

PEARSON

移动开发经典丛书

MIT App Inventor

完全上手

 不会编程也能“乐高式”构建Android应用的编程神器

Learning MIT App Inventor:
A Hands-On Guide to Building Your Own Android Apps

[美] Derek Walter
Mark Sherman
靳晓辉

著
译

PEARSON

清华大学出版社

第 1 章

编程入门

从本质上说，编程就是告诉计算机要做什么。现代的程序员正在更多地集中精力集中在为我们普遍持有的便携式计算机——智能手机——构建应用。移动设备的普及引爆了移动App的流行热潮。

理解编程过程可以为使用MIT App Inventor构建优秀Android App所需的思考方式打下基础。虽然在App Inventor中不会使用基于文本的编程，但需要用到相同类型的逻辑思维、解决问题的方法和创造力，这些都是专业的应用开发者在编写程序时所要使用的。

创建App会涉及编写计算机代码。然而学习一门编程语言是一个复杂难懂的过程，这通常是为那些计划成为专业程序员的人而准备的。MIT App Inventor的强项在于可视化的块就是代码。你可以通过拼接一些拼图块和使用拖放工具来构建Android App并实现其功能。

不过，不要将这误认为是“非主流”或低级版本的编程。在App Inventor中创建程序需要具有与在传统编程中相同类型的逻辑思维、推理方式和创造力。学会以此种方式思考并知晓在智能手机上所见情形背后发生的事情将使你能够更高效地创建伟大的应用。如果打算深入研究计算机科学并掌握一门或更多门编程语言的话，这也将会给你打下一个极好的基础。

计算机式的思维需要一系列不同的基础技能。优秀的编程要求能够创建出计算机和应用会遵从的有逻辑的、步进式的命令。拼接指令并赋予应用处理诸多数据和事件的能力有助于让用户了解到他们正在使用的是真正的智能机器。

1.1 操作系统

要在某种操作系统(OS)上构建健壮的应用，至少要具备一些关于该操作系统的大体知识，这会对你很有帮助。久而久之，这将让你成为一位更加优秀的程序员并增强你构建伟大应用的能力。虽然本书的关注点是Google的移动操作系统——Android，但我们首先会探讨所有操作系统都必备的通用部分。

简单来说，操作系统是管理计算机设备的软件。它处理全部的核心功能，例如执行应用、运行任务并控制所有的外围设备。

如今的智能手机都称得上是计算机——它们承载了以前只有更大体积的机器才能够具有的全部特性和组件。它们有持续的网络连接并且能够在程序发出请求时获取各种类型的数据，例如通过GPS传感器使用你的位置来确定手机的倾斜程度。操作系统由多个独立的进程构成。使此概念形象化的一种好方法是把所有这些进程看成一个更大层次结构中的各个层级。

操作系统的最底层是内核，它提供了对设备硬件最基本的控制。它管理内存访问并决定哪些程序可以访问各种各样的硬件资源，例如拨号器或照相机。

即使这对于你来说是个新概念，你也能够开始发现Android操作系统在设备和你的App之间提供了一层接口。这很重要，因为它使得你的App可以在各种设备上工作。作为程序员，你只需要编写能够在Android操作系统上工作的App，不用担心，Android已经为你省去了不得不处理不同可用硬件选项的麻烦(参见图1-1)。

一系列的库和应用程序框架构成了操作系统的下一个层级，它们负责管理各种数据集。操作系统在应用和设备的硬件之间创建了一层接口，以确保高效的信息共享和内存管理(参见图1-2)。尽管现在还不必关心，但掌握操作系统管理其内存资源的方法终究是很重要的。

决定操作系统表现优劣的一个重要因素是它的内存管理。所有的设备都只有有限数量的内存，这些内存必须在各种资源(例如，操作系统的关键功能和应用)之间进行分配。当编写更高级的程序时，关键的一点就是确保App不会使用过量的内存。



图 1-1 Android 设置中显示了关键的操作系统信息，例如 Android 的版本



图 1-2 操作系统管理着由各种应用运行的后台进程。Android 允许你监控并采取行动来保护系统资源

传统的程序以线性方式工作，在每个时间点仅执行一个动作。当程序需要一条来自用户

的信息时，整个程序会停止，直到用户输入该信息，接着程序继续执行。然而，现今的程序都是多任务的。举例来说，在使用智能手机时，用户可能正在地图上查找方向。设备接下来会通过GPS访问某个位置、拉取地图数据并执行各种其他的后台任务，例如发送电子邮件或者接收即时通知(参见图1-3)。

