中国大学生计算机设计大赛



交互媒体设计创作文档

作品编号：　　　　　　2024035705

作品名称：　　ICmd-指令行交互应用开发通用框架

作　　者：　王永淳 蒋伟豪 蒋忱迅 任川渝 罗钧元

版本编号：

填写日期：　　　　　2024年4月25日

填写说明：

1. 本文档适用于所有涉及软件应用与开发类的各个小类作品，包括：（1）Web应用与开发（2）管理信息系统（3）移动应用开发（非游戏类）（4）算法设计与应用（5）信创软件应用与开发（6）区块链应用与开发；
2. 本文档为简要文档，不宜长篇大论，需简明扼要，建议设计二级目录，逻辑性强；
3. 一级标题采用二号黑体，居中，二级标题采用三号黑体，靠左，根据需要可以设计三级标题，正文一律用五号宋体；
4. 提交文档时，以PDF格式提交本文档；
5. 本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目 录

[第一章 作品简介 6](#_Toc100040660)

[第二章 概要设计 6](#_Toc100040661)

[第三章 详细设计 6](#_Toc100040662)

[第四章 测试报告 6](#_Toc100040663)

[第五章 安装及使用 6](#_Toc100040664)

[第六章 项目总结 6](#_Toc100040665)

[参考文献 7](#_Toc100040666)

# 行业背景与意义价值

1.1当前各类交互界面背景

物联网已被国务院列为我国重点规划的战略性新兴产业之一，在相关政策带动下，我国物联网产业呈现高速发展的态势。同时2021 年 3 月 23 日，国家发展改革委等 13 个部门发布《关于加快推动制造服务业高质量发展的意见》。《意见》中强调了格蕾信息技术的重要性，强调了智能制造的发展需求和前景。

在工业互联网以及物联网中，大量的设备都并非使用的传统常见的Windows等操作系统一样拥有图形化界面的功能，例如服务端多是使用centos之类的linux操作系统，此类操作系统通常只有命令行界面以及简单的颜色显示能力。

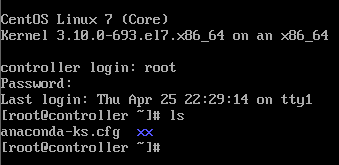


图1.1 centos仅有命令行界面显示

到终端的各类机器，其主要功能并不需要复杂的显示，更有使用单片机之类的终端，仅仅只有字符显示的功能，并不具备搭载各类图形化界面或者交互式界面的功能，只能做到简单的字符显示。

所以，当前工业互联网和物联网市场中存在的的终端与服务端的机器机器普遍存在着缺乏图形化界面的难题。这样的难题要求各行各业的操作工人要能够对现场的终端，服务器显示的参数和信息拥有敏锐的观察能力。这不仅提高了人工培训的成本，也增加了出现误判的概率，降低了对于突发事件的处理效率。

所以当前的各式设备中，存在着如上亟待解决的痛点。

1.2需求分析

随着物联网的发展，无人机，单片机，各类传感器等等各类设备也在不断迅猛发展，在这些硬件系统中，缺乏了可以图形化，或者是可以增强界面显示能力和交互能力的界面系统。

与此同时，此类设备的操作系统通常为只拥有命令行界面的操作系统，同时此类设备通常并不拥有专门的渲染硬件，只有类似于CPU的计算硬件，这些客观条件决定了此类设备并不具备安装并运行诸如QT等图形化界面的能力。

而一个完整的拥有较强显示功能和交互功能的界面系统，将会很好的为设备赋能，并提升使用者的开发和设计效率。同时也能起到提升开发效率，降低维护成本的效果。

因此，针对当前并非所有设备都拥有图形化交互界面的难题和痛点，一个能够在命令行界面实现便于理解的界面和便于使用的交互功能的框架是诸多系统的需求。

我们开发的指令行交互应用开发通用框架（ICmd）便针对了这样的现状，提出了使用在命令行界面中通过字符打印和监控各类输入方式的形式，实现通用开发框架的解决方案，旨在为各类系统提供通用的框架的形式，在较低的系统占用的前提下，实现图形化界面，并从此角度实现图形化交互界面的革新。

