的脚本编程来编制出更加复杂、拥有动态效果的程序。本文从最基本的指令开始介绍, 系统阐述 GGB 脚本编程的概念、原理和用法。

【关键词】 Geogebra; GGB; 指令; 脚本

1 指令和脚本简介

GGB 的使用方法有 3 个维度, 分别是: 工具、指令和脚本。其中工具就类似于一般制图工具里的绘图功能。指令就是函数, 前面的工具在本质上都是函数操作。脚本就是把多条指令放在一起, 另外再加一些选择和循环结构。这样看来, GGB 的脚本编程跟一般的编程其实几乎一样。

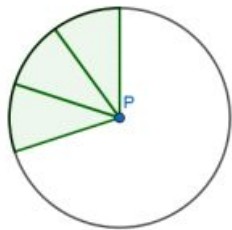
指令是脚本的主要组成部分, 也可认为指令就是最简单的脚本。指令本身分为两种:

1.1 构造型指令: 即用于构造对象的指令, 返回值是对象名。如 $c = \text{circle}((3, 1), 1)$ 表示以 (3, 1) 为圆心, 以 1 为半径做圆。

1.2 动作型指令: 即用于执行某动作的指令, 无返回值。如 $\text{translate}(f, (1, 2))$ 表示将对象 f 平移向量 (1, 2)。

2 指令的嵌套

指令的嵌套就是指令里面套有指令。现有一例, 在已有圆上作 3 个扇形, 其图形效果和对应的指令如下所示:



$\text{Rotate}(\text{Sequence}(\text{CircularSector}(P, \text{point}(p, (t-1)/10), \text{point}(p, t/10)), t, 1, 3), -\pi/2, P)$

对于该嵌套指令的分析如下:

point 函数的参数有很多种, 对应着不同的描点方式和条件, 这里使用的是: 描点(对象, 路径值)。即在有限长度的对象上描点, 用路径值(其实是一个小于 1 的分数)来表征该点在对象上的位置(即在全程的几分之几处)。

CircularSector 函数的参数有很多种, 对应着在圆或椭圆上画扇形的不同方式和条件, 这里使用的是: 扇形(圆/椭圆, 点 1, 点 2), 两点的方向为逆时针。

rotate 函数的参数有很多种, 对应着不同的旋转方式和条件, 这里使用的是: 旋转(对象, 角度, 中心)。

本嵌套指令的层次是: 旋转里面套着序列, 序列里面套着扇形的生成, 扇形的生成里面套着点的生成。

指令的嵌套有个规律, 即被某指令嵌套的指令或操作所产生的对象只是临时的, 不会被保存。

对于本例而言, $\text{point}()$ 是被 $\text{CircularSector}()$ 嵌套的, 那么 $\text{point}()$ 生成的点没被保存。

$\text{CircularSector}()$ 是被 $\text{sequence}()$ 嵌套的, 为什么 $\text{CircularSector}()$ 生成的扇形却还在呢? 原因是 $\text{sequence}()$ 不是一个指令, 而只是一个循环操作。它的作用只是让 $\text{CircularSector}()$ 执行了 3 遍。为了以示区别, 将其倾斜, 且取消加粗。在研究指令嵌套时, 完全可以无视 sequence , 而只看它所包裹的多个操作或生成指令。

$\text{Sequence}()$ 所包裹的 3 个并列的 $\text{CircularSector}()$ 是被 ro

$\text{tate}()$ 直接嵌套的, 那么 3 个并列的 $\text{CircularSector}()$ 所生成的 3 个原始扇形没被保存。最后我们只看到了经过 $\text{rotate}()$ 旋转到新位置的 3 个扇形。

利用嵌套的这个特性, 我们可以去掉如旋转、对称、平移等操作的复制效果。即把要操作的对象的生成语句, 嵌套到操作指令中。

3 选择结构

GGB 的选择结构跟其它编程语言一样, 也是使用的关键字 $\text{if}()$ 。

If 语句的完整结构是: $\text{if}(\text{条件表达式 } 1, \text{语句 } 1, \text{条件表达式 } 2, \text{语句 } 2, \dots, \text{条件表达式 } n, \text{语句 } n, (\text{默认}) \text{语句})$