因此，现代编程讲求最大程度地使用这些已联网的设备并巧妙地分配智能手机的资源。移动设备持续地从多个源访问数据，这使得App具有更强的交互性。例如，Google Now从你的Google账号拉取数据来提供关于本地活动、天气状况更新和感兴趣的新闻条目的建议(参见图1-4)。

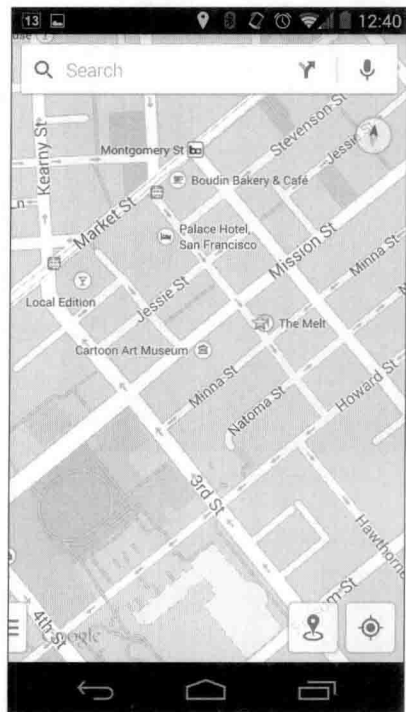


图 1-3 Google 地图在展现某个区域时使用位置作为上下文

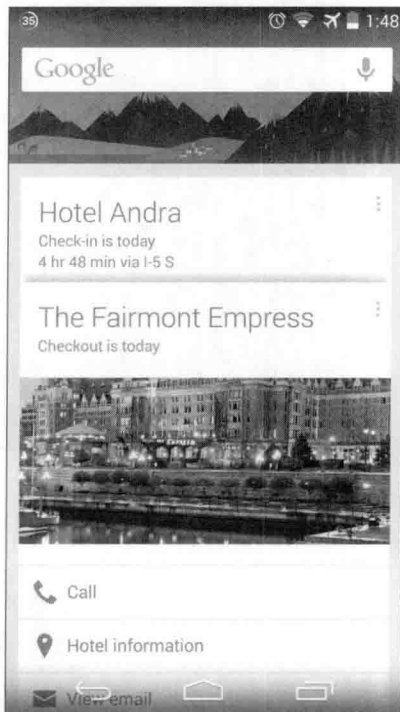


图 1-4 Google Now 使用多种资源来拉取上下文数据。现代操作系统为了满足用户的需求都在不停地处理数据

用户界面

用户界面(UI)是一款操作系统的外观。好的界面会把用户放在第一位，让他们容易发现给出命令和访问App的方法。对于程序员来说，了解界面的工作原理以及它对应用设计的影响是非常有用的。

Android提供了一个易于操作的界面，包括可以展示App和Widget(小组件)的桌面(参见图1-5)。一同展现的Google搜索栏使得搜索和访问Google Now(图1-4所示的上下文信息工具)都很容易。

由于Android允许大量地进行定制，因此不同设备的用户界面有时会大相径庭。例如，大多数主流的Android设备制造商都会创建一套定制的用户界面(有时称为“皮肤”)。因此，尽