1.3竞品分析

在当前的市场上，已经存在了大量类似的图形化界面，这里选用同样适用C/C++的QT和EASYX进行比较。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能\框架名 | EASYX | QT | ICmd |
| 主要应用 | 实现图形库 | 制作GUI | 基于字符画绘制GUI |
| 操作系统 | 基于Windows | 多平台兼容 | 未来将多平台兼容 |
| 生态 | 完善度低 | 允许组件编程 | 允许组件编程且提供了丰富的自定义接口 |
| 对资源占用 | 占用较大 | 占用较大 | 占用低，适配各类机型 |

表1.3 竞品分析

与其他的常用图形化框架相比，ICmd拥有更好的适配性，打印字符的图形渲染方式虽然会在清晰度上略有缺失，但会占用更少的系统资源，可以更好的在一些设备上适配。

# 作品概述

2.1作品简介

指令行交互应用开发通用框架（以下简称ICmd）是一款基于C++语言，为指令行界面所设计的，拥有XML支持的，用于进行交互应用开发的通过打印字符的形式实现图形化的通用框架。当前已可在Windows平台中运行，通过修改基础的原生库的调用，就可以在其他系统中使用。此框架通过在指令行窗口中打印字符的方式进行图形和界面的渲染，避免了常用的图形化界面需要使用各类框架并占用大量内存与计算资源的问题，使得用户可以在指令行窗口中就能够实现可视化的界面，以及完成鼠标点击，键盘操控，等等一系列操作。从而实现在占用较少资源的情况下进行交互式应用的开发，并可以在许多原本缺少图形化界面的操作系统中（如centos）依旧可以通过字符串的形式实现交互式应用的开发。

2.2创作立意

但此类设备不同于我们常用的Windows系统一样的拥有图形化的界面，通常都是使用的类似centos一样只有指令行界面的操作系统，缺少了便利性，对于用户的使用有一定的门槛，且需要一定的学习成本。而比起单纯的指令行界面，一个图型化的界面很明显会让用户更好的理解各种程序的运行状态，并让用户拥有更好的使用体验。

因此我们设计了这样的一个交互应用开发通用框架，基于C++语言和CMAKE技术使得此框架可以在各种操作系统中拥有较好的适配度，从而可以满足在各种不同操作设备中的的图形化交互的需求。系统通过ICMD实现图形化交互将只占用较少的资源，使用可视化的形式进行交互，降低指令行系统的学习成本。从而助力我国物联网产业的加速发展，为无人机等各类机械的发展提供助力。

2.3开发原则与创新点

1.脱离原生系统，降低跨平台成本

本框架仅仅在命令行输出与监听鼠标/键盘的地方使用了原生系统自带的库，其他成分包括图形库，消息循环以及各种UI的底层实现，底层渲染，事件调度，以及更多的对于一个成熟UI框架必不可少的部分，均为使用C++20标准库，由开发人员从0开发，有效降低了对各类函数库的依赖性。本文档也会在以下篇幅中扼要地阐述每个部分的设计思路。

2.减少心智负担，追求每个开发者最舒服的开发姿态

拥有Android或者其他大型项目开发经验的开发者，都会对一个开发友好型框架有着深切的向往。此框架也提供了一整套面向开发者的开发工具包（ICmd Development Kit），不仅仅支持XML进行布局设计，也提供了高度响应式，高度拓展性以及高度简明性的用户接口，力图在保证性能优秀的同时，给予每个应用开发者一个舒服简便的开发环境。

3.优化的编程手段，先进的编程思想

采用了元编程技术，函数式开发思想，节点式布局设计，在优化了整体程序的同时，也增加了框架的容错性，提高了可拓展性，提供了丰富的自定义方式，并减少了开发者的心智负担。

