

第三章 创建一个彩色方块

传说上古时女娲造人,是得益于众神的帮助(黄帝生阴阳、上骈生耳目、桑林生臂手),历经 70 次的尝试与改变,最终创造了人类。女娲按照自己的模样造出了人类,同时也赋予了人类喜欢创造的天性。然而时过境迁,在物质极大丰富的时代,消费带来的欢愉取代了创造的快乐,我们的创造力一天天退化。

为了挽救这即将泯灭的创造力,让我们从创造方块开始我们的游戏开发之旅。

第一节 画布上的静态方块

画布(Canvas)组件在组件面板的“绘图动画”分组中。如果把游戏比作一场足球比赛,那么画布就是足球场地,而运动员——那些方块,就要靠我们自己去创造。我们要在画布上画出这些颜色及形状各异的组块,如图 3-1 所示,但首先要画出右上角的那个灰色方块。



图 3-1 颜色及形状各异的组块

一、添加画布组件

如图 3-2 所示,在设计视图中,从组件面板的“绘图动画”分组中将画布组件拖放到工作区的预览窗口中。

二、为画布组件设置属性

如图 3-3 所示,当画布被拖放到预览窗口之后,画布自动停留在屏幕的左上角,这时组件列表中的 Screen1 下方多了一个“画布 1”,而属性面板中也显示了“画布 1”的属性,这些属性的值被称为默认值,也可以按照自己的需要进行重新设置。这里要对画布的宽、高以及画笔的颜色和线宽进行设置,如表 3-1 所示。设置完成后如图 3-4 所示。



图 3-2 在设计视图中向用户界面预览区拖入一个画布组件



图 3-3 在设计视图对画布的属性进行设置

表 3-1 设置画布的属性值

属性	宽 度	高 度	画笔颜色	线 宽
值	300 像素	400 像素	浅灰	24 像素
说明	水平方向可以容纳 12 个 25×25 的方块	垂直方向可以容纳 16 个 25×25 的方块		方块为 24×24 像素，方块之间间隔 1 像素



图 3-4 画布的属性设置完成

三、编写代码

在画布左上角绘制一个 24×24 的方块。

(1) 屏幕初始化事件。点击“编程”按钮切换到编程视图,在“代码块”分组中点击 Screen1,打开 Screen1 组件的代码块抽屉,选中“当 Screen1 初始化时”块,如图 3-5 所示。



图 3-5 选择 Screen1 初始化事件

(2) 在画布上画线。在“代码块”分组中点击“画布 1”,打开画布 1 的代码块抽屉,选中“让画布 1 画线”块,如图 3-6 所示。





图 3-6 让画布执行画线功能

(3) 代码的组合。将“画线”块拖放到“初始化”块中,注意当前者靠近后者时,后者的内框突起处显示黄色线条,这说明两个代码块可以组合在一起,此时松开鼠标,两个块将自动拼装到一起,并发出清脆的“咔嚓”声,如图 3-7 所示。

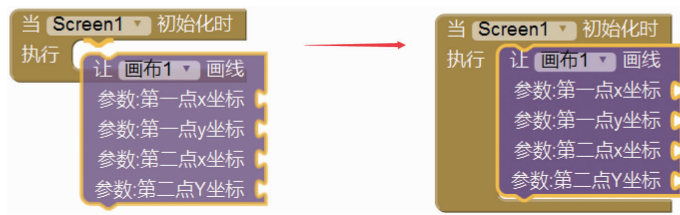


图 3-7 代码块的拼装

(4) 添加参数。点击“代码块”分组中的“数学”类,将打开数学类代码抽屉,第一个块就是数字块“0”,拖曳“0”并将其放置在“画线”块的“第一点 x 坐标”的插槽中,如图 3-8 所示。执行相同的操作为“第一点 y 坐标”“第二点 x 坐标”“第二点 y 坐标”添加数字,并将数字 0 改为需要的值,如图 3-9 所示,第一点坐标为(0,0),第二点坐标为(24,0)。