管它们都运行着Android，但三星设备上操作系统的图标和整体界面外观可能会与LG或HTC所生产设备上的不相同。

此外，Android 允许你使用自定义的桌面启动器(参见图 1-6)。



图 1-5 Android 用户界面



图 1-6 Yahoo! Aviate 自定义启动器

某些Android设备甚至支持用户在不同的启动器之间切换(参见图1-7)。

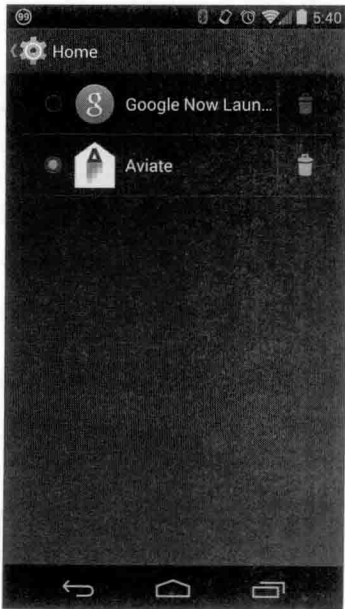


图 1-7 Android 的设置允许用户选择自定义的启动器

幸运的是,使用App Inventor构建的App应该很容易运行在多种设备之上。本书后面会讨论如何在特定的设备上测试App,赋予你解决任何潜在问题的能力。

1.2 Android 的优势

Android有一些特别的优势,令其成为一个编写程序的理想操作系统。由于它是世界上最流行的移动操作系统,因此App开发人员有着让他们的应用得到大量受众的机会。

可以从Google Play商店下载Android App(参见图1-8)。基于网页的店面和Android设备上的Play Store App能够确保你的应用得到广泛展现。

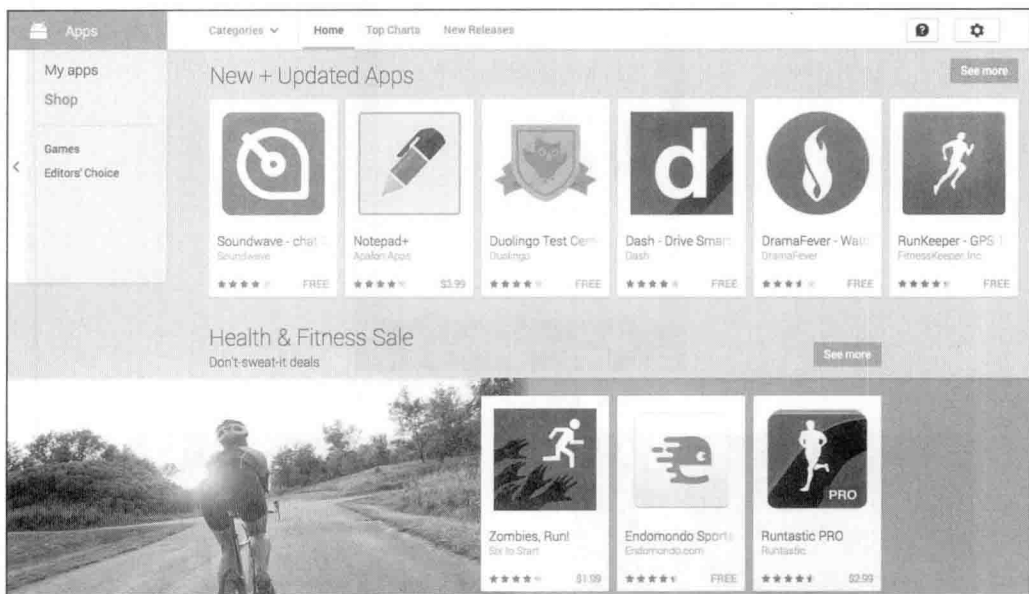


图 1-8 Google Play 商店中集合了可供用户下载到自己设备上的 Android App

Google还在Android中内置了许多用于帮助开发者的服务,例如App使用情况统计以及不经过审核流程就能上传App的功能。

1.2.1 扩展 App 的功能

与那些为更加封闭的系统(例如Apple的iOS)编写的App相比,Android App可以访问操作系统的更多组成部分(当然是在用户许可的条件下),使得它们拥有额外的权限来访问源自于其他App的数据。Android App还能够很好地彼此交互,这为开发者提供了多种选择。举例来说,Android的Intent特性允许一个App访问另一个App的共享功能——例如,使用内置照相机App还是你自己的图片选择App(参见图1-9)。

这种灵活性延伸到了用户,允许他们选择用于某些操作的默认应用。例如,用户可以在任意浏览器中或者在完成该功能的App中打开一个网页链接。这对于开发者是个好消息,因为这将你的App与其他App,甚至是Android设备自身提供的那些App都放在了同一起跑线上(参见图1-10)。

此外，可以将App的功能扩展到桌面Widget(如图1-11所示)，它可以显示实时的、与上下文关联的信息(目前还无法使用App Inventor创建Widget)。然而，考虑如何将App扩展为Widget只是开始培养App开发者思维的一部分，随着Android的发展，还需要考虑可穿戴设备和电视。

