

Android 英文朗读功能的实现

徐尤华, 熊传玉

(广东松山职业技术学院, 广东 韶关 512126)

摘要: 给出了 Android 下朗读英文文章的实现方法。

关键词: Android; 语音; 应用

The Realization of the English Reading in Android

XU Youhua, XIONG Chuanyu

(Guangdong Songshan Polytechnic College, Guangdong Shaoguan 512126)

Abstract: This article gave the realization method of the English reading in Android.

Key words: Android; speech; application

1 引言

英文朗读软件目前市面上应该算是比较多的,但这些软件多数都只能在 PC 机上使用,或者做成了 PDA (Personal Digital Assistant) 的形式,在智能手机迅速普及的今天,可以把英文朗读的功能移植到智能机上,这样就可以让用户随时随地地学习英语或者收听英文文章,并且不需要额外的硬件支持。

2 应用界面

应用运行界面如图 1 所示。

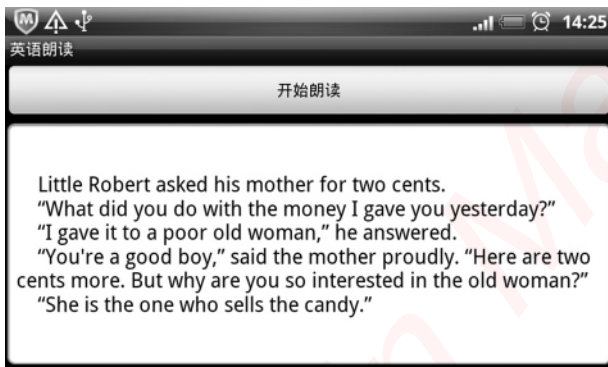


图 1 应用运行界面

3 功能分析

应用运行界面上提供两个控件: 一个按钮和一个文本域。用户先在文本域中输入要朗读的英文内容,然后点击按钮,系统就会开始朗读。

4 功能实现

Android 环境下英文的朗读可以通过 android.speech.tts.TextToSpeech 类来实现。

下面给出 Eclipse 下该应用的实现过程。

4.1 建立项目

新建一个 Android Project, 项目名设为 TestSpeech, 主 Activity 名为 SpeechEnglish。

4.2 布局设计

打开 layout 文件夹下的 main.xml 文件, 删除系统自动创建的标签, 并在该布局文件中加入一个按钮和一个文本域控

件, 并设置好控件 ID。

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <Button android:layout_height="wrap_content"
        android:layout_gravity="center" android:layout_width="match_parent"
        android:text="开始朗读" android:id="@+id/button1" ></Button>
    <EditText android:layout_width="match_parent" android:id="@+id/editText1"
        android:hint="输入要朗读的英文文章" android:layout_height="match_parent" >
    </EditText>
</LinearLayout>
```

4.3 Activity 设计

打开 SpeechEnglish.java 文件, 在该活动中提供朗读方法的实现。

android.speech.tts.TextToSpeech 类提供了英文的朗读方法, 该类构造方法声明如下: public TextToSpeech (Context context, TextToSpeech.OnInitListener listener), 要求提供两个参数: 第一个是实例所在的 Context, 一般用 this 表示 TextToSpeech 类的实例在当前 Activity 中运行; 第二个参数要求是一个 TextToSpeech.OnInitListener 的监听器, 用于对 TextToSpeech 引擎初始化。

要朗读英文内容, 只要调用 TextToSpeech 的 speak () 方法即可, 该方法声明如下: public int speak (String text, int queueMode, HashMap<String, String> params), 要求提供 3 个参

(下转到 126 页)

作者简介: 徐尤华 (1977-), 男, 讲师, 硕士, 研究方向: 计算机应用与软件; 熊传玉 (1979-), 女, 讲师, 硕士, 研究方向: Web 开发。

收稿日期: 2011-10-26

所以它们默认就是静态的。

举例创建一个长方形物体，第一步是定义一个形状。

```
var bigLongShapeDef:b2PolygonDef = new b2PolygonDef();
bigLongShapeDef.vertexCount = 4;
b2Vec2 (bigLongShapeDef.vertices [0]) .Set (0, 0) ;
b2Vec2 (bigLongShapeDef.vertices [1]) .Set (550, 0) ;
b2Vec2 (bigLongShapeDef.vertices [2]) .Set (550, 10) ;
b2Vec2 (bigLongShapeDef.vertices [3]) .Set (0, 10) ;
```

上述程序定义了一个多边形的形状，同时定义了该形状有 4 个顶点和每个顶点的位置。

```
bigLongShapeDef.friction = 0.5;
```

这用来定义两个对象之间的摩擦，可以在 0.0-1.0 之间调整它们。

```
bigLongShapeDef.restitution = 0.3;
```

设定碰撞弹性系数。

```
bigLongShapeDef.density = 0.7;
```

设置密度，在碰撞的等式中使用密度 * 面积=质量，密度如果是 0 或者 null，将会是一个静止的对象。

第二步是定义一个物体。

```
var rigidBodyDef:b2BodyDef = new b2BodyDef () ;
rigidBodyDef.position.Set (0, 390) ;
```

第三步，在世界中创建物体。

