



创新创业实践课之

移动应用开发

—使用App Inventor开发Android应用

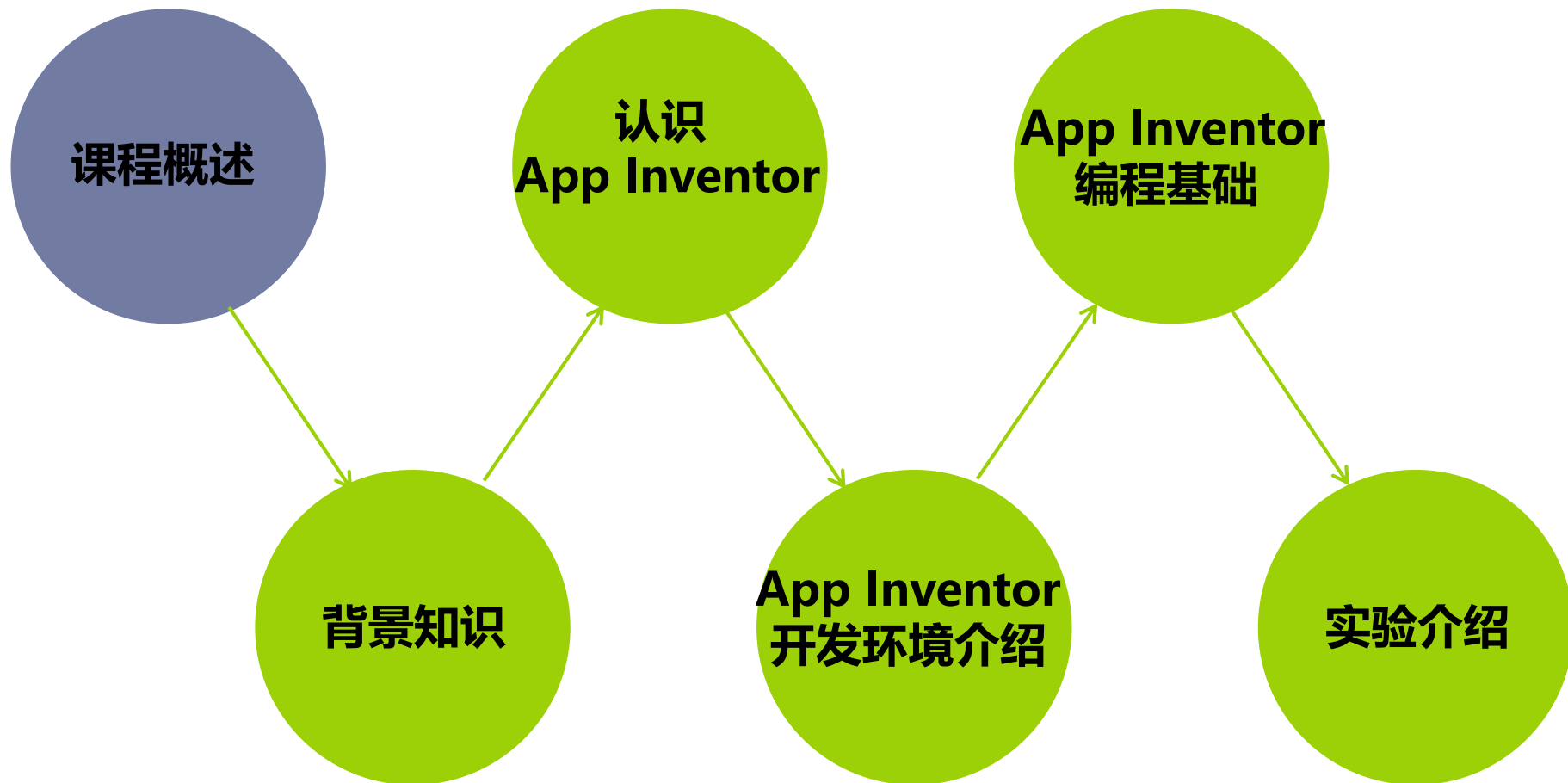
刘健培

ljp@bupt.edu.cn

北京邮电大学 计算机学院



课程内容





课程安排

- ▶ 理论（3课时）
- ▶ 实践（移动应用开发实验指导书）
- ▶ 内容
 - ▶ 固定实验1-5 or
 - ▶ 自主实验6
- ▶ 方式
 - ▶ 线下实验上机（6课时） or
 - ▶ 如果因为疫情防控，无法线下上机，可以在本机安装实验环境，选择线上验收

	班级 2021211-	313 314 315	301 302 303 304
	QQ群号	973375373	973367979
	时间	周一-3-5节	周一-9-11节
		3~17周	3~17周
	理论课地点	S118	S118
校历周次	实践课地点 每周一日期	教学实验综合楼北 D-N01/03, D-N05/07, D-N09/11,	
3	9月13日	绪论+机器人（课）	绪论+机器人（课）
4	9月20日	中秋假期	中秋假期
5	9月27日	机器人（实验）	机器人（实验）
6	10月4日		
7	10月11日	创新讲座	创新讲座
8	10月18日	大数据（课）	大数据（课）
9	10月25日	大数据（实验）	大数据（实验）
10	11月1日	大数据（实验）	大数据（实验）
11	11月8日	双创对接会	双创对接会
12	11月15日	移动应用（课）	移动应用（课）
13	11月22日	移动应用（实验）	移动应用（实验）
14	11月29日	移动应用（实验）	移动应用（实验）
15	12月6日	创新培训	创新培训
16	12月13日	补充实验/验收	
17	12月20日		
18	12月27日		



考核方式 – 二选一

▶ 常规方案：专题指定实验

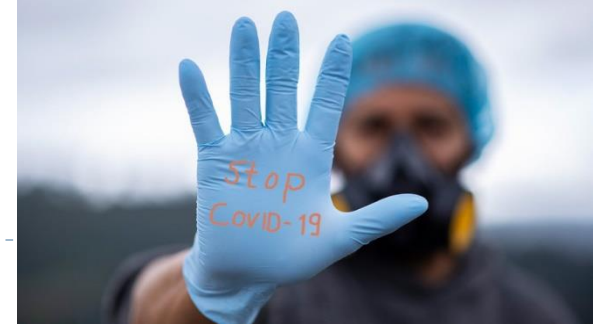
- ▶ 实验方式：线下在机房上机完成实验，or 在本机安装实验环境线上完成实验（疫情防控）
- ▶ 验收方式：线下现场运行程序，由实验指导老师验收 or 线上验收（方式待指导老师通知）
- ▶ **提交材料**：实验报告+项目源文件（.aia文件）+可安装文件（.apk 文件），材料要求参见ftp上实验报告模板要求
- ▶ 成绩构成：基本实验成绩+提高实验成绩

▶ 个性化方案：自主创新实验

- ▶ 题目、内容、语言、工具不限。如项目较大，可多人合作组队完成。可线上完成。
- ▶ 验收方式：演示应用+简单答辩
- ▶ **提交材料**：提交项目源文件+可安装文件+实验报告等相关文档
- ▶ 成绩构成：验收成绩+奖励成绩
 - ▶ 如好的创意、好的用户体验、具有现实意义、已发布到应用市场等



个性化方案 – 2021题目推荐



- ▶ **Fighting Coronavirus**
- ▶ 开发一个应用程序，帮助用户应对新型冠状病毒挑战
- ▶ 例如，内容可以是：
 - ▶ 以试题形式，测试用户对新型冠状病毒的知识
 - ▶ 以游戏形式，帮助用户提高对新型冠状病毒的认识
 - ▶ 以游戏形式，鼓励用户保持良好的健康习惯
 - ▶ 以多媒体互动方式，展示/引导如何预防病毒，如：如何正确洗手、佩戴口罩、保持社交距离等
 - ▶ 更新关于新型冠状病毒的新闻
 - ▶ 跟踪新型冠状病毒，本地爆发时向用户发出警报
 - ▶ 。 。 。

欢迎同学们提出自己的创意！



个性化方案 – 往届示例

- ▶ 2-36进制转换
- ▶ 配色工具
- ▶ 汉诺塔
- ▶ 排序算法可视化
- ▶ Fighting Coronavirus

2~36进制任意转换

16 进制 | EEEE

清除 开始转换

10 进制 | 61166

二进制 | 1110 1110 1110 1110

八进制 | 167 356

十进制 | 61166

十六进制 | EEEE

G/M/K/B | 0GB 0 MB 59KB 750B

0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
1	1	1	0	1	1	1	0
15	14	13	12	11	10	9	8
1	1	1	0	1	1	1	0
7	6	5	4	3	2	1	0



个性化方案 – 往届示例

- ▶ 2-36进制转换
- ▶ 配色工具
- ▶ 汉诺塔
- ▶ 排序算法可视化
- ▶ Fighting Coronavirus





个性化方案 – 往届示例

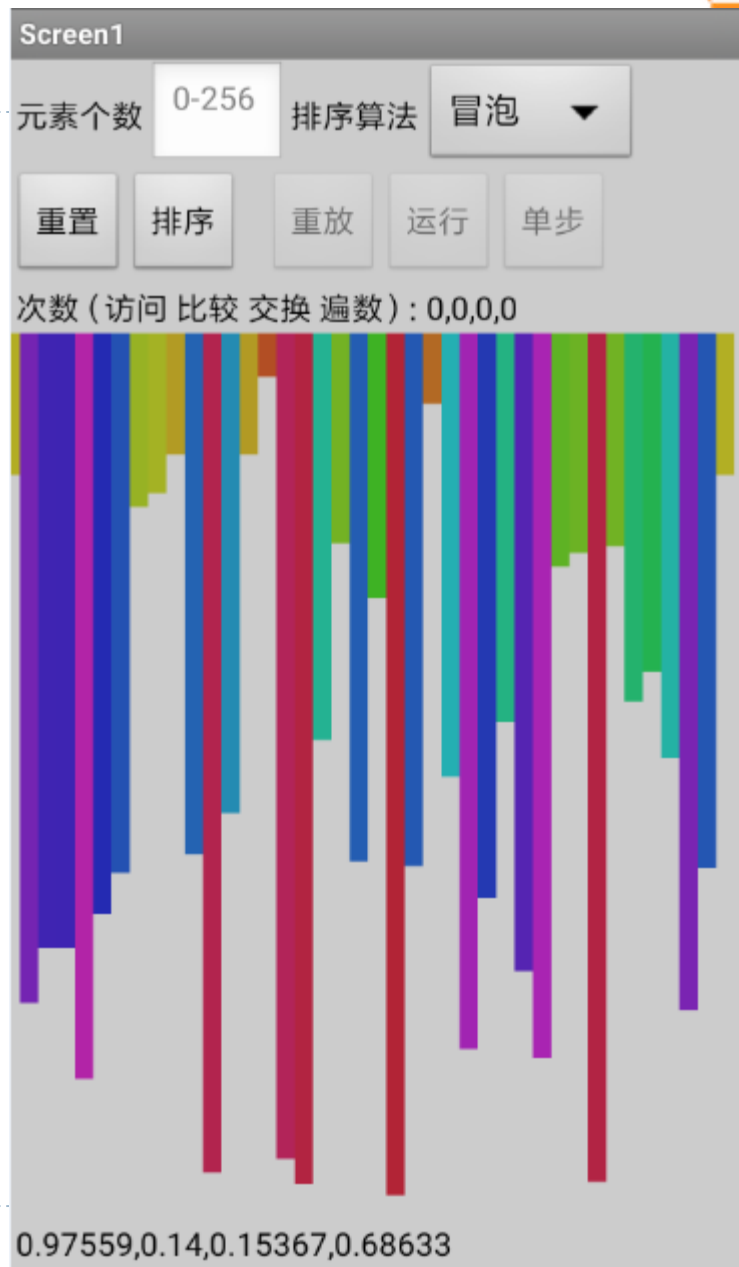
- ▶ 2-36进制转换
- ▶ 配色工具
- ▶ 汉诺塔
- ▶ 排序算法可视化
- ▶ Fighting Coronavirus





