**程序设计简单教程**

**第一课《打鸟游戏》**

今天我们就来谈谈儿童编程方面的话题。儿童编程主要是指面向高中以下的孩子所设计的编程学习类产品。Scratch2.0是由美国麻省理工学院“终身幼儿园研究组”（MIT Lifelong Kindergarten Group）专为少年儿童设计的、充满趣味性的计算机编程工具。

下面我们就共同研究使用Scratch2.0来设计《打鸟游戏》这个作品。

《打鸟游戏》程序设计实践操作要求：

1. 教师要对scratch2.0有一定的了解和实际使用或教学经验。

2. 教师要对程序设计中的基本语句、语法要有所认知。

3. 教师要能正确理解积木模块指令所体现出来的程序设计思想。

**一、角色的上传、设置**

教师们在熟练掌握scratch2.0基本操作的基础上，导入事先准备好的背景图和6个角色图。如下图：



【提示】

1.鸟的造型是2个 : 鸟1.png 鸟2.png

2.猎手的造型是3个:   

3.我们导入角色之后要看一看角色的中心点是不是需要调整一下。

**方法**：点击相应角色，单击【造型】标签，点击造型中心.png设置造型中心。

4.要把每一个角色放到舞台的相应位置，角色不要重叠，并且要调整好角色的大小，使我们的整体布局更美观，效果更逼真。

**二、角色的脚本设计(6个角色)**

**脚本（Scripts）**是用来描述程序逻辑结构并控制角色行为的模块集合。

**（一）太阳脚本**



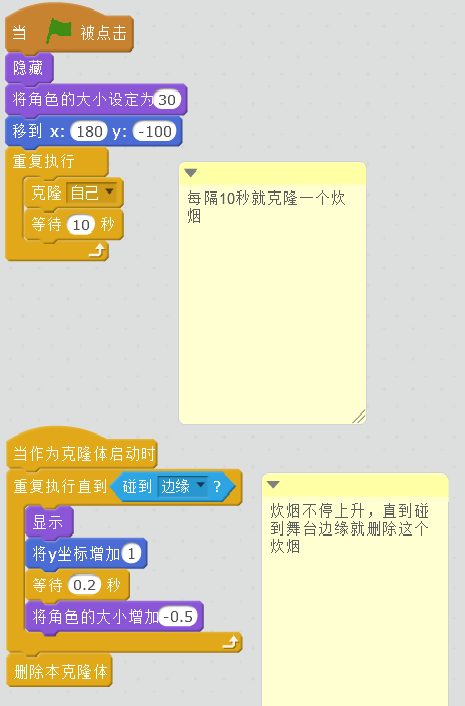
这个脚本块是实现太阳从从升起到落下沿着一个弧线在移动。

【提示】

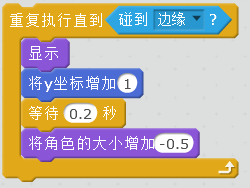
这个脚本块有13个移动指令移动指令.png，且每一个移动指令中x和y的坐标信息是变化的，这是因为太阳的移动位置是变化的。为了准确实现太阳的移动轨迹，我们教师要把太阳移动的整个过程（13个位置）在舞台上演示一遍，每移动一个位置，移动指令就记录下了太阳当前位置信息，我们就可以直接拖曳移动指令.png到脚本区，记着是移动一次，拖曳一次。

显示.png和隐藏.png是实现太阳出现升起和落下消失的功能。

**（二）炊烟脚本**



我重点讲一讲下面的脚本块：



这是一个条件式的直到型循环，直到碰到边缘就退出循环，也就是说炊烟在没碰到舞台边缘之前，一直在显示并向上移动，调整炊烟角色的大小（即越往上飘动越小）。

**克隆的独特之处：**

克隆的3个指令克隆.png  删除克隆.png 是scratch2.0的独特之处，它可以实现一个角色多次复制，并能达到同等行为效果。

以上2个角色脚本的设计基本上可以实现单一角色行为效果，我们教师可以单击绿旗.png看一看动画效果。

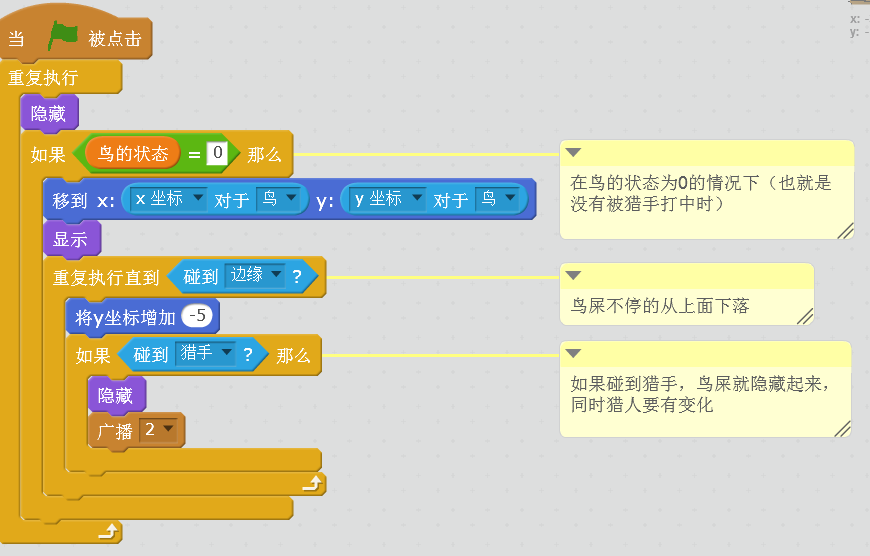
下面我们来共同研究角色和角色间协同实现行为效果的程序设计方法。

**（三）鸟的脚本** 

**并发线程的使用：**

鸟这个角色共有2个脚本块（2个当点击.png），如果我们设定了两个程序块，那么他们会同时执行，这叫“并发线程”或叫“并发执行”

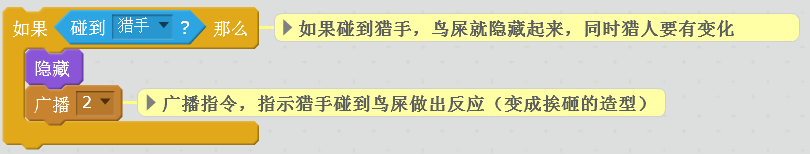
**（四）鸟屎脚本**



**重点讲**：“鸟的状态”这个变量的作用

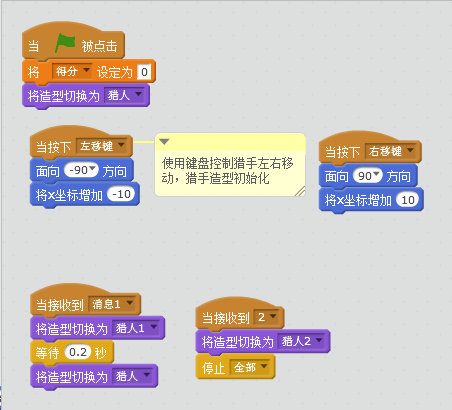
在鸟的脚本块有一个很具讲解的地方就是标志性变量“鸟的状态”的建立，这个变量起到在鸟没有被打中前值为0，只有“鸟的状态”为0的情况下，鸟屎这个角色才会移到鸟的坐标位置上去（也就是执行移到.png）然后不停跟随鸟的位置下落鸟屎，当鸟被打中时（值为1），鸟屎就不在鸟的位置显示下落，而是处于隐藏状态。

**重点讲**：协作和同步的实现

在上面的脚本中，当鸟屎下落砸中猎人，那么猎人会有什么反应，并且如何实现呢？我们就需要使用事件指令组中的广播1.png 和广播2.png来实现。这两个指令可以实现多个角色之间协作，同步不同角色的行为。 

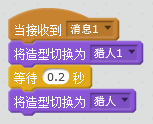
通过广播1.png和广播2.png来实现鸟屎角色和猎手角色的互动效果，产生相应的角色行为。

**（五）猎手脚本**



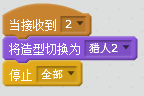
**猎手和子弹子弹造型.png互动同步**：

下面的脚本块实现与子弹脚本中指令广播4.png形成互动同步效果，子弹角色和猎手角色产生相应行为（即发射子弹的同时，猎手变成第二个造型）



**猎手和鸟屎鸟屎造型.png互动同步**：

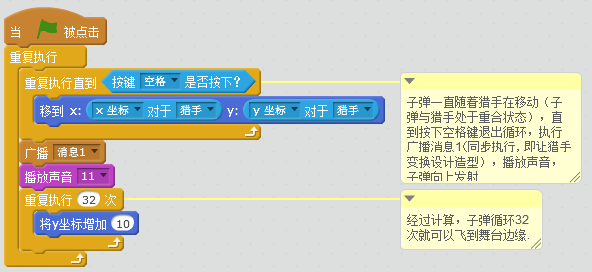
下面的脚本块实现与鸟屎脚本中指令广播1.png形成互动同步效果，鸟屎角色和猎手角色产生相应行为（即鸟屎下落砸到猎手的同时，猎手变成第3个造型）

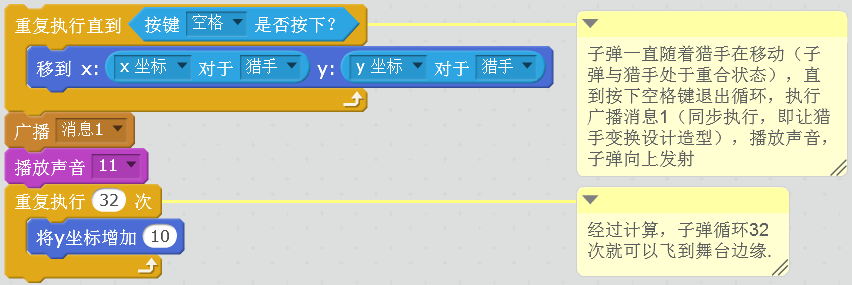


**（六）子弹脚本**

**重点讲**：**协作和同步的实现**

在上面的脚本中，当我们按下空格键后，执行广播4.png，那么猎手会有什么反应，如何实现呢？我们就需要使用事件指令组中的广播4.png 和来实现。这两个指令可以实现(子弹和猎手)多个角色之间协作，同步不同角色的行为。

****

****

通过广播4.png和来实现子弹角色和猎手角色的互动效果，产生相应的角色行为。

至此，我们盼望已久的《打鸟游戏》这个作品设计完成，点击当点击.png试一试，看看效果如何？

Scratch创始人，Mitchel Resnick（瑞斯尼克）教授说“计算机不应该只用来提供信息，而是应该让孩子们展现自己，并与其中最重要的思想碰撞”。身为信息技术老师，程序设计教学的要务应该是让学生感受到学信息技术是有趣的，并从中体会信息技术的作用和不断发展的历程。无论是哪个版本的程序设计语言，我们都要根据学生当前对知识的认知水平，选择恰当的内容进行教学设计和教学实践，学生一定能学得很精彩。

