第 1 节 制作倒计时器

格致猫自从学会了 Scratch 编程，就着了迷。平常碰到什么问题，他都想：能不能用编程的方法解决呢？这不，观看跨年晚会时，他看到零点钟声响起前的倒计时，就在想：能不能利用 Scratch 编一个倒计时器小程序呢？

想到不如做到，说干就干！格致猫拿出一张纸就把自己的思路写了下来：

1. 先输入一个时间。
2. 把这个时间存在一个变量里。这个变量可以命名为“倒计时”。
3. 既然是倒计时器，那么每隔 1 秒，变量“倒计时”就得减少 1，直到变量“倒计时”的值变为 0 为止。

有了思路，我试着用积木把它们表示出来吧！

格致猫自言自语。

我可以用 

积木输入倒计时的时间数。新建一个变量“倒计时”，并通过 

积木给

赋值。每隔 1 秒，让变量“倒计时”的值减少 1，直到变量“倒计时”的值变为 0 为止（见图 1-1）。应该就这样，我先把所有积木组装起来，运行一下看一看效果如何。

说着，格致猫就把所有的积木组装了起来，程序如图 1-2 所示。

格致猫迫不及待地单击了绿旗，并按程序的提示输入 10，舞台上的倒计时变量真的从 10 开始倒计时了！看到程序运行无误，格致猫十分高兴，蹦蹦跳跳地去找小麦，分享自己的作品。

图 1-1

图 1-2

小麦看了格致猫的作品后，对格致猫的进步感到十分高兴。但小麦发现了一个问题。

格致猫，如果要想倒计时 2 分钟怎么办呢？

这……，是不是输入 120 就行了？因为 2 分钟等于 2×60=120 秒。

嗯，这倒也行，不过真正的倒计时器是“时”“分”“秒”分别进行倒计时的。每满 60 秒，“分”就减 1；每满 60 分，“时”就减 1。

这应该怎么编写程序啊！

别着急，我们一起思考一下！在询问环节，可以分别询问需要倒计时多少“分”、多少“秒”，把两次回答分别存储在变量“分”与变量“秒”里，像图 1-3 这样。再建立一个变量“总时长”，把“分”与“秒”的总秒数赋给它。假设倒计时 2 分 20 秒，那么总时长为 2×60＋20＝140 秒，积木组合为 

。

图 1-3

这不是还是以秒数倒计时吗？怎么把分钟的倒计时也体现出来呢？尤其是 2 分 20 秒中的 20 秒倒计时完成后，前面的 2 分钟怎么自动变成 1 分钟呢？

这就需要给这几个变量之间建立“联动机制”。“联动机制”是指一个变量发生变化，引发其他变量同时发生变化。在这个程序中，可以想办法给变量“总时长”与“分”“秒”之间建立这种“联动机制”。格致猫，你再想一想，刚才我们是怎么把变量“分”与“秒”的值转换成“总时长”的值的？

我们用的是“分”×60＋“秒”＝“总时长”的方法。

对，因此我们可以利用这个运算的逆运算把“总时长”的值再分别赋给“分”与“秒”。用“总时长”除以 60 的商是不是就是“分”？余数是不是就是“秒”？

嗯，是这么回事，但我还是不太明白它们之间是怎样互相联动的。

格致猫，我们一起分析一下。当“总时长”是 140 时，140 除以 60，商是不是 2？余数是不是 20？当“总时长”是 120 时，120 除以 60，商是不是 2？余数是不是 0？当“总时长”是 119 时，119 除以 60，商是不是 1？余数是不是 59？你看这时“分”是不是由“2”变为了“1”？

哇，还真是呢！

有了这个“联动机制”，我们再在你的程序上稍作修改，就变成了一个更正规的倒计时器了！

程序改动如图 1-4。

图 1-4

格致猫修改完程序后赶紧单击绿旗，看一看到底行不行。他输入了倒计时 2 分 20 秒，果然，不仅实现了“秒”的倒计时，也实现了“分”的倒计时。但是他发现了一个问题——倒计时器在最后一秒停止了。这是怎么回事？