个性化方案 – 往届示例

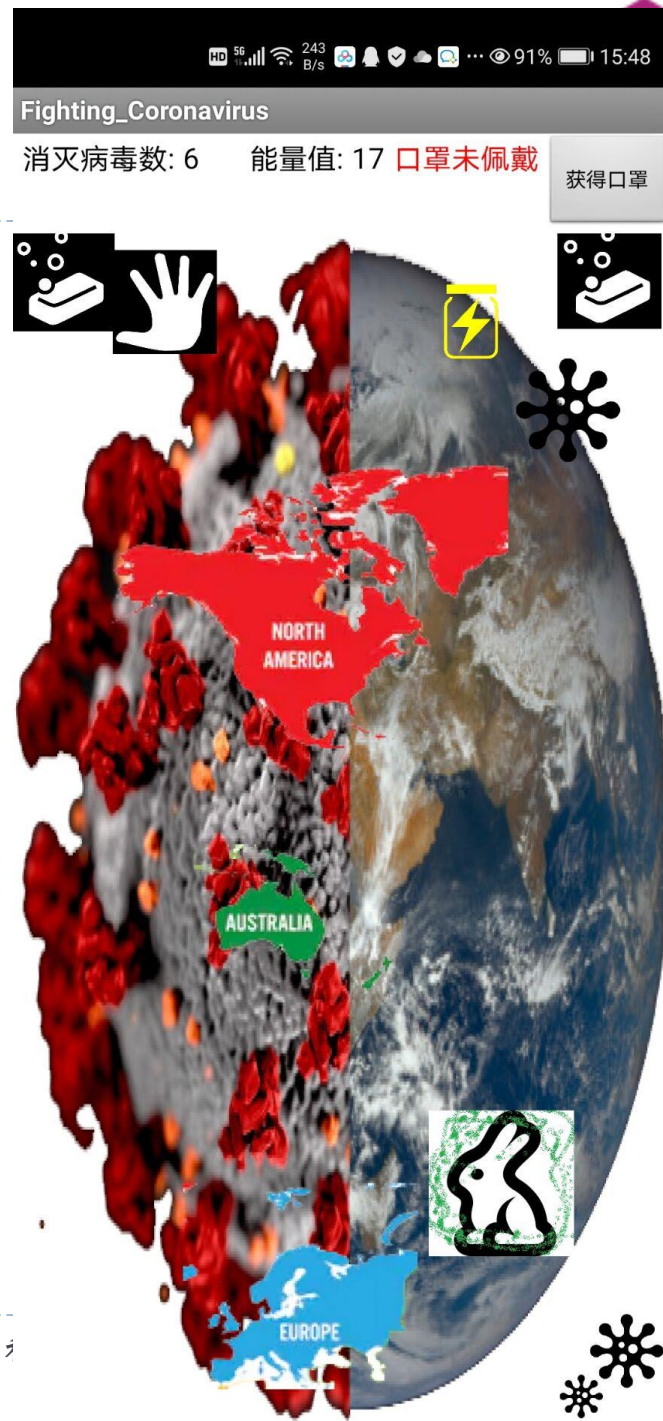
- ▶ 2-36进制转换
- ▶ 配色工具
- ▶ 汉诺塔
- ▶ 排序算法可视化
- ▶ Fighting Coronavirus





个性化方案 – 往届示例

- ▶ 2-36进制转换
- ▶ 配色工具
- ▶ 汉诺塔
- ▶ 排序算法可视化
- ▶ **Fighting Coronavirus**





个性化方案 – 更多示例



► <https://gallery.appinventor.mit.edu/>

The screenshot shows the MIT App Inventor Gallery homepage. At the top, there's a navigation bar with the MIT App Inventor logo, a 'Create Apps!' button, and links for 'About', 'Educators', 'News', 'Resources', 'Blogs', and a 'Donate' button. A small blockly snippet is visible on the right. Below the navigation bar is a large teal banner with the text 'MIT App Inventor Gallery'. Underneath the banner, it says 'You are logged in as: liu jp' with a link to 'Account Information'. There's a search bar with a 'Search' button. Below the search bar are several links: 'Sort by Name', 'Your Apps', 'Sort by most recent', 'Next >>', 'Featured Apps', and 'Gallery Home'. The main content area displays two app cards. The first card is titled 'AID5', dated 'Nov. 12, 2021, 4:45 a.m.', with 'Likes: 0' and a heart icon. It features a diagram of a network with nodes and edges. The author is 'hwsertgq' and the credit is 'Michael David Lopez 892968'. The second card is titled 'FoodChase_template2', dated 'Nov. 12, 2021, 2:09 a.m.', with 'Likes: 0' and a heart icon. It features a screenshot of a mobile app interface. The author is 'fgh'. Below the app card, there are links: 'Load App Into MIT App Inventor', 'Other projects by same author', 'Report Project', and a 'Permanent link' which is 'https://gallery.appinventor.mit.edu/?galleryid=fcd38a8-9dee-45d2-8f3b-b718e23c4968'.



课后作业 - 实验准备

▶ 提前预习实验指导书

- ▶ 移动应用开发 实验指导书 - 2021.pdf
- ▶ <ftp://student:asdf1234@10.105.240.91/>
- ▶ 子目录：移动应用开发

▶ 搭建开发环境

- ▶ <ftp://student:asdf1234@10.105.240.91/移动应用开发/开发环境安装文件/>
 - ▶ AppInventor2018PersonalEdition_Win.zip
 - ▶ AI2Companion_Win7_32bit.zip/AI2Companion_Win7_64bit.zip
 - ▶ 免安装，解压可用

▶ 完成实验指导书上的实验一

- ▶ 目的：熟悉App Inventor开发环境



参考资源

▶ 网址

- ▶ <http://appinventor.mit.edu/>
- ▶ <http://www.l7coding.net/>
- ▶ <http://ai2.appinventor.mit.edu/>
- ▶ <http://app.gzjkw.net/>
- ▶ <http://www.appinventor.cn/>
- ▶ <http://www.imagnity.com/>
- ▶ <http://www.puravidaapps.com/>

▶ 参考书目

- ▶ 《写给大家看的安卓应用开发书:App Inventor 2快速入门与实战》
- ▶ 英文电子版: <http://www.appinventor.org/book2>
- ▶ 中文电子版: <https://book.l7coding.net/>





动手试一试



▶ 登录ftp

- ▶ 在文件管理器中打开<ftp://student:asdf1234@10.105.240.91/>

▶ 下载资料并解压

- ▶ 移动应用开发 实验指导书 - 2021.pdf
- ▶ ApplInventor2018PersonalEdition_Win.zip
- ▶ AI2Companion_Win7_32bit.zip/AI2Companion_Win7_64bit.zip

▶ 打开App Inventor

- ▶ ApplInventor2018PersonalEdition_Win/启动APPIinventor.cmd

▶ 试用

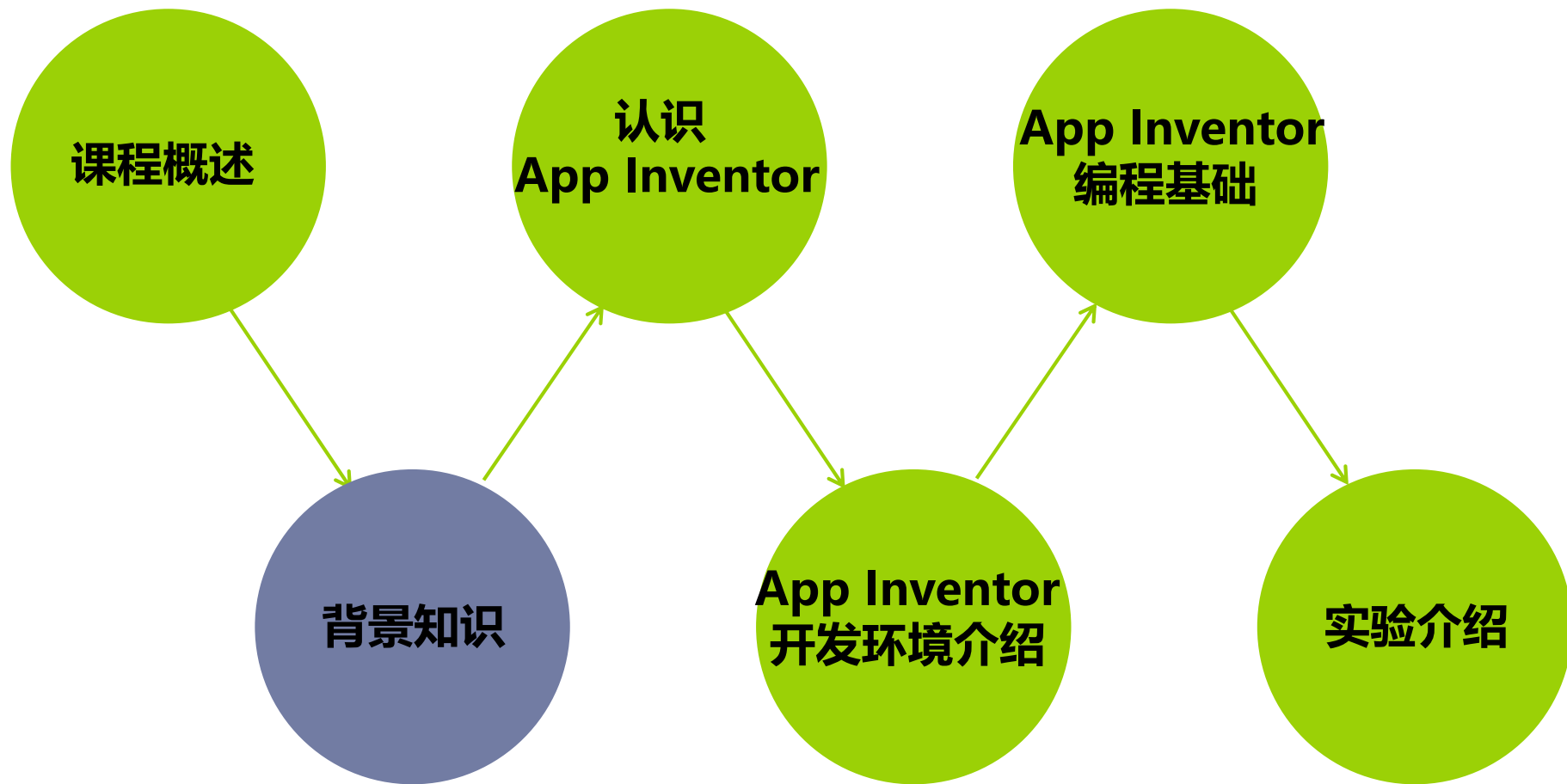
- ▶ 参考《移动应用开发 实验指导书 - 2021.pdf》的1.2-1.3节

▶ 导入参考示例

- ▶ ftp: 移动应用开发/其余资料/写给大家看的安卓应用开发书-随书资料/aia.zip



课程内容





移动：手机、操作系统、应用



▶ 智能手机 (Smartphone)

- ▶ A smartphone is a handheld personal computer with a mobile operating system and an integrated mobile broadband cellular network connection for voice, SMS, and Internet data communication.

▶ 移动操作系统 (Mobile OS)

- ▶ A mobile operating system (or mobile OS) is an operating system for phones, tablets and smartwatches or other mobile devices.

▶ 移动应用程序 (Mobile App)

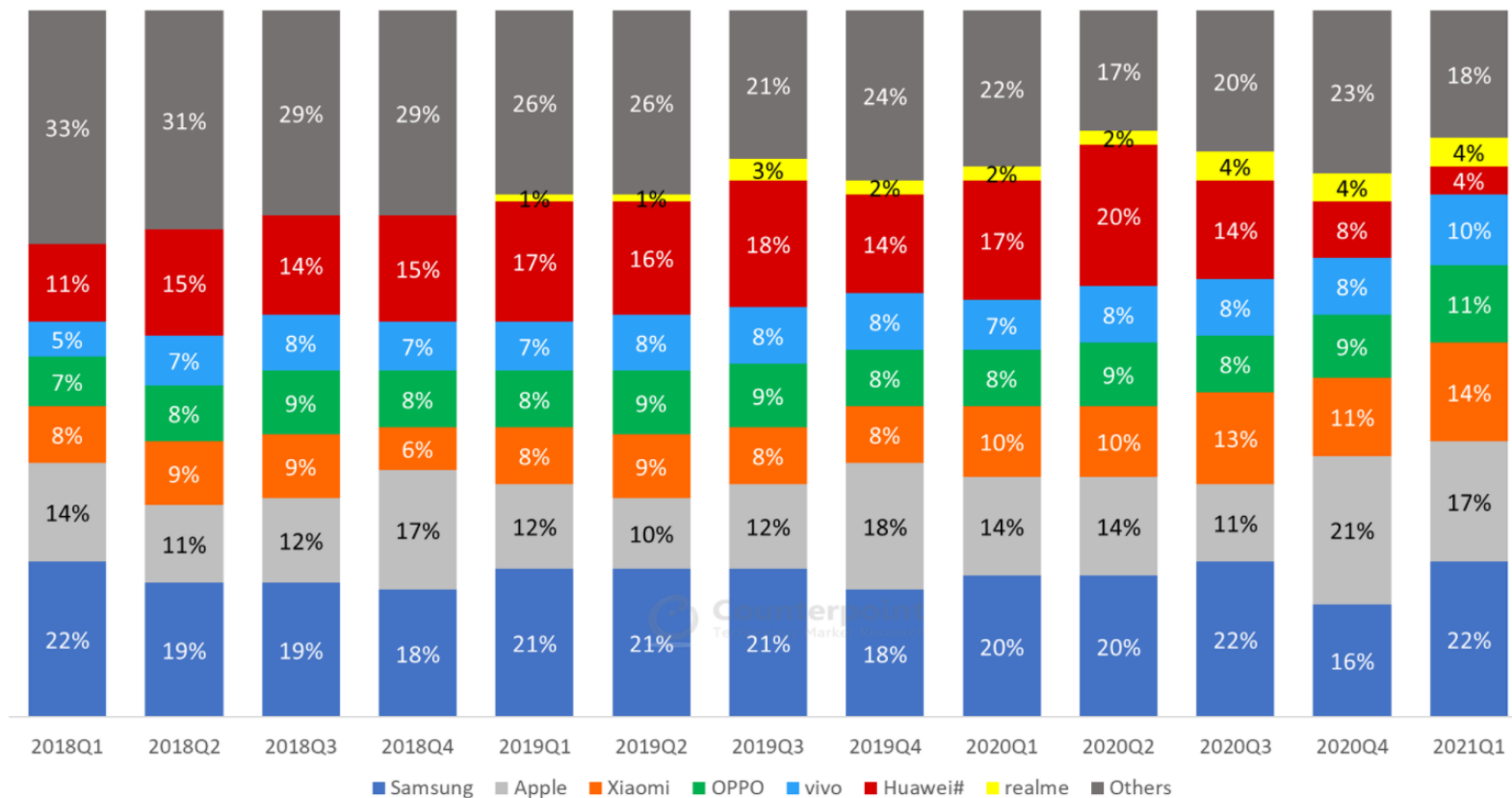
- ▶ A mobile app is a computer program designed to run on a mobile device such as a phone/tablet or watch.