在分析 GGB 的选择结构时, 完全可忽视 $\text{if}()$, 只看条件后的表达式。 $\text{if}()$ 只起到一个添加条件的作用。

如输入 $y = \text{if}(x > 0, x^2)$, 会绘制出抛物线的右半段。

再看一个分步演示的例子: 实现 5 个对象通过变量控制能依次出现。

创建一个变量 a , 范围为 0 到 5, 步长为 1。

在第 1 个对象的属性——高级——显示条件里填写: $a > 1$ 。在第 2 个的显示条件里写 $a > 2$, 以此类推。

创建“下一步”按钮, 脚本为: $\text{setvalue}(a, a+1) \text{ if}(a > 5, \text{setvalue}(a, 5))$, 也可一起写成: $\text{if}(a > 5, \text{setvalue}(a, 5), \text{setvalue}(a, a+1))$, if 的作用是让 a 不超过 5。

创建“上一步”按钮, 脚本为: $\text{setvalue}(a, a-1) \text{ if}(a < 0, \text{setvalue}(a, 0))$, 也可一起写成: $\text{if}(a < 0, \text{setvalue}(a, 0), \text{setvalue}(a, a-1))$, if 的作用是让 a 不小于 0。

从上面例子我们都不难看出, $\text{if}()$ 的使用很灵活, 不仅能用到纯数学表达上, 还能用到逻辑处理上。

4 列表和序列

4.1 列表: 集合

List 就是一个大集合, 里面可以有不同类型的多个对象, 里面的元素是有顺序的, 可重复的

使用列表的例子: 在输入框中输入: 多项式拟合 (11, 2)。参数中的 11 代表已经存在的一个点列表, 2 表示二次函数

或者输入: 多项式拟合 ({A, B, C, D}, 2) // 这是现场创建一个临时列表

4.2 序列: for 循环 + 数列

序列是特殊的、有规则的列表。

序列这种列表里的元素是由通项公式用 for 循环算出来的。所以序列就是由 for 循环实现的数列。

数列是一种集合, 所以序列是一种列表。序列在代数区中的表示就是列表 list。

序列在绘图区是个整体, 不能个别删除, 在代数区体现为一个集合。

序列的通式为: $\text{sequence}(\text{表达式}, \text{变量}, \text{初值}, \text{终值}, \text{步长})$

步长: 当为 1 时, 可不写。步长也能是一个变量, 这时该变量和上面的 n 都有了应对滑动条。但滑动 n 不会有任何变化, 应该是系统的一个 bug。

(下转 203 页)

教师惩戒权的使用方式,帮助学生避免再次发生不合规行为。高校可通过线上线下教育方式,结合新媒体教学手段全方位开展普法教育。例如通过线上方式举办与教师惩戒权相关法律方面的知识讲座,通过线下方式展开集中讨论,可以是师生间的讨论,也可以是学生间的讨论,也可以是教师、家长与学生间的讨论,确保教师惩戒权的合理运用。此外,家长也要不断更新自身传统的教育观念,增强自身法律意识。学校可以定期举办主体鲜明的家长洽谈会,就教师惩戒权的使用方式与使用标准展开讨论,有针对性的制定相关惩戒方案。这种方式有助于指引学生更好的面对教师的不良惩戒行为,并逐渐形成维权意识,通过相关法律途径保护自身的权益。

3.3 明确教师惩戒权的范围、方式与程度

教师惩戒权旨在防止学生出现不合规行为。因此,教师使用惩戒时,应根据等级与方式选择。首先,最高的惩戒不能侵害学生最基本的人格权利。例如,开除或劝退学生。其次,教师使用惩戒权时,确保等级与违规程度相匹配。而且,教师惩戒的对象时学生的不合规行为,而不是学生自身。教师惩戒权主要有三个等级,分为“一般”、“较重”、“严重”三个方面。因此,学校要根据学生不合规行为程度选择合适的惩戒手段,但要避免惩戒过度。且要注意惩戒的对象。教师惩戒权主要是针对学生不良行为的惩戒,而不是成绩或者学生自身。最后,选择惩戒的方式要恰到好处。教师形式惩戒权时,既不能影响正常教学秩序,也不能影响学生课堂学习效率。教师要多维角度发挥惩戒权的作用,分轻重缓急,选择合适的时机与惩戒手段对学生不良行为实施惩戒。

4 结语

随着教育方式与手段的转变,高校里师生间关系逐渐由控制性变为合作型。教师逐渐从授课的主体转变为引导学生成为课堂主体,并不断变革传统教学观念,提高课堂教学质量与学生参与度。因此,在当前教育过程中,赏识教育逐渐成为高校中重要的教学方式,惩戒式教学逐渐减少。然而教师惩戒权是不可忽视的。教师惩戒权作为一种强制性较弱的教学手段,能够保证减少学生在校期间的不合规行为。教师惩戒权随着时代的变迁,也逐渐从传统的体罚转变为有明确的边界与具体限制。

