

基于 Elastos 操作系统的图形引擎的设计与实现

张金焕, 陈 榕, 张曼夕

(同济大学 基础软件工程中心, 上海 200092)

摘 要: Elastos 图形系统 Vinci 是 Elastos 操作系统的一个子系统, 负责在屏幕上绘制像素、文本和用户界面(包括窗口、菜单、对话框)等。图形引擎的体系结构决定了此系统的简易性、灵活性、效率和稳定性, 所以它的设计又显得至关重要。分别介绍了基于 Elastos 操作系统的图形系统 Vinci、此图形引擎的设计需求, 在此基础上根据图形系统的引擎功能设计需求结合 Elastos 图形系统 Vinci 自有的特点, 实现了 Vinci 引擎的设计, 使得基于 Vinci 系统的 Elastos 操作系统可以完成各种各样的操作。

关键词: 图形系统 Vinci; 图形引擎; 设计需求

中图分类号: TP316

文献标识码: A

文章编号: 1673-629X(2008)11-0044-03

Design and Realization of Graphics Engine Based on Elastos Operating System

ZHANG Jin-huan, CHEN Rong, ZHANG Man-xi

(System Software Engineering Centre of Tongji University, Shanghai 200092, China)

Abstract: Elastos graphics system Elastos Vinci is a subsystem of the operating system, responsible for drawing pixels on the screen, text and the user interface (including windows, menus, dialog). Graphics engine determines the architecture of the same easy-to-use, flexibility, efficiency and stability, so it is essential. This paper introduces the graphics-based operating system Elastos Vinci system, the design of the graphics engine demand, and according to graphics system and the engine function design demand and Elastos graphics system Vinci's own characteristics, realize the Vinci engine design which makes the operating system Elastos based on Vinci system can completed a wide range of operations.

Key words: Vinci; graphics engine; engine demand

0 引 言

通常, 把引擎看作是一种动力装置, 图形引擎也一样, GDI(Graphic Device Interface)之所以可以完成各种各样的操作, 就是依赖于底层图形引擎的支持。Elastos 图形引擎^[1]由五部分组成: 窗口管理、事件管理、绘图、字体和图像处理, 它提供了完成窗口、事件、绘图、字体管理和图像处理所需的各项操作, GDI 就是将这些底层操作封装起来, 使用户可以在不了解底层实现的基础上方便地调用底层的函数, 完成各项操作。介绍了 Elastos 操作系统中图形系统中图形引擎的设计及具体实现过程。

1 Elastos 图形系统—Vinci

1.1 Vinci 系统简介

Elastos 图形系统 Vinci 是 Elastos 操作系统^[2]的一个子系统, 负责在屏幕上绘制像素、文本和用户界面(包括窗口、菜单、对话框)等。Elastos 图形系统还负责和硬件设备的通信, 所以在 Elastos 操作系统上编写图形应用程序时, 应用程序不需要直接访问硬件设备。

1.2 Vinci 系统整体架构

从整体结构上看, Elastos 图形系统 Vinci 是分层设计的, 见图 1。底层是驱动层; 中间层为 Vinci 的图形引擎层, 其中包括了窗口系统必不可少的各个模块, 负责创建 Window(实际对应上层 Form), 收发消息, 以及一些基本绘图操作; 顶层是控件集^[3], 是用户可见的, 与用户进行交互的主要一层, 多种控件为用户建立丰富的用户界面提供了保障。

在 Vinci 系统 UI 模型中, Gfx.dll 作为低层图形引擎为绘图引擎 eladgi.dll 和控件组件 elactrl.dll 提供

收稿日期: 2008-03-13

基金项目: 国家“863”基金项目(2001AA113400)

作者简介: 张金焕(1983-), 女, 山东人, 硕士研究生, 研究方向为嵌入式操作系统、图形系统; 陈 榕, 博士生导师, 教授, 研究方向为嵌入式系统、构件技术。

了底层服务。elagdi.dll 处于顶层控件于引擎进行绘图及窗口处理的交互中间态,也提供一些独立的 API,可以直接通过这些 API^[4]与引擎进行交互;elactrl.dll 作为 UI 控件构件为应用程序的 UI Application 提供了应用界面服务。而控件^[5]中的所需要进行的绘图操作需 elagdi.dll 引擎中的 font、image 的支持。