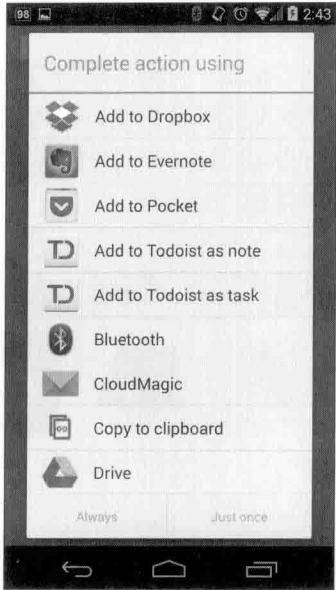


图 1-9 Android App 可以轻松地彼此共享信息

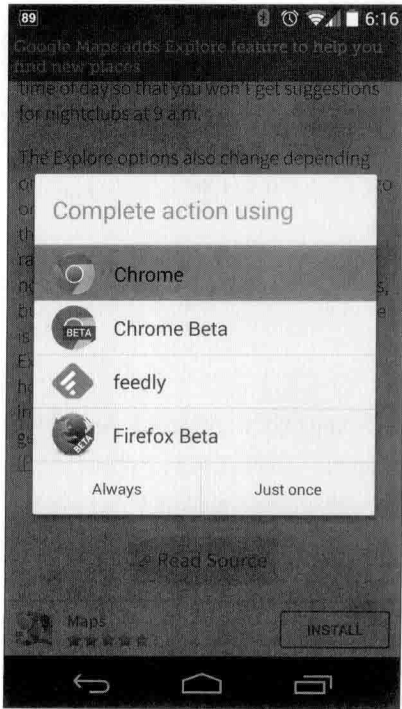


图 1-10 Android 的 Intent 特性在将何种应用绑定到特定动作方面给予了用户更多的选择



图 1-11 Widget 是将 App 的实用功能扩展至桌面的另一种方式

所有这些都在构建App时开放了许多可能性。短期来说，你主要关注的仅仅是构建新的App，但后期的目标将是最大化对Android的使用，创建出构成用户体验核心部分的应用。

1.2.2 Google 服务

随着Android的每一次迭代，Google愈加深度植入了其自身的服务。当进入一台Android设备时，用户登录到他们的Google账号，这会使他们的设备与其Gmail、Drive、Chrome、地图数据以及其他Google服务同步。Android目前包含一个庞大的原生Google App套件(参见图1-12)。

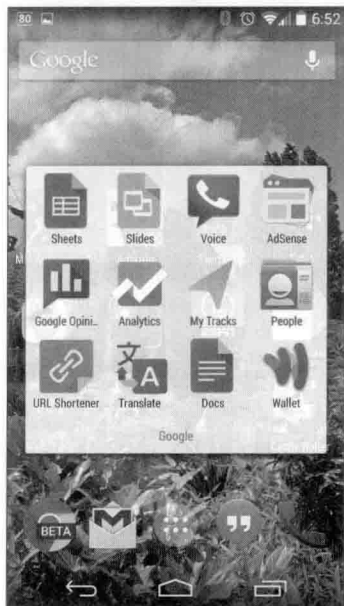


图 1-12 Google 的 App 核心套件是 Android 的关键组成部分，与 Google 服务交互可以创建出更强大的应用

类似地，你的Google账号是使用App Inventor所需的一个重要部分：如果计划提交App以便在Android商店中发布，那么需要登录并连接到Google Play。

Google 在 Android 上的开发节奏非常快，而且该公司持续地增强 Android 与 Google 服务之间的关联。幸运的是，作为程序员，你可以在应用中更多地获取 Google 所提供的强大数据支持。

由于Android持续地前推Google服务并使其中心化，因此思考如何利用这些服务是创建成功App的一个好策略。第11章会探讨连接到其他Web服务的方法，让你的App成为用户更好的信息源。

1.3 应用

智能手机的流行引入了“应用”这个新概念，现在人们也经常称其为App(参见图1-13)。“应用”这个术语一直以来描述的都是一段在特定任务中提供交互式用户体验的程序。而今，文化上的新词App，以及它在手机上的出现，恰恰体现了它的用途和公众对计算领域的兴趣。



图 1-13 智能手机用户期望将多种多样的优质 App 下载到他们的设备上

Android 用户可以通过 Google Play 商店下载 App (参见图 1-8)。使用 MIT App Inventor，也能构建出稍后可以置于 Android 商店中的 App。