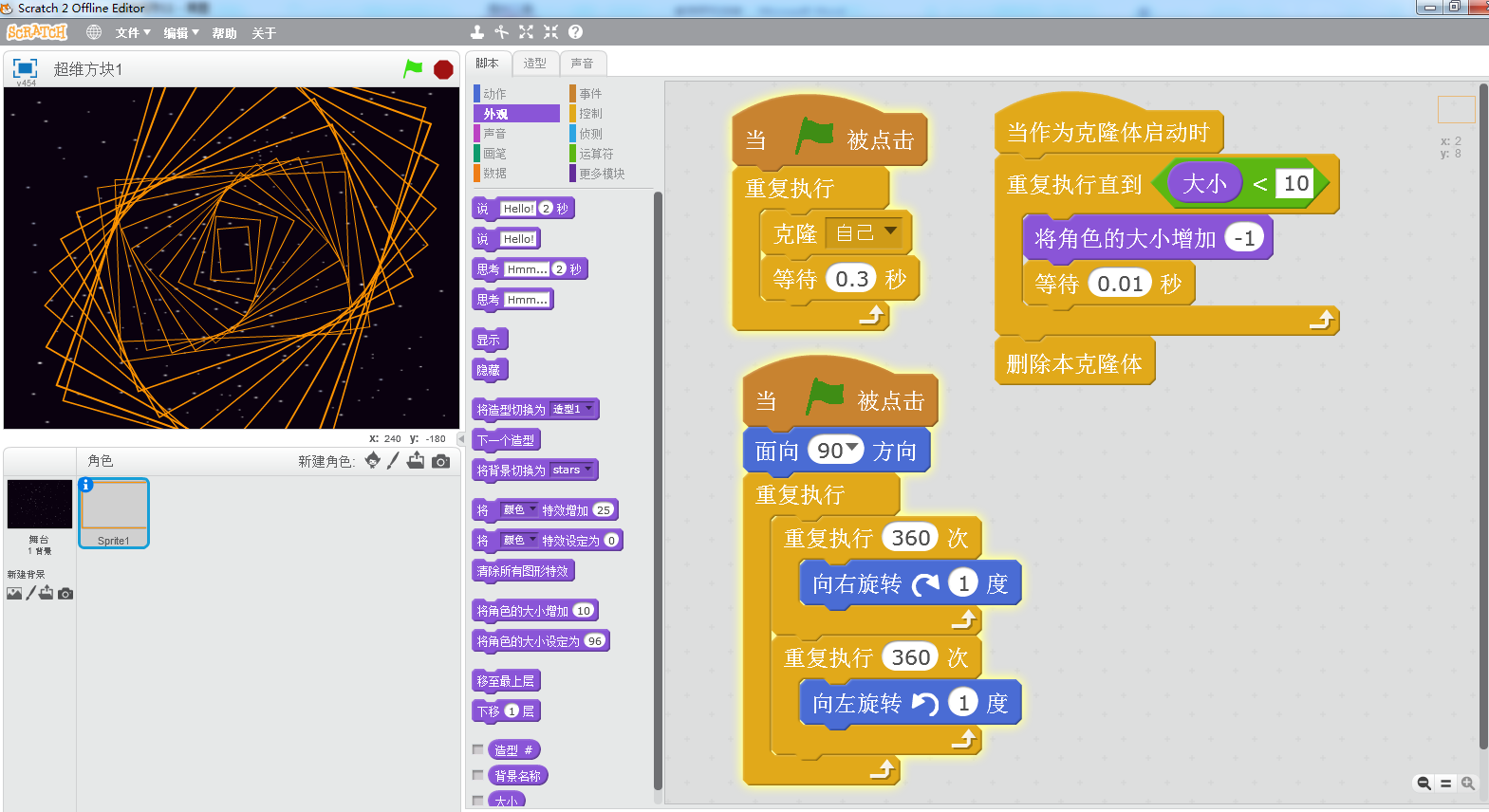
总之，辩证看，务实办！谢谢大家！

**第二课“克隆体”**

Scratch 最突出的特点便是提供给用户一套由积木系统组成的编程指令，开启了积木式编程学习模式的先河。

**1、使用对比法，阐述“克隆体”**

克隆体是Scratch2.0的独有之处。通过欣赏事先准备好的作品，教师随即说出“克隆体”的作用：我们在设计一个作品时，你需要多个相同的角色，其动作脚本、造型完全相同，不同的仅仅是这些角色的位置。在scratch1.4或之前的版本中，该如何实现呢？只能先将角色的动作脚本、造型什么的先设计好，再一个个去复制，需要多少个就复制多少个，这些角色都密密麻麻的排在角色列表上。这还不算麻烦，如果好不容易复制好了，结果却发现脚本有bug需要修改或者想修改一下其功能就悲催了，那只能一个个去修改脚本，或者先将复制好的角色删除等修改好了之后再重新一个个的复制。有了Scratch 2.0，麻烦就大大减轻，只需要使用克隆功能就可以一下子搞定。这样教师的认知积极性就调动起来了。



超维方块主界面

**2、采用任务驱动创设情境**

首先先播放作品，让老师们欣赏一下，提升教师对克隆体的兴趣，如果我把克隆体脚本摘了，就没有这种效果了。克隆其实就是复制，克隆体有着与本体完全一样的属性，但它不是新的角色，而仅仅是一个克隆体而已。让教师们感受到采用“克隆体”技术能达到事半功倍的效果，进而能使用“克隆体”创作出更加完美的作品。

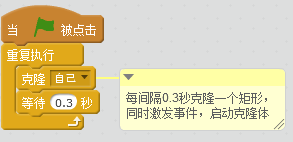
Scratch 2.0实现克隆功能用到了三个block：

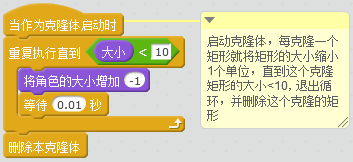
（1）指令模块：创建克隆体（ Create Clone） 克隆体1.png

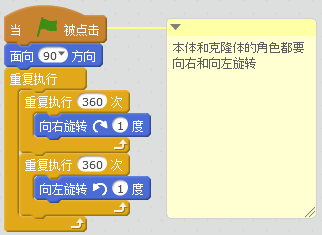
（2）事件模块：当作为克隆体启动时（When I Start As a Clone)克隆体2.png

（3）指令模块：删除克隆体 （Delete this clone)克隆体3.png

当一个克隆体被创建时，它就会自动启动，同时激发事件：当作为克隆体启动时(When I Start As a Clone)。当克隆体运行结束时，可以使用指令“删除克隆体(Delete this clone)”将自身删除。这是克隆体指令的整个执行过程。通过作品的实际演示教学，让教师们通过实践进而再根据研究性学习的特点，把教学过程逐步演变为独立设计和创作作品的过程。







教师在讲解上述3个脚本块时，要注重脚本间程序的执行顺序，内在逻辑联系和层次叠加关系。教师的讲解绝不是简单的知识传授和操作步骤的指导，而是要注重方法的指导和能力的培养。

**3、采取递进教学法，提高教师认知度**

上面提到的克隆功能的用法仅仅是其中的一个而已，更多的用法期待着你自己去发现。当然可以递归创建克隆体，为了避免无限制创建克隆体将系统资源耗尽，目前限制只能克隆300次。

充实资源。通过大量教学范例、教学课件、练习素材等教学资源，来搭建资源库。展示【弹弹球】【泡泡机】【旋转的花】【切水果】4个作品，同时把脚本文件发给老师们，不仅能自己制作，还可以修改其中的脚本，产生不同效果的艺术作品，这样能更进一步体会到“克隆体”在实际作品中的应用是十分广泛的。

总之，教学相长，我们选择恰当的教学内容进行教学设计和教学实践，采取任何方法，都是让大家使用、感受到Scratch2.0的魅力，带给我们无比广阔的设计空间。

**第三课 飞起来**

目的：

让某个角色飞起来。计划是让巫师从舞台左下角飞到舞台右上角，达到右上角以后，回到左下角原点。后来有添加了螃蟹、蝙蝠、鬼怪。打算巫师是对角飞，螃蟹是在下方左右翻滚移动，鬼怪在中间左右平飞，蝙蝠在舞台中间自下而上飞，同时蝙蝠会逐渐变大。

先完成巫师的对角飞行。介绍舞台的X方向和Y反向。舞台的正中央是零点。如果要想让某个角色从左到右移动，那么横向的X是增加还是减少？答案是X++，比如一次加1或者一次加4，加的越多，就飞得越快。然后巫师还需要往上飞，因此Y方向要向上移动，即Y++。如果要往左飞，则X–，往下飞则Y–。舞台最右边是X=240，最左边是X=-240。舞台最上面是Y=180, 最下面是Y=-180。负数就是比零还小的数字。

有了这个认识以后，我们进入脚本区域。介绍第一个分类是“事件”。在事件中，我们添加了“当小绿旗子被点击”，就是开始按钮，然后执行一些操作。

1）因为默认的巫师图片比较大，所以每次开始时，我们都先设定巫师的大小为60。这样怎么操作？进入“外观”，里面有一个控件“将角色大小设定为XX”，XX空格中可以录入数字。将这个控件拉过来，把开始小旗子磁贴粘贴在一起。这和乐高的WEDO类似。就是几个控件依次放在一起。

2）要实现巫师往右飞，我们使用了“运动”中的控件“将X坐标增加XX”，我们设定了增加4.

3）要实现巫师往上飞，我们使用了“运动”中的控件“将Y坐标增加XX”，我们设定了增加3.

这时候已经可以开始运行了，但是每次点击运行，巫师只会飞一点点，需要不停点击，才能往右上角飞。

那如何让巫师连续的飞的？答案是使用循环。

4）循环属于“控制”，在“控制”分类中，选择重复执行这个框，将X，Y的两个控件都包括在内。

因为是重复无数次执行，因此我们实现了巫师对角飞行。但是问题又来了。巫师碰到右上角以后，就飞出去不动了，因为到底了。如何解决这个问题？

这时候我们讨论，每飞一次，我们就检查一下，如果X坐标大于240，我们就然巫师重新回到左下角。因此我们增加了一个判断。

5）判断还是属于“控制”，在控制中增加“如何…..那么”的判断。

6）判断有条件，条件属于“运算”，我们到“运算”中，拖一个“某某>某某”的判断控件到这个“如果…那么”上。

然后获取巫师的当前位置，这可以在“运动”中找到“X坐标”。判断条件是：

X坐标> 240.

7）如果条件满足，我们就要让巫师回到原点，即X坐标等于-240，Y坐标等于-180。

运行，巫师就开始对角飞行，飞到右上角以后，又重新回到坐下角，不停的重复。

后来我们按照相同的方法，实现了旁边从左到右，里面增加了一个运动动作“右转XX度”。

鬼怪的动作我们选择了，从左到右，但更简单，没有使用判断，只是往右执行120次，X轴每次增加4，执行完毕以后，继续执行120次，但是每次增加-4，这样就实现了不停来回的移动。

对于蝙蝠，我们加上了大小变换。从下往上移动，还是使用判断，到达顶部以后回到底部。蝙蝠往上移动的时候，我们然大小逐渐变大，然后回到原点的时候，重新设置为开始时的大小。

我们是使用浏览器，在MIT网站上直接使用Scratch的，最后我们将项目保存在MIT网站中，同时保存一份到本地计算机桌面上。

小孩说要设计一个能发射炮弹的游戏。那我们下次，打算让一个飞机从左往右飞，然后飞行过程中能够向下扔炸弹。飞机是一个角色，炸弹是一个角色，关键是计算位置，要实现炸弹是从飞机上扔下来的。

要继续学习，就得好好吃饭，先做好作业。果然，有了这个期待，今天的娃还是很听话的。

总结今天学习的内容：

1. 舞台：会自己添加舞台背景了。

2. 角色：会自己添加角色了。

3. 事件：开始

4. 外观

a. 将角色大小设定为60

b. 将角色大小增加2

5. 运动

a. 将X坐标增加4

b. 将Y坐标增加3

c. 右转4度

d. X坐标：读取X坐标的值

e. 移动到X=-240, Y=-180:回到左下角。

6. 控制

a. 重复执行：无限制，永远重复

b. 重复执行140次：每次X+2

c. 如果？？那么？？：如果X>240，就回到原位。

对于这些分类，第一次还是记不住的，后面希望每周花半天时间，通过练习，记忆，能够自行完成这些作业。其他时间，可以讲解各种设计、控件和基础知识。

找一个球、水果或物品，从屏幕顶部位置按照随机的方向落下；屏幕下方设置一块板，板可以跟随鼠标左右移动；如果球落下时碰到板，球即往上按照随机的角度反弹；球触碰到边框以后，折返，继续落下；玩家继续移动鼠标接球。如果没有接住，球则掉入下面的火坑，游戏结束。

https://scratch.mit.edu/projects/196363356/ 最终效果可以参考这个网址。

实现方式

根据上次学习的技能，添加了一个球的角色，但后来并没有选择球，而是选择了橙子，然后选择一个按钮板，后选择了一根红线，球碰到红线游戏即结束。后来我在网上找了一个熊熊烈火的图，在红线下方加了烈火图。意思是球碰到红线了，掉入火坑中烧掉了，游戏结束了。

