

移动学习平台的多图层管理技术研究

邢 邗¹ 孙连英² 石晓达¹ 葛 娜¹

¹(北京联合大学信息学院 北京 100101)

²(北京联合大学自动化学院 北京 100101)

摘 要 针对在 Android 移动设备实现具有“多图层”管理功能的绘图软件的过程中,位图(BitMap)过度占用内存资源导致出现卡顿与尾随的现象,甚至出现 OOM(Out Of Memory)错误,提出一种“三层”管理机制,不仅实现“多图层”管理功能而且确保绘图操作流畅。通过设计“画笔”类与内部数据结构,将包括坐标在内的用户绘图数据保存至 XML 格式的文件,达到跨平台共享数据的目的,不仅提高了互操作能力,同时改善了基于 Android 平台的移动学习系统性能。

关键词 Android 系统 “三层”管理机制 XML 文件 移动学习

中图分类号 TP311.52 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2017.08.018

RESEARCH ON MULTI-LAYER MANAGEMENT TECHNOLOGY OF MOBILE LEARNING PLATFORM

Xing Han¹ Sun Lianying² Shi Xiaoda¹ Ge Na¹

¹(College of Information Institute, Beijing Union University, Beijing 100101, China)

²(College of Automation, Beijing Union University, Beijing 100101, China)

Abstract During the process of implementing drawing software with “multi-layer” management on Android mobile devices, BitMap over-occupied memory resources results in caton and trailing phenomena, and even OOM (Out Of Memory) errors. Therefore, we presented a kind of “tri-layer” management mechanism that not only to achieve the “multi-layer” management function, but also ensured the smooth operation of drawing operation. We designed the “brush” class and the internal data structure, including the coordinates of the user to save the data to the XML format file, so as to achieve the purpose of sharing data across platforms. Our approach can not only improves the interoperability, but also improves the performance of a mobile learning system based on Android platform.

Keywords Android system Tri-layers management mechanism XML file Mobile learning

0 引 言

随着移动应用设备性能的不不断提升和价格的迅速下降^[1],移动学习软件的广泛应用促使人们的学习方式发生重大改变,从传统的教科书到 PC 端的数字化学习(E-learning)^[2],再到使用平板电脑等移动终端设

备进行的移动学习(Mobile learning)^[3-4]。移动学习是指在通信网络技术和无线移动通信设备的帮助下,不受时间与地点限制的学习^[5-6],是一种时间零散化的学习,是一种具有个性化和情绪化的学习,在培养学生的学习兴趣、学习热情和自学能力等方面有着其他学习方式无法比拟的优点^[7]。在学习过程中,用户遇到的操作不流畅或软件崩溃会破坏其对移动学习软件

的信任并减少其学习时间^[8],最终导致用户流失。所以移动学习软件的设计要体现以用户为中心满足用户功能需求和审美需求的思想^[9]。移动学习软件还需要满足教师与学生、学生与学生之间双向交流的功能,要求移动学习软件拥有保存用户的学习结果数据和跨平台共享学习数据的功能。

1 软件需求

在 Strategy Analytics 公布的 2014 年第二季度智能设备操作系统市场份额报表中,Android 系统占据 84.6% 的份额,占据市场主导地位^[10]。本文针对移动学习软件的需求,设计一款基于 Android 系统的移动绘画学习软件。本软件具有画笔、颜色、擦除、添加外部图片资源与文字等基本功能外,着重设计并实现“多图层”管理功能,方便用户通过图层管理绘画操作。“多图层”管理功能包括图层合并、添加与删除图层、移动缩放旋转图层和更改图层透明度等功能,用户在更改某一图层时,对其他图层无任何影响。

在 Android 系统中实现“多图层”功能的方法有很多,例如采用 FrameLayout 布局,使用多个 View 叠加实现“多图层”功能。创建一个 View 对象,在 View 中为每一个图层创建一个 BitMap 同样可以实现“多图层”功能。采用 LayerDrawable 对象也可以实现显示多张图片叠加的效果。当应用程序仅仅需要一个或两个图层时,占用的系统内存相对较少,不同方法之间呈现的应用效果近乎相同,同样可以满足用户的需求。但是当用户创建十几个、乃至创建几十个图层时,采用以上方法将占用大量内存资源,导致软件运行缓慢,甚至出现 OOM(内存溢出)现象,用户体验将急剧下降。本文针对内存资源管理问题,提出“三层”管理机制,实现“多图层”管理功能的基础之上确保用户绘图操作流畅,并且达到绘图操作流畅程度与用户创建的图层数量无关的目的。在使用“三层”管理机制的同时,软件兼具实现使用 XML 文件保存用户绘画学习结果数据功能。1997 年,W3C 就发布了 XML1.0 规范^[11-12],用来简化信息传输。XML 语言是当今处理结构信息的有效工具,多种编程语言支持 XML 文件的生成与解析。本文采用 XML 格式文件实现保存与恢复用户的绘画学习信息,有利于实现跨平台共享用户绘画数据和用户双向交流学习结果数据的功能,有利于提高移动学习乐趣。