小麦，这是怎么回事？为什么在最后一秒停住了？

小麦若有所思地看了看这个程序，发现……

格致猫，我们一起分析一下最后的“重复执行直到”语句，这个循环停住的条件是变量“总时长”的值为“0”。我们可以把“总时长”值为“1”时带入这个循环试一下。此时循环依次先给“分”赋值为“0”，再给“秒”赋值为“1”，再进行“总时长”减“1”的运算。格致猫，你注意一下，这时候“总时长”减完“1”之后，它的值就变为了“0”，因此循环就结束了，所以“秒”的值就停留在“1”上了。

那怎么才能让这个循环再进行一次呢？

小麦笑了笑，用鼓励的眼神看着格致猫。

聪明的格致猫，动动脑筋，你一定会想到答案的！

格致猫仔细地观察着这个程序，突然拍了一下脑袋。

我想到了，把这个循环的结束条件改为变量“总时长”的值小于“0”就行了！这样就会比刚才多一次循环，“秒”的值就会变成“0”了！

他们赶紧进行了尝试——成功！两人还进一步进行了修改，使这个倒计时器的程序变得更完善。程序如图 1-5 所示。

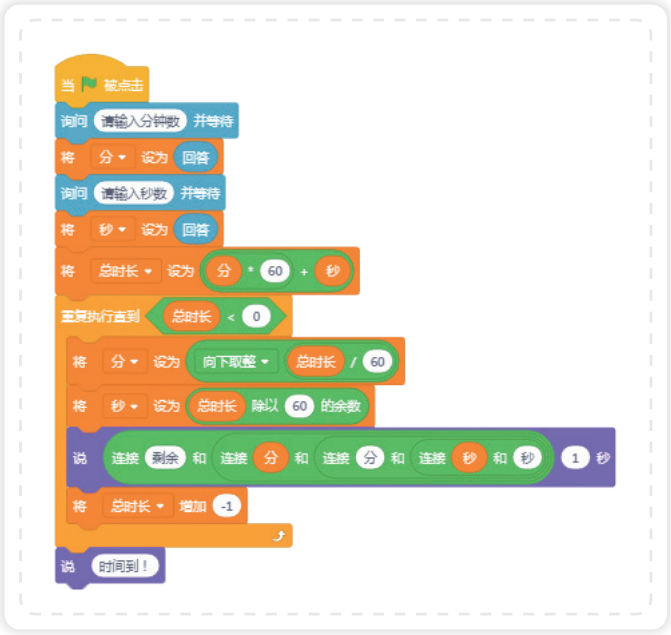

图 1-5

成长日记

我们学会了利用“重复执行直到”加条件控制循环的执行；利用时间单位之间的换算与取余的方式使“总时长”“分”“秒”三个变量之间建立“联动机制”。

第 2 节 画正多边形

假期，格致猫外出旅行时发现了一种以前没有见过的建筑——水塔！更令他惊讶的是，圆柱形的水塔竟然是用方砖垒成的。“方”怎么变成“圆”了呢？

带着这个疑问，一回到家，格致猫就去找小麦，想弄明白这到底是怎么回事。

小麦听了格致猫的描述后，思考了一会儿，然后打开了计算机中的 Scratch 软件。

Scratch 应该可以告诉我们原因。

什么！ Scratch 知道为什么？

嗯，当然了，不过你先别着急。在 Scratch 揭晓答案前，我先问你几个问题，你知道什么是正方形吗？

这还不简单！正方形有四个角，每个角都是 90° ，有四条边，四条边长度都一样。

对，那你能用 Scratch 画一个正方形吗？

我想想，应该很简单！可以让小猫先前进 100 步，然后右转 90° ，再前进 100 步，再右转 90° ，再前进 100 步，再右转 90° ，再前进 100 步，再右转 90° 。