4.指令行中也可以有丰富的交互

在传统指令行界面中只有输入与输出以及黑屏和白字，用户和开发者在使用时常会出现疲惫和倦怠感，此框架在鼠标交互提供了非完备的监听机制，在颜色与样式更是实现了高度的风格化，让用户可以对界面的色彩，样式等进行自定义，让原本平淡无奇的指令行可以为用户带来如应用程序UI般的体验。

# 技术难点与算法实现

3.1难点一：元编程

1.简述

ICmd实现了字符串为索引的元组。

众所周知元组的类型是编译时确定的，也是编程中编译时多态性的体现（Compile-time Polymorphism），而传统元组是以数字为索引的，例如mTuple.get<1>()，本框架将类似于mTuple.of<”hello”>()的代码成为了可能。

2.实现

元编程可以和运行时的代码相互比较学习，例如using A = int与int i = 0这种赋值语句类似， is\_enable<true, A>::type与printf(“xxx”)这种调用语句类似。通过学习元编程中的基本概念，可以在C++中进行类的任意多态处理

定义编译时字符串struct CompTimeStr

将编译时字符串的模板（template <CompTimeStr... Keys>）和类型模板（template <typename... Args>）以嵌套结构体方式放进字符串索引元组的定义

定义中声明元组tuple\_raw作为背后的容器

定义中嵌套tuple\_helper，用模板类多态推断（Type Deduction）的方法，将Keys遍历，成员为Key，设置静态常数index，获取当Key与传来的Key值相等时的迭代器I的值并给予index

获取tuple\_raw中位置为I的元素与类型，返回。

3.元编程在ICmd中的应用

字符串索引元组可非常方便地作为每个节点或节点组件的参数的容器，替代的std::map，允许多类型，且在运行前完成类型推导，既保证了输入值的类型检查，也保证了取出值的类型推断，全程无类的转换与指针参与，是一种非常优雅而实用的参数寄存方法。

3.2事件机制的实现

3.3图形库渲染的实现

3.4算法实现

# 作品功能和实现说明

在对此框架进行设计时，分为内核层与用户层。从而在保障框架的稳定运行的同时暴露足够多的可用接口给用户从而实现更加丰富的自定义组件。

2.1内核层实现说明

内核层主要分为了3个线程，2条队列，分别是内核线程，渲染线程，还有监听器线程。队列则分为了帧循环队列和消息循环队列。帧循环队列用于更新帧和处理各类信息，消息循环用于处理窗体收到的各式消息。