在多任务领域，App 的功能变得更加复杂。在用户切换到不同的 App 之后，Android 允许 App 在后台运行某些进程。随着你能更加熟练地构建 App，识别即使 App 处于后台也需要继续运行的进程 (例如播放 MP3 文件或者使用 GPS 跟踪当前位置) 将成为编程过程中的一个重要部分 (参见图 1-14)。

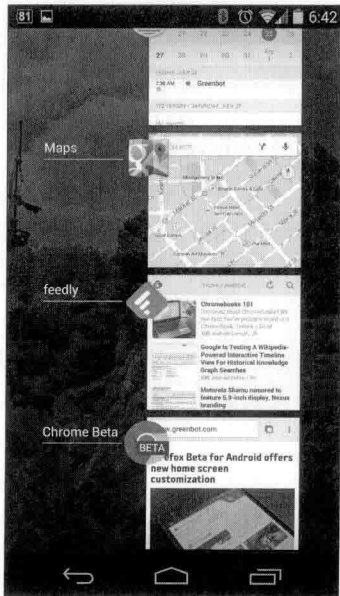


图 1-14 Android 多任务菜单。鉴于其他的 Android 特性，看起来可能会稍有差异，这取决于运行在特定设备上的自定义用户界面

1.4 编程语言

学习传统的、基于文本且难以理解的编程语言已经成为许多想要学习应用编程的人的极大障碍。然而，App Inventor设计的初衷就是为了扫除这些障碍。它简化了编程过程，并允许充满抱负的程序员关注于App的目标。

你将在App Inventor中学习的可视化编程语言为编写Android应用提供了一种快速入手的方法。如果你的目标是进一步学习传统的编程语言，例如Java或Objective-C，那么它是一个极好的入门工具。即使不想深入研究，它也是一种了解更多关于移动设备 and 应用彼此之间交互方式的好方法。

可视化编程的另一个优势在于它可以作为使用当前开发者工具体系的桥梁。在掌握了App Inventor之后，你将为学习更高级的编程做好更充分的准备，例如，使用Android软件开发包(Software Development Kit, SDK)以及使用Android Studio这样的软件来构建Android App(参见图1-15)。



图 1-15 在已经掌握了 App Inventor 的使用之后，或许是时候更进一步了——
获取 Android SDK，以便能够开始构建更加强大和复杂的应用

此外，可视化编程所要求的那种思考方式对于许多其他类型的工作也有帮助。编程中所需要的解决问题的方法和有逻辑的理性思维在当今的商业环境中都是高度受欢迎的技能。

不管未来的道路通向何方，对编程语言、操作系统和计算机设备有着基本的了解将会让你能够更高效地使用MIT App Inventor创建App。

1.5 小结

编程就是让设计良好的应用看上去就像在变魔术。多亏了其较低的入门门槛，MIT App Inventor使你能够非常快速地开始为Android编写App。在进一步完成练习和实验之后，你将能够创建更加实用和复杂的App(参见图1-16)。事实上，Google Play商店中的许多App都是使用App Inventor创建的。

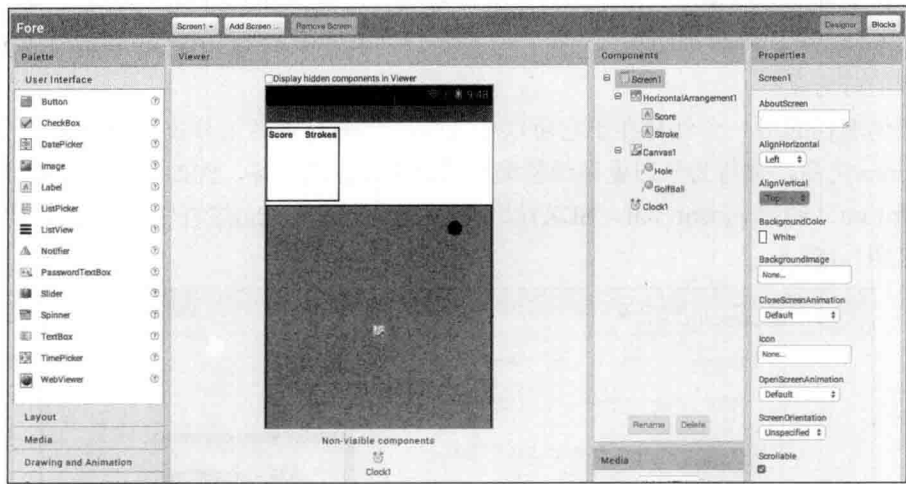


图 1-16 当把 App Inventor 的全部潜能发挥出来时，你可以创建游戏、动画以及各种各样的 App

在考察了操作系统和应用的结构之后，下一步就是学习App Inventor编程的基础。我们要探讨它提供的功能以及如何开始使用它来构建App。

下一章涵盖使用App Inventor构建App的方法，并带领你完成最初的若干个步骤。在这个过程中，你将会看到本章所讨论的计算机科学原理是如何在实践中发挥作用的。