智能手机



Global Smartphone Market Share (2018 Q1 - 2021 Q1)



模拟手机

数字功能手机

早期智能手机

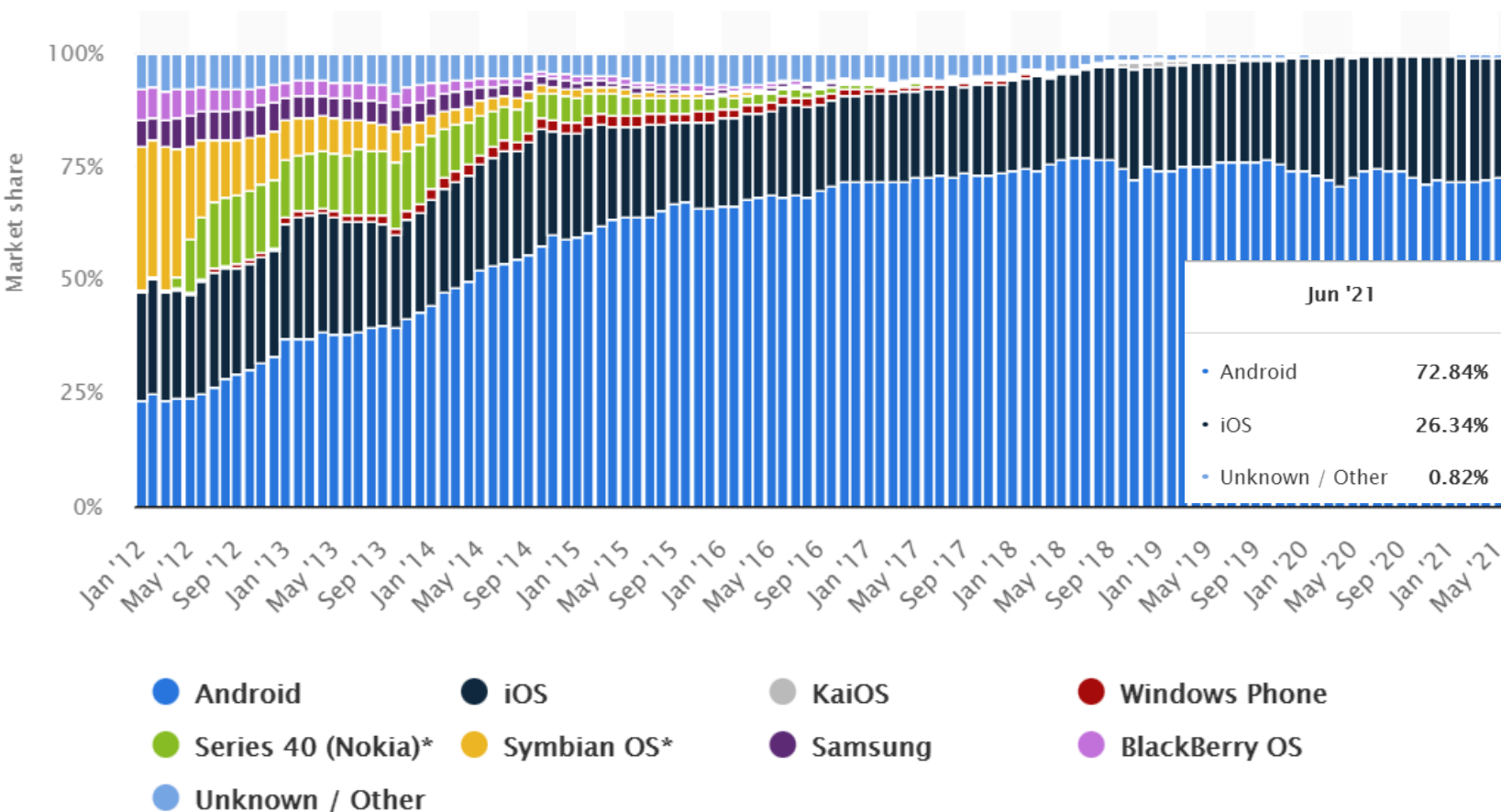
现代智能手机





移动操作系统

- ▶ Android (Google)
- ▶ iOS (Apple)





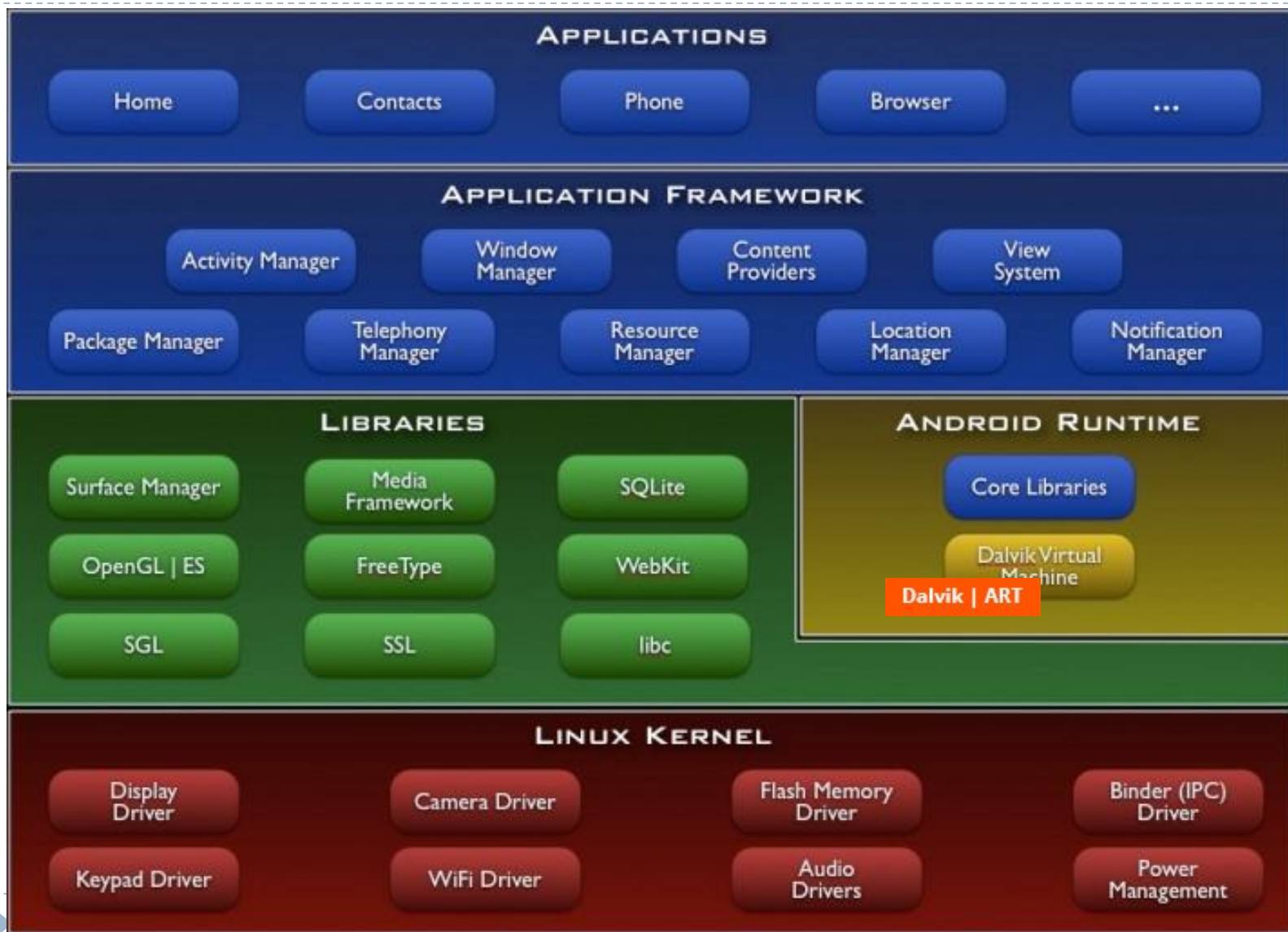
移动开发平台对比



Mobile platform	Android	iOS
Company	Google, Open Handset Alliance	Apple Inc.
Market share	75%	25%
Current version	14	11
OS family	Modified Linux kernel based	Darwin
Supported CPU architecture	ARM (32-bit ARMv7-A and 64-bit ARMv8-A only), x86, x86-64	64-bit ARMv8-A only
Programmed in	C, C++, Java, Kotlin	C, C++, Objective-C, Swift
Official app store	Google Play, etc	App Store



Android操作系统结构

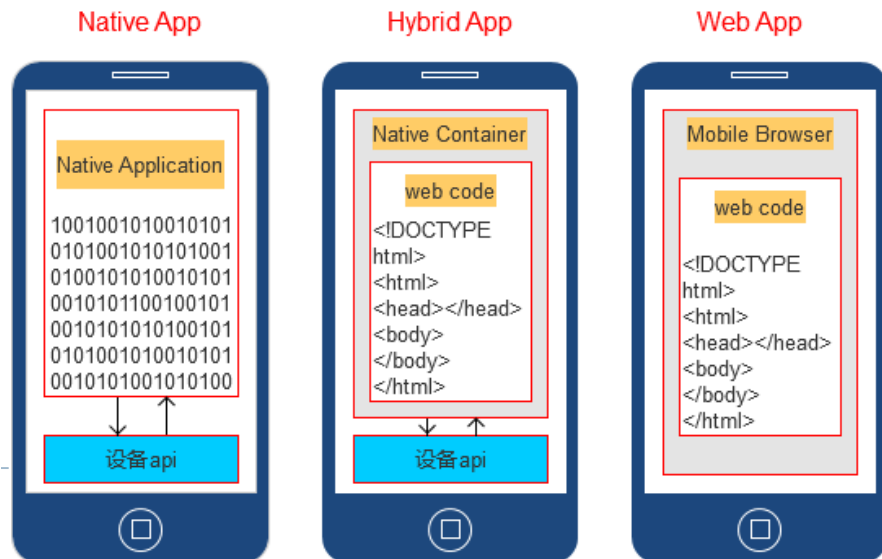




移动应用 - 按技术分类

- ▶ 原生应用程序 (Native App)
- ▶ Web应用程序 (Web App)
 - ▶ PWA (Progressive Web App)
- ▶ 混合式应用程序 (Hybrid App)
 - ▶ HTML5 WebView + 原生 (微信小程序、Cordova、Ionic)
 - ▶ JavaScript 开发 + 原生渲染 (React Native、Weex)
 - ▶ 自绘UI + 原生 (Flutter、QT Mobile)
 - ▶ 托管平台: Xamarin

} 跨平台技术





原生应用程序（Native App）开发

- ▶ 由移动操作系统厂商，自行提供原生SDK单独开发的本地应用，一般采用针对操作系统的特定语言进行编写。

例如

- ▶ Android App: 使用Android Studio与Java/Kotlin开发，上架到Google Play，或直接执行apk安装文件
- ▶ iOS App: 使用Xcode与Objective-C/Swift开发，上架到App Store

▶ 优点

- ▶ 可善用系统API及平台特性访问手机的提供的功能（例如GPS、相机、传感器等）
- ▶ 与系统高度切合，可获得更快的运行速度和性能，支持丰富的图形和动画
- ▶ 可针对不同平台提供不同的体验，针对平台特性去做用户体验优化
- ▶ 可访问本地资源，支持离线工作，节省用户的流量成本
- ▶ 可在应用商店轻易地找到应用及安装