本文实现的移动绘画学习软件采用 Android 系统中 Android. Graphics 包提供的二维图形处理 API(应用程序编程接口)实现。它包括图形处理相关的 Canvas(画板)、BitMap(图像)、Paint(画笔)、Matrix(矩阵)和 Point(点)类等。其中可以将 Canvas 理解为“画板”,BitMap 理解为画板上的“图像”,一块“画板”可以拥有多张“图像”,Paint 是用户绘画使用的“画笔”,Paint 可以设置颜色、大小和透明度等多种属性,更改 Paint 属性可以达到呈现多种绘画效果的目的。在软件运行过程中,首先创建一个 View 视图并获取 Canvas 对象,通过视图 View 的 View. touchEvent 函数捕获用户的绘画操作。其次 Canvas 通过 setBitmap 方法将 Bitmap 与 Canvas 进行绑定,用户当前的绘画操作显示在绑定的 Bitmap 中,最后调用视图 View 的 View. onDraw 方法,动态显示绘画效果。

2 “三层”管理机制

2.1 图层概念

在绘图软件中,“图层”的概念非常重要,许多效果可以通过“图层”的直接操作而得到。“图层”叠加后的效果是呈现在用户视觉中的最终效果。每一个绘图操作必须对应一个操作“图层”。以移动设备的屏幕为 X、Y 坐标轴平面,以垂直屏幕朝向用户方向为 Z 轴正方向。“多图层”可以理解为多个在 Z 轴坐标值不同的“图层”构成,“图层”本身透明不可见,如图 1 所示。“图层”在 Z 轴坐标值越大,代表该图层越接近用户,对 Z 轴坐标值小于它的“图层”有遮盖效果。所有“图层”按 Z 轴正方向依次叠加在一起就形成了屏幕上显示的最终效果。如果用户对其中一个“图层”做出缩放、旋转、混合效果等操作的时候,不会对其他“图层”造成任何影响。

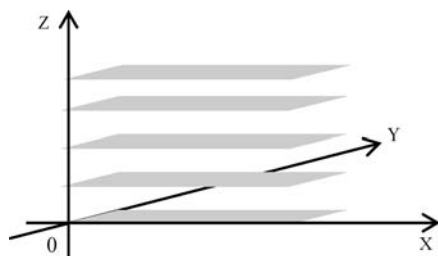


图1 “多图层”示意图

本软件中使用 PopWindow 组件实现图层管理窗口。点击图层管理按钮,在弹出的图层管理窗口内,包含添加图层功能的按钮和图层列表。图层列表中每一项展示本图层内绘画操作的缩略图、图层名称、图层是否可见和更改图层顺序按钮,用户正在编辑使用的图

层使用深灰色标识,如图 2 所示。



图 2 图层管理窗口

2.2 “三层”管理设计

“多层”管理功能包括新建图层、删除图层和图层功能三类,图层功能包括图层合并、更改图层顺序、清空图层内容、复制图层和更改图层透明度等。长按图层列表,弹出对应的图层功能窗口,如图 3 所示。用户在更改当前图层时,对其他图层无任何影响。



图 3 图层功能窗口

如果使用一个 BitMap 显示所有绘画操作,在更改图层属性时会对其他图层造成影响,违背“多层”管理功能的需求。如果为每一个图层建立一个 BitMap,需要占用大量的内存。通过实验发现,当创建四个或四个以上的 BitMap 并显示在屏幕上时,用户在绘画过程中出现尾随和卡顿的现象。“尾随”是指软件呈现绘画效果与用户手势移动存在一定时间延迟,如图 4 所示。“卡顿”是指软件呈现绘画效果不够流畅。需要显示的 BitMap 数量越多,占用内存越多,尾随和卡顿的现象越明显,直至出现内存溢出的错误,导致程序崩溃。