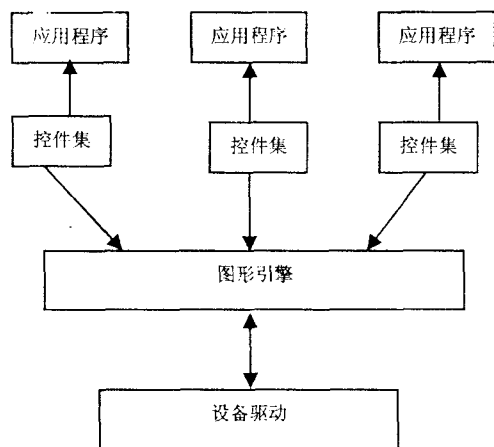


图 1 Vinci 整体结构图

2 Vinci 图形系统引擎设计

2.1 Vinci 图形引擎功能需求

2.1.1 窗口的创建和销毁

由 CreatWindow 创建窗口。CreateWindow 可以创建用户定义属性的窗口,包括窗口大小、像素格式、窗口属性等。函数返回一个窗口指针,用于以后的窗口操作。由于窗口上的绘图操作都是在对应的 surface 上进行的,所以与 CreateWindow 对应的 CreateSurface 方法,用于创建 window 对应的 surface,与 CreateWindow 类似的是,创建 surface 时也可以指定一系列参数,包括 surface 大小与 surface 像素格式等。

2.1.2 在窗口中绘图

每一个窗口对应一块内存,即 surface,在窗口内绘图就是将相应的像素内容填入 surface 内存中,当然,首先要将像素格式转换为目的显存支持的格式。这一系列转换格式及写入内存的操作都是引擎处理的重点。

2.1.3 消息队列的处理

Vinci 不直接与硬件交互,通过驱动层与硬件交互。图形系统的消息除了来自驱动层传入的触笔和键盘消息,还来自窗口的变换等,例如,窗口的切换和拖动将触发重绘消息;如果用户使用了定时器,还存在定时消息的处理等等。

2.1.4 窗口的绘制

屏幕上出现多个窗口时,就存在 Z 序处理问题^[6]。

2.2 引擎设计

综合上述引擎功能的需求,初步将引擎结构设计为由以下几个接口分工合作完成:ILayer 接口:负责 window 和 surface 的创建,surface 的配置,layer 透明度的设置等。IWindow 接口:负责 window 的显示、隐藏、移动、定时器的设置与销毁、透明度的设置、向消息队列增加新消息等。ISurface 接口:负责简单的图形绘制以及窗口信息的获得等。Eventbuffer 接口:负责管理消息,包括增加消息、获取消息等处理消息的操作。Windowstack 接口:负责窗口的管理,多个窗口创建后以队列形式被管理。窗口的创建、销毁、移动等都需要在 windowstack 中留下记录。

3 Vinci 引擎的实现

3.1 显示层 ILayer 的实现

显示层主要负责创建 window 和 surface,另外还需要设置方法以进行显示层的切换和透出等特殊效果。ILayer 接口中,主要使用到以下几个重要的数据结构:LayerFormat:该数据结构记录了某一层的基本信息。当用户要求 Enable 某一层时,将以此结构体类型的变量为参数传入 Layer 的 Enable()方法。DisplayInformation:该数据结构记录了层的显示信息。在 Enable Baseplane、Overlay1 或 Overlay2 任一层时,需要对 DisplayInformation 的各项赋值,将用于 windowstack 中。类 buffer:主要用来记录一些层的信息,用于 surface 的绘制。每一个 window 都对应了一块 surface(即内存空间),如果 window 是 Doublebuffer 模式,则独占一块内存空间,否则和其它 window 共用一块内存空间。上层对于 window 或 Form 的绘制操作都是在内存中进行的,最后通过 Flip 操作将内存中的数据 copy 到当前层的显存上,才能显示在屏幕上。

3.2 窗口队列管理的实现

windowstack 中使用了一个结构体 FusionVector 记录各项参数,包括 windowstack 中的 window element,window 的数量以及 windowstack 的最大容量。每层 layer 具有一个 windowstack 队列,各个窗口被创建后,将被加入相应 layer 的 windowstack 中,作为一个 Fusion 元素。各窗口在队列中的次序按照优先级排列,如果用户没有特别指定窗口优先级,后创建的窗口默认优先级较高,对应的视觉效果就是显示在上层,覆盖先创建的窗口。windowstack 中主要的接口方法是对于添加删除窗口的处理以及各种窗口操作引起的焦点切换和窗口刷新操作。