格致猫边说边在纸上画，一个正方形就出现在了纸上。

用程序把这个过程表现出来就应该能画出正方形了。

说着，格致猫就在计算机上开始编写程序了。程序如图 2-1 所示。

格致猫一单击绿旗，一个正方形就出现在舞台上了。

这个程序中有四个 

积木组合，我们可以用一个“重复执行 4 次”来简化这个程序。像图 2-2 这样就好多了！

图 2-1

图 2-2

真不错！格致猫，那你能再画一个正三角形吗？

当然可以了，正三角形就是等边三角形，有三个角，每个角都是 60°，三条边相等。我把重复次数改为 3 次，把旋转角度改为 60° 就可以啦！

格致猫信心十足地单击了绿旗，但结果却大大出乎他的预料。显示结果如图 2-3 所示。

这是怎么回事呢？

格致猫，你仔细观察一下，你的小猫其实是向右转了多少度呢？

小麦启发着他。

你再看看图 2-4 所示正三角形应该是旋转多少度？

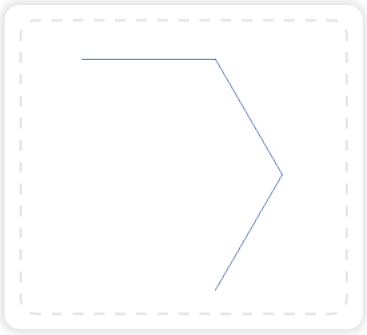

图 2-3

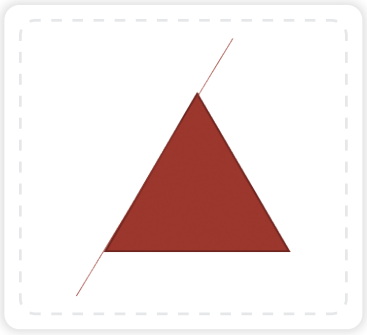

图 2-4

通过对比这两幅图，格致猫恍然大悟。

应该是旋转 120° 啊！我赶紧把程序修改一下。

修改后的程序如图 2-5 所示，这次一单击绿旗，一个“可爱”的正三角形就出现在舞台中央了。

小麦，正三角形也画出来了，可这和圆有什么关系呢？

别着急嘛，格致猫。你再观察一下图 2-6 这几个图形，用量角器（见图 2-7）分别量一下正五边形、正六边形需要旋转多少度？你能通过这几个图形把表 2-1 填一下吗？

图 2-5

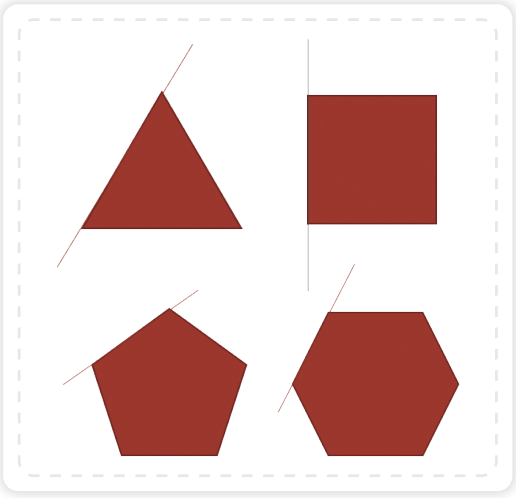

图 2-6

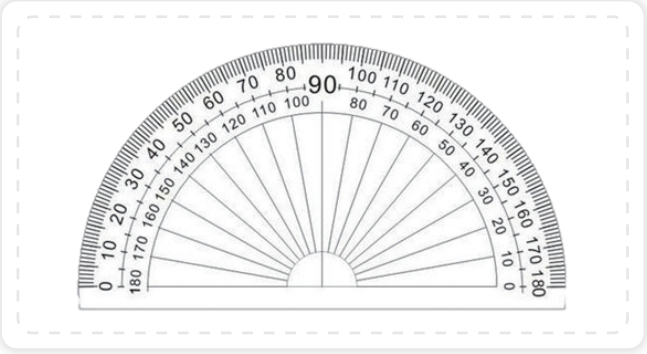

图 2-7

表 2-1　正多边形

正多边形	边数	需要旋转角度	边数 × 旋转角度
正四边形	4	90°	4 × 90°=360°
正三边形	3	120°	
正五边形	5		
正六边形	6		
正七边形	7		