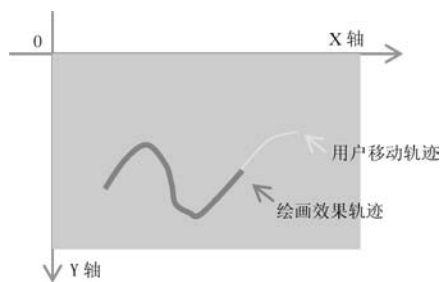


图 4 “尾随”现象示意图

“三层”管理机制将用户绘画操作绘制在三个 BitMap 中以实现“多层”效果,完全避免因“多层”功能需求创建 BitMap 导致的内存过度占用问题,支持用户创建无限多“图层”,并且用户绘图操作流畅。无论用户创建多少个图层,系统内部共需建立三个 BitMap,分别命名为顶层、当前操作层和底层,对应的 BitMap 分别命名为 UpBitMap、CurrentBitMap 和 DownBitMap,如图 5 所示。Canvas 可以通过 setBitMap 方法更改绑定操作的 BitMap。

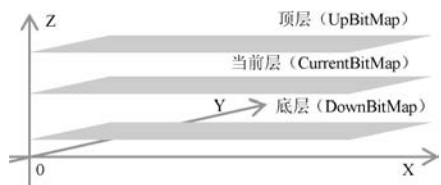


图 5 “三层”管理机制示意图

“三层”管理机制是将用户当前的绘画操作绘制在当前层,顶层中绘制大于当前层图层编号的所有图层对应的绘画操作,底层绘制小于当前层图层编号的所有图层对应的绘画操作。顶层对当前层和底层有遮盖效果,当前层对底层有遮盖效果。并且在用户操作当前层时,对其它图层无任何影响。

软件在新建一个绘画项目时,默认拥有一个图层,图层编号默认等于一。用户每新建一个图层,图层编号加一。创建一个 $\text{Map} \langle \text{Integer}, \text{List} \langle \text{Paint} \rangle \rangle$ tu-ceng 对象,保存用户绘图操作的 $\text{List} \langle \text{Paint} \rangle$ 序列与图层编号一一对应。

“三层”管理机制中必须遵守两个原则,一是用户绘图操作的始终是当前层。二是图层编号小于当前层图层编号的绘画内容绘制在底层中,图层编号大于当前层图层编号的绘画内容绘制在顶层中。

2.3 当前操作层编辑

假设共有 N 个图层,当前层为 $n (1 \leq n \leq N)$,用户期望当前层为 $m (1 \leq m \leq N, m \neq n)$ 。本着用户绘图操作的始终是当前层原则,更换当前层的步骤如下:

第一步 清空三个图层分别对应的 BitMap,可以调用 recycle 函数回收三个 BitMap,之后重建相同大小的三个 BitMap。

第二步 从 `tuceng` 对象中获取图层编号为 m 的 `List < Paint >` 序列对象,将 `List < Paint >` 序列中的绘画操作绘制到当前层对应的 `CurrentBitmap` 中。

第三步 从 `tuceng` 对象中获取图层编号小于 m 的所有 `List < Paint >` 序列对象,依照从小到大的顺序,依次将 `List < Paint >` 序列中的绘画操作绘制到底层对应的 `DownBitmap` 中。

第四步 从 `tuceng` 对象中获取图层编号大于 m 的所有 `List < Paint >` 序列对象,依照从小到大的顺序,依次将 `List < Paint >` 序列中的绘画操作绘制到顶层对应的 `UpBitmap` 中。

在更换当前层的过程中,存在两种特殊情况。一是期望当前层 m 等于图层总数 N ,即期望当前层是所有图层中的顶层。二是期望当前层 m 等于 1,即期望当前层是所有图层中的最底层。

当 m 等于图层总数 N 时,三个 `Bitmap` 依旧全部存在, `UpBitmap` 中没有任何绘画操作,图层编号小于 m 的用户绘图操作全部绘制在 `DownBitmap` 中,如图 6 所示。