3.3 窗口管理的实现

窗口对事件的管理是根据不同的窗口类型进行不

同的操作的。窗体有两种类型,即 WindowType-Common 和 WindowType- Main, 对于 WindowType- Main 型的窗体,会在窗体初始化的时候,为该系列 window 创建一个 eventbuffer。而对于 WindowType-Common 型的窗体,它与同 environment 的主窗体共用一个 eventbuffer。一个应用可能有多个窗体,其中只有一个主窗体,只要这些窗体具有相同的 enviroment,所有相关的事件统一进入主窗口创建的 eventbuffer。

3.4 窗口绘制的实现

Window 对应的内存可绘制区域是由 surface 管理的,通过类 buffer 主要用来记录一些层的信息,用于 surface 的绘制。在 layer 初始化的时候,会创建 windowstack,并且在 windowstack 中记录了 primary-surface,primarysurface 是记录显存地址的一块特殊的 surface。当窗口内容改变时,引擎修改窗口的 surface 相应内存地址的内容,经过窗口的一系列处理,最终通过 Blit 操作使更新后的内容显示在屏幕上。

上层的一些绘图操作实际上是将颜色信息写入 window 对应的 surface buffer,这只是在内存中进行操作,没有反映在屏幕上。经过一系列判断和处理之后,此时内存中已经是应该显示的结果,只需要将内存中记录的颜色信息拷贝到显存中即可。primarysurface 的 buffer 中记录的就是当前 layer 在显存中的首址,Enable Overlay1 时需要修改 primarysurface 的 buffer 中的值,将 buffer 中记录的首址改为驱动为 Overlay1 分

配的显存首址。确定了显存的地址后,开始拷贝操作,这里涉及到格式转换的问题。不同格式之间的拷贝要注意格式转换。

4 结束语

基于此图形引擎的 Vinci 系统实现了具有高效的事件处理机制和面向构件的体系结构,这样大大提高了图形程序与用户交互效率,同时它为用户灵活地实现多种图形界面提供了保障。

参考文献:

- [1] Koretide Website[EB/OL]. 2008-03. <http://www.koretide.com.cn/>.
- [2] 陈 榕,刘艺平.基于构件、中间件的因特网操作系统及跨操作系统的构件与中间件运行平台[R].上海:上海科泰世纪科技有限公司,2003.
- [3] 上海科泰世纪科技有限公司. CAR 构件大全[EB/OL]. 2008-01-01. <http://www.elastos.com.cn/>.
- [4] Elastos, Inc. Elastos 2.1 Information Repository[EB/OL]. 2008-03. <http://www.koretide.com/download.php>.
- [5] Chen Rong. The Application of Middleware Technology in Embedded OS[C]//Workshop on Embedded System, in Conjunction with the ICYCS(6th). Hangzhou: [s. n.]. 2001: 1-3.
- [6] Yuan Feng. Windows 图形编程[M]. 北京:机械工业出版社,2002.

(上接第 43 页)

表 2 某井天然裂缝识别结果

序号	深度(米)	神经网络输出结果	成像显示	文中方法
1	1507.25	0.9981	1	1
2	1513.00	0.0020	0	0
3	1525.12	0.9976	1	1
4	1551.37	0.3540	1	0
5	1560.5	0.9876	1	1
6	1594.37	0.0040	0	0
7	1818.87	0.0120	0	0
8	1842.75	0.9021	1	1
9	1869.37	0.0325	0	0
10	1876.87	0.9640	1	1
11	1880.25	0.5637	1	0
12	2010.56	0.0011	0	0
13	2204.32	0.0010	0	0

3 结束语

采用粗糙集方法能够挖掘出反映天然裂缝特征的有用的常规测井曲线属性,这样可以得到最简样本集;

再采用具有容错性、抗干扰性和自适应性等优点的神经网络作为后置的信息识别系统能够很好地识别天然裂缝。实例表明,文中提出的基于粗糙集和神经网络的智能识别方法切实可行,在天然裂缝的识别中效果显著。

参考文献:

- [1] 张仕强,焦 棣,罗平亚,等.天然岩石裂缝表面形态描述[J].西南石油学院学报,1998,20(2):45-49.
- [2] 郑雷清,刘东付,左新玉,等.测井资料在砂岩裂缝识别中的应用[J].吐哈油气,2003,8(3):79-82.
- [3] Pawlak Z. Rough sets[J]. International Journal of Information and Computer Science, 1982,11(5):341-356.
- [4] 曾黄麟.粗糙集理论及其应用[M].重庆:重庆大学出版社,1998.
- [5] 韩力群.人工神经网络理论、设计及应用[M].北京:化学工业出版社,2002.
- [6] 刘光中,李晓峰.人工神经网络 BP 算法的改进和结构的自调整[J].运筹学学报,2001,5(1):81-88.