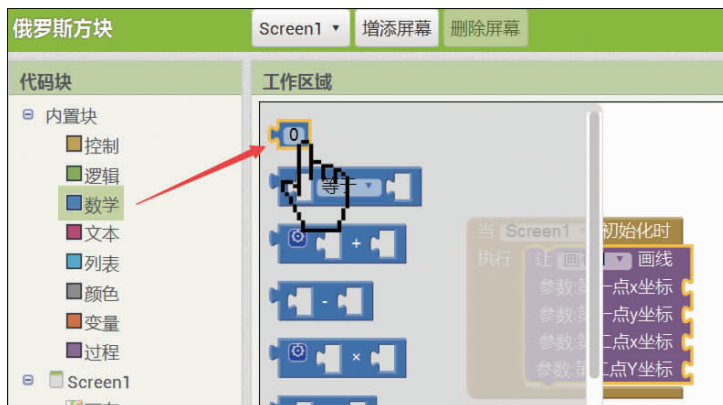


图 3-8 在数学抽屉中选中数字块“0”

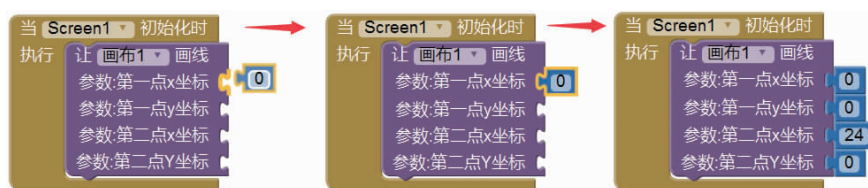


图 3-9 为“画线”块添加所需的数字参数

(5) 程序测试。此时第二章中讲到的知识该派上用场了。按照第二章第二节提示的操作方法,在手机上进行程序的测试。测试结果如图 3-10 所示。

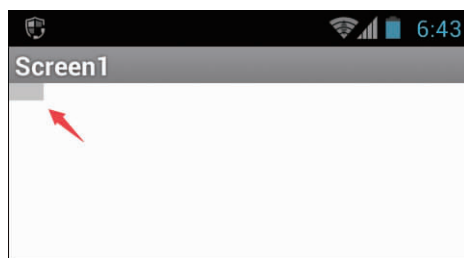


图 3-10 只有半个方块

(6) 修正。将第一点坐标改为(0,13),第二点坐标改为(24,13),并在设计视图将画笔颜色改为红色,再来看看结果。如图 3-11 所示,屏幕上出现了完整的红色方块,此时会发现,在红色方块与屏幕顶端之间留有空隙。

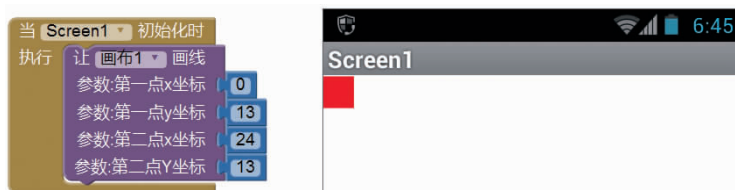


图 3-11 将坐标的 y 值改为 13 之后终于画出了完整的方块

(7) 再修正。将第一点坐标改为(0,12),第二点坐标改为(24,12),并将画笔颜色改为绿色,再来看看结果。如图 3-12 所示,绿色方块紧贴着屏幕的左上角,顶端的空隙消失了。

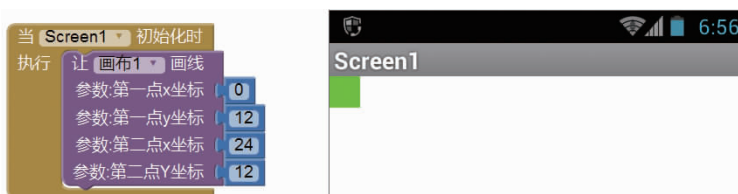


图 3-12 将坐标的 y 值改为 12 后方块与屏幕顶端之间的空隙消失了





第二节 对程序的解释

经过以上 7 个步骤,在画布上完成了画一个方块的任务,这个过程看似简单,却包含了一种很重要的数学知识——坐标,这是绝大多数游戏中都会用到的知识,因此从这里开始讲解。

一、坐标

先来看图 3-13,这是一个围棋棋谱,记录了一盘棋的对弈过程,棋子上的数字表示黑白双方落子的顺序。在比赛的同时,选手们还要记录棋谱,他们是如何记录这盘棋的呢?有这样一种记谱方法:①(Q, 4);②(D, 3);③(D, 17);④(P, 16),等等,前面圆圈里的数字表示落子的顺序,后面括号里的字母表示水平方向的位置,数字表示垂直方向的位置,这是运用坐标描述一个点在平面上位置的典型例子。想必读者已经对坐标的概念有了一点感性认识,下面解释几个数学概念。