1.6 练习

Google在developer.android.com站点上为Android开发者创建了极好的资源。以下是一些需要探讨的领域，借此来感受一下优秀的Android应用设计和开发应该是什么样子(参见图1-17)。

1. 应用设计和编程需要有创造性的思维，同样也需要理性和逻辑。浏览Android开发者站点上的Creative Vision(创造性构想)一节：<http://developer.android.com/design/get-started/creative-vision.html>。探讨Google在Android平台上的App构建中所追求的设计原则是什么。
2. 本章讨论了用户界面的一些元素，但你也可以研究一下Google对此如何解读：<http://developer.android.com/design/get-started/ui-overview.html>。
3. 花些时间来研究MIT App Inventor的网站：<http://appinventor.mit.edu>。查看Blog(论坛)和Support(支持)两节，找到一些可用的资源。很快你就将开始使用App Inventor更深入地开展工作并构建自己的App了。



图 1-17 Android 开发者网站上拥有相当多的资源

使用 MIT App Inventor 构建应用

了解某个App的功能只是编程的一部分。程序员还需要关注一些特性以及如何实现它们。

在构建 App 的正式工作开始之前，简要概述一下应用的执行方式是有意义的。让我们透过与之交互的漂亮屏幕和图像，看看背后真正发生的事情以及如何让 App 按照设想的方式执行。在这之后，你就能够理解 App 请求信息、从 Internet 上拉取数据以及与其他应用交互的原理。

2.1 MIT App Inventor 网站

与诸如 Gmail 和 Google Drive 等其他在线工具产品一样，MIT App Inventor 是基于 Web 的。在使用 App Inventor 之前，你不需要下载任何软件或者将文件保存到自己的硬盘上(参见图 2-1)。

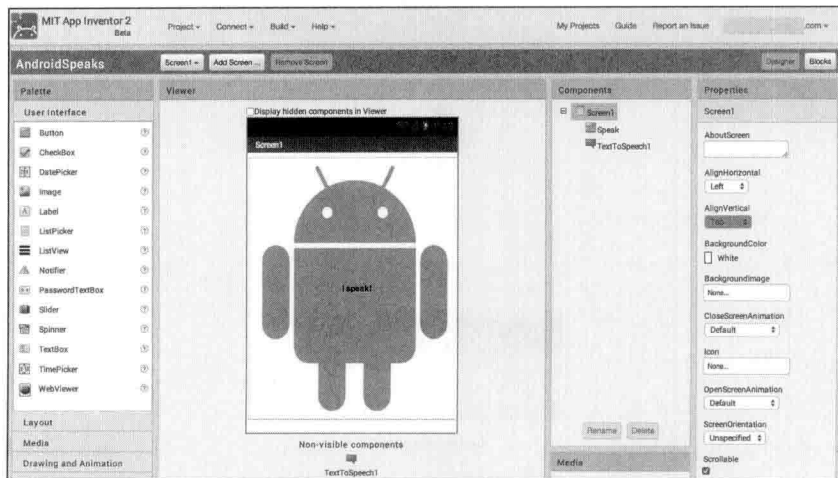


图 2-1 MIT App Inventor 中的 Designer 界面

网页浏览器的选择很重要，App Inventor 团队推荐使用 Google Chrome 或 Firefox。选择其他的浏览器，例如 Internet Explorer，可能会在使用 App Inventor 时遇到错误或其他复杂情况。

浏览App Inventor的网站是获知可用资源的一种好方法。作为开始，启动你的浏览器并访问appinventor.mit.edu(参见图2-2)。首页上包含了App Inventor工具的入口，还有大量在线教程和其他有帮助的资料。