```
var rigidBody:b2Body = _world.CreateBody (rigidBodyDef) ;
```

(上接第 118 页)

数：第一个是要朗读的内容；第二个是语音消息队列的管理策略，有 QUEUE_ADD 和 QUEUE_FLUSH 两种取值，设为 QUEUE_ADD 表示将发音任务加入到当前任务队列之后，设为 QUEUE_FLUSH 表示中断当前实例正在运行的任务；第三个参数用于传入一些数据，一般设为 null 即可。

当 Activity 关闭时最好调用 TextToSpeech 的 stop () 方法来停止当前朗读任务，并接着调用 shutdown () 方法释放 TextToSpeech 引擎所占用的资源。

核心代码如下：

在类中定义成员：

```
TextToSpeech speech;
```

在 onCreate () 方法中构造 TextToSpeech 对象：

```
speech=new TextToSpeech (this,this) ;//当前 Activity 必须要
```

//实现 TextToSpeech 的 OnInitListener 接口

按钮点击后的响应代码：

```
speech.speak (edit.getText () .toString () , TextToSpeech.
QUEUE_FLUSH, null) ;
```

//上行中出现的 edit 为 UI 中的文本域控件。

TextToSpeech.OnInitListener 接口的 onInit(int status)方法：

```
if (status==TextToSpeech.SUCCESS)
```

```
{
    int result=speech.setLanguage (Locale.US) ;
    //设置语音为美式英语
```

```
if (result==TextToSpeech.LANG_MISSING_DATA ||
result==TextToSpeech.LANG_NOT_SUPPORTED)
```

最后，在物体上创建形状。

```
rigidBody.CreateShape (bigLongShapeDef) ;
rigidBody.SetMassFromShapes () ;
```

3.4 让世界运动

使用世界的 Step 函数来计算世界中物体的位置，执行后，物体的位置、角度、速度等信息更新。

3.5 显示世界

由于 Box2D 中所有对象都是不可见的，要创建可见对象得用刚体定义中的 userData 属性来创建自己的图形，如果不创建自己的图形，则用 b2DebugDraw 类来实现物体的可见，方便调试。

4 结语

Box2D 是一个非常优秀的 2D 开源物理引擎，虽然它主要是为游戏而开发的，但是现在使用范围却越来越广泛。介绍了使用 Box2D 物理引擎进行刚体运动和碰撞的模拟步骤和方法，并举例加以解释。

参考文献

- [1] 李健. 游戏中物理动画应用的研究 [D]. 重庆：重庆大学, 2008.
- [2] <http://box2d.org>.
- [3] Erin Catto.Box2D user manual.2007-2011.

```
{
    Toast.makeText (this, " 不支持指定语言" ,
    Toast.LENGTH_LONG) .show () ;
}
```

Activity 关闭后的处理代码：

```
protected void onDestroy () {
    super.onDestroy () ;
    if (speech! =null)
    {
        speech.stop () ;//终止当前朗读任务
        speech.shutdown () ;//释放引擎占用的资源
    }
}
```

5 结语

只是给出了 Android 系统下英文文章的朗读方法的实现，在实际开发过程中可以根据具体需求开发相应功能的应用，例如在游戏中提供语音提示功能、个性闹钟的报时功能等。

参考文献

- [1] 王向辉. Android 应用程序开发. 北京：清华大学出版社, 2010.
- [2] 林城. Google Android 2.X 应用开发实战. 北京：清华大学出版社, 2011.
- [3] <http://www.iteye.com>.
- [4] <http://developer.android.com>.

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)
30. [总线协议中的 CRC 及其在 SATA 通信技术中的应用](#)
31. [基于 FPGA 的 SATA 硬盘加解密控制器设计](#)
32. [Modbus 协议在串口通讯中的研究及应用](#)
33. [高可用的磁盘阵列 Cache 的设计和实现](#)
34. [RAID 阵列中高速 Cache 管理的优化](#)

35. [一种新的基于 RAID 的 CACHE 技术研究与实现](#)
36. [基于 PCIE-104 总线的高速数据接口设计](#)
37. [基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)
22. [基于 VxBus 的高速数据采集卡驱动程序开发](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)

7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)
24. [掌握 Android 7.0 新增特性 Quick Settings](#)
25. [Android 应用逆向分析方法研究](#)
26. [Android 操作系统的课程教学](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

17. [Windows CE 环境下无线网卡的自动安装](#)
18. [基于 Windows CE 的可视电话的研究与实现](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)
11. [基于 PowerPC 处理器 SMP 系统的 UBoot 移植](#)
12. [基于 PowerPC 双核处理器嵌入式系统 UBoot 移植](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)

15. [CortexA8 平台的 \$\mu\$ C-OSII 及 LwIP 协议栈的移植与实现](#)
16. [基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计](#)
17. [ARM S3C2440 Linux ADC 驱动](#)
18. [ARM S3C2440 Linux 触摸屏驱动](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
13. [基于 UEFI 固件的恶意代码防范技术研究](#)
14. [MIPS 架构计算机平台的支持固件研究](#)

Programming:

1. [计算机软件基础数据结构 - 算法](#)