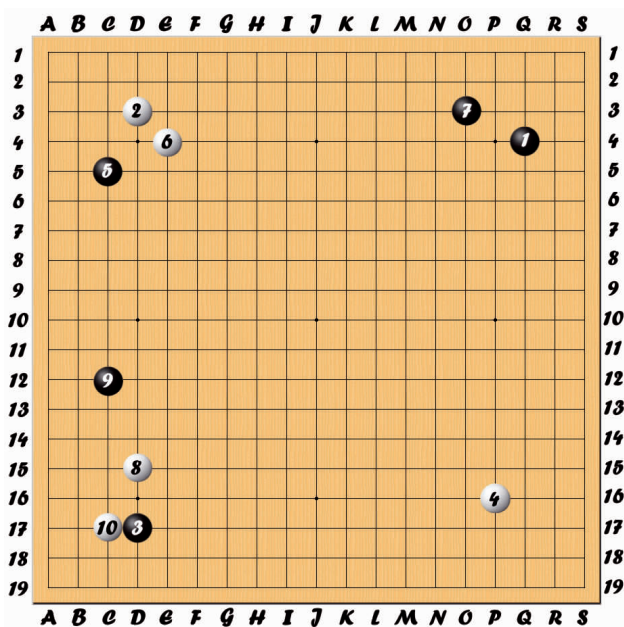


图 3-13 用字母和数字来表示棋子的位置

(1) 平面。数学中的平面与生活中的平面有相似之处,可以把它想象成一个桌面、一块操场或一张棋盘,不同之处是生活中的平面是有边界的,而数学中的平面是人们假想出来的,或者说是一个抽象的概念,它没有边界,无限延展,面积无限大。

(2) 坐标轴。这是一条直线,人为规定了原点(值为 0 的点)、正方向及标尺。与现实中的直线相比,数学上的直线同样是一个抽象的概念,它可以无限延长。坐标轴上的标尺用来描述某种事物的数量,如温度、海拔高度、资产数量等,可以是 0,也可以是大于 0(正)或小于 0(负)的数。坐标轴的正方向是数量增加的方向。





(3) 直角坐标系。当两个或 3 个坐标轴的原点重合,且彼此之间的交叉角度为直角时,这些坐标轴共同组成了一个直角坐标系。两个坐标轴构成平面直角坐标系,也称为二维直角坐标系,它构成了一个二维平面,用于标注平面上点的位置;3 个坐标轴构成三维直角坐标系,它构成了一个三维空间,用于标注三维空间中点的位置。在平面直角坐标系中,通常将水平坐标轴称为 X 轴,垂直坐标轴称为 Y 轴,如图 3-14 所示。

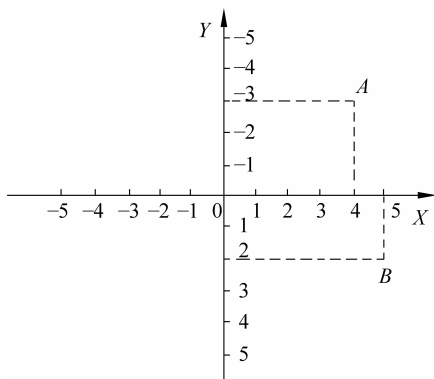


图 3-14 用两个数字表示一个点在平面上的位置

(4) 坐标。如图 3-14 所示,在平面直角坐标系中有 A 、 B 两点,如何表示它们在平面上的位置呢? 方法是从一点分别向 X 轴和 Y 轴画垂线,垂线与坐标轴的交点就是它们在该轴上的坐标,因此,这两点的坐标分别为 $A(4, -3)$ 及 $B(5, 2)$,至于括号内两个数字的顺序,这是一种人为的约定, X 轴坐标在前, Y 轴坐标在后。需要特别说明的是,坐标轴的正方向也是人为规定的,在数学课本上, Y 轴的正方向向上,而这里向下,这是因为在即将讲解的画布组件中, Y 轴的正方向就是向下的,为了保持知识与其运用的一致性,这里将 Y 轴的正方向规定为向下。

二、画布组件的画笔功能

画布组件的名称形象地说明了组件的功能,即可以在其上绘制图形。为了绘制需要的图形,可以随时对画笔的颜色和宽度进行设置和修改。画布组件具有画点、画线、画圆的功能,这里使用了画线的功能来画一个方块,具体的操作步骤在上一节中已经实践过了,这里主要是解释为什么要这么做。

(1) 画布的坐标系。

- ① 原点: 即坐标为 $(0, 0)$ 的点,在画布的左上角。
- ② 正方向: X 轴的正方向向右, Y 轴的正方向向下。
- ③ 标尺单位: 像素。
- ④ 坐标系的可见范围: X 轴从 0 至 300(画布的宽度), Y 轴从 0 至 400(画布的高度)。