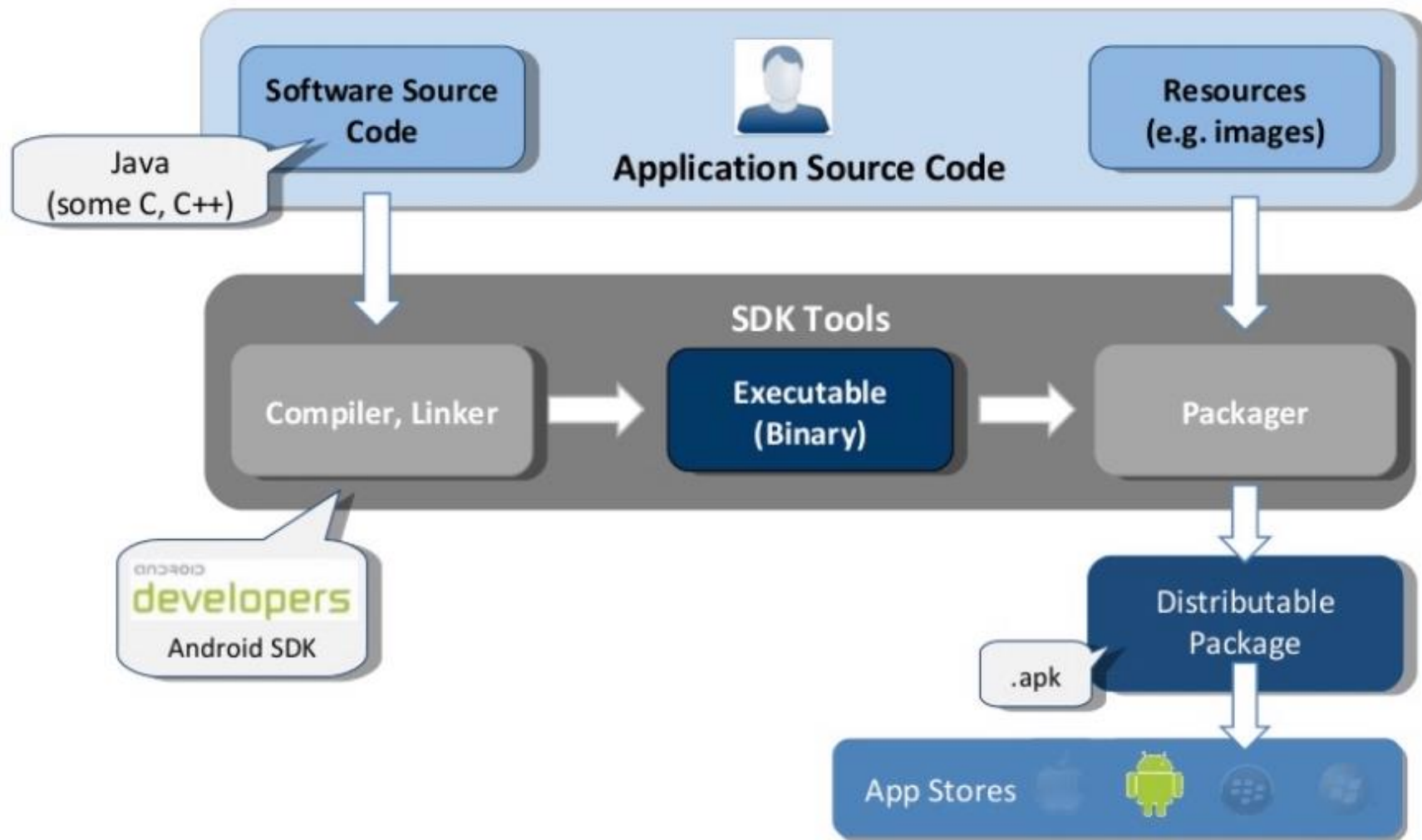
▶ 缺点

- ▶ 开发成本较高，开发周期长，不同平台有不同的开发语言和界面适配
- ▶ 维护成本高，使用者须手动下载更新版本
- ▶ 更新缓慢，上架时间（应用商店审查程序）较长
- ▶ 下载应用可能需要付费





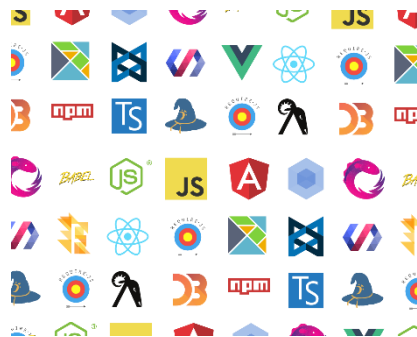
Android Native App 开发





Web应用程序（Web App）开发

- ▶ 使用HTML(5)、CSS及Java Script等Web网页技术开发的应用程序,通过浏览器访问,可看作触屏版的网站
- ▶ 优点
 - ▶ 只需要浏览器,适用范围广,覆盖所有智能手机,可以跨多个平台和终端
 - ▶ 开发成本较低
 - ▶ 方便快速地部署,无需用户安装,无需提交到应用商店
 - ▶ 更新快,使用者无须更新即可使用最新版本
- ▶ 缺点
 - ▶ 浏览体验短期内还无法超越原生应用
 - ▶ 速度较慢,大部分需要链接互联网
 - ▶ 无法充分运用手机的特性,限制用户使用功能
 - ▶ 开发者较难藉由使用者下载应用而获利
 - ▶ 用户留存率低





混合式应用程序（Hybrid App）开发










- ▶ 以Web技术开发App，再透过混合开发框架跟移动设备互动，最后包装上原生应用程序的外壳，上架至应用程序商店。是一个“嵌入”了浏览器的原生程序。
- ▶ 优点
 - ▶ 兼具“Native App良好用户交互体验的优势”和“Web App跨平台开发的优势”
 - ▶ 相比原生应用程序，开发、维护、更新更加便捷
 - ▶ 适用于应用程序商店
 - ▶ 部分支持离线功能
- ▶ 缺点
 - ▶ 仍须上架时间
 - ▶ 速度及使用者体验不如原生应用程序
 - ▶ 技术点多，学习范围较广





渴望力量?



- ▶ <https://developer.android.com/docs?hl=zh-cn> 
- ▶ <https://developer.apple.com/documentation/technologies> 
- ▶ <https://developer.mozilla.org/en-US/docs/Web> 
- ▶ <https://flutter.cn/docs> 
- ▶ <https://reactnative.cn/docs/getting-started> 
- ▶ <https://developers.weixin.qq.com/miniprogram/dev/framework/work/> 
- ▶ <http://doc.weex.io/zh/guide/introduction.html> 
- ▶ <https://dotnet.microsoft.com/apps/xamarin> 
- ▶ <https://www.qt.io/product/mobile-app-development/> 



微信小程序（ Wechat ）



- ▶ 一种不需要下载安装即可使用的应用，它触手可及，用完即走。
- ▶ 非常适合相对简单、低频、性能要求不高的应用。
- ▶ 微信小程序提供了一个简单、高效的应用开发框架和丰富的组件及API，帮助开发者在微信中开发具有原生 APP 体验的服务。
 - ▶ 小程序定义了一套微信自有的组件并根据运行环境的不同（iOS、Android、PC），将这些组件编译转化为对应平台的可运行组件。
- ▶ 优点
 - ▶ “轻”，即用即走，随手可得
 - ▶ 基于微信，跨平台
 - ▶ 接近Native App的体验，能访问原生组件
 - ▶ 上手容易，开发逻辑较简单
 - ▶ 丰富的流量资源入口，微信天然的关系链与获客能力
- ▶ 缺点
 - ▶ UI较简单
 - ▶ 开发环境封闭，无法使用已有的JavaScript框架、类库等



微信小程序开发流程

注册账号

- 1. 在微信公众平台注册小程序，完成注册后可以同步进行信息完善和开发。

完善信息

- 2. 填写小程序基本信息，包括名称、头像、介绍及服务范围等

开发代码

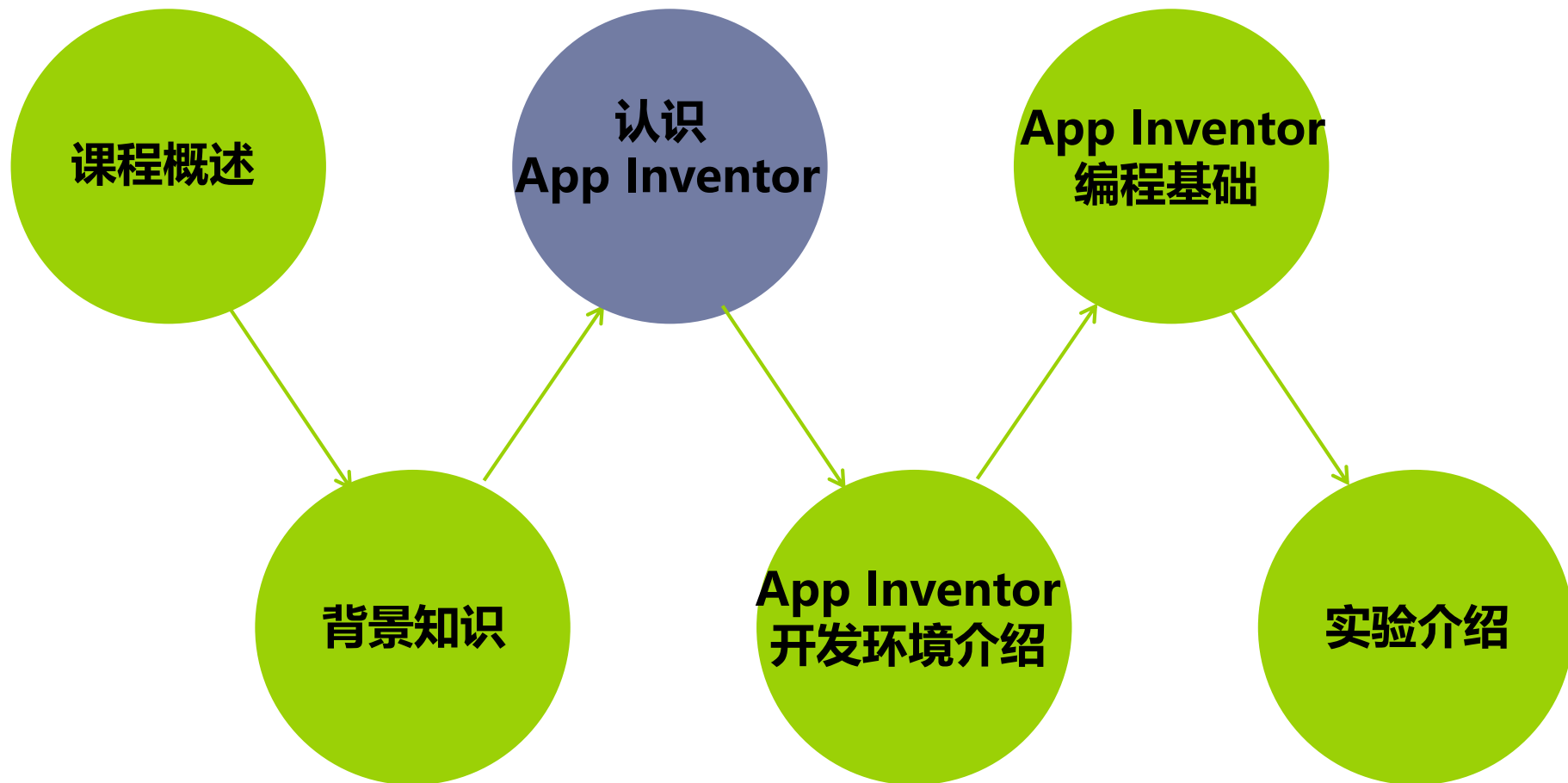
- 3. 完成小程序开发者绑定、开发信息配置后，开发者可下载开发者工具、参考开发文档进行小程序的开发和调试
- <https://mp.weixin.qq.com/cgi-bin/wx>

审核发布

- 4. 完成小程序开发后，提交代码至微信团队审核，审核通过后即可发布



课程内容





什么是App Inventor?