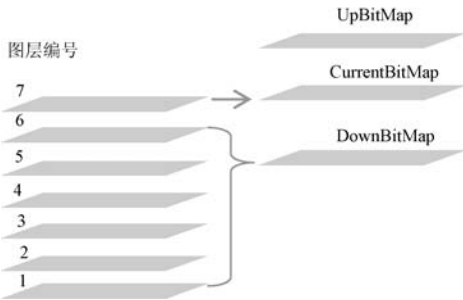


图 6 当前层为最顶层

当 m 等于 1 时, `DownBitmap` 中没有任何绘画操作,图层编号大于 1 的用户绘图操作全部绘制在 `UpBitmap` 中,如图 7 所示。

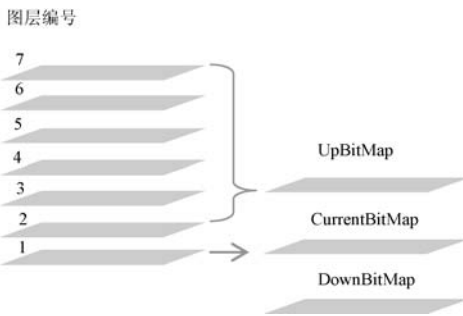


图 7 当前层为最底层

3 画笔与内部结构

3.1 画笔类设计

在本软件中,将用户的每一个绘画操作可以看作

是“画笔”(Paint)沿手势移动轨迹在“画布”(Bitmap)上的一次绘画。为了能够重现用户的绘画操作,需要保存“画笔”数据和手势移动轨迹,其中主要包括以下三类数据。

(1) 手势移动轨迹

手势移动轨迹可以使用一系列二维坐标表示。Android 移动设备屏幕坐标系以左上角为坐标轴原点,水平向右为 X 轴正方向,垂直向下为 Y 轴正方向。通过触屏事件 `Event` 获取用户触摸屏幕的坐标。触屏事件可以分落下 (`touchDown`)、移动 (`touchMove`) 和抬起 (`touchUp`) 三种事件。Android 系统默认获取坐标值是 `Float` 类型,为使用 `Point` 对象保存坐标值,将捕获的坐标值数据强制转换为 `Int` 类型。实例化一个 `List < Point >` 对象,将捕获的坐标点依次添加进 `List < Point >` 序列。代码如下:

```
X = (int)Event.getX();
Y = (int)Event.getY();
Point. x = X;
Point. y = Y;
List < Point >. add( Point );
```

(2) Paint 参数

Android. Graphics 包中 `Paint` 类提供更改画笔半径、颜色、透明度等属性方法。在画笔类中创建 `Paint` 对象和半径 (`size`)、颜色 (`color`)、透明度 (`alpha`)、样式 (`style`)、`Join`、`Cap`、`Align` 等变量,通过 `Paint` 对象的 `set` 方法设置各类属性,更改相应属性对应的变量值,可以呈现不同绘画效果。

(3) “画笔”类型

根据绘画效果可以将“画笔”类型分为点、曲线、基本几何图形、图片和文字五类,调用的绘画方法各不相同。“画笔”类型为“点”表示在捕获的手势坐标点的位置绘画一个点,并不连接相邻的坐标点。调用 `Canvas.drawPoint()` 方法,参数包括坐标点 X 与 Y 轴坐标值和实例化的 `Paint` 对象。“画笔”类型为“曲线”是使用 Android. Graphics 中提供 `Path` 类连接相邻的捕获坐标点,形成轨迹对象。连接相邻坐标点函数为 `Path.moveTo(x, y)` 和 `Path.quadTo(x1, y1)`,其中 (x, y) 和 (x_1, y_1) 表示相邻两个坐标点坐标值。调用 `Canvas.drawPath` 方法绘制轨迹,参数包括 `Path` 对象和实例化 `Paint` 对象。“画笔”类型为“基本几何图形”表示可以绘画包括直线、矩形、圆等基本几何图形。不同几何图形对应各自的绘画方法,例如绘画直线采用 `Canvas.drawLine` 方法,绘画矩形使用 `Canvas.drawRect` 方法,绘画圆采用 `Canvas.drawCircle` 方法。各自的参数中包括起始坐标和终点坐标,表示最终绘画几何图形的位置。

置只与捕获的落下事件坐标点和捕获的抬起事件坐标点有关,与移动过程无关。“画笔”类型是图片时,使用 `Bitmap.decodeResource` 方法读取指定图片资源成为 `Bitmap` 类型数据。调用 `Canvas.drawBitmap` 方法将图片绘画到指定坐标。“画笔”类型是文字时,获取用户动态输入的文字,调用 `Canvas.drawText` 方法,即可在指定坐标点或指定路径绘画文字。

3.2 “撤销”与“重画”功能实现

当一个绘画操作完成后,也就是用户的手势从落下到抬起后,需要将此绘画操作进行保存。创建一个 `List <Paint >` 序列对象命名为“ALL”,将每一个绘画操作创建的画笔添加进“ALL”序列对象。通过“ALL”达到保留所有用户操作的目的。