(2) 方块的尺寸。在设计游戏时,要考虑一些简单的算术问题,如在 300 宽的画布上可以摆放多少个完整的方块。为了使方块的尺寸为整数,可以让它的宽度为 20、25、30 或 50,但考虑到高度为 400,为了保证在垂直方向上能够摆放整数个方块,因此 30 这个宽度





不合适。经过在测试手机的屏幕上实地观察,发现 25 这个宽度效果比较好,因此决定让方块的宽度和高度均为 25 个像素。

(3) 画笔的宽度。既然方块的宽度(高度)为 25,那么为什么将画笔的线宽设置为 24 呢?而且画线时,第二点的 X 坐标也是 24,这相当于画了一个宽度和高度均为 24 的方块。这个问题的答案在图 3-15 中。如果不留出一个像素的空隙,这个方块就会与周围的方块连接在一起,无法分辨单个方块的位置。

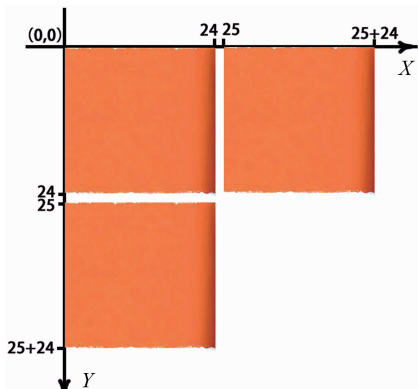


图 3-15 对画笔宽度的考虑

(4) Y 轴坐标与方块的位置。由于是在水平方向上画线,因此从起点到终点的 X 轴坐标增加了 24,而 Y 轴坐标保持不变。比较图 3-9 ~ 图 3-12 可以发现,Y 点的坐标影响了方块在垂直方向上的位置:当 $Y=0$ 时,只画出半个方块;当 $Y=13$ 或 12 时,画出了完整的方块,但当 $Y=13$ 时方块与屏幕顶端之间有一条空隙,这说明画笔的起点在画笔的中间,而非画笔的上端,如图 3-16 所示。

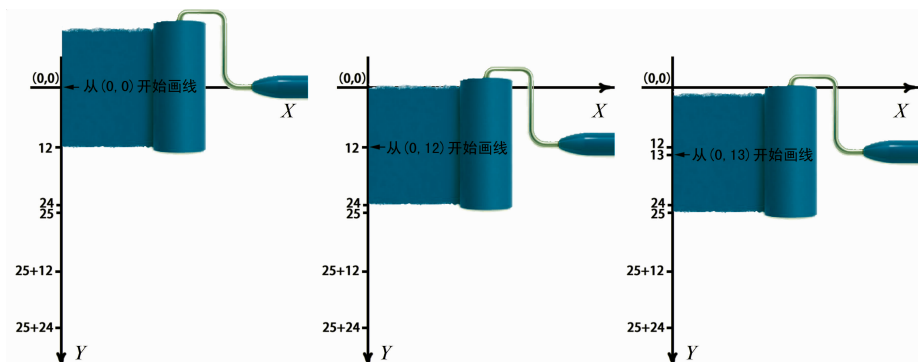


图 3-16 画笔在水平方向画线时画笔的 Y 坐标对画线位置的影响

第三节 逐步了解开发工具

App Inventor 是一个可视化、可拖曳的编程工具,用于构建安卓平台上的移动应用。利用基于浏览器的图形化用户界面生成器,可以设计应用的用户界面(外观),然后像摆弄拼图玩具一样,将“块语言”拼在一起定义应用的行为。这里所说的“块语言”就是在编程视图中所使用的代码块,本节简单介绍代码块的分类、数据及其处理方法。

一、代码块的分类

在编程视图中看到了代码块的分组,分组与分类是两个概念。我们会把周围的人划分为家人和外人,这是按照血缘关系来划分的,是对人群的一种分组;也会对人群有另一





种划分方法,如老师、学生、工人、农民等,这是按照他们的职责(或功能)来划分的,是对人群的分类。对于代码块来说,分组的原则是代码块的普遍可用性:内置块可以在任何项目中使用,块的种类和数量不会因项目的不同而不同;而与组件相关的块则不然,只有在项目中添加了某个组件,才能使用与该组件有关的代码块,不同的组件对应不同的代码块集合。再来说说代码块的分类,分类的原则是代码块的职责,有控制类、功能类、读写类及数据处理类,分别对应于代码块的不同颜色。具体分类见表 3-2。