第一步：红线和烈火是固定的，最为简单，什么脚本都不需要。

第二步：设置角色按钮板的程序脚本：

1. 设置开始事件

2. 直接进入无限制循环

3. “移动”到“鼠标”，这一个控件即可实现角色跟随鼠标移动；

4. 设置按钮板的Y轴固定位-95，即实现只能左右移动，不能上下移动

5. 运行测试。鼠标左右移动，按钮板也跟着左右移动

“移动到”是一个很重要的动作，可以移动到随机位置，即打地鼠一样随机出现，也可以跟随鼠标移动，还可以移动到其他角色的位置上。

第三步：设置角色橘子的脚本，这是最复杂的部分。当然可以先从简单开始，逐步加入复杂一些的功能。

1. 设置开始事件

2. 位置设置为舞台上放，所以选择了“移动到X：随机数，Y：180”，横轴选择从-240到240随机选择一个数。

3. “面向XXX方向”：橘子开始运动前，需要确定运行的方向，可以在一定范围内让系统随机选择一个方向。XXX这里可以让系统随机选择一个某范围内的数字。

因此，我们使用了运算中的“在XX到XXX中随机选择一个数”，我们设置了135到255。

初始化就做好了，下面开始循环往前移动。

4. 然后进入无限制循环

5. 首先我们需要让橘子往前走，因此需要使用“移动5步”。通过修改每次移动的步数，可以实现橘子移动的速度，增加游戏的难度。经过测试，速度为10比较合适，速度为15-20，难度就大一些了。

6. 移动之前，我们就可以判断是否碰到了按钮板角色。因此我们需要增加一个“如果…那么…”的判断。 如果“碰到XX”,那么，就改变方向，这时候方向要朝上，所以选择在-65到65度角度上随机选择一个方向；然后继续往前移动。

7. 球在往上运动时，会碰到舞台边缘，执行“碰到边缘就反弹”的动作。

8. 如果不接住橘子，橘子将继续往下，直到碰到红线角色，游戏停止。所以这里需要增加一个判断，如果“碰到红线角色”，则说话“没接住，游戏结束啦！！”，这个提示话语停留1秒钟。

9. 移动50步，这样橘子，就从掉入下方火焰中了。

增加声音

后来我们增加了声音：

1. 在橘子碰到按钮板的时候，播放声音POP；

2. 在碰到红线游戏结束时，播放咚咚咚，可以选择演奏的乐器-钢琴、小提琴、鼓等，然后弹奏何类鼓声，持续多少节拍，中间可以添加几节拍的休止符，实现咚、咚、咚的效果。

后来我们增加了沙漠的背景图

后续改进

1. 增加计分器，没接住一次球，就增加1分，这样可以进行比赛了；

2. 设置游戏难易程度：大人、中学生、小学生、幼儿园小朋友（高中低），选择高中低以后，设置橘子移动的速度。

完整的图。

演示地址：

https://scratch.mit.edu/projects/196363356/

总结：

这些学习了不少内容：

1. “移动到-鼠标、或者某个角色”上

2. 随机产生某个范围内的数字；

3. 设定角色运动的方向；

4. 碰到边缘就反弹

5. 播放声音POP

6. 设置演奏器材为XXX

小朋友这周的表现还算不错。周末多数时间都由我陪(bi)着(zhe)做课本上的数学题，后来还学了英语。任重道远啊，语数外都还得加强，还远不到自己就能取得好成绩的阶段。

上周说好这周要做一个发射炮弹的游戏。最后在周日晚上八点学习完英语以后才开始。

**第四课 宇宙飞船**

目的

目的：设置一艘宇宙飞船或其他角色，实现上下左右移动。每按一次空格键即发射一枚炮弹，同时播放声音，炮弹往屏幕上方飞，直到飞出屏幕。发射子弹、炸弹是很常见的游戏功能。

实现方式

宇宙飞船角色的上下左右移动是比较简单的。我们在接球运动中已经学过了。只需要不停循环，检测是否有按键按下。如果左箭头按下，X轴就减少10；如果右箭头按下，X轴就增加10；如果上箭头按下，Y轴就增加10；如果下箭头按下，Y轴就减少10。对于这个步骤，小孩表示理解。

这是我们第一次使用发射炮弹的功能。首先要找到一个炮弹的图，可以选择球类或者其他物品， 最后我们选择了一个闪电，但要将闪电转一下，让尖尖的头朝上。

Scratch提供了克隆角色的功能，这是一个强大的功能，就是克隆出另外一个副本出来。克隆体启动时，可以执行单独的脚本，即相当于一个子函数，控制这个炮弹克隆体。每按一次按钮，炮弹就克隆复制自己。然后炮弹往屏幕上放飞行就比较简单了，就是向上移动一定的位置，或者Y轴增加一定的距离。不停循环即可往上飞行。

逐步改进

制作程序的过程，是一个不断测试改进的过程。我们在制作过程中，发现了一些问题，并且逐步解决了。

1. 宇宙飞船的范围确定。 如果不设置范围，那么左右上下会移出屏幕。因此我们后来增加了一个判断，即只有在X轴大于-220时，才继续往左移动，否则就不往左移动了。Y轴类似。这样通过四个箭头的移动，保证宇宙飞船一直都不会超出屏幕，同时上方保留一定的空间。

2. 克隆体一直往上飞行，到顶部以后，炸弹会停留在那边，留下一个尾巴。我们增加了判断。如果碰到边缘，就“删除克隆体”，这样克隆体碰到边缘后就消失了。

3. 播放声音：每次发射炸弹后，系统会播放声音。播放声音不能直接在循环中做，因为那样会做完以后才能继续发射下一个炮弹，影响效果。因此广播一个消息，然后在消息中播放音乐。最后我们选择了一个敲打低音鼓的声音，这样发射炸弹以后，就发出类似爆炸的声音了。

制作步骤

第一步：飞船角色上下左右移动。

1. 加入宇宙飞船角色；选择星空背景。

2. 造型中，将飞船大小调小一点；设置飞船的中央点为中心。

3. 设置开始事件

4. 直接重复执行，无限循环

5. 加判断，然后使用侦测 “按键 左移键是否按下”，那么X坐标增加-11。即飞船向左边移动。

6. 后来我们增加了一个判断。如果X坐标大于-220，才将X坐标增加-11，否则就不增加。即到达左边框以后，飞船不会继续往左边移动，保持在屏幕范围内。

7. 向右，向下和向上，方法类似。

游戏做出来后，小小朋友要来玩。结果她一直都按住箭头，飞船会超出屏幕。为了解决这个问题，我们就增加了边框的判断。

–飞船：上下左右移动–

第二步：克隆炸弹

8. 加入闪电角色。

9. 造型中，将闪电调小一点；设置飞船的中央点为中心。然后转动了一下。

10. 设置开始事件

11. 直接重复执行，无限循环

12. 每个循环，都将子弹 “移动”到“飞船”。因为飞船是移动的，所以发射炸弹时，要从飞船位置上发射出来。

13. 增加如果那么的判断，检测空格按键是否被按下。

14. 如果空格键被按下，那么就克隆自己。（这里可以克隆其他角色-例如飞船）。

15. 然后设置：控制>克隆体启动时。执行这个克隆体的脚本。

a. 首先，外观>隐藏自己。 （开始的时候，我们是没有增加隐藏、显示的步骤的。这样炸弹就在飞船上面能够看见。）

b. 然后选择：运动>面向0°方向，即朝上。这里可以改变方向，例如向右发射炸弹等。（可以选择不同的方向，测试效果）

c. 然后外观>显示自己。

d. 重复执行；移动8步。炸弹即往上飞行。（可以改变移动的速度。）

e. 增加判断，如果碰到边缘，那么，删除克隆体。（如果不删除，就留一个尾巴在屏幕上方。）

16. 增加声音：

a. 克隆自己以后，广播消息1；

b. 当收到消息1时

1）演奏乐器设置为18-鼓

2）弹奏鼓声2, 长度为0.25节拍。（要使用广播，否则飞船发射炸弹就要等待声音播放，导致不连续。）

-炸弹：克隆自己、克隆后移动、广播播放声音。

后续改进

1. 打算在上方随机出现另外一个角色，例如鬼怪，鬼怪也可以克隆自己。鬼怪按照随机的方向自上往下运动。（角色随机从上往下运动，在上一球类发射的课程中已经学过）。如果鬼怪碰到子弹，鬼怪则消失，并且播放打中的音乐，最后播放一个炸弹爆炸的效果。

2. 增加计分器，每消灭一个鬼怪，就增加10分，这样可以进行游戏了。

分享地址：https://scratch.mit.edu/projects/196376484/

浏览器中需要Flash播放器。

这个游戏，并不是由小孩独自完成。是在家长指导下，由小孩点击操作完成。目的依然是培养兴趣，了解逻辑过程，了解要实现什么功能，使用什么控件，对控件如何操作。

**第五课 听话的小猫**

目标

理解坐标和角色的移动。通过键盘或鼠标控制小猫的移动，按左箭头就往左走，按右箭头往右走，按上箭头往上走，按下箭头往下走。

实现基本的移动功能以后，进一步增加对边框的限制，使得小猫不能超出边框。

适用对象：本练习适合一年级初学者使用。在老师指导下，学员基本能够独立完成。对坐标的判断，难度较大一些，需要理解坐标的大小比较。

上机操作题

（三个角色，对应三种方法。）

练习题1：用键盘控制角色的移动，实现：

按上箭头↑，角色往上移动；

按下箭头↓，角色往下移动；

按左箭头←，角色往左移动；

按右箭头→，角色往右移动。

习题2：练习1中，当用键盘将角色移动的舞台边缘时，角色将被遮挡一部分。要求实现：角色不被遮挡，即移动到接近边缘时，如果继续按键，则角色不再往舞台外移动。如此，角色只能在一定的范围内移动，不能超出边框。

按上箭头↑，角色往上移动；

按下箭头↓，角色往下移动；

按左箭头←，角色往左移动；

按右箭头→，角色往右移动；

控制小猫不能越界，X，Y轴需要增加什么限制？

习题3：练习1和练习2使用键盘作为输入，实现对角色的移动。本练习实现用鼠标移动角色，即角色跟随鼠标不停移动。

练习关键知识点

1. 事件：开始。可以设置一个或多个开始事件。

2. 事件：当按下？？键。检测某个安检是否按下，使用这个指令，可以实现角色跟随上下左右箭头的移动。

3. 外观：将角色大小设置为？？如果角色过大，可以减少角色的尺寸，通常在启动时设置角色的大小，50即为角色原大小的一半。

4. 运动：X坐标增加？？很常用的运动方式，X坐标增加则往右移动，增加为负数则往左移动。注意系统没有X坐标减小的指令，因此需要转换成增加负数的指令。

5. 运动：Y坐标增加？？同X坐标增加。

6. 运动：移到？？？：移动到鼠标、某个角色上，例如发射子弹时，子弹从坦克、飞机等身上发射，则先将子弹移动到飞机上。移动到鼠标，则跟随鼠标移动。

7. 运动：移到X:？？ Y：？？开始时，使用这个指令，可以将角色移动到零点或者某一个角落。

8. 运动：X坐标（获取X坐标值）。在如果…那么…的判断中，可以判断X坐标是否在一定的范围内。例如X坐标<-220（左边），则X坐标不允许再次减小，即不能继续往左移动。例如X坐标>220（右边），则X坐标不允许再次增加，即不能继续往右移动。使用时，需要考虑使用大于还是小于的判断符。

9. 运动：Y坐标（获取Y坐标值）

10. 控制：如果…那么…

11. 侦测：按键？？是否按下。可以在重复执行中检测按键是否被按下。这种方式也可以实现角色跟随上下左右箭头的移动。

12. 运算：？？> ??

13. 运算：？？< ??

14. 控制：重复执行

练习讲解要点

练习一：

方法一：使用按键按下的事件，如果收到上下左右的箭头按键，则X坐标或Y坐标增加10或-10.