一般情况下,用户要求绘图软件拥有“撤销”和“重画”的功能。“撤销”就是取消最后一次或最后几次的绘画操作,“重画”就是将“撤销”的操作重新绘制到相应图层中。此时需要另一个与“ALL”序列对象同样类型的序列,命名为“SOME”。将取消的最后一次或最后几次操作依次添加进入“SOME”,并从“ALL”序列中删除,代码示例为:

```
SOME.add(ALL.get(ALL.size()-1));
```

```
ALL.remove(ALL.size()-1);
```

可以设定一个“撤销”功能允许的最大撤销步骤数目为 $M(M \geq 1)$ 。当用户绘画操作数为 $A(A > M)$ 时,将第一步绘画操作直到 $A - M$ 步绘画操作保存至文件并从保存绘画操作的序列“ALL”中删除,即可减少内存占用。

清空当前层对应的 `CurrentBitmap`,将“ALL”序列中的绘画操作从第一位开始,依次绘画到 `CurrentBitmap` 中,刷新显示,即可实现“撤销”功能。当需要“重画”功能时,将“SOME”中保存的绘画操作依次绘画到 `CurrentBitmap` 中,将重画的操作添加进入“ALL”序列中,并从“SOME”中删除,即可实现“重画”功能。

4 项目保存与恢复

在 Android 移动设备中使用 `Bitmap.compress` 方法将 `Bitmap` 保存成图片是每一个绘图软件的必备功能。用户之间共享的数据仅仅是一张图片,不具备绘画项目重新编辑功能,降低了移动学习的乐趣与效率。为实现绘画项目重新编辑的目标,解决数据共享中跨平台和兼容性的问题。本软件实现将具有“图层信息”的用户绘画操作数据保存为 XML 文件的功能,达到共享用户绘画操作信息的目的,提高用户数据交流和移动学习水平。保存的 XML 文件中标签根据“画笔”类

中参数定义,例如本软件中包含图层编号标签、“画笔”标签、坐标标签和颜色标签等,保存的 XML 文件示意图如下所示:

```
<tuceng number="1">
  <Paint1>
    <coordinate>
      <x>100</x>
      <y>100</y>
    </coordinate>
    <color>red</color>
  </Paint1>
  <Paint2>
  <Paint2>
</tuceng>
<tuceng number="2">
  <Paint1>
    <coordinate>
      <x>100</x>
      <y>100</y>
    </coordinate>
    <color>red</color>
  </Paint1>
  <Paint2>
  <Paint2>
</tuceng>
```

在打开与其他用户的绘画数据时,根据 XML 文件标签顺序,依次读取相应数据并更改内存中数据值。在读取数据结束后,将读取的绘画操作依次绘制到相应 `Bitmap` 中,实现项目恢复功能,满足不同用户之间学习交流的需求。

5 结 语

在 Android 移动应用开发中,终端有限的内存与对内存不断提升的需求始终是矛盾的。移动学习软件采用“三层”管理机制,实现了绘图软件的“多图层”管理功能,确保用户绘图体验流畅。并且通过设计“画笔”类保存用户绘画操作数据,实现“撤销”与“重画”功能。在文件保存和用户交流中,利用多平台支持 XML 文件生成与解析的特点,实现项目恢复与跨平台用户交流绘画数据的功能。与仅将用户绘画数据保存

(下转第 161 页)

4 结 语

本文研究了多小区场景下无线携能通信的功率分配问题,大规模天线技术的引入使得基站与接受端之间的通信链路不再受到小尺度衰落的影响,从而提高了下行链路数据传输的可靠性和稳定性。本文提出的功率分配方案在满足接收端获得的信干噪比和能量采集约束的基础上,通过削减基站端给信道状态良好用户的发送功率,大大降低了基站发射的合功率。一方面,根据本文推导出的信干噪比下限,仿真结果表明在发射端配备大规模天线的携能传输技术降低了基站端功耗;另一方面,本文提出的功率分配方案优于平均功率分配,为基站端节约了大量能源。