- ▶ 用于手机软件开发的“块”语言
 - ▶ 2010年12月15日。正式公开发布，团队由Hal Abelson教授（SICP一书作者之一）和Mark Friedman领导。
 - ▶ 2011年下半年。谷歌发布源代码，项目移交给麻省理工学院（MIT）移动学习中心。
 - ▶ 2012年3月。MIT版本的App Inventor被推出并开放使用。
 - ▶ 2013年12月。MIT发布了App Inventor 2。
- ▶ App Inventor是一款采用拖拽操作的可视化编程工具，主要用于构建运行在安卓平台上移动应用。
 - ▶ App Inventor提供了基于Web的图形化的用户界面设计工具，可以设计应用的外观；
 - ▶ 再像玩拼图游戏一样，将“块”语言拼在一起，来定义应用的行为。
- ▶ 不需要程序设计的经验，对初学者与业余开发者非常友好
- ▶ 学习程序设计的基本方法，培养计算思维



App Inventor适合做什么？

- ▶ 创建原型应用
- ▶ 教与学
- ▶ 基于兴趣或者游戏的心态开发手机软件
- ▶ 创建贴近自己需求的个性化应用
- ▶ 比较适合创建的应用有
 - ▶ 手机游戏
 - ▶ 信息与教学类应用：小测验、互动学习反馈等
 - ▶ 高科技应用：条码扫描、文字识别、音乐制作、视频播放、语音识别等
 - ▶ 位置感知应用：丰富的手机传感器
 - ▶ 基于web的应用



App Inventor开发方式：开发+测试

▶ 开发环境

▶ 在线开发：使用online服务器

▶ MIT: ai2.appinventor.mit.edu

▶ 国内: app.gzjkw.net

▶ 离线开发：

▶ 使用离线版: AppInventor2018PersonalEdition_Win

▶ 测试环境

文档: <http://appinventor.mit.edu/explore/ai2/setup.html>

▶ Wifi局域网 + Android手机 + Ai伴侣App

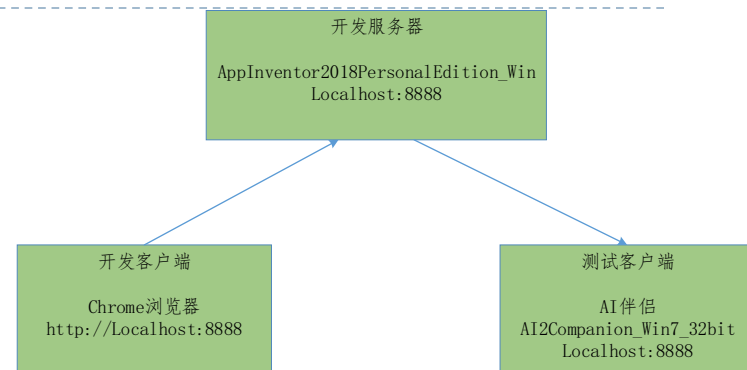
▶ USB数据线 + aistarter软件 + Android手机 + Ai伴侣

▶ Android模拟器 + aistarter软件 + Ai伴侣App

▶ 实验室机房使用的方式

▶ AppInventor2018PersonalEdition_Win + AI2Companion_Win7_32bit

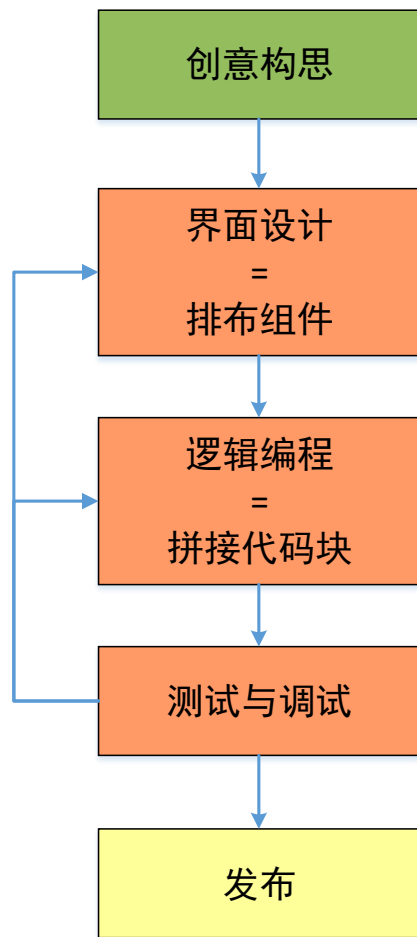
▶ 机房安装路径: C:\CxcyCourse\





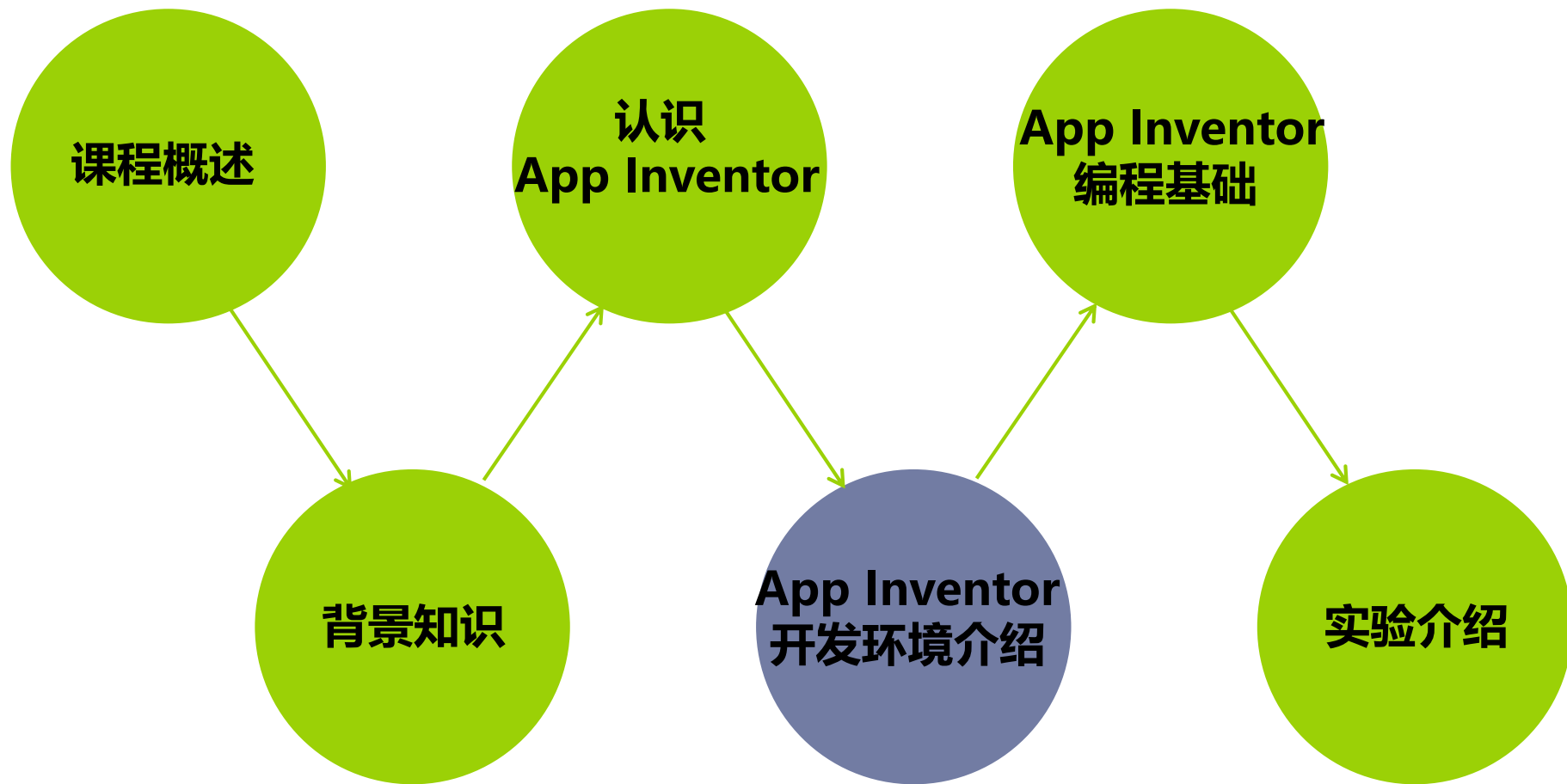
App Inventor开发的主要过程

- ▶ 1. 打开开发环境
 - ▶ 可以使用在线版或者离线版
- ▶ 2. 创建项目
 - ▶ 可以新建项目、导入项目或通过另存复制项目的方式创建
- ▶ 3. 在设计视图中设计界面
 - ▶ 选择组件
 - ▶ 拖放、排布组件
 - ▶ 设置组件属性
- ▶ 4. 在编程视图中增加代码
 - ▶ 选择代码块
 - ▶ 拖放、连接代码块
 - ▶ 为组件添加行为，完成事件处理
- ▶ 5. 打开测试环境进行测试
 - ▶ 可以通过Wifi、USB数据线连接手机，或者Android模拟器
- ▶ 6. 创建APK文件，发布APP
 - ▶ 可以打包apk文件，或者导出项目源代码（.aia文件）





课程内容





App Inventor开发环境



▶ 组件设计视图/Designer

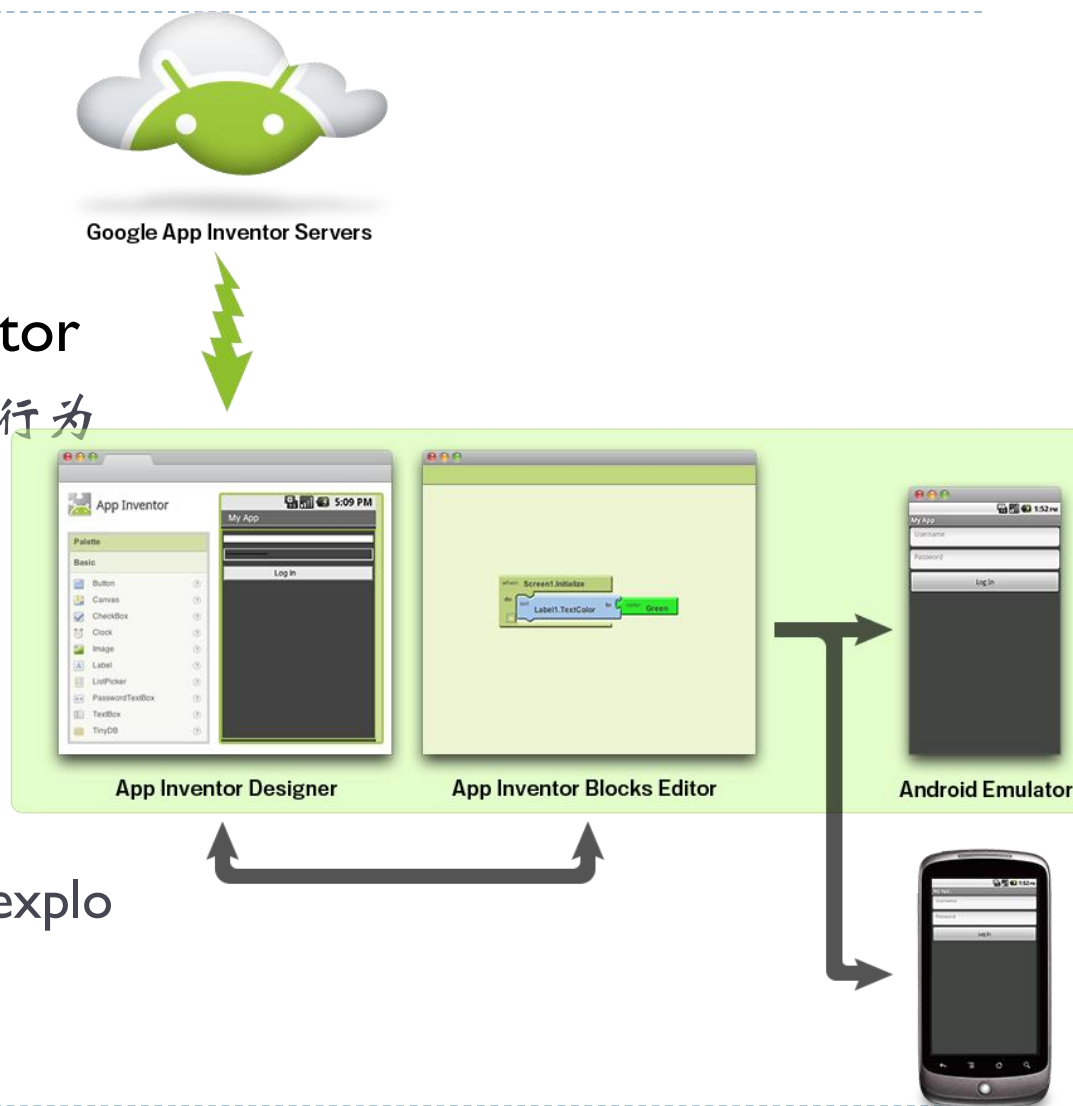
- ▶ 用户GUI构建器
- ▶ 应用的界面部分（静态）

▶ 逻辑编程视图/Blocks Editor

- ▶ 提供支持GUI界面元素的行为
- ▶ 应用的编程部分（动态）
- ▶ 基于MIT OpenBlocks库和Scratch语言

▶ 模拟器

- ▶ 实时测试
- ▶ <http://appinventor.mit.edu/explore/ai2/setup-emulator>





组件设计视图/Designer



 MIT App Inventor 2
汉化测试版

项目 ▾ 连接 ▾ 编译 ▾ 帮助 ▾ 系统功能区 我的项目 向导 反馈 简体中文 ▾ test@example.com ▾

你好猫咪 项目名称

Screen1 ▾ 增添屏幕 删除屏幕

屏幕操作

切换视图 设计 编程

组件面板

用户界面

组件布局

多媒体

摄像机 ①

照相机 ①

图片选择框 ①

音频播放器 ①

音效播放器 ①

录音机 ①

语音识别器 ①

语音合成器 ①

视频播放器 ①

语言翻译器 ①

绘图动画

传感器

社交应用

数据存储

通信连接

乐高®机器人

所有可用组件

工作区域

☐ 显示隐藏组件

☐ 选中此项显示平板预览效果

你好猫咪



用户界面预览
[可视组件]

非可视组件

非可视组件

音效播放器1 加速度传感器1

组件列表

Screen1

按钮1

音效播放器1

加速度传感器1

项目中
所有组件列表

重命名 删除

素材

kitty.png

meow.mp3

项目中所有资源

属性面板

Screen1

应用说明

水平对齐
居中 ▾

垂直对齐
居上 ▾

应用名称
HelloPurr

背景颜色
白色

背景图片
无...