格致猫一边用量角器测量多边形的度数一边回答。

画正五边形需要旋转的度数是 72°，画正六边形需要旋转的度数是 60°。它们之间有什么关系呢？小麦，我好像发现了点什么——正四边形的边数是 4，需要旋转的角度是 90°，它们的乘积是 360°；正三角形的边数是 3，需要旋转的度数是 120°，它们的乘积也是 360°；正五边形的边数是 5，需要旋转的度数是 72°，它们的乘积还是 360°。小麦，它们的乘积都是 360°。不用说，那画正六边形应该也是这样，正六边形的边数是 6，需要旋转的度数是 60°，6×60° =360°。

哇，太神奇了！根据前面边数与需要旋转的度数之间的关系，画正七边形时需要旋转的度数应该是 360° /7。原来正多边形的边数与需要旋转的角度的乘积都是 360° 啊！

格致猫，你真棒！这么短的时间就发现了正多边形边数与旋转度数之间的关系。如果想画一个任意边数的正多边形，假设是一个 n 边形，那需要旋转的度数就为 360° /n。知道了这个关系是不是就能画出更多边数的正多边形了呢？格致猫，你试着根据这个规律画一下正七边形、正八边形、正九边形、正十边形。

好嘞！

说干就干，不一会儿的工夫格致猫就把这四个图形画出来了。程序及运行结果如图 2-8 ~ 图 2-11 所示。

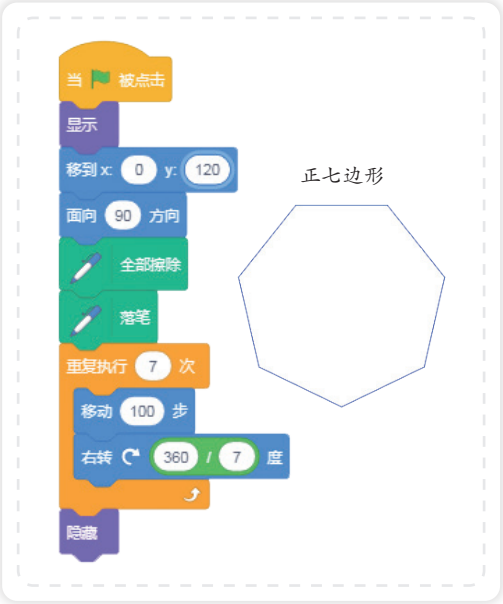

图 2-8

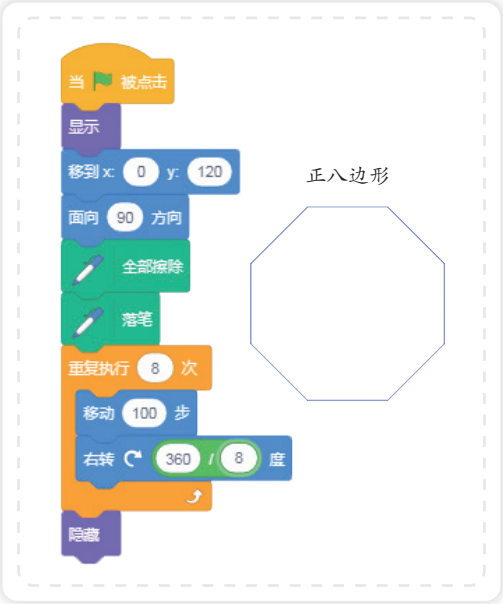