消息循环：由looperThread负责，其中是消息循环，用于处理线程间的通讯，生命周期与资源加载



图形渲染：由renderThread负责，其中是图形库与渲染程序，用于处理各种图元信息，以及输出帧信息，字符打印



监听循环：由hookThread负责，其中是对于键盘与鼠标的循环监听

2.2用户层重要概念讲解

Node（节点）: 应用层面渲染的最小单位，相当于Android中的View，但是有着比View更高的响应性与组件性

NodeComponent（节点组件）：规范每个节点行为的单位。每个节点的渲染，布局，监听都可由节点组件提供

在进行组件的自定义编写时，通常通过继承这两个基类实现。

2.3用户创作的应用程序的生命周期

1. 资源加载 – 内存地址的里应外合：
   1. 程序开始时，首先加载的资源是用户定义的onCreate方法，窗口/应用参数，以及XML文件，资源性文件（图片，音频等等），和监听器回调统一进入**上下文管理器**（Context）中进行热加载。
   2. 上下文管理器将加载结果（内存地址）封装成回调，通过调用**内核主函数**的方法给予内核，封装进**内核资源上下文**（CoreContext）中
2. 内核的初始化 – 消息握手的左拥右抱：
   1. 创建**帧队列**（FrameQueue）与**消息队列**（MessageQueue），将三大线程，互斥锁等资源进行初始化，并将**资源上下文信息**（CoreContext）与**应用信息**（CoreInfomation）给三个线程
   2. looperThread（消息循环线程）中资源加载完毕后，发送该线程初始化完毕（LOAD\_OVER）的信息给renderThread（渲染线程）
   3. renderThread收到消息后，初始化**命令行窗口**（Console），发送初始化完毕（LOAD\_OVER）的信息给looperThread；发送安装Hook函数（INSTALL\_HOOKS）的信息给hookThread（监听线程）
   4. looperThread收到消息后，从内核资源上下文提取用户的onCreate方法并执行，并将消息循环句柄以参数形式反射到应用层面
   5. hookThread收到消息后，开启监听循环，侦测鼠标和键盘的状态
   6. 至此内核启动并持续运行
3. 节点的加载 – 应用层面的翩翩起舞
   1. 节点在构造中捆绑**上下文**（Context）
   2. 将必须组件加载到节点上（例如Background组件使得节点有纯色背景）
   3. 将节点附属到上下文中的**顶层节点**（parent）中
   4. 顶层节点进行更新
      1. 调用layout函数，自父到子调用每个子节点的layout函数，确定每个节点的绝对坐标与长宽
      2. 调用draw函数，自父到子调用每个子节点的draw函数，实现提取节点的绘制结果，渲染为帧信息后，封装为一个更新（UPDATE）消息，发送给renderThread
   5. renderThread将帧信息输出到命令台，结束节点到命令台的渲染全流程
4. 节点的监听 – 交互响应的心有灵犀
   1. hookThread中的循环侦测到事件后，将事件各种参数进行包装
   2. hookThread调用内核资源上下文中传来的全局回调
   3. 全局回调中将事件进行二次包装，包装为提供给用户事件回调的**鼠标事件**（MouseEvent）与**键盘事件**（KeyEvent）
   4. 全局回调中遍历所有顶层节点的子节点，并查找他们的**监听器**(Listener)，将符合事件条件的监听器回调触发
   5. 至此从底层到应用层面的事件处理全部完成。
5. 应用的结束 – 华丽呈现的爽快收尾

各个Node析构，将上下文中各种数据类型进行释放，发送SHUTDOWN消息至looperThread，内核开始释放各种资源，退出各种线程，最后关闭窗口

# 运行结果

# 应用场景与未来展望

5.1应用场景

1.初学者入门C/C++

当前，为面向科学技术的现代化，C/C++语言课程已经成为了大部分工科学子必定会学习的课程，而在大部分工科学子学习相关内容时，也将会面临需要使用图形化交互界面的程序设计要求。但对于初学者而言，诸如QT等复杂的图形化界面框架有着过高的学习成本，难以快速掌握。在这种时候，像此框架一样可以快速学习的图形化界面将会更好的帮助他们完成相关程序设计。

同时，此框架的丰富的组件基类提供了大量的可使用接口，拥有强大的自定义能力，可以为初学者提供很好的面向对象编程范例，在用户使用的同时，也能从中学习和实践到C/C++语言的语言特性，从而更好的掌握相关的语言。

2.虚拟机以及服务器应用

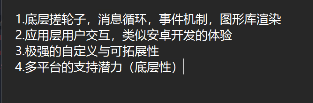
随着云计算相关科技的不断深入和发展，虚拟机在各方面都变得更加常用，而其中大多数的虚拟机或是服务器都采用的是centos一类的linux操作系统，通常都没有甚至缺乏图形化的界面，只能通过传统的命令行界面进行交互。在这种情况下，本框架将可以提供更加直观的交互界面。

5.2社会效益

5.3未来展望

# 参考文献

【请按照标准参考文件格式填写】

1. 当前-需求-痛点-我们的产品的优势
2. 简介-创新点：支持xml，拓展性，创作自己的窗体
3. 创作难点：底层的各个逻辑，支持xml，
4. 作品功能和各个实现概括
5. 应用场景：画饼，图片
6. 安装和使用