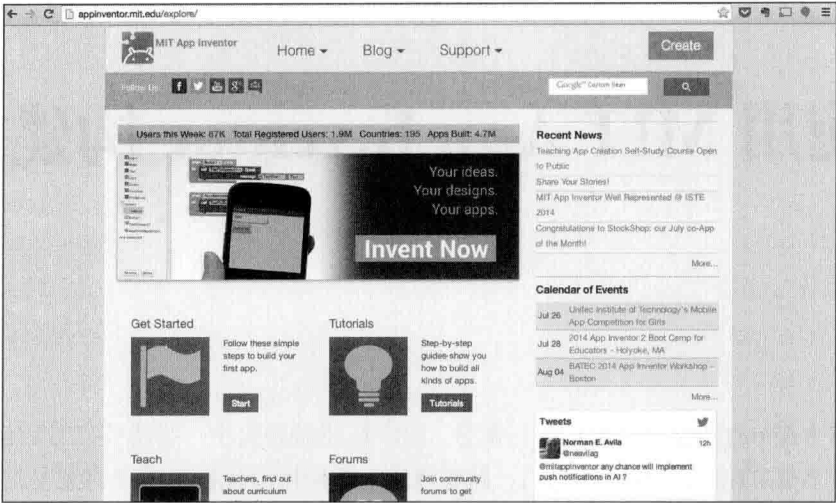


图 2-2 App Inventor 首页——单击 Create 按钮来开始

2.1.1 登录

要开启App Inventor会话，单击位于首页顶部的Create按钮(参见图2-2)。接下来，App Inventor会要求连接到Google账号的权限。这里可以是个人Google账号(以@gmail.com结尾的邮箱地址)，也可以是由大学、公司或其他类型的组织管理的Google App 账号(参见图2-3)。

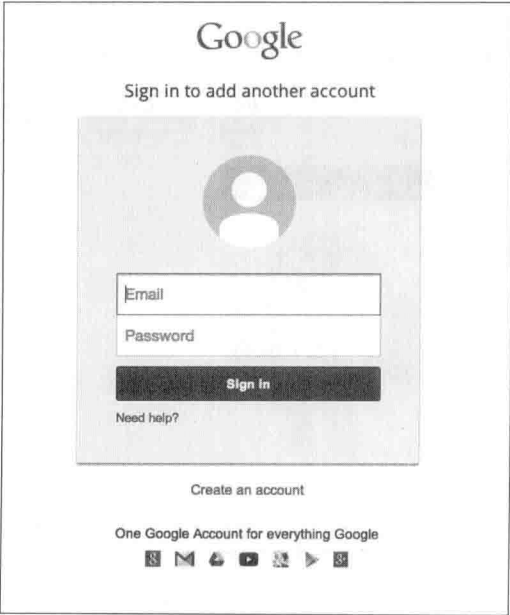


图 2-3 添加 Google 账号以连接到 App Inventor

在使用Google账号登录之后，必须授权App Inventor访问你的Google账号，以便它能够验证登录信息。如果选择了Remember this approval for the next 30 days(在30天内记住此授权)，那么你在返回App开发工作时将不必重复此步骤(参见图2-4)。你只需要在30天结束的时候重新授权访问。



图 2-4 授权你的 Google 账号

下一个界面是保存App Inventor项目的文件系统。在创建了多个项目之后，你可以在这里找到它们，就像计算机上保存所有本地文件列表的文件夹一样。

接下来，单击New Project，并在文本框中输入CatWatch(不允许输入空格)。将会显示Designer界面。

2.1.2 Designer

App的构建在Designer中开始。你要在这里创建用户界面，或者创建称为App的“外观”。还要添加接收用户输入以及向用户显示输出或信息所需的组件。Designer还是你指定App将要使用哪些不可见组件(例如拨号器、GPS或SMS)的地方。注意，由于我们在Designer中，因此图2-5中界面右上角的Designer按钮是浅灰色的。这个按钮和它旁边的Blocks按钮共同标识正在使用的编辑器。

