中国大学生计算机设计大赛



软件开发类作品文档简要要求

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)

# 作品简介

1.1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。当前已在Windows平台上可以正常运行。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。

1.2创作立意

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。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，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。

1.3作品特色

基于指令行界面使用字符渲染：使用打印字符的形式进行渲染，对内存空间占用小，同时也节省了计算资源。

使用元编程的形式定义元组：可以直接使用字符串与需要的数据进行静态对应，解决了C++中对各种数据类型需要通过繁复的类型转换的问题。

丰富的自定义组件功能：为用户提供了开放的接口和基类用于自定义组件，可以让用户实现自定义组件从而更好的满足用户自己的设计需求。

# 使用技术

本作品主要使用了C++和CMAKE进行开发。

# 概要说明

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

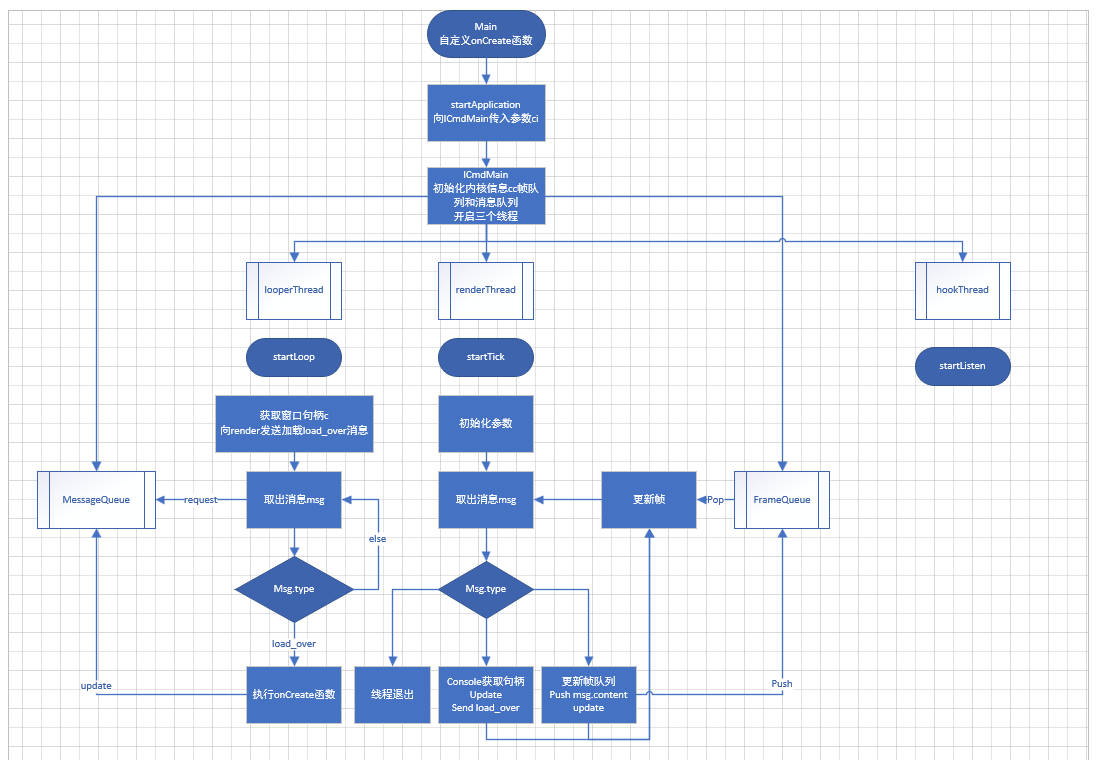


图2.1 指令行交互应用开发通用框架内核层框体图

由图可知内核层主要分为了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。

2.2用户层

用户层主要是layout, listener, os, utils, view, widget六部分

layout 主要设计了绝对布局，每个子layouts的位置都基于其左上角的位置。

listener主要包括了对键盘，鼠标，计时器等等各项内容的监听和处理。

os 用于定义context以及基于xml文件创建node节点的函数

utils 包括了canvas和util，canvas主要用于定义画布并绘制矩形线段字符等。util主要用于给绘制提供接口。

view中包括了node，node可以与监听器，布局等各类进行绑定，同时也提供了每个节点的更新，绘制等接口。

widget中包括了对背景，图案，字符的绘制。

# 设计难点

【填写说明：包括但不限于：界面设计、数据库设计(如果有)、关键算法。界面设计建议用作品实际界面，建议包括典型使用流程；数据库设计建议用表格、ER图或UML方式，说明文字简明扼要，违背范式的设计建议请说明理由；关键算法也可以替换为关键技术、技术创新等。本部分不宜大篇幅铺陈，建议突出重点痛点难点特点。】

# 安装及使用

【填写说明：简要说明安装环境要求、安装过程、主要流程等。建议包含默认安装和典型使用流程。】

# 项目总结

在项目制作的过程中，在项目协调方面，通过使用github来进行进度管理，并使用了todo list的形式来管理每个人正在制作的部分，并保证可以互相正在完成的东西是否有冲突，并互相帮助完成任务。在任务分解方面，我们将任务总体分为了前端和后端两部分进行协调和任务分解，互相提供需要的接口和提出需要实现的功能，从而实现框架的基本构建。

在项目的后续升级演进方面，我们准备向多平台适配上更加努力，尽可能实现更多平台的可用性，同时也不断拓展新接口，为用户提供尽可能多的使用方式。

# 参考文献

【请按照标准参考文件格式填写】