如果不增加X或Y坐标的大小判断，角色将会移动舞台。

蝙蝠的运动：使用了当按下???键的事件，因此放4个事件。如果不判断出界，那就很简单了。点击开始时，增加造型切换的指令。为了控制蝙蝠翅膀拍打快慢，造型切换的循环中增加等待时间。

方法二：开始后直接进入循环，在循环中如果…那么…判断，判断条件中检测上下左右箭头被按下，如果按下，则X坐标或Y坐标增加10或-10。效果和方法一类似，但比方法二要复杂一些。

地球移动：大循环中，如果那么中判断是否有按键按下。

练习二：角色跟随上下左右箭头移动时，如果不增加X或Y坐标的大小判断，角色将会移动舞台。因此本练习是添加X坐标和Y坐标的大小判断。

如果X坐标小于220，那么X坐标增加10.即如果X坐标小于220，允许继续右移。否则，不允许右移。实现不超出右边框的功能。

如果X坐标大于-220，那么X坐标增加-10.即如果X坐标小于-220，允许继续左移。否则，不允许左移。实现不超出左边框的功能。

上下通过Y坐标大小控制，注意正负数和大小判断即可。

练习三：角色跟随鼠标移动。开始后进入循环，然后每个循环都移动到鼠标即可。

如果要增加判断，例如到某一个边界不允许继续移动，则增加练习一和练习二中的坐标判断。

球的移动：这个最简单。重复执行中一个指令：移动鼠标指针即可。为了更换球的造型（球的颜色会依次变换），循环中增加了时间等待。当然这会导致移动不够顺畅。所以可以新增一个启动事件，单独处理造型的变换。

填空选择题

1. 舞台坐标有两个轴，分别是横向坐标（ ）轴和纵向坐标（ ）轴。

2. 用X，Y两个坐标，能够唯一确定一个点。舞台坐标中心点，x=（ ），y=（ ）。

3. Y坐标最大为舞台最上方，坐标值为（ ）。

4. Y坐标最小为舞台最下方，坐标值为（ ）。

5. X坐标最大为舞台最右方，坐标值为（ ）。

6. X坐标最小为舞台最左方，坐标值为（ ）。

如果角色往右水平移动，下面哪个正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往左水平移动，下面哪个正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往上垂直移动，下面哪个正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往下垂直移动，下面哪个正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往右上方斜向移动，下面哪两个答案是正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往右下方斜向移动，下面哪两个答案是正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往左上方斜向移动，下面哪两个答案是正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

如果角色往左下方斜向移动，下面哪两个答案是正确？（ ）

A．X增加 B. X减少 C. Y增加 D. Y减少

当X=0时，X坐标增加10步，x=10；X坐标再增加10步，x=（ ）；即每执行一次增加10步。每次增加步数越多，说明（ ）

A．跑得越快 B. 跑得越慢

小猫往左移动（←），X增加值为（ ）

A．正数 B. 负数

小猫往右移动(→)，X增加值为（ ）

B．正数 B. 负数

小猫往上移动(↑)，Y增加值为（ ）

C．正数 B. 负数

小猫往下移动(↓)，Y增加值为（ ）

D．正数 B. 负数

后续练习

依然是角色的移动，但需要实现角色的自动移动。例如角色从中央开始，然后自动往右移动，到右边框时，自动折返，然后碰到左边框时，自动往右，如此往复。

如果更复杂一些，可以实现角色在屏幕中按照长方形的方式移动。如开始时，角色位于右下角，然后自动往上、往左、往下、往右回到起点。

第六课 抽奖

目的：抽奖游戏运行时会不断的快速切换很多奖品，点击停止，画面停在某个奖品上，即为用户所抽到的奖品。这节课学习如何制作抽奖游戏。通过抽奖游戏，理解背景、角色、造型、随机和广播的功能。

适用对象：适合一年级初学者使用。在老师指导下，学员基本能够独立完成。

背景：之前给小孩玩过一会“果果老师”编写的抽奖游戏。一天下班回来后，小孩跟我说他自己做了一个抽奖的游戏。我很惊讶，不大相信。开机，检查，还真的是一个很不错的抽奖的程序。当然这是一个简单的抽奖游戏，还有很多可以改进的地方。因此我就顺着这个抽奖游戏雏形，逐步深入讲解和制作抽奖游戏。

本游戏共享在：https://Scratch.mit.edu/projects/200876858/

实现方式

最简单的抽奖游戏：这一个版本就是小孩自己琢磨出来的。即将所有的奖品都放在一个角色中，然后点击绿旗子开始时，立即进入一个大循环，然后在循环里面不停的切换造型。点击结束时，系统就会将奖品停在某个造型上，就是最后的奖品。因此只需要三个指令：开始、循环、下一个造型。

抽奖游戏逐步改进：上述抽奖游戏是最简单的版本，是有一些缺陷的。

第一个大问题是奖品的出现不是随机的，是依次按顺序更换造型。虽然小孩加入了42个造型，但是每次运行时，系统是按已经固定的礼品顺序抽取的。这不大公平，因为用户可以记住奖品的大概顺序。虽然奖品跑得很快，但还是能够知道奖品的前后大概顺序的。要解决这个问题，就需要使用随机的功能。即从两个数字之间随机挑一个数字。Scratch提供了“在XX到YY之间随机选一个数字”的功能。在更换造型中，可以使用“更换造型XX”。要使用随机数，就要把角色的造型重新命名为数字，否则以前的英文单词形式，是不能随机抽取的。其实我们之前看的“果果老师”的抽奖程序也是没有使用随机功能的。

第二个问题是抽奖的开始和结束问题。现在的抽奖是使用系统的开始结束按钮。如果需要在游戏页面内部自己制作按钮，然后点击开始按钮即开始抽奖，点击停止按钮就停止。用户可以多次点击开始和结束。因为按钮和奖品属于两个角色，因此要使用广播功能，开始按钮通知奖品开始摇奖，停止按钮通知奖品停止摇奖。

第三个问题是一个小问题。小孩自己做的抽奖程序中，一开始奖品就是显示的。实际抽奖过程中，奖品肯定是先隐藏的。这个改进很简单，即程序开始时，将奖品设置为隐藏，抽奖开始的时候才显示出来。

第四个问题是后续改进。抽奖是一个激动人心的过程，因此通常要增加激动人心的音乐，停止的时候，还可以切换一些灯光，播放停止的音乐。切换灯光可以通过变更背景实现。播放音乐的功能打算放到以后课程内再讲。

功能1：角色（孩子已熟悉）

一个图片或物品都可以作为角色，角色有一个脚本，程序写在角色的脚本中。这是Scratch的基本功能。

可以新建角色，默认角色是一个小猫咪。先删除小猫角色，系统提供了动物、奇幻、字母、人物、物品、交通等角色分类。

例如我们添加了小绿旗子和停止按钮两个角色。“当小绿旗子被点击”，就是开始按钮，然后执行一些操作，即开始抽奖。点击停止按钮时，抽奖结束。再次点击开始按钮，可以重新抽奖。

功能2：造型

每一个角色可能有一个或多个造型。造型是实现动画效果的重要方式。例如角色“猫”。猫有两个角色，通过切换造型实现小猫奔跑的动作，切换造型会实现动画的效果。在一个角色中放多个造型是Scratch的常用方式，否则可能会编写很多重复的代码。例如实现0-9数字加减法，打字程序a-z 26个字母，如果每个数字或字母都设置成单独的角色，那么这些角色中很多代码是完全相同的。使用一个角色，然后加上克隆功能，就可以实现打字、加减法游戏。

例如，在该抽奖游戏中“旗帜”的造型分两个，一个正的一个斜的。见下图所示，

切换时产生运动的效果，旗帜摇动起来。系统中的小旗子其实是只有一个造型的。讲解了可以复制一个出来，然后稍作调整，例如旋转一定的角度。然后切换造型，就可以实现一定的动画效果。

<小猫的两个造型，不停的切换造型，看上去，小猫就奔跑起来了。>

系统指令：下一个造型，实现切换功能。

下面通过造型做出抽奖游戏。这个版本也就是小孩自己制作的最原始的抽奖游戏版本。这个脚本放在奖品上，一个大循环，然后下一个造型，没有随机。奖品即不停循环显示，点击停止，就会停在某一个奖品上。

然后同理，将这个脚本放在小猫咪角色上，小猫咪就跑动起来了。当然如果不加等待时间，小猫咪的跑动就实在是太快了，快被累坏了。因此设置了等待0.1秒，才更换下一个造型。这样小猫咪的跑动就会正常一些。

制作步骤

第一步：选择舞台背景：这个舞台背景是小孩自己选的。

加入了小猫咪，在不停的跑动；加入了一个西瓜，西瓜也切换造型。说这是小猫咪在吃西瓜。然后加入了雪人，说雪人和小猫咪在舞台前面观看抽奖。加入了两个雪花角色，当装饰使用。这些是小孩自己加上去的。

后来我们在改进环节，加入绿旗子开始按钮，和橙色的停止按钮。绿旗子和停止按钮都复制了一个造型，稍微转动一下，实现动画效果。这样程序开始时，就知道这两个地方是可以点击了。

第二步：设置奖品角色的造型

1. 首先随便选择一个物品作为奖品角色。然后在这个角色中增加其他造型，即其他奖品。

【操作步骤】选择奖品角色，“从造型库中选取造型”。选多少个即有多少个奖品。这些造型都同属于一个角色。如下图所示，

孩子添加了42个造型，这42个造型都同属恐龙这一个角色，随着奖品的切换，角色上的图也会跟随切换。

2. 为了使用随机函数，我们造型进行数字编号，更改造型名称为数字。选中造型，右上角更改录入数字，按照顺序录入数字即可。

问题：目前多个造型是按顺序排列执行切换，不是随机的。能够猜出奖品，可以控制奖品。我们要做成随机的。所以要进行以下拓展操作。如果在更换造型中增加等待2秒，那么能够很明确的看到，系统是按顺序更换造型的。加入随机功能以后，系统就从42个奖品造型中随机抽取一个了。用户就无法猜测系统下次将显示哪一个奖品了。

第三步：设置随机显示造型

1. 随机的介绍：即系统中提供这样的指令，从一个数到另外一个数之间随机选一个数。例如，如果有42个造型，每次都是从中随机挑选一个。因此，需要从1到42中，随机挑选一个数字。

2. 【操作】点击奖品角色的脚本，添加“将造型切换为在1-42之间随机选择一个数”。我们需要将随机数放在造型切换指令中。因此我们需要将这两个指令拼在一起。这是一种很常用的方式，需要熟练掌握。

开始时，请先忽略广播。在最简单的抽奖游戏中，也可以使用这个随机功能。

【操作步骤1】点击脚本—外观，选择

【操作步骤2】点击脚本—运算，添加指令 。

造型是有编号的，系统默认编号为英文字母，我们之前对造型进行编号，该游戏中有1-42个造型编号。已更改编号为1-42。

【操作步骤3】把 拖到 中。即

这样，每次执行就会随机显示一个造型。

第四步：实现对奖品角色的开始和结束广播功能。（如果不自己制作按钮实现开始和结束的话，抽奖程序会简单很多。）

Scratch中，角色就是我们需要使用程序操作的对象。角色和角色之间是相对隔离的。如果一个角色需要通知另外一个角色要做什么事情，那么我们可以使用“广播”的功能。“广播”就是对程序中每个角色“大喊一声”，这样其他角色才能听到，听到以后，他们就去做一定的事情。否则别的角色是不知道其他角色要他做什么的。在这里，开始和结束是两个按钮角色，点击以后，要告诉奖品角色，开始或结束抽奖。开始按钮喊“开始抽奖”，抽奖的角色接收到这个消息以后，就显示奖品，开始抽奖。结束按钮喊“结束抽奖”，抽奖的角色接收到这个消息以后，就停止抽奖，显示抽中的奖品。