界面的左侧是Palette(参见图2-5)，顾名思义，这里囊括了创建工作所需的全部工具(第3章会详细介绍整套组件)。

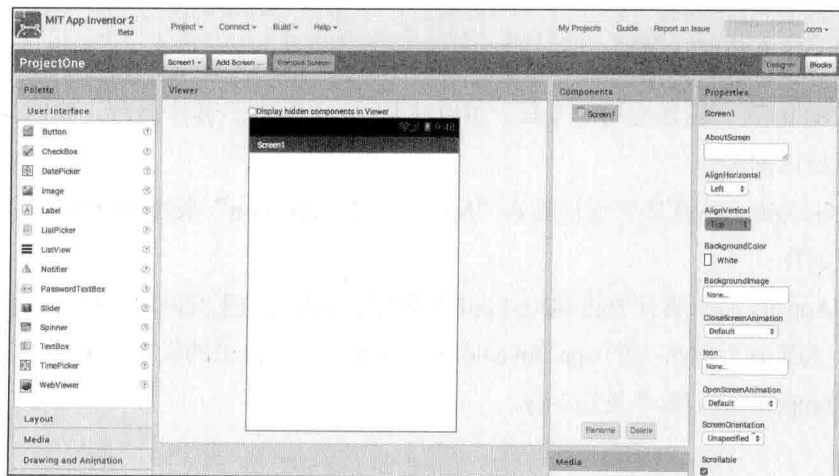


图 2-5 App Inventor 的 Designer 界面

2.1.3 Blocks Editor

Blocks Editor是通过编程来指定App行为的地方(参见图2-6)。你将要在这里添加完成App功能的命令。正如刚刚提到的那样，你可以通过单击右上角的Blocks按钮来访问它。

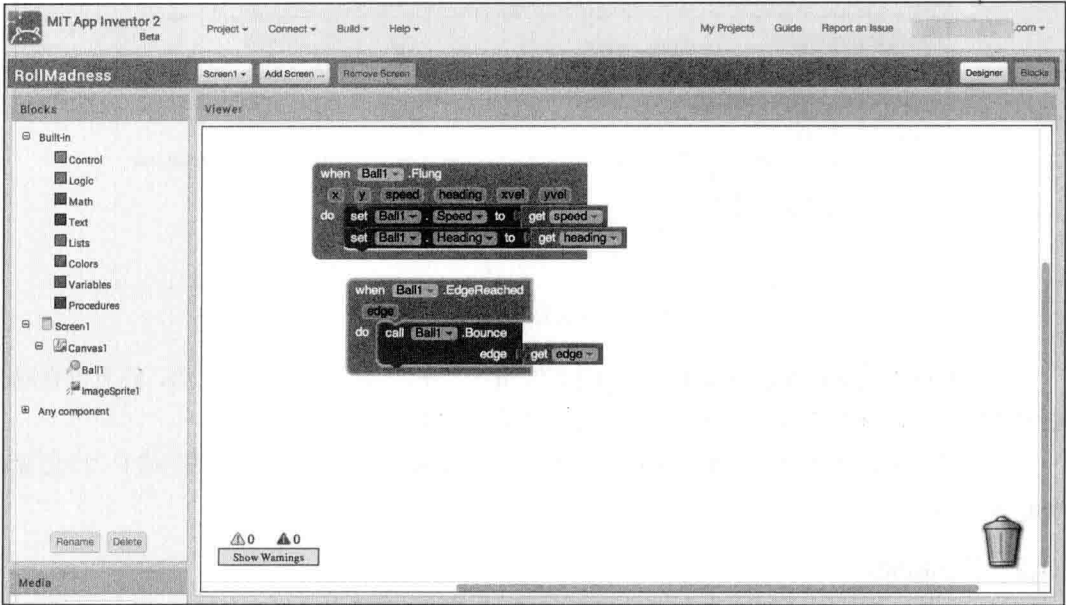


图 2-6 在 Blocks Editor 中进行更细化的编程

MIT App Inventor使用像抽屉一样的弹出菜单来收纳用于编程的拼图块。在Blocks调色板的Built-in(内置)子项下的每个条目都可以被看成一个抽屉。抽屉里包含着像拼图一样的小块。编程是通过连接这些拼图块来完成的。尽管这看上去很简单，但是App Inventor拥有许多能够让用户构建出复杂应用的强大功能。

要更好地理解App编程需要做哪些事情，了解应用的内部原理会很有用。

2.1.4 AI2 Companion App

App Inventor有一款很有用的工具，可以在开发过程的每一步中持续、实时地在Android设备上看到自己的App。

可以在Google Play商店中通过搜索“MIT AI2 Companion”来找到MIT AI2 Companion App(参见图2-7)。

当构建App时，必须将计算机和Android设备连接到相同的无线网络中(台式机可能会使用有线连接)。想要在计算机上的App Inventor中将App连接到自己的设备，请单击Connect选项卡上的AI Companion选项(参见图2-8)。