图 2-9

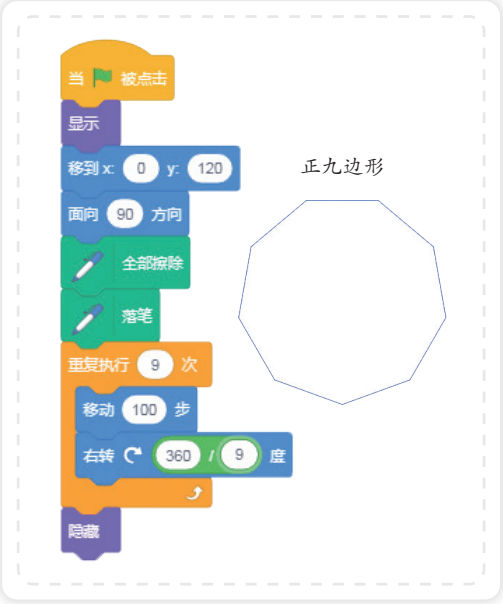

图 2-10

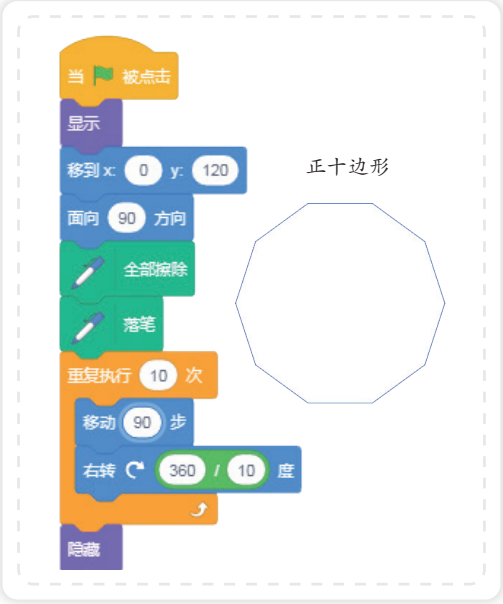

图 2-11

小麦，在编程时，为了不让图形画到舞台之外，我对程序做了一些调整，给小猫定了初始坐标，并随着边数的增加，减少了小猫移动的步数。

格致猫，你考虑问题真严谨，给你点个赞！你再观察一下画好的几幅图，能发现随着边数的增加，正多边形有什么变化吗？

好像是越来越像一个圆了！

对了，那你现在再画个正 100 边形，看一看结果是什么样的？

好，我试试！只要修改两个参数就可以了，so easy（太容易了）！

程序及运行结果如图 2-12 所示。

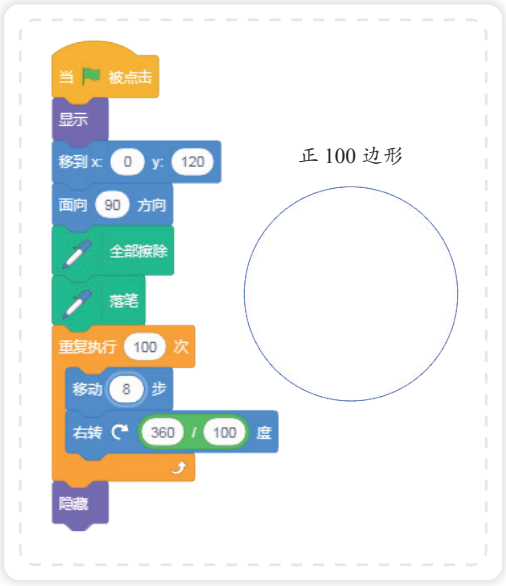

图 2-12

小麦，你快看，它是一个圆！

这下你该明白“方”是怎么变成“圆”的了吧！我国魏晋时期的数学家刘徽早在 1800 多年前就发现了这个规律，并命名为“割圆术”。通过“割圆术”，刘徽计算到了正 192 边形、正 3072 边形，得到了圆周率为 3.1415 和 3.1416 这两个近似值。“割圆术”是我国古代极限思想的体现。

明白了，明白了！数学与编程的结合真是太神奇了！

成长日记

我们学会了利用“重复执行（）次”积木画正多边形；了解了正多边形边数与旋转角度之间的关系；对“割圆术”有了初步的认识。