目的，用点击按钮的方式实现对一个角色的开始结束动作。

1. 添加旗帜图标角色 表示开始；

2. 添加八边形图标角色 表示结束。

3. 设置造型，开始和结束角色图标都有两个造型，切换造型产生动画效果。

4. “等待”操作：点击新增的开始按钮时，奖品角色开始快速变化。如果不想让它动那么快，在脚本-控制中选择

，例如更改时间为0.1秒。

要设置等待多少秒。就不会变化太快。

功能3：广播

1. 对开始角色进行广播

例如，开始的时候要告诉角色奖品开始抽奖了，当角色被点击这个事件开始时，做一个广播，即喊一声开始抽奖。通过广播的形式，对方（奖品）接到抽奖信息开始随机切换奖品。广播就是让一个角色喊一声。

广播可以新建消息，比如让小旗帜喊一声开始抽奖，奖品收到信息后便开始切换造型。

点击开始旗帜角色 ，脚本—控制，如下图，

当开始角色被点击时，广播“开始抽奖”。

奖品角色收到这个广播消息以后，就开始抽奖。

当结束角色被点击时，广播“结束抽奖”。

奖品角色收到这个广播消息以后，就结束抽奖。结束时，我们使用了一个巧妙的指令“停止角色的其他脚本”，即系统将这个角色中的其他脚本停止掉，也就是说系统会将开始抽奖的那一段无限制循环停止运行。如果停止所有脚本，系统将停止所有角色的脚本，那么开始、结束、小猫咪等角色将被停止运动。我们想用户可以继续抽奖，所以这些角色继续执行。还可以选择停止当前脚本，就是停止收到结束抽奖后的其他脚本，这里其实是没有其他脚本的。如果不想使用这个指令，那就要使用一个全局变量了。变量功能，将在后续课程中介绍。

2. 对结束角色进行广播

奖品一开始设置为隐藏。当点击结束 时，结束抽奖，奖品收到结束抽奖消息时，停止该角色的其他脚本。即只停止奖品这一个角色的脚本。奖品运动结束，小猫和西瓜等其他角色仍然在动不受影响。

点击结束八边形角色，脚本—控制，如下图，

只有点击系统自带的右上角脚本旁边的停止按钮时，才会停止全部角色的脚本。

以后的提升练习：添加播放声音，比如添加开始和结束时的声音。

本章重点：

1，造型：每一个角色可能有一个或多个造型。

2，切换造型：如果不想变化很快，添加等待指令。

3，随机：随机找一个造型：对造型进行数字编号，添加随机指令，将造型切换为在1-X之间随机选择一个数。

4，广播：广播就是让一个角色喊一声，对方收到广播后执行命令。

**第七课 打字游戏**

目的

制作字母打字游戏。字母随机从屏幕上方出现并往下坠落，坠落过程中，按下对应的字母键，该字母即消失，得分。

[结果运行图]

实现方式

主要技术是克隆、移动和随机。本游戏只需设置一个角色，里面包含26个字母造型。开始时，每隔一段时间即克隆该角色本身，克隆体启动时，不停往下移动。监控是否有按键按下，按下以后，对应的克隆体消失。为了让字母从屏幕不同位置出现，因此对于角色的X轴需要随机取值。

制作步骤

第一步：添加角色和造型。

1. 加入一个字母角色

2. 在这个角色中，按字母顺序依次加入其它25个字母造型，总共有26个造型。

【角色，我们重命名了造型，但其实并没有什么用。获取造型时，得到的是造型编号，而不是名称】

第二步：克隆角色，坠落。

3. 开始后重复执行，无限循环

4. 每一次循环都克隆一个字母角色，因此循环里面可以增加时间，如等待2秒。等待时间越短，克隆的字母就越多。

5. 克隆体出现以后，进入循环，X轴保持不变，Y轴不停减少。即将到达屏幕时，删除克隆体。

第三步：接受按键，消除字母

6. 新设一个克隆体启动事件，启动即不停循环检查是否有按键。（一个无限大循环后面就不能添加其他代码，因此需要另起炉灶，新增一个克隆体启动事件）

7. 如果A被按下，判断那造型编号是否为1，如果是1，则计分，并且删除克隆体，增加得分。造型获取中并没有获取造型名称的语句，只有获取造型编号的语句，因此造型需要按顺序排列。

第四步：随机字母

8. 克隆自身之前，将造型切换为1-26的随机数字，即在26个字母中随机选择一个字母。如果不进行随机处理，那么字母将按顺序依次出现。

后续改进

本打字练习，仅随机选取26个字母。在实际的英语录入是按照单词句子进行的。如果能够存储一定数量的单词，然后依次显示某个单词中的字母，形成单词中的字母雨，那将更贴近实际。同时在应用技术上，能更上一层楼。

**第八课 太空大战**

目的

目的：变量是程序最为基本的要素之一。本程序制作了游戏“太空大战”。初始时，超人的生命值为3，碰到火龙，超人生命值减1。生命值为0则超人牺牲。超人在飞行中获取金币时得分，集满10个金币，生命值加1。本游戏使用了一些变量，通过变量存储分数、金币数量、生命值，在运行过程中修改相关变量。

适用对象：适合一年级初学者使用。本游戏的初稿基本由小孩一个人完成，后续在讨论指导下，丰富了其他功能。

本游戏共享在：https://Scratch.mit.edu/projects/195739577/

实现方式

简单描写一下过程吧。

一、变量

均为跨角色的变量，即全局变量。

分数：每收集一枚金币，分数加1。

生命值：初始值为3，超人有三条命。三条命用完，超人牺牲，游戏结束。每碰到火龙一次，就减1。相当于三条火龙等于一个超人。

金币数量：超人收集的金币数量，数量满10（或5），立即兑现一条命，即生命值加1，金币数量归零。

四个角色

二、超人角色

1. 1. 在造型中，自己用画笔简单画了一个超人。

2. 2. 开始时，设置各变量的初始值。

3. 3. 开始的时候，超人在左中间。否则可能就找不到了。这是后来加上去的。

4. 4. 实现上箭头和下箭头的按键，超人上下移动。这种移动放在一个大循环中，已经很熟练了。

5. 5. 如果碰到火球（金币），分数加1。分数的变量已经用过好几次，已经比较熟练了。

6. 6. 游戏结束处理：如果生命值为零，则超人隐藏，结束游戏。

这里有改进，开始时并没有生命值的概念，只要碰到火龙就结束。后来我们说设置三条命，碰到三次火龙，才结束游戏。

游戏结束时，切换一下游戏结束的背景，告诉玩家游戏结束了。

游戏结束后，停止所有脚本。否则飞龙和火球会继续飞出来。

简单画了一个超人。

超人脚本，上移，下移。生命值为零时，结束游戏。碰到火球，分数加1.

游戏结束画面，太难看啦。

三、火龙角色

1. 1. 火龙角色从右边飞往左边，因此需要不停的克隆。这一步已经很熟悉了。

2. 2. 克隆体启动时，重复执行，不停的从右边到左边，因此X减少10。

3. 3. 克隆体启动时，Y方向需要随机出现一个位置，否则位置就固定了。移动到X=240，Y从-180到180随机选择一个坐标值。这一步也熟练了。

4. 4. 如果碰到左边的竖线，就删除克隆体。开始的时候，小孩使用了隐藏功能。在克隆方式中，还是使用删除克隆体为好，否则该克隆体还是存在的，会逐步拖累系统的速度。

5. 5. 如果碰到超人，将生命值减少1。超人生命减少。同时删除火龙的克隆体。就是三条火龙的命能够消灭超人。

火龙脚本，碰到超人，生命值减1

画了一个火球。

四、金币（火球）角色

火球角色和飞龙的角色功能类似。

1. 1. 火球角色从右边飞往左边，因此需要不停的克隆。

2. 2. 克隆体启动时，重复执行，不停的从右边到左边，因此X减少10。

3. 3. 克隆体启动时，Y方向需要随机出现一个位置，否则位置就固定了。移动到X=240，Y从-180到180随机选择一个坐标值。

4. 4. 如果碰到左边的竖线，就删除克隆体。

5. 5. 如果碰到超人，就是被超人收集了，分数加1。金币数量加1删除克隆体。如果金币数量等于5（10），生命值增加1，同时将金币数量重置为0.

金币脚本，碰到超人，金币数量加1。金币数量等于5，加一条命。

五、竖线角色

竖线角色不用添加任何脚本。碰到竖线就删除克隆体，这也是一种常用的方式。在击球运动中，碰到球板则反弹，使用了横线。

**第九课 跳跃的小猫**

目的

目的：跳跃是很常用的游戏功能。不少游戏都是通过跳动前进、躲避障碍物，获取得分。这一次我阅读了Scratch上关于跳跃的说明和举例。以此为蓝本，由浅入深，练习各种跳跃运动。

适用对象：适合有一定编程和数学基础的用户，例如一二年级学生。通过之前约10次的练习，一年级学生能够独立完成本课程中多数功能，或者在家长指引下完成练习。

本游戏共享在：https://Scratch.mit.edu/projects/216224542/

实现方式

本程度使用简单的猫咪角色，同时在屏幕最下方放置一红线，即地板。通常的跳跃是要在一定的物体上才能跳跃，例如地面、某个运动的挡板上。空中的跳跃一般称为飞翔。当然开始的时候，也可以随意跳动，并且跳动的速度完全相同。在现实世界中，人跳起来的时候，因为受到重力的影响，速度会逐渐减小直到为零，然后落下，落下的过程中，速度不断增加，达到地面的时候，停止。除了垂直的跳动以外，还可能是往前跃进的跳动(跳远)。

参考文档：https://en.scratch-wiki.info/wiki/Jumping

下面描述几种跳动方式。

一、最简单的跳跃方式

点击数字0，小猫跳动一定的距离，然后停留一小会，落下。上升和下降都是“瞬间位移”的，即立即改变位置。并且在任何地方都可以跳动。

测试这种跳跃方式，然后提出问题：我们的实际跳动果真是如此的吗？是瞬间跳到最高处吗？当然明显不是这样，因此需要继续改进。

二、匀速跳跃

实现往前跳跃，即上前方跳动，并且落下。匀速上升和下降。

1. 程序开始时，将小猫咪移动到左下角，等待跳动。

2. 点击上箭头，小猫跳动往上跳动，跳动过程中，循环16次，每次Y轴增加10，同时X轴增加2。这实现了匀速的上升跳跃，和之前的“瞬间位移”好了一些。让小孩思考人的实际跳跃运动是怎么样的。往前跳跃是小孩想出来的，这比较符合实际。

3. 跳到最高点，可以停留零点几秒，然后开始下降。小孩提出想法，到达最顶端后翻一个跟斗，因此重复执行10次，每次右转36°，即旋转360度。

4. 下降过程和上升过程相同，循环16次，每次Y轴增加-10，同时X轴增加2。开始的时候设置为X轴增加-2，下降过程中，又回到原点了。这不符合常理。小猫在跳动过程中，一直往前走，因此X轴一直都在增加。

5. 小猫往右跳动，快到边缘时重新返回到最左边，否则就跳出屏幕了。这个功能很早就熟练了。

6. 跳动之前和落地之后，更换了一次造型。如果造型合适，能够实现落下后下蹲然后站起的效果。

改进：如果熟练，这里也可以引入变量，例如通过循环次数、单词循环Y轴位移变量，决定跳跃的高度。

三、匀速跳跃2

实现往前跳跃，即上前方跳动，并且落下。匀速上升和下降，但上升和下降的速度不同。

按下数字1，实现跳跃，实现方法同上，仅是下降的时候速度比上升的速度低，差异较大时能感觉到漂浮下降的过程。同时小孩选择了，反向翻跟斗。这个翻跟斗是小孩自己想出来的，做出来以后，让三岁小妹妹玩了玩，引起了一片笑声。

四、从地面上匀速跳跃

功能和第二类似，但增加了地面的判断，只有在地面上才能跳跃，在空中不能跳跃。

1. 增加一条红线的角色，放在屏幕最下方，模拟地面。即小猫只有碰到这条红线才允许跳跃。

2. 点击数字2，同时判断是否碰到某个角色或某个颜色，只有满足这两个条件才能跳跃，否则不能跳跃。在“侦测”中，有碰到某个角色，或碰到某种颜色的指令。

3. 为了测试离开地面后小猫无法跳动，增加上移和下移的操作，按A上移，按Z下移。测试的时候，将按键1的无地板判断的跳跃和按键2带地板判断的跳跃进行比较。

五、加速跳跃（考虑重力）

功能和第二类似，但更改了速度。小猫在上升跳跃过程中，速度逐渐减小，直到速度为零。到达最高点以后，下降，速度逐步加快。因此看起来，就比较符合现实场景了。

1. 为了便于控制，引入了y轴的速度变量。

2. 点击数字3，跳动时，速度设定为15，即每次Y轴增加15，然后Y轴速度增加-0.5.重复执行30次。到达顶部的时候速度已经为零。

3. 到达顶部的时候，停留0.1秒。

4. 下降的时候，将Y的速度增加0.5，即Y坐标的每次增加值为0.5的倍数。实现加速下降的功能。

六、两次连续跳跃（考虑重力）

功能上面带重力的跳跃类似，只是这里实现了连续跳两次的功能，每次的高度相同。

1. 为了便于控制，程序块-积木。在积木中实现跳跃代码，带重力功能。这样能够减少主程序的代码量。

2. 点击数字4，调用一次积木，然后接着再调用一次积木。实现两次跳跃。

七、多次连续跳跃（考虑重力，高度逐步降低）

功能与上面两次跳跃类似，这里实现了多次跳跃的功能，每次的高度逐步降低。通过这种方式，能够模拟扔皮球的动作。即往上前方抛球，球往前弹起，然后落下，高度逐渐降低。

1. 依然使用了积木块，增加了高度的变量。

2.