参 考 文 献

- [1] Yang S, Hanzo L. Fifty Years of MIMO Detection; The Road to Large-Scale MIMOs[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 1941 - 1988.
- [2] Marzetta T L. Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas[J]. IEEE Transactions on Wireless Communications, 2010, 9(11): 3590 - 3600.
- [3] Ngo H Q, Larsson E G, Marzetta T L. Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems[J]. IEEE Transactions on Communications, 2013, 61(4): 1436 - 1449.
- [4] Gao X, Tufvesson F, Edfors O, et al. Measured propagation characteristics for very-large MIMO at 2.6 GHz[C]//2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, 2012: 295 - 299.
- [5] Zhang Q, Jin S, McKay M, et al. Power Allocation Schemes for Multicell Massive MIMO Systems[J]. IEEE Transactions on Wireless Communications, 2015, 14(11): 5941 - 5955.
- [6] Varshney L R. Transporting information and energy simultaneously[C]//IEEE International Symposium on Information Theory. IEEE, 2008: 1612 - 1616.
- [7] Grover P, Sahai A. Shannon meets Tesla: Wireless information and power transfer[C]//IEEE International Symposium on Information Theory Proceedings. IEEE, 2010: 2363 - 2367.
- [8] Zong Zhiyuan, Feng Hui, Yu F R, et al. Optimal Transceiver Design for SWIPT in K-User MIMO Interference Channels[J]. IEEE Trans. Wireless Communications, 2016, 15(1): 430 - 445.
- [9] 楼亦厦. LTE-Advanced 系统中大规模天线技术研究[D]. 上海交通大学, 2015.
- [10] Huang K, Zhou X. Cutting the last wires for mobile communications by microwave power transfer[J]. IEEE Communications Magazine, 2014, 53(6): 86 - 93.
- [11] Hong Y, Marzetta T L. Performance of Conjugate and Zero-

Forcing Beamforming in Large-Scale Antenna Systems[J]. IEEE Journal on Selected Areas in Communications, 2013, 31(2): 172 - 179.

- [12] org. cambridge. ebooks. online. book. Author@ dd. Convex Optimization[J].
- [13] Bi S, Ho C K, Zhang R. Wireless powered communication: opportunities and challenges [J]. IEEE Communications Magazine, 2014, 53(4): 117 - 125.
- [14] Tulino A M, Verdú S. Random Matrix Theory and Wireless Communications[J]. Communications & Information Theory, 2014, 1(1): 1 - 182.
- [15] Lv T, Yang S, Gao H. Semi-Blind Channel Estimation Relying on Optimum Pilots Designed for Multi-Cell Large-Scale MIMO Systems[J]. IEEE Access, 2015, 4: 1190 - 1204.

(上接第 101 页)

为图片的功能相对比,无疑提高了编辑绘画项目的效率和灵活性,增强了移动学习系统的性能。

参 考 文 献

- [1] 潘志庚,何高奇,杨柏林,等. 移动设备图形技术的研究进展[J]. 计算机辅助设计与图形学学报, 2008, 20(3): 273 - 280.
- [2] 冯瑶,冯锡炜. 基于本体的教育资源推理查询原型系统设计与实现[J]. 计算机应用与软件, 2016, 33(10): 14 - 18.
- [3] 李向东,杨昕. 基于 Android 平台的移动学习系统的设计与实现[J]. 计算机时代, 2014(4): 24 - 26.
- [4] 刘帅,李建伟,勾学荣. 互动式移动学习系统的设计与开发[J]. 北京邮电大学学报(社会科学版), 2014, 16(6): 111 - 116.
- [5] 赵丞兵. 基于 Android 平台的移动学习系统设计与实现[D]. 北京邮电大学, 2012.
- [6] 孟祥武,史艳翠,王立才,等. 用户对移动网络服务偏好学习技术综述[J]. 通信学报, 2013(2): 147 - 155.
- [7] 陈锦昌,刘菲,陈煌,等. 基于工程制图课程投影知识的移动学习游戏设计[J]. 东华大学学报(自然科学版), 2016(4): 608 - 616.
- [8] 刘建设,李青,刘金梅. 移动学习研究现状综述[J]. 电化教育研究, 2007(7): 21 - 25.
- [9] 罗仕鉴,龚蓉蓉,朱上上. 面向用户体验的手持移动设备软件界面设计[J]. 计算机辅助设计与图形学学报, 2010, 22(6): 1033 - 1041.
- [10] 杨红丽,姜皓,秦胜潮,等. Android 应用能耗漏洞分析技术综述[J]. 计算机应用与软件, 2016, 33(9): 1 - 6.
- [11] 郑黎晓,王成. XML 模式推断研究综述[J]. 电子学报, 2016, 44(2): 461 - 467.
- [12] Zhao H, Chen M, Qiu M, et al. A novel pre-cache schema for high performance Android system [J]. Future Generation Computer Systems, 2016, 56(C): 766 - 772.