关屏动画
默认效果 ▾

图标
无...

开屏动画
默认效果 ▾

组件属性设置



组件设计视图/Designer

- ▶ **项目名称：**显示当前正在开发的项目名称（编程视图中也可见）；
- ▶ **屏幕操作：**一个项目可以是单一屏幕或多屏幕的应用，在此可以为项目添加或删除屏幕（该功能在编程视图中也可见）；
- ▶ **视图切换：**在“设计视图”及“编程视图”之间进行切换(编程视图中也可见)；
- ▶ **组件面板：**开发操作区左侧，分组陈列出所有可用组件，开发者可将其拖拽到用户界面预览区中（非可视组件也必须拖拽到预览区中）；
- ▶ **用户界面预览区：**组件面板右侧，用于显示用户界面上所有可视组件；
- ▶ **非可视组件区：**用户界面预览区下方，陈列出项目中所有非可视组件；
- ▶ **组件列表：**位于用户界面预览区的右侧，显示项目中的所有组件，当选中某一组件时，可以修改组件名称，或删除该组件；
- ▶ **素材：**位于组件列表下方，列出了项目中加载的所有资源文件，当选中某资源文件时，可以下载或删除该资源文件；
- ▶ **组件属性面板：**当在组件列表中选中某个组件时，属性面板中将显示该组件的所有属性，供开发者查看及设置。



逻辑编程视图/Blocks Editor

MIT App Inventor 2
汉化测试版

项目 ▾ 连接 ▾ 编译 ▾ 帮助 ▾ 系统功能区 我的项目 向导 反馈 简体中文 ▾ test@example.com ▾

你好猫咪 项目名称 Screen1 ▾ 增添屏幕 删除屏幕 屏幕操作 切换视图 设计 编程

代码块

系统内置块

内置块

控制

逻辑

数学

文本

列表

颜色

变量

过程

Screen1

组件相关块

按钮

音效播放器1

加速度传感器1

任意组件

任意组件块

代码块分组

重命名 删除

工作区域

代码块抽屉

如果 则

针对从 1 到 5 且增量为 1 的每个 数

执行

针对列表 中的每一 项

执行

只要满足条件 就循环执行

如果 则 否则

执行 并输出结果

求值但不返回结果

显示告警

打开屏幕

代码块放置区

当 按钮1 ▾ 被点击时

执行 让 音效播放器1 ▾ 播放

当 加速度传感器1 ▾ 被晃动时

执行 让 音效播放器1 ▾ 振动

参数:毫秒数 200

垃圾桶

素材 资源文件列表

kitty.png

meow.mp3

上传文件



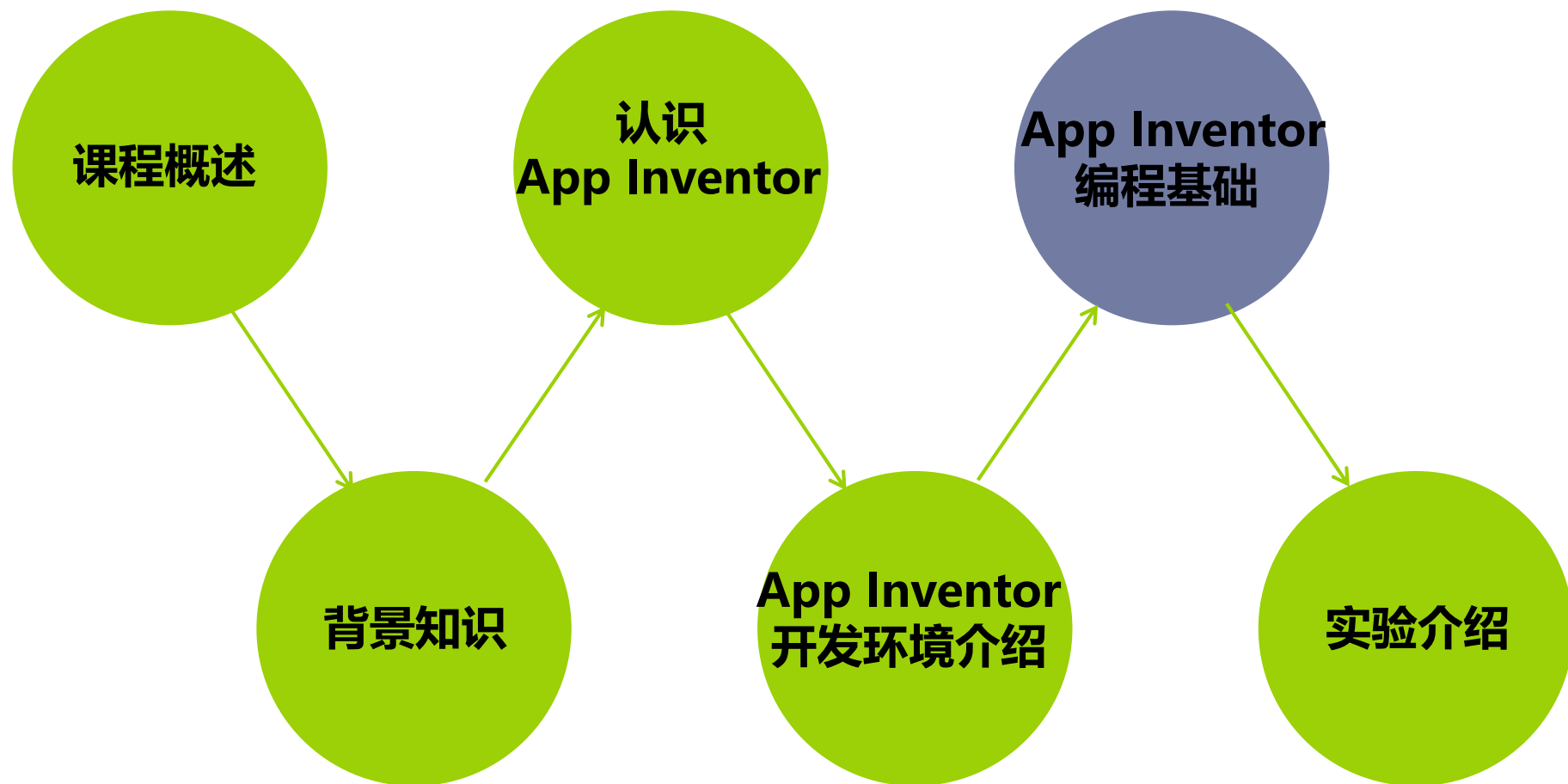
逻辑编程视图/Blocks Editor



- ▶ **代码块分组：**所有可用的代码块都存放在此，并划分为以下三组：
 - ▶ **内置块：**包含8类代码块，这些代码块与项目及项目中的组件无关，是编程语言的核心；
 - ▶ **与项目中组件相关的代码块：**这里列出了项目中的所有组件，选中任意组件，可以打开与该组件相关的代码块抽屉；
 - ▶ **与某一类组件相关的代码块：**这组代码块用于实现应用中的高级功能，即在程序运行过程中，动态设置某个组件的属性，或调用某个组件的内置过程。
- ▶ **代码块抽屉：**开发者可以选中任何一类代码块，即可打开该类代码块的抽屉，可以从抽屉中选择所需要的具体代码块；
- ▶ **代码块放置区：**当开发者从代码块抽屉中选中某个代码块后，抽屉将自动关闭，而被选中的代码块将自动被添加到该区域中；
- ▶ **垃圾桶：**在代码块放置区的右下角有一个垃圾桶，可以将需要删除的代码块拖拽到垃圾桶中（注意：桌面版没有撤销功能，在移动代码块时，注意避开垃圾桶，否则代码块一旦删除，将无法恢复！）；
- ▶ **素材：**显示项目中加载的所有资源文件。