3.

4. 点击数字5，设定跳跃高度，然后调用一次积木，然后再设置一个跳跃高度，再调用一次积木。这里跳了五次，高度从30，20，10，5，3。如果水平好一些，可以多加一个循环进行处理。

上面列了7种跳跃方式，通过这节课，对跳跃应该比较好理解了。

**第十课 演奏的钢琴**

目的

目的：编写一个简易的钢琴演奏程序，了解Scratch对声音的处理（播放声音、弹奏鼓声、弹奏音符，控制节拍，选择乐器，控制音量等）。

适用对象：适合有一定编程和数学基础的用户，例如一二年级学生。在家长指引下小完成多数练习，用时约2小时。

本游戏共享在：https://Scratch.mit.edu/projects/217924108/

实现方式

尽量使用简单的语句和方法，实现一个简易的钢琴。首先自己制作一个钢琴白色按键，有两个造型，只是颜色有点区别。按下按键的时候，切换造型，显示按下的效果。实现从1,2,3,4,5,6,7，i八个音节，因此复制角色即可。方法相同，制作5个黑色按钮角色。（为了简单，并未使用克隆功能，因此代码是重复的。这种方式适合入门的用户。）

一、钢琴白键：造型、按键和播放声音

1）自己制作一个角色，用填充白色的长方形即可。制作完毕以后，复制这个造型，修改填充颜色。白键的两个造型即完成。

2）添加按键、播放音节的脚本：开始时切换造型到1（即白色）。重复执行，判断按键A被按下，则切换造型2（浅蓝色），然后“弹奏音符60 0.5节拍”，如果没有A被按下，则切换回造型1（白键）。

3）其他按键类似，只是弹奏的音符不同，1,2,3,4,5,6,7，i八个音符的数字依次为：60,62,64,65,67,69,71,72.

4）为了使按键间隔一致，初始化时移动到X Y，Y为0，X从-180开始，每个按键增增加50。为了简单，这里并未使用变量。级别高的用户可以使用变量，以便在整体往右移动按键的时候会简单些。

二、钢琴黑键：造型、按键和播放声音

1）使用相同的方法制作黑键。两个造型，高度稍短。位置和钢琴上的黑键相同，即3,4之间没有黑键。

改进

中高级用户可以实现这些改进功能：

1） 使用变量统一控制钢琴按键在屏幕中的相对位置。

2） 使用克隆，减少角色数量，使用一个白色角色、黑色角色，通过克隆，即可实现所有按键。（代码虽有所减少，但单个角色中的代码会增加，程序会复杂一些）

3） 考虑按键时长：这种循环中，直接判断某个按键是否被按下，存在严重的缺陷。即用户如果连续按着按键，系统会再次播放这个音节。相当于每次都是固定的0.5拍，不会随着按键快慢而变换节奏。

**第十一课 植物大战僵尸**

目的

实现植物大战僵尸游戏。游戏比较大，需要分几次课程才能完成，因此这里先讲述第一部分：种植豌豆射手、豌豆射手射出豌豆，种植向日葵，僵尸从右边往左边移动，到房子边上开始“吃脑子”。

适用对象：适合有一定编程和数学基础的用户，例如一二年级学生。在家长指引下能完成多数练习，每次课程约2个小时，制作完整的植物大战僵尸需要三四次课程。

游戏主界面。

本游戏共享在：https://Scratch.mit.edu/projects/219832179/

实现方式

分步骤实现植物大战僵尸游戏。我让小孩自己画了植物大战僵尸的各种图标、草坪，但颜色较浅，效果不好，最后还是使用了网上的一个版本的图片。将来可以将这些图片替换掉。

一、角色说明

1） 三个卡牌：阳光卡牌、向日葵卡牌、豌豆卡牌。 卡牌位置固定。阳光卡牌下方有阳光数量变量。收集阳光以后，阳光数量增加。使用阳光，可以终止豌豆射手或者向日葵，种植以后，扣减对应的阳光数。

2）背景：草坪、围栏、终点是房子。如果僵尸走到房子边上，就开始吃“脑子”，脑子被吃完，游戏则结束。

3）阳光：阳光会定期随机飘落，可以点击，收集阳光。

4）豌豆射手：会摆动，会发豌豆子弹，打击僵尸。

5）豌豆子弹：种植豌豆以后，豌豆开始发射豌豆子弹。

6）僵尸：分成两个角色，一个是普通前进的角色；一个是碰到向日葵、豌豆射手或到达房子时，吃植物或脑子动作的角色。开始吃的时候，切换成开始吃的角色。

7）向日葵：向日葵会产生阳光。

二、铁桶僵尸

僵尸被打中两次即消失。

僵尸从屏幕右边随机出现，向左移动。到达左边房子以后，记录位置，广播开始脑子，然后显示第二个开吃脑子的僵尸。如果碰到子弹，打中次数减少1，如果打中次数为零，则打死了，删除克隆体。这里打中次数就是僵尸的生命值。

三：吃脑子的僵尸

收到吃脑子的广播以后，克隆自己，移动到原僵尸位置，显示，开吃脑子。这里设置了吃十次以后，僵尸消失。

四：种植豌豆射手（豌豆射手卡牌）

阳光数大于100，可以种植豌豆射手。种植时，立即克隆一个豌豆射手，减少阳光数，否则提示阳光不足。

五：豌豆射手

豌豆射手启动时，切换造型，实现动画效果。

豌豆射手启动时，跟随鼠标移动，当鼠标被按下以后，记录鼠标的位置，这个变量为角色变量，非全局变量。种植豌豆射手一段时间以后，开始发射豌豆。这里有个关键点，就是鼠标按下，广播发射豌豆以后，要停止当前脚本，即豌豆射手固定在草坪上。

六：豌豆子弹

接收到豌豆射手发射子弹的广播以后，克隆自己。然后移动到豌豆射手位置，开始向右移动，碰到边缘或者僵尸，重新回到原点，注意不是删除克隆体。

如果打中了僵尸，则子弹需要移回豌豆射手。（这里可改进：如果碰到僵尸，需要考虑射手和僵尸的距离，经过一段时间后再发射下一个豌豆子弹。否则僵尸靠近射手时，子弹的发射速度将加快，这不合理。）

七：种植向日葵（向日葵卡牌）

阳光数大于50，可以种植向日葵。种植时，立即克隆一个向日葵，减少阳光数，否则提示阳光不足。

八：向日葵

向日葵启动时，切换造型，实现动画效果。

向日葵启动时，跟随鼠标移动，当鼠标被按下以后，记录鼠标的位置，这个变量为角色变量，非全局变量。经过一段时间以后，要停止当前脚本，即豌豆射手固定在草坪上。还未制作向日葵产生太阳的功能。

九：阳光

初始化阳光。每5秒克隆产生一个阳光。

克隆后，阳关在一定范围内随机产生，然后在一秒内，随机滑行到另外一个地方，停留三秒。点击阳光时，阳光滑回卡牌，阳光增加50。

植物大战僵尸的游戏，初步可以玩玩了。

**第十二课 时钟**

目的

制作一个带秒针、分针和时针的时钟，秒针每秒走一小格，分针每60秒走一小格，时针每12分钟走一小格。

适用对象：这个游戏比较简单，具有初步Scratch基础的学员在教师的指导下均能完成。考虑到程序时间和讲解需要时间，安排2-3个小时会比较合适，适合一二年级小孩。

这是在朋友家里完成的。对方刚好也有一个一年级小孩，具有初步的Scratch基础，因此两个小朋友都顺利的完成了任务。虽然任务简单，但我们加了不少内容，最后还是花了两个小时左右。

首先：我带他们看钟表实物，介绍时钟。客厅刚好有一个挂钟。但 小孩看不懂上面的数字，因为这个朋友家是大House，装修超级豪华，能让我这种住老小房子的人感到惭愧的豪宅。时钟上是罗马数字哦。

我在白板上介绍了一下时钟，特别是小时、分钟、秒的关系。整点在哪里，七点干什么，八点干什么，几点下课放学等。通过询问回答，能够起到热身的作用，逐步进入程序制作环节。时钟是一个大圆圈，分成60份。每一秒，秒针就需要走一小格，即向右旋转一定的角度。这个角度要引导才能算出来。即360/60=6°。

教师手稿图~~~

不带指针的时钟背景图，之前我就准备好了。然后自己选择想要的角色当做时分秒三个针。

其次：开始进入MIT Scratch系统，开始制作程序。用到角色中心十字星的功能，十字星要放在针的顶端。先制作秒针，对准12点的位置，设置指针中央位置，然后大循环，每等待1秒就右旋转6度。

制作完毕秒针，小孩就兴奋了，叫爸爸妈妈来看看。然后继续完成分针。分针和秒针类似，只是每60秒，才转一小格。时针其实是比较复杂的，要准确计算时针每走一小格的秒数。时针每走一大格是一个小时，即60分钟，然后分成5个小格，因此是60/5=12分钟；12分钟=720秒。

其实制作时针之前我们吃了一顿午饭，午饭后写作业，然后继续完成下列任务。

第三：做完以后，然两位小朋友上台轮流讲解。这是家长们是学生，小孩是老师。小朋友表现还是很不错的。这一个环节很重要，能够体现小孩的思路是否清晰，表达力如何。

第四：练习关联的英语。写了几个单词Clock, Time, Second, Minute, Hour, What time is it? It’s 8 o’clock.等。

游戏主界面：

本游戏共享在：https://scratch.mit.edu/projects/225754329/

实现方式

背景-时钟；三个角色：时分秒三个针。方法在上面基本上都说明了。

一、背景

背景图是在网上找的，要找不带指针的时钟。

二：秒针

初始化的面向很重要，不过可以引导小孩逐步尝试。通过增加和减少方向数字，确保开始的时候在12点的位置。

然后移动XY的指令，很重要，确保指针的十字中心点（头部）在时钟背景中心点。因为原来时钟的图形是方的，Scratch布景是长方形的，因此时钟会靠左一点。

然后是主程序，每等待1秒，右转6°。

三：分针

和秒针类似，每60秒，右转6度。

四：时针

和秒针类似，每720秒，右转6度。

五：学生讲解什么是时钟，今天程序做了什么，怎么做的。

第一个小朋友，讲解的很好

我家，讲的也还行，还有另外一张，没上载好。

**第十三课乘法问答**

目的

目的：通过编写一个简易的九九乘法问答程序，理解变量、问答和广播。本想编写一个输出九九乘法表的程序，以便能够练习循环、克隆和对象的相对位置。最后决定先做一个乘法问答小程序。