教师惩戒权虽然被广泛应用于高校之中,但是不可避免的是仍然存在诸多障碍。例如社会成员、家长、学生对于教师惩戒权的误解,认为惩戒就是对于学生的体罚。该种情况导致教师惩戒权难以顺利的在高校中实施,也无法解决人们担心的问题。基于此,本文站在多维视角上,对教师惩戒权的实施提出几点建议,确保教师能够合理运用惩戒手段规避学生出现的不良行为,进而改善教育事业中存在的诸多问题。

参考文献:

[1] 王晓芳,覃欢,熊和妮.专业主义视角下中小学教师惩戒权的保障与规制[J].福建教育,2019(45):21-22.

[2] 陈国明,胡惠闲.多维视角下的教师教育与教师专业发展——第二届“上海课程圆桌论坛”纪要[J].全球教育展望,2014(5):125-128.

作者简介:

石泽翰(1968.05—),男,籍贯:湖南吉首市,学历:湖南吉首大学本科,学校:吉首大学师范学院讲师,研究方向:音乐教育。

(上接201页)

列表和序列的最大区别是:列表是最一般的集合,可以手地地包罗万象。而序列是用for循环实现的数列,是用通项公式表示的。

5 事件与脚本

脚本是指令、选择结构和循环结构的组合。任何对象都可以有自己对应的脚本,脚本又跟事件是一一对应的。

事件一般有3种:

5.1 某对象(特别是按钮对象)被点击,对应着“单击时”事件,会触发“单击时”脚本(以后简称单击脚本)执行一次。

5.2 某运动对象每移动屏幕1mm的距离,对应着“更新时”事件,就会触发“更新时”脚本(以后简称更新脚本)执行一次。

如果是下拉列表对象,“更新时”事件指的是选取一个选项。

5.3 对应于全局JavaScript脚本,不针对具体对象,或者说针对所有对象的。要用javascript来写,用得比较少。

任何一个对象都有脚本属性,在里面写了具体代码后,该对象就随时等待相应的事件来触发这些脚本。

为对象A添加了某脚本后,能触发脚本执行的事件肯定是针对对象A的。也就是说写脚本的对象,和事件针对的对象肯定是同一个。脚本的执行,可以是针对对象A的,也可以是针对其它对象的。

脚本的常见任务其实是固定的,一般而言有如下几种:

5.3.1 改变对象的属性:可通过函数setcaption(标题)、setcolor(颜色)、setlinestyle(线型)、setlinethickness(线宽)、setbackgroundcolor(背景色)、setpointsize(点径)、setpointstyle(点型)来直接修改属性值。或者通过setvalue函数来改变属性变量的值(前提是该属性已被设置成了变量,但setvalue(A,B)除外,这里是直接赋值坐标)。

5.3.2 启动/暂停动画: startanimation()

5.3.3 隐藏/显示:

5.3.4 改变环境状态:用得少

5.3.5 数据计算

这里有必要将脚本中最常见的赋值语句进行一下详细举例和介绍:

setvalue(a,a+1):将a加1后,赋给a,跟c语言里的c=c+1一样。

setvalue(A,0):将0点的坐标赋给A点,即让A到0点去
setvalue(A,(0,0)):跟上面的功能一样

setvalue(x(A),0):不会有结果,因为x(A)是一个函数(返回),不是变量,因此不能接收0,因此赋值是无效的,不报错,也不会有反应。

为一按钮添加点击脚本zoomin(1.5),即视角扩大1.5倍。这个可以认为事件是针对按钮的,而脚本是针对绘图区(这个对象)的。

6 总结

只有掌握了脚本编程,才能最大限度地发挥GGB的优势和能量。有编程功底的用户应该不难驾驭,没编程经验的朋友,在先掌握了一定编程基础知识后,相信也能很快地进入到丰富多彩的GGB世界。

参考文献:

[1] 曹盼.高中物理虚拟学习平台易化物理开发与应用研究[D].2019.

[2] 庞智.脚本编程在工控软件中的应用[J].机电信息,2019(24).

作者简介:

蔡渊(1984.10—),男,汉,湖北武汉人,硕士,武昌工学院信息工程学院讲师,软件工程。