课程内容





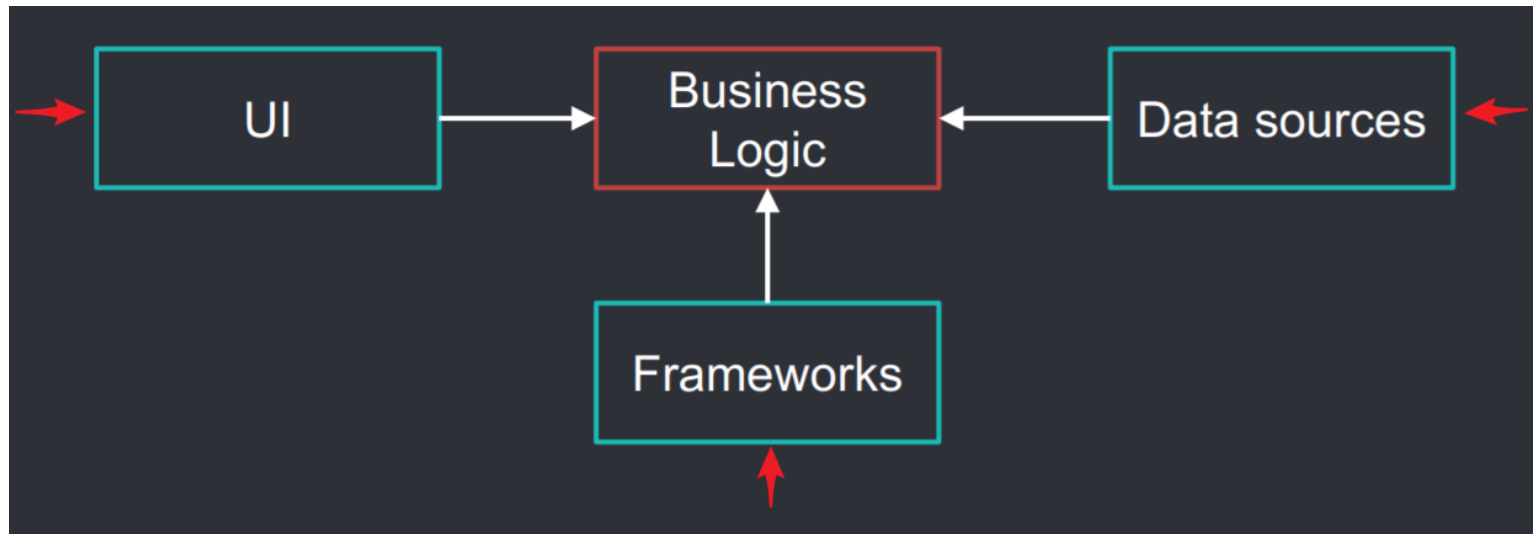
App Inventor编程基础

- ▶ GUI应用程序的基本“套路”
- ▶ 理解应用的内部结构
- ▶ 编程基础元素
- ▶ 组件



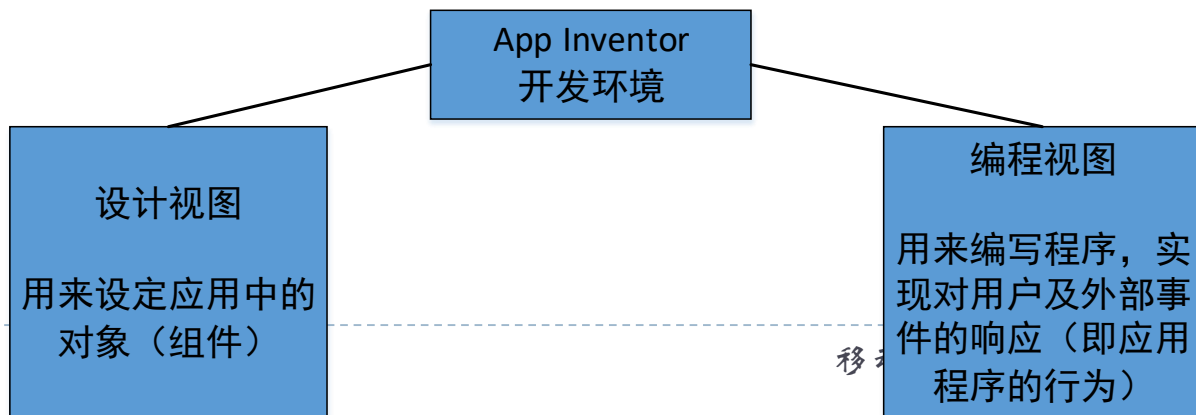
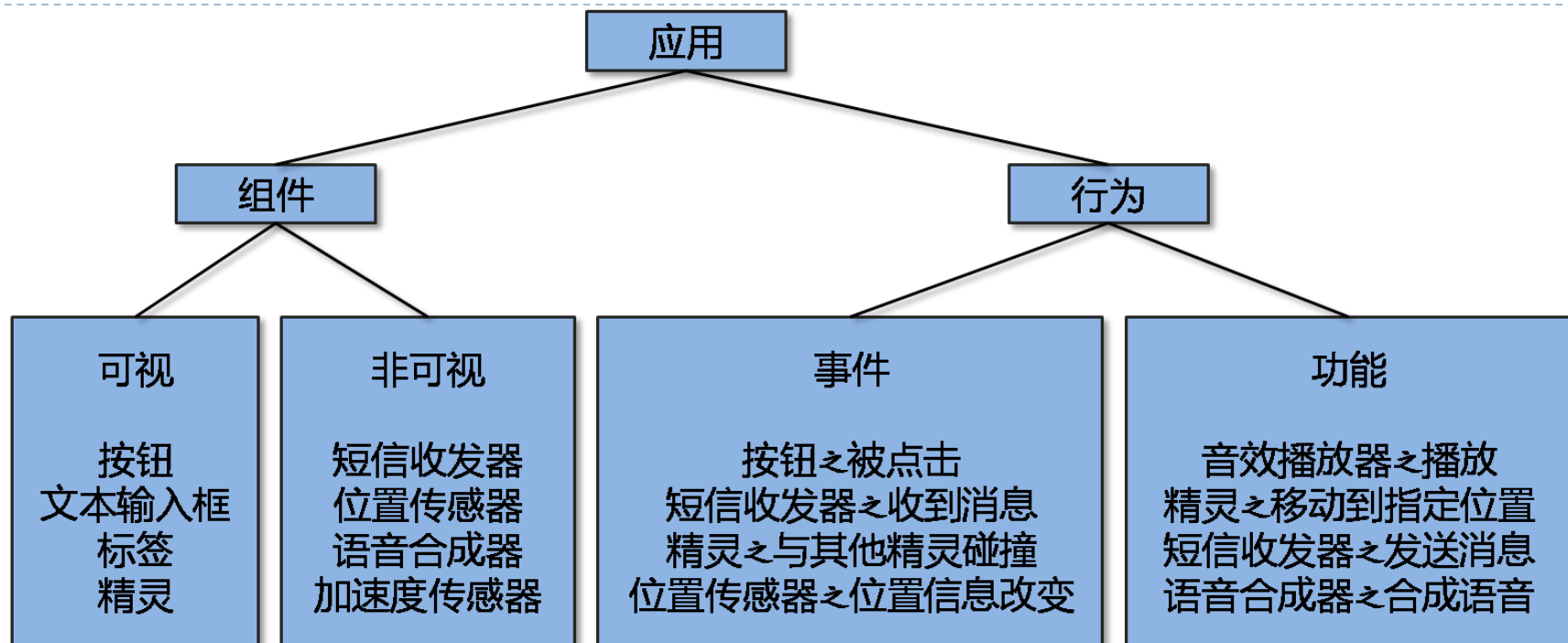
GUI应用程序的基本“套路”

- ▶ 关注点分离：UI交互+业务逻辑
 - ▶ 更多架构模式：MVC、MVP、MVVM...
- ▶ 事件驱动





应用的内部结构：组件 + 行为





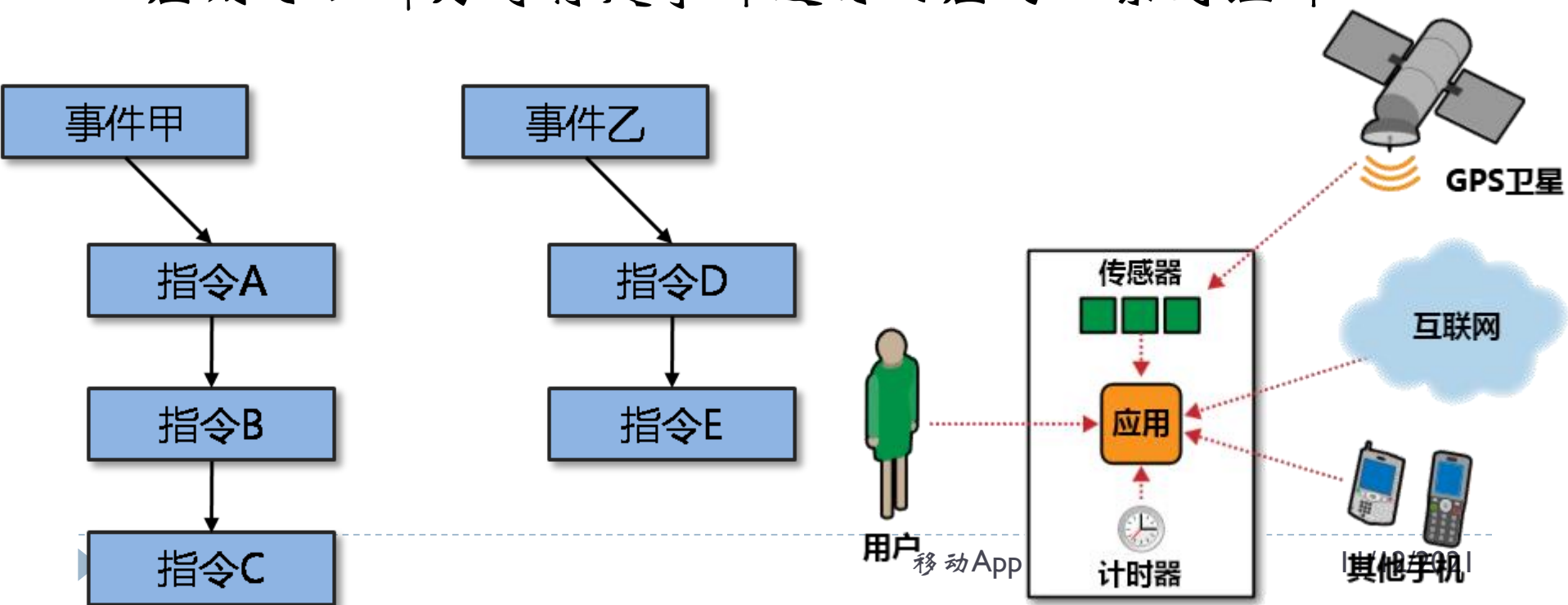
应用的组件：一组数据（属性）

- ▶ 应用中的组件分为两大类：
 - ▶ 可视组件：用于构成用户界面
 - ▶ 非可视组件：用于访问设备的内置功能
- ▶ 无论是可视组件还是非可视组件，都是由一组数据构成的，这组数据被称作属性。属性相当于组件信息的存储器。
 - ▶ 如可视组件的宽度、高度及对齐等属性，共同定义了组件的外观。
- ▶ 在设计视图中设定属性的值，来定义组件的初始样式。
 - ▶ 这些样式可以在编程视图中用代码来进行修改。



应用的行为：一组事件处理程序

- ▶ 现在的应用不再是单一的顺序执行一系列指令，而是对事件做出响应。当事件发生时，应用调用执行一系列的指令来实现对事件的响应。对事件进行响应的一系列指令统称为事件处理程序（Event Handler）。
- ▶ 应用可理解为对特定事件进行响应的一系列组件。





事件驱动编程方式

► 事件的类型









事件类型	举例
用户引发的事件	当用户点击确定按钮时，执行...
初始化事件	当应用启动时，执行...
计时器事件	当10毫秒过去时，执行...
动画事件	当两个物体碰撞时，执行...
外部事件	当收到来电时，执行...



编程基础元素

- ▶ 基本数据类型
 - ▶ 逻辑、数学、文本、颜色
- ▶ 基本控制结构
 - ▶ 顺序、分支、循环
- ▶ 数据的组织
 - ▶ 列表
- ▶ 控制的组织
 - ▶ 过程/函数
- ▶ 变量
 - ▶ 局部变量、全局变量
- ▶ 组件
 - ▶ 事件、属性和方法

内置块

-  控制
-  逻辑
-  数学
-  文本
-  列表
-  颜色
-  变量
-  过程



组件

- ▶ 用户界面组件
- ▶ 界面布局组件
- ▶ 多媒体组件
- ▶ 绘图动画组件
- ▶ 传感器组件
- ▶ 社交应用组件
- ▶ 数据存储组件
- ▶ 通信连接组件
- ▶ 乐高机器人
- ▶ 外部插件

组件面板

用户界面

界面布局

多媒体

绘图动画

传感器

社交应用

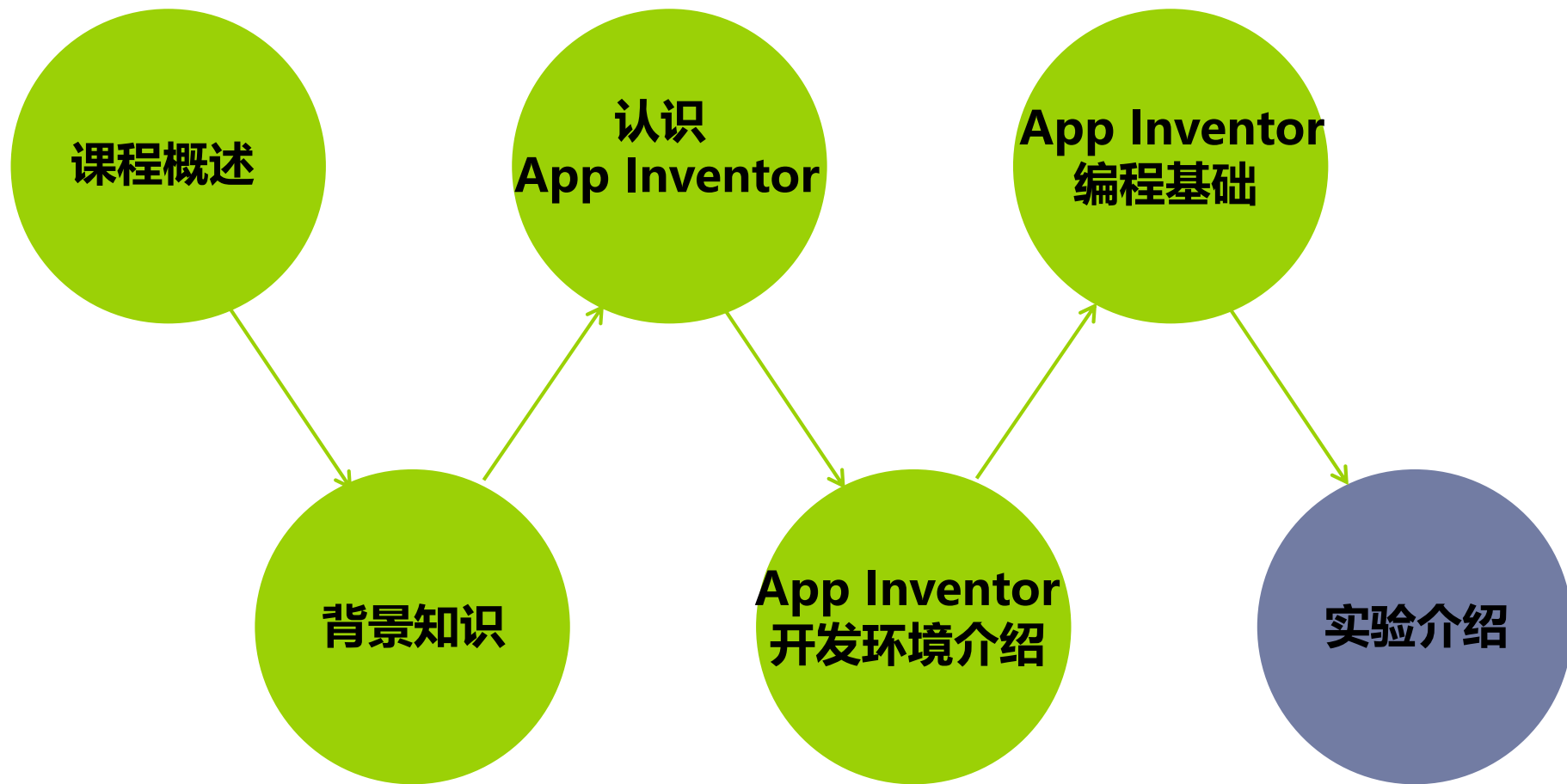
数据存储

通信连接

乐高机器人®



课程内容





实验题目



- ▶ 实验一、快速入门 —— 摇一摇登录校园网
- ▶ 实验二、Flappy Bird 1 —— 扬翅的小鸟
- ▶ 实验三、Flappy Bird 2 —— 下坠的小鸟
- ▶ 实验四、Flappy Bird 3 —— 避障的小鸟
- ▶ 实验五、Flappy Bird 4 —— 飞扬的小鸟
- ▶ 实验六、自主实验 (选做)