适用对象：适合有一定初步Scratch编程经验的用户，例如一二年级学生。在家长指引下学员能独立完成大部分功能，用时约3小时。

本游戏共享在：https://Scratch.mit.edu/projects/211180722/

实现方式

九九乘法问答思路：

1） 点击魔术师出题，系统从1-9随机选择两个数，并且相乘，得到答案。

2） 魔术师显示被乘数、乘于的操作符、乘数、等于，然后显示问号。

3） 魔术师随机询问，答案是多少，显示录入框。

4） 用户录入答案。系统判断用户录入是否和系统计算一致；

i. 一致：隐藏问号，显示录入的值；显示正确打钩角色；

ii. 不一致: 隐藏问号，显示录入的值；显示错误打钩角色。

5） 用户点击“查看答案”按钮。

a. 系统显示“正确答案”标签；

b. 在旁边显示正确答案。这样用户可以将自己录入的值和系统答案相比较。

• 变量设置

定义四个变量，均为全局变量。

• X：被乘数

• Y：乘数

• R：系统结算结果

• Input：用户回答录入的值

2. 魔术师角色

1） 初始化：隐藏输入和结果变量。初始化变量值。

2） 当角色被点击时：

a. 隐藏结果和录入变量

b. 随机选择1-9数字，并存入X

c. 随机选择1-9数字，并存入Y

d. 计算结果：X \* Y, 并且写入到结果变量R中

e. 广播“显示问号”。问号角色收到以后，就显示问号。

f. 询问答案，并等待用户回答。用户需要录入答案。

g. 用户录入答案确定以后

i. 将用户录入的值写入Input变量。

ii. 如果回答正确（Input = R），则广播“对”。打钩的角色收到后，显示打钩。

iii. 如果回答错误（Input <> R），则广播“错”。错误标识的角色收到后，显示错误标识。

iv. 显示用户录入的变量Input。

3. 显示答案角色(Answer)

1） 当这个按钮被点击时，广播两个消息：隐藏问号、显示“正确答案”的标签、显示答案变量。

2） 问号角色收到后隐藏问号。其实之前如果用户有录入的话，已经会隐藏掉，显示用户实际录入的值了。

4. “正确”角色

1） 收到回答正确的广播时，显示。

2） 收到回答错误的广播时，隐藏。

3） 收到显示问号时，即出题完毕，也要隐藏。

5. “正确答案”标签角色

4） 这是一个标签，标签是一个图“正确答案”。

5） 程序开始时隐藏。

6） 点击魔术师出题时会隐藏；点击“Answer”按钮显示答案时会显示。

后续改进

1） 增加难易程度按钮，高中低三挡。选择高，则从20-99随机；选择中，则从10-19随机；选择低，从1-9随机。

2） 将用户录入的值，按照图片角色的形式显示出来。因为显示变量的样式是固定的，过于简单，通过显示图片，可以实现各种效果。

**第十四课 排序**

目的：

使用冒泡排序算法，对数组中的数值由大到小进行排序。通过比较冒泡，了解冒泡排序的原理。初步了解数组链表功能，往链表中添加数据，读取链表的数据，删除链表数据，获取链表数据条目数量。

适用对象：适合有一定初步Scratch编程经验的用户，例如一二年级学生。本算法单独完成的难度比较大，需要家长讲解。

本游戏共享在：https://Scratch.mit.edu/projects/213542724/

微信订阅号：ABO\_CRM

实现方式

冒泡排序制作思路：

1） 背景图：既然是冒泡，那就加个海底的冒泡的背景图。

2） 火龙角色：为了增加趣味性加入了一个火龙角色，开始时清空数据，然后随机填充15个数字，接着进行排序。火龙负责说话，提示当前系统在做什么。

3） 数组（链表）变量一个：Data

4） 设置三个变量：循环次数，当前位置，临时变量。

变量和链表。

5） 清空数据：程序开始时，清空链表中的数据。

a. 火龙提示“开始清空数据。”

b. 使用链表删除指令，可以删除链表指定位置条数条目，也可以全部删除或者删除尾部的数据。因此如果简单，可以直接全部删除。不过为了体现操作过程，我们使用了重复执行，读取条目总数，如果条目总数>1就删除第一条，然后等待0.1秒。这样就能看见系统依次删除链表中的数据。

c. 火龙提示，“数据已清空”。

清空，填充数据。

6） 随机填充数据：

a. 火龙报“开始随机填充数据”

b. 强制加了一条最小的数0到链表；

c. 然后执行14次，在1-999中间随机选择一个数字，填充到链表中。填充中间未等待，因此一会填充好了15个数字。

（初始化后开始冒泡）

7） 冒泡排序：大循环，逐个比较。

a. 设置当前位置变量为15，从最后一个开始，和前面一个数相比较，如果比前面的数据大，就要往上冒泡，否则就不冒泡。

b. 设置循环次数：为当前位置减1，即第一次大循环，总最下面的第15个数字开始，一直往上比较，比较14次。15和14比，14和13比，最后是2和1比较。第一次将15个数字中，最大的数据冒到最上面，即位置1。第一次大循环结束。

c. 因为第一个数字已经是最大数，因此只要对2-15的数字进行第二轮排序即可。这样第二个大循环，重复比较的循环次数就可以减少1。因此第二次比较了13次，15和14比，14和13比，最后是3和2比较。即14个数字中找最大。第二次大循环结束。

大循环。

d. 依次类推，每次都找一个最大的，搬到最上面去。

e. P和P-1位置的数字相比较时，如果P> P-1，即下面的数字大，就要交换。因此先将P-1存在临时变量里面；然后P挪到P-1，然后临时变量挪回P，实现P往上冒一格。火龙报“需要冒泡”或者“不需要冒泡”。

f. 冒泡比较后，等待0.5秒，这样能够看见在比较，冒泡，交换数字。

g. 每个大循环之间等待2秒，让火龙报“下一个大循环开始。”

排序结束

非专业程序人员编写，希望没有什么错误 ：）

**第十五课 贪吃蛇**

目的：完成一个贪吃蛇的小游戏，了解角色图片制作、克隆、运动方向、碰到障碍物的处理。

游戏说明：游戏开始时，蛇不停的往前移动，身体逐渐变长。玩家可以通过上下左右四个按键改变蛇运动方向（向上、向下、向左和向右）。蝙蝠是蛇要吃掉的小动物，可能随机出现在屏幕任何一个地方，挥动翅膀，但保持位置不动（当然也可以让蝙蝠飞起来）。当蛇碰到蝙蝠时，蝙蝠消失，即蛇将蝙蝠吃掉，获取得分；蛇吃掉蝙蝠以后，其中的一节身体变成蝙蝠的颜色。经过几秒后，（另外一只）蝙蝠重新出现在另外一个位置。游戏继续。

这个游戏属于“炒冷饭”的范畴，因为很多小游戏的基本实现指令和方法都是类似的。

适合人群：一二年级学生。在家长或老师指导下完成。

用时：2-3小时。

一、角色设置

设置两个角色。一个是蛇，分成蛇头和蛇身。蛇的身体由一块一块组成。

蝙蝠：蛇的食物，可以挥动翅膀(，也可以飞翔移动)。

微信订阅号：ABO\_CRM

二、蛇头的制作

这是Scratch标准角色库中提供的球，有不同颜色的球。我们以这个球为基础，制作蛇头。

蛇头造型1：在角色中，使用“变形”，出现出现几个可以拉动的小点点。拉动圆形右边的点，最后形成尖尖的蛇头形状。

蛇总要有眼睛，因此使用用粗线点两下，或者用画圆的功能，画两个圆，填充红色。这是蛇的造型1.

蛇头造型2：因为蛇在上下左右移动时，要保持头的方向一致，因此复制一个图，然后上下翻转，形成另外一个蛇头，造型2.

三、让蛇头运动起来

因为一个多月没有做运动类的程序了，今天已经比较生疏。为了让蛇头前后左右移动，显示使用了右边的X或Y坐标增加一定位置的循环指令。

后来改成左边的运动和方向。

重复循环，每次都移动1步。默认的时候，蛇头是往右边移动的。如果按了上箭头，面向0度，即往上走。如果按下箭头，面向180度方向，即往下走。

测试。

但这时候我们会发现，蛇头的方向是不对的，因此，下移的时候，需要将蛇头的造型切换为2. 除非蛇头上下对称，否则就有这个问题。

用类似的方法，处理左右移动。

现在，这个蛇头可以顺畅的动起来了。

四、蛇身体的制作-让蛇长出身体

也不能只有这一个蛇头啊。怎么做出蛇的身体呢？那就一节一节克隆吧，没格一段时间，克隆出一节“蛇身体”来，然后间隙小一些，看上去，就成一条连续的蛇了。

这里再次使用一遍“开始”指令，蛇头每隔一秒，就克隆一个自己。注意克隆等待时间和运动步数，蛇的身体要保持连续。

试玩一下。

随着蛇的移动，蛇真的变长了， 长出来了。但是有个问题，有很多蛇头，因为我们是克隆了蛇头。Scratch允许我们克隆300份，因此可以一直玩，到了300份蛇头，就不会继续克隆了。

到这一步，小孩的兴致已经上来了，一定要把满屏都擦成蛇头为止。小妹妹也上来玩了。

五、蛇身体的制作-一节一节的蛇身

为了解决复制出来的都是蛇头的问题，我们需要一节蛇的身体，就是没有眼睛的头。

这个很简单，只需要复制出另外一个造型，把蛇的眼睛去掉即可。造型3：一节蛇的身体。

六、蛇头变成蛇身

蛇头复制出来以后，克隆体启动时，需要将造型切换成3，即将蛇头变成蛇的身体。如果不变换，那就会出来很多蛇头。

再玩一下。

这时候就只有一个蛇头了。还是可以克隆300份。KK想把整屏都盖上蛇，到后来就不会继续克隆了。这也体会了只允许300份克隆体的限制。在实际应用中，要将不需要的克隆体删除，而不能仅做简单的隐藏操作。

克隆体启动后，等待8秒，就删除本克隆体。这样蛇的身体就不会“无限”长了。

七、蝙蝠角色-随机出现

本来贪吃蛇的游戏是让蛇吃掉球或豆子的，结果说要吃掉蝙蝠。就加了一个蝙蝠角色。

首先让蝙蝠挥动翅膀，这个只要不停更改造型就可以了。

然后，让蝙蝠出现在屏幕的随机位置上。 X和Y都使用一个随机数，当然不能太靠近边缘。

八、蛇吃蝙蝠：侦测碰到某种颜色

这是关键的一步。我们使用了碰到某种颜色，蝙蝠则隐藏。选择“碰到颜色”以后，可以在屏幕中点击蛇的眼睛， 这样能够获取的红色。

当碰到红色的时候，蝙蝠隐藏。蛇将蝙蝠吃掉了。

九、蛇吃蝙蝠：游戏得分

吃掉蝙蝠以后，分数增加。因此设置一个全局变量。开始时，将分数设置为零。蝙蝠被蛇吃掉后，分数增加。

十、蛇吃蝙蝠：蝙蝠在蛇的肚子里

吃掉蝙蝠以后，蛇的肚子有一节会变成黑色，甚至那一节还可以变大一点。这个实现稍微复杂一点。

首先制作一节黑色的蛇的身体，还是蛇的角色，造型2.

其次，设置一个黑色的全局变量。默认是0.如果被吃掉了，设置为1.

