《认识Scratch3.0》拓展资源

一、Scratch3.0动作模块功能详解

如果要制作游戏或者带有动画的程序，使用动作模块中的积木移动角色是最常见的操作。所谓角色移动，是指命令角色移动到舞台中的某一个具体的坐标点，或者旋转到一个特定的方向。

**1、绝对动作**

  后面会讲到，舞台是一个480\*360的矩形网格，其中线点是坐标（0,0）。在Scratch的动作模块中，共有四个绝对动作的积木。

这些积木能精确的把角色移动到舞台的某个具体位置。

  下面我们通过一个小例子来演示绝对动作。假设火箭角色Rocket要击中目标角色Target，目标角色的坐标位置是（200,150）。最简单的方法就是使用积木，

如图1所示，X坐标告诉角色在舞台水平移动的距离，Y坐标告诉角色垂直移动的距离。

图1：使用移动到x：y：积木吧角色移动到舞台的任何位置

  火箭角色Rocket在移动时有两个问题，首先它没有面向角色Target，而是向上方移动；其次是直接从点（0,0）移动到了点（200,150）。我们先来解决第二个问题。为了让火箭看起来是缓慢移动，而不是瞬间移动，我可以使用积木。这样就能根据时间的长短来看到火箭移动的轨迹。

**2、相对动作**

  如图2所示，和之前不同的是图中没有任何可以参考的坐标，那么火箭要如何才能击中目标呢？如果火箭Rocket会说话，它一定会说：“向前移动三步，向右转，再向前移动2步”。

图2：可以使用相对动作积木移动角色

 这些都是相对动作积木，图3展示了Scratch种各个方向对应的度数。


图3：Scratch中0是向上，90度是向右，180度是向下，-90度是向左

  使用积木就能将角色旋转到任意的一个角度。如果只是要面向上下左右，那么使用该积木种的下拉菜单即可快速选择。如果要更精确的方向，则需要在参数中制定。角度甚至可以是负数。

  现在你应该明白了Scratch中方向的概念，你可以自己试试各个动作积木是如何工作的。

**3、其他动作**

  我们继续来学最后四个动作积木：



  下面我们来创建一个简单的火箭撞击太阳的小程序来演示这4个动作积木的作用。

图4：火箭一直追着太阳移动

  这个程序包含了角色Sun和Spaceship，每个角色都有一段自己的脚本。当点击绿旗后，角色Sun就会随鼠标移动，而角色Spaceship则会追着Sun缓慢移动。

二、扩展阅读：Logo语言与scratch语言

已经不止一次有人提过这样的观点：“未来，每个人都要会一点编程”。不论是美国在线编程教育网站 code.org 发起的“编程一小时”活动，还是教育部召开新闻发布会，宣布在此次「新课标」改革中，正式将人工智能、物联网、大数据处理划入新课标的新闻，都有相同的中心思想：下一代教育里，编程可能跟英语一样重要。那么时至2020年，少儿编程又会有哪些新的动向呢？  其实少儿编程往前追溯，可以最早追溯到LOGO编程语言。

上世纪 90 年代，台湾推出繁体版的 LOGO 语言，称之为：【葛拉 /Gerlabau】，其后国内也推出了中文化的 LOGO 语言，分别是“易乐谷 /ELOGO”、“创新 CX-LOGO/ 微世界”和“Flash 版 Logo 语言”，发扬了中文编程的优势。

Papert 提出儿童编程最重要的，不是学习所谓的知识，而是通过编程来改变思维模式。“不是 learn to code，而code to learn”。 这是他创立 LOGO 语言的初衷。秉承着这样的理念，在 Seymour Papert 创造了 LOGO 语言的四十年后，2007 年，当初他在麻省理工带过的一个博士生 Mitch，创造了 Scratch。官方网址是麻省理工学院网站的一个分支，这个软件的开发团队称为“终身幼儿园团队”（Lifelong Kindergarten Group）。  后来最为世界所广泛熟知的莫过于Scratch编程软件了。

说到Scratch编程，Scratch有以下这些优点：入门简单，无关原有编程基础。通过拖拽块创建代码，而不是手动编写基于文本的代码，去除了枯燥的编程语法。内容丰富，针对有兴趣、喜爱绘画的学生，提供角色绘制设计功能。通过使用Scratch，让学生在动画、游戏设计过程中逐渐形成逻辑分析、独立思考创新的思维方式。相比其他编程软件，其更加直观，学生能比较容易地看到自己的劳动成果。  对于Scratch编程课程体系的考量，逻辑性强；独创模块/代码随时互换，。自然过渡，有效衔接，从入门模块编程到实战代码编程。通过游戏化互动式教学的形式，让学生与老师 零距离交流，从根源上减少孩子对编程的陌生感。创新的主题游戏，易于掌握的学习方法让孩子在轻松愉悦的学习环境中快速掌握编程的力量。让孩子将编程当做感知世界的工具。循序渐进地引领学生一层一层突破学习难关，再潜移默化中形成编程思维，实现独立编写代码的能力。