本课程规划了6个实验，前5个为给定题目的实验，最后一个自主开放实验。

第一个实验的目的是快速入门，熟悉App Inventor开发环境。此实验需要上机前提前完成。

第2-5个实验是循序渐进的，难度依次增大。最终目标是开发一款类似FlappyBird游戏的App，后一个实验需要在前一个实验的基础上增加功能，最后在第5个实验的时候完成最终的应用。

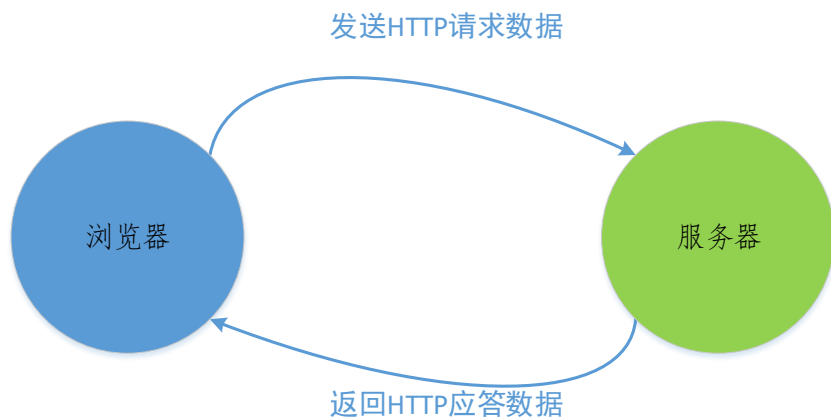
实验二三是基础实验，有比较详细的指导。供所有学生完成。

实验四五是提高实验，只给出基本步骤与提示。供学有余力的学生完成。



实验1

► 摇一摇登录校园网



北邮校园网登录

用户名 XXXX

密码

登录

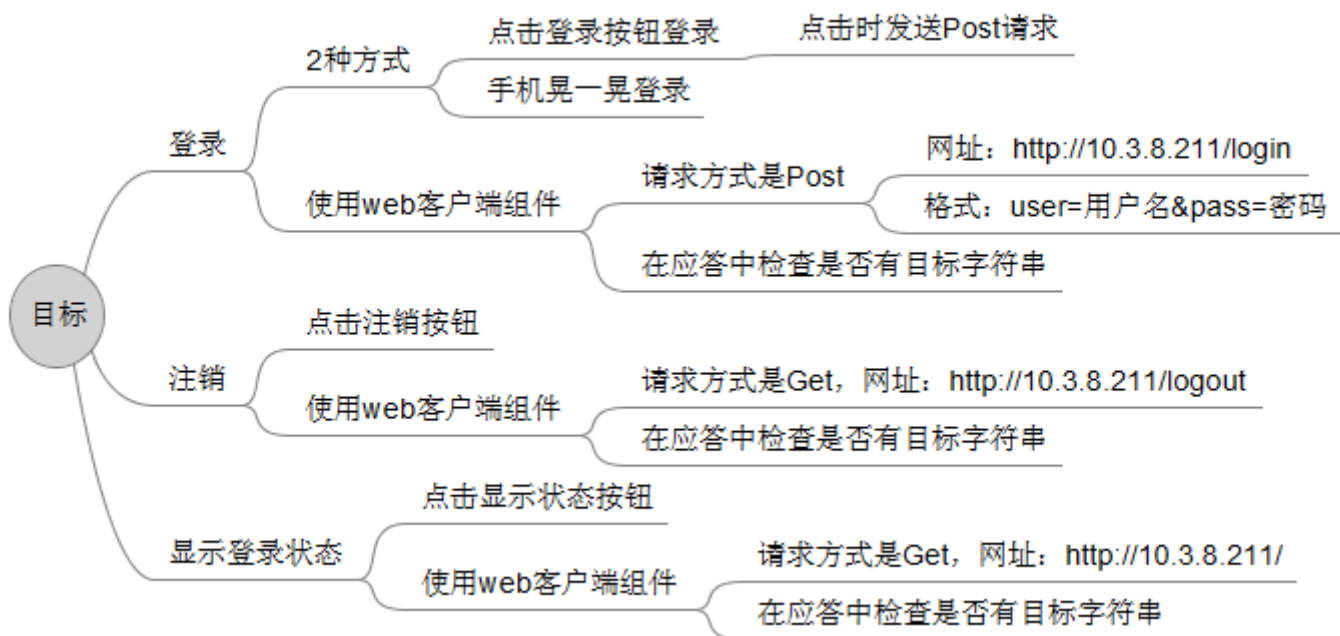
注销

显示状态





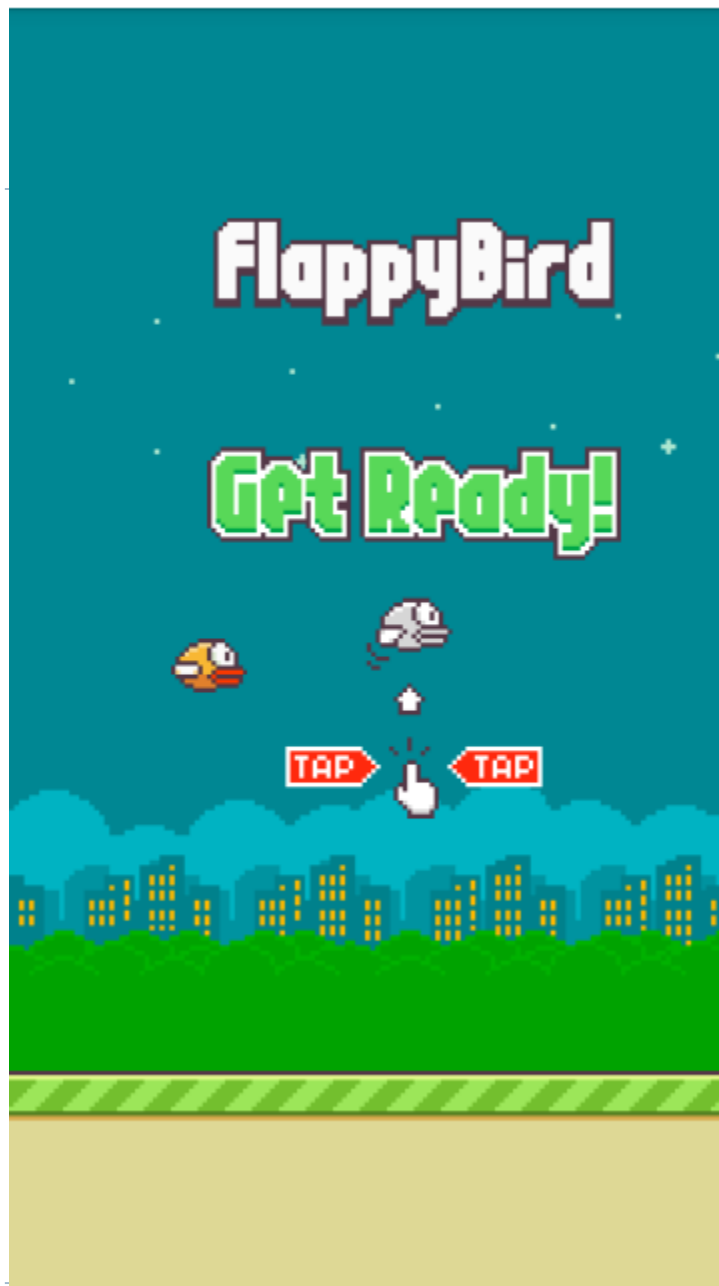
实验1 – 分析





实验2-5

► Flappy Bird





实验原理- 精灵动画

▶ 动画的实现

▶ 使用的动画技术：精灵动画（Sprite Animation）

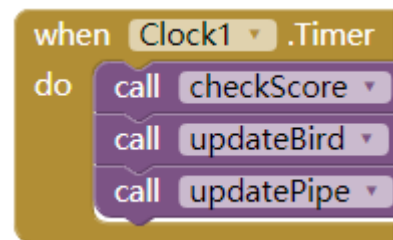
- ▶ 精灵动画是一种电子版的序列帧动画，是以快速连续播放不同的静止画面/帧，来产生角色的动态效果，就像电影胶片一般，通过快速切换来产生动画效果。
- ▶ 精灵（Sprite），就是一张张大小相同的位图，叠加在背景图像上。每一张图像都代表这个精灵在某一静止时刻的动作体现。

▶ 游戏的时间线

▶ 通过定时器（Clock）实现

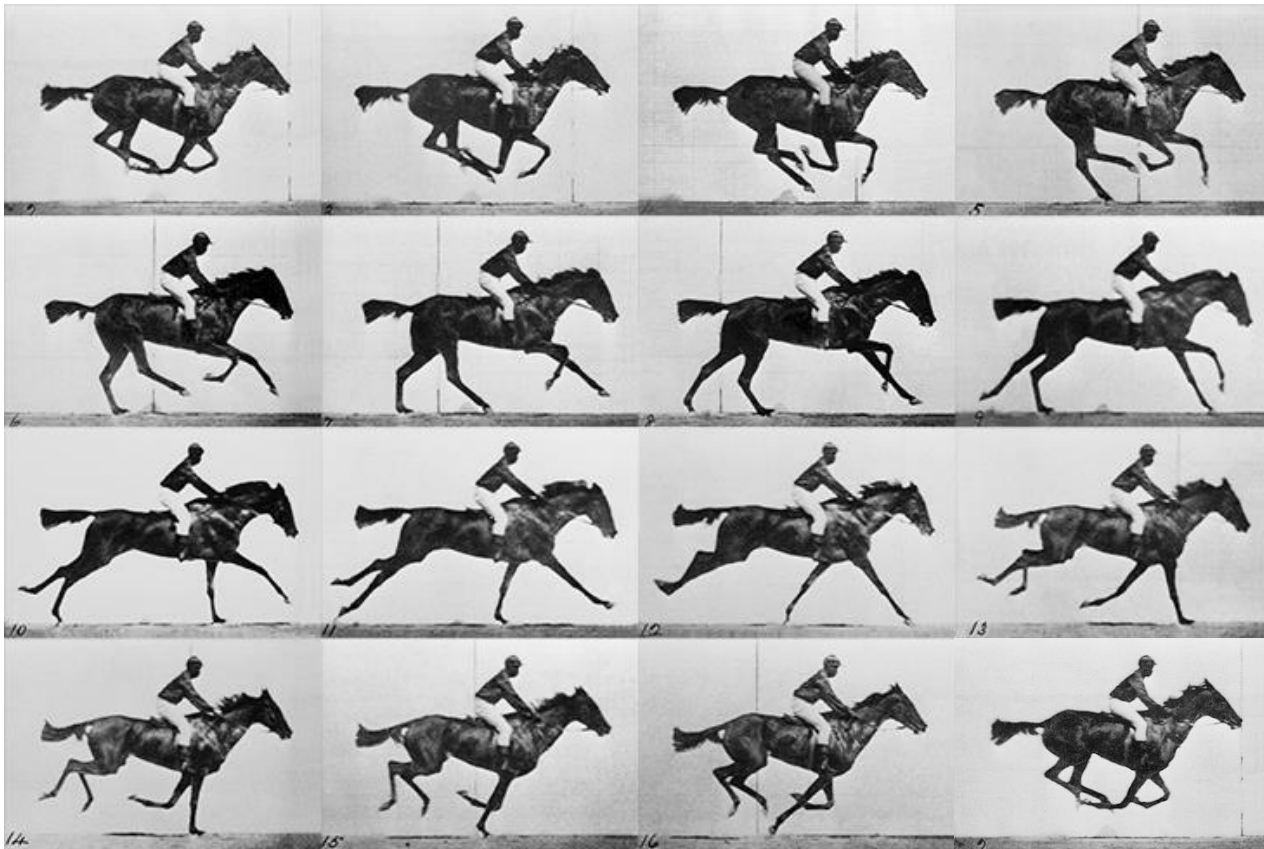
▶ 游戏场景

- ▶ 初始场景、游戏场景、结束场景
- ▶ 背景层（canvas）+前景层（sprite，小鸟、柱子）





实验原理 - 精灵动画





实验原理



▶ 扇动翅膀

- ▶ 实验二使用3张位图表示小鸟翅膀的上中下3个位置，再配合以小鸟Y坐标的增减，可模拟出扇动翅膀的动画效果。

▶ 垂直飞翔

- ▶ 实验三中小鸟在垂直方向做垂直上抛运动，通过给小鸟上升的初速度和向下的重力加速度，利用位移差分公式得到每次超时小鸟相对于上一次的Y方向位移改变量。在超时间隔固定的情况下，位移的改变量可通过Y方向速度的改变量来体现。

▶ 水平飞翔

- ▶ 实验四小鸟需要在水平方向通过柱子，实际上小鸟在水平位置是不动的，是柱子不停的在屏幕中从右往左移动，模拟小鸟在往前飞的效果。

Thank you!