在克隆体启动时，读取合格黑色的全局变量标识，如果是0，则为普通蛇身，造型为2.如果是1，即刚吃掉一只蝙蝠，需要变成黑色，造型3。切换成造型3以后，需要将黑色的变量设置为0，否则后面就全是黑色的了。

总结测试：

最后效果图这个图。

还可以改进，例如多条蛇、多只蝙蝠，蝙蝠会飞翔，不能碰到边缘等。蛇的身体，在转弯的时候，覆盖的还不大好。

目标：两个球随机运动，然后将两个球之间画一条直线，球在不停运动同时，画笔就不停的画线。同时可以改变球的速度，改变画笔的颜色。

效果图：两个球在舞台上不停运动。碰到边缘就反弹。

三个角色：两个球，球1和球2，还以一支笔，也是一个球，不过大小设置为最小。

画面效果图。

1. 篮球1代码

1） 开始的时候随机选择一个方向；

2） 创建一个全局变量，速度1，将速度1设置为4-21的随机数。

3）重复执行，移动XXX步，因此，改变速度1就可以改变球的运动速度。

4）碰到边缘就反弹。

2. 篮球2代码

和篮球1几乎一样，只是速度为速度2的变量。

1） 开始的时候随机选择一个方向；

2） 创建一个全局变量，速度2，将速度2设置为4-21的随机数。

3）重复执行，移动XXX步，因此，改变速度2就可以改变球的运动速度。

4）碰到边缘就反弹。

3. 画笔代码1：画线

1）这是3.0的代码，因此要将画笔从“拓展”中放出来；

2）擦除所有，否则会留下上次运行的结果。

3）移动到篮球1；落笔；

4）移动到篮球2；落笔；

5）将画笔的颜色增加1.

然后不停循环。

4. 画笔代码2：擦除重置

随着程序的运行，屏幕会被画满。因此经过一段时间后要重新擦除。

1）等待15秒；全部擦除；

2）将速度1设置为某个随机数；

3）将速度2设置为某个随机数。

这样下一次运行时，速度就会不一样。

目标：给出一组数据，做出柱状图图表，柱子的高矮代表数值的大小，这是最基本的一种数据图形展示方式了。

共享位置：https://scratch.mit.edu/projects/276761007/

Scratch 3.0,可以直接在手机上打开这个地址。

效果图：显示了三个变量，一个列表，小猫咪画出了柱状图，还有一根X轴的轴线。

角色:只有一个，默认的小猫，但大小设置为小一些。

手稿：制作前，先讲清楚目标和原理。

1）高度：高度是变化的，需要从列表中取出来

2）宽度：柱状图的宽度，固定的，如果宽度是20，那么就是画20条线，这20条线连接在一起，就成一个柱子。

3）间隔：柱子之间有固定的宽度。

因此设置三个变量，这样后续方便调整。

数据：使用列表，直接将四个数据填写在列表中。

数据列表，数值填充。

制作步骤：

一、初始化

初始化高度、宽度、间隔，同时填充列表数据。另外，需要设置小猫的起点位置。

二、画X轴坐标线

1）间隔为20；

2）全部擦除；否则下次运行时，还存在这些图画。

3）移动起点位置；

4）落笔，就是下笔，如果有运动，就会画线。

5）直接将X坐标增加到330，就是画了一根横线了。如果需要高度横线，可以这样画出来。

6）到最右边以后，画图结束，就需要抬笔。

7) 回到原点，准备画柱状图。

三、画第一根柱子

1）宽度50；

2）高度：从列表中取第一项；

3）因为里面的高度数据太小，所以需要放大，这里x16，放大16倍。

4）重复执行宽度次。每个柱子都是一条一条线画出来的。这个概念要理解，画笔默认宽度是1.

5）画每根线都是从底部-140开始，然后往上走。

6）落笔；将Y增加高度值；抬笔。就画了一根线了。

7）x坐标加一，继续画，即画（宽度）次数。

这样一根柱状图就画出来了。 然后我们可以使用循环。但为了贪图方便，小孩选择了，复制四份代码。那就先使用四份代码，这样代码会很长，重复的代码会多。如果要修改里面的某行代码，那很可能要修改四个地方。这些代码是可以服用的。借此，后面转成循环方式，让他体会一下区别。如果要画20根柱状图，那就复制20份吗？也许，我应该让他复制20分代码，然后一个一个核对顺序，然后让他体会一下重用代码的好处。

当中还出现某两根线之间没有间隔，贴在一起的情况，那怎么解决呢，为什么呢。

我们的y坐标线其实是后来加上去的。画完坐标，如果不回位，柱状图就在右边挨着画了。

改进：组成循环，Block。、

**第十六课 图表**

一、目的:

上次做了一个柱状图，了解了柱状图的特点，可以将数字转化成图，可以发现规律和趋势。终于有了时间，周末和二年级小孩一起做了一个饼图，饼图能够发现几个数据的占比，例如哪块占的最多，哪块最小。柱状图和饼图是最近的数据可视化图形格式，重要性再怎么强调都不为过。

二、难度：

理解列表数据的存取，多重循环对七八岁的小孩来说还有一定的困难。还是需要在草稿纸上举例说明，将每次循环的变量都记录下来，然后如何使用变量获取列表某行的数值。多重循环和列表操作是重点，需要多加练习。我打算做一些PPT，让小孩深刻理解列表，熟练存取列表中的数据。

三、结果：

列表中有几个数据，然后画出饼图。

一个一个扇区画。

最后就画出这样的饼图，当然还没有百分比图例。

共享在这里：https://scratch.mit.edu/projects/276709292

四、角色：

一个角色即可，只有一根线，注意可以设置为矩形，中心点为一端中央。

这是一个矩形，可以调整高度和宽度，中心点是一端。

五、定义变量：

最重要的是定义变量。

这里列表，里面填充了8个数字。

主要是获取列表的第XXX项。

两个变量：所有这些值的加总-总和，还有一个计数器，用于获取当前数据。然后每一扇区占用几度（总共360°）是计算出来的。

六、程序：

1）读取列表项目数，里面有几个记录，就循环几次。

2）循环的时候，将数据加入到“总数”中，然后计数器加1.

1）重置计数器；还是循环列表数据项目数的次数，即每个数据都是一个扇区。

2）可以先活着之后更改颜色，使用特效即可，随机颜色。

3）关键是计算出每一个数据对应的扇区度数。总共是360°，即将一个圆分解成360°。每一个扇区就是这根线旋转对应的度数。

度数为数据/总数X360°。每次画线的时候如果是1°，就x360。考虑到这根线不够厚，图形太大时，周边会有间隙，因此也设置1440，每次右转0.25度。

4）每个扇区，循环，每一个循环就盖一个图章，即复制这一根线；复制完毕后，右转0.25度。相当于一个时钟转一圈，转的过程中留下足迹（盖章）。

盖章和克隆很类似，只是盖章只是画图复制一份，不会再被激活。而克隆体，是有克隆体启动脚本，每个克隆体都是一个对象。

360度的话，每次转1度，周边有间隙。还能看见“衍射”图。

这是720份，每份0.5度，也有衍射图。

调整角色图像大象，宽一点，窄一点，也能有此效果。

小一点的图，就比较近似圆形了。

功能：二年级下学期第一单元就是数据收集和统计，本想让小孩记录一个月的天气温度变化情况，无奈时间太长。还是相信科学，直接在网上搜索，找到北京12月份的平均温度数据。我们的目标，就是将这12个月的温度曲线画出来。这样能够看到每个月的温度变化趋势，能够看到哪些月份低温、哪些月份高温。

体会：

如果一个学生能够熟练掌握列表的存取操作、数字计算和逻辑比较，那么基本上就可以说这个人具有编程能力了。数据存取、数据计算和处理、逻辑比较，就是程序的核心灵魂。对于列表和数字计算、逻辑比较，还是需要通过诸多训练，掌握一定的数学知识才能完成的。

因此，别急，别急。别想听一两个小时课程或者一刻钟半小时的网课，就能把Scratch编程学好，那是不可能的。当然如果小孩具有初高中水平，或高年级小学生，理解起来可能会容易一些，大人理解起来可能很简单。但对于一二年级的学生来说，理解这些还是有一些困难的。在学习过程中，需要尽可能的拿出A4纸，学会数学计算，将相对位置、计算坐标写下来，然后转成变量，转成计算机代码程序。

本想本周一会做好的程序，花了不少时间，还能完全消化完毕，还得继续练习。

一、效果图：

小孩总是闹着要做游戏，当然不能每次都做游戏。我先在草稿上画了坐标，然后让小孩把这12个月的温度在上面标出来。到计算机上，先画横坐标，这还是可以的。设置起点，y轴不变，到最右边。结果说要找一个人打球，球打出来以后，球走过的轨迹，就画线。

那好吧，他加入了棒球的角色，还改变造型，看似挥动球杆了。还有模有样的。

二、角色：

球的角色，经过的轨迹，就画线。棒球手纯粹为了好玩。

三、列表：

一个温度列表，记录了12个月的温度数据。

四、变量：

为了能够更灵活一些，设置了月份数，起始点x，起始点y，间隔。

P1，P2是相邻的两个点，即1月份为P1，二月份为P2，然后二月份为P1，则三月份为P2，以此类推。其实这四个变量可以不要，只是为了更好的理解这些点是怎么计算出来的。

五、开始：

一堆初始化。

初始点和间隔是关键。

六、画横坐标

起始点在左下方一点，因为低温少，高温多。然后y轴不变，x轴到最右边的某个地方。落笔，抬笔，当中使用了滑行功能，这样能够看见动画效果。画完以后，盖个章，在尾巴处盖个章。

坐标还没有刻度线，下一个版本中再加入刻度线吧。刻度线其实也可以做，在适当的地方盖章即可。不过要计算好位置。

七、画温度线：

这是最为关键的，起始点相当于零点，然后1厘米的地方是一月份的温度，2厘米的地方是二月份的温度。

因此x轴是起始点x轴+间隔X月份。月份从1-12，这样x轴坐标就确定出来了。

然后是y轴，y轴要取当前月份的温度值，但温度比较小，放大8倍，然后要加上起始的y。

从1循环到12月份，我们就可以找到每个月份的x坐标和y坐标。这个最好在纸上写下来。

由于我们是从1月份画线到2月份，0-1是不画的。因此我们重复循环11次即可，从2月开始。

这样要计算出1月份的点（当前月份-1），然后是2月份的点（当前月份），然后从P1到P2滑行。这里依然使用了滑行运动操作，这样能够看到画线的过程。

这个程序能够自己做出来了，基本上就可以说会写程序了。

总图。

一、最终效果：

开始后，如果点击“选择乐器”，系统弹出草莓，每一个草莓就是一种乐器。

选中乐器以后,这些草莓消失。然后可以用这种乐器演奏了。

二、造型制作：

制作琴键造型，白色，灰色。画框，复制即可。

制作乐器类型，写字就行。

8个琴键，一个三明治选择按钮，三种乐器类型。

三、加入音乐扩展

需要加入音乐扩展，才可以使用音乐指令。

扩展，很漂亮 。

四、定义变量：

一个选择的乐器编号，一个记录乐器名称。

五、琴键代码：

点击后演奏对应的琴键。点击后更换造型。

这里我们学习 一个功能：复制琴键前先写好代码，这样代码能够跟着被复制过去。否则会麻烦一些。

六、选择乐器：

两行代码，很简单，点击后做一个广播。

七、乐器选择：

默认是隐藏的。用户点击乐器选择按钮时，才显示（用广播）。单击后，设置参数，并且隐藏起来。

因为我们要将所有的三个（或更多）乐器按钮都隐藏起来，所以使用了一个已选中的广播。如果要做21个乐器类型，这样做就会太麻烦，可以使用克隆功能，克隆出21种乐器出来，不过那样的代码更难一些，属于Scratch代码中级水平了。

一、最终效果：

田鼠(松鼠）出现在草地上，点击后可以发出“喵”的一声，然后消失，得分，过一段时间（2秒），地鼠重新出现。

二、角色和背景：

加入一个和田地相关的背景，加入一个动物。图片库里面没有地鼠，就使用类似的松鼠吧。

三、程序：

1. 先定义一个变量，可以将默认的“我的变量”修改成“分数”。

2. 启动时，先出现在中央，然后进入循环，每2秒重新隐藏、显示。

这里使用了移动的指令，能够限制地鼠在一定范围内出现，否则就会出现在屏幕边缘了。

3. 当被点击时，发声，增加积分，然后隐藏。因为“启动”的主程序还在循环，等待一段时间以后，地鼠又会出现。