表 3-2 代码块的分类

分组	类型	子类型	颜色	职责(功能)	举 例
内置	①控制类	执行顺序	黄	控制程序的执行方式:顺序、分支、循环	
组件相关	①控制类	事件响应	黄	对组件的事件进行响应	
自定义	②功能类	自定义过程	紫	用户自行创建的一段程序,可以被反复调用	
组件相关	②功能类	内置过程	紫	组件本身具备的功能	
内置	③读写类	变量读写	橙	变量的声明、变量值的读取及改写	
组件相关	③读写类	属性读写	绿	组件相关属性值的读取及改写	
内置	④数据处理类	文字	玫瑰红	文字类型数据的生成及处理	
内置	④数据处理类	数字	蓝	数字类型数据的生成及处理	
内置	④数据处理类	颜色	浅灰	颜色类型数据的生成及处理	
内置	④数据处理类	逻辑	浅绿	逻辑类型数据的生成及处理	
内置	④数据处理类	列表	青	列表类型数据的生成及处理	





二、数据及其类型

在表 3-2 中,有几个标记为粗体的词,这些是编写程序时经常会遇到的概念,就像学生们离不开书本和文具一样,编程离不开这些基本要素。本章只介绍数据及其处理,余下的概念将在后续章节中逐一介绍。

就像我们烧菜做饭,是要加工某些蔬菜和粮食一样,编写程序就是要加工处理各种类型的数据。在 App Inventor 中有以下几种类型的数据。

- (1) 数字,包括正整数、0、负整数、正小数及负小数。
- (2) 文字,包括各种人类自然语言的文字及符号。
- (3) 颜色,包括 1600 多万种颜色($256 \times 256 \times 256$),可以用数字的形式来合成颜色。
- (4) 逻辑,包括两个值——真与假(true 与 false)。
- (5) 列表,是任意多个以上各种类型数据的集合。

三、数据处理方法

编写程序就是要对以上各种类型的数据进行加工,目的是获得所需要的结果,这个过程就是数据处理。不同类型的数据采用不同的加工处理方法。

(1) 数字。包括各种数学运算、小数取整、求各种函数的值,如图 3-17 所示。游戏中最常用的是求随机数,如求 1~19 的随机整数,每次运算都会随机生成一个 1~19 的整数。对于小学生来说,有些运算方法是陌生的,如三角函数(正弦、余弦等),但是这些不妨碍学习编写程序,在需要时再做详细的解说。

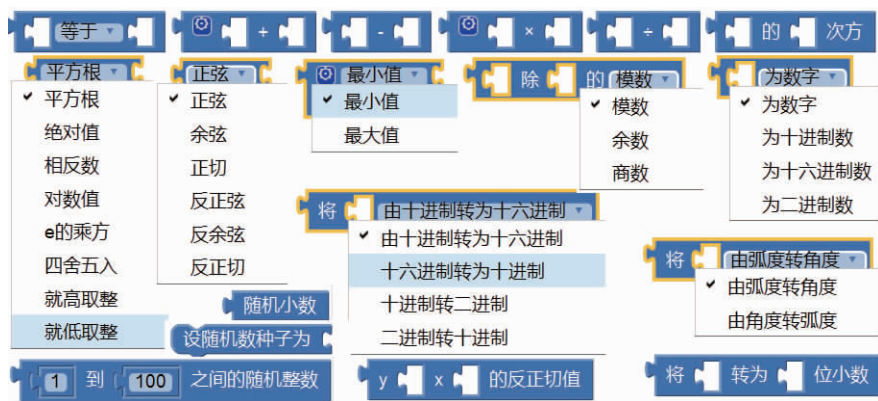


图 3-17 数字类数据的处理方法

(2) 文字。在计算机以外的世界里,极少有对文字的处理,但程序编写过程中,文字处理是一种很常见的数据处理方式(见图 3-18),如用分隔符“,”对文本“苹果,香蕉,菠萝”进行分解,这项数据处理的结果是生成一个列表,如图 3-19 所示。

(3) 颜色。在计算机中,颜色都是用数字来表示的。App Inventor 中的颜色用 3 个十进制的整数来表示,被称为 RGB 颜色模型,或红、绿、蓝颜色模型。R 代表红色,G 代表绿色,B 代表蓝色。每种颜色的饱和度都有 256 个级别,用十进制数 0~255 表示,0 表示

