

【麦当劳点餐系统-图形化 OJ 版概要设计书】

版本号：v2.1
初稿编制时间：2024 年 5 月 31 日
最后更新时间：2024 年 6 月 10 日
编制人员：刘南府
图形化技术：Qt6

1. 图形化界面设计

图形化程序的界面主要分为两个界面：开始界面与点单界面

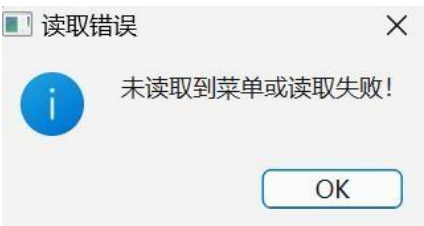
1.1 开始界面



开始界面的主要功能是载入菜单，点击“读取菜单”后，若正确读取菜单，则会提示



若读取菜单失败，则会提示



当用户未读取菜单或读取菜单失败，但试图进入点餐系统时，程序将提醒



当用户成功读取菜单后，点击“开始点餐”，则进入点餐界面

1.2 点单界面



点单界面有返回上一级菜单、选择食物/套餐并下单两个功能，同时，点餐系统内置程序时间，程序 时间从 07：00：00 开始，程序内时间的运行速度可自由变换，其上限为正常时间流速的 100 倍，下限为 正常时间流速
若时间流速超出了限定范围，程序将会提示



界面中有三个核心显示窗口：单品窗口、套餐窗口、订单状态窗口

其中，套餐窗口会在套餐名称后显示套餐内所包含的单品

单品窗口与套餐窗口会显示从菜单中读取到的食物和套餐，菜单更换后，两个窗口所显示的内容也将改变

单品

	单品	单品名称	当前备餐数量
1	<input type="checkbox"/>	Coke	0
2	<input type="checkbox"/>	BigMac	0
3	<input type="checkbox"/>	CaesarSaladC...	0
4	<input type="checkbox"/>	ChocolateSauce	0
5	<input type="checkbox"/>	Croutons	0

套餐

	套餐	套餐名称	所包含食物
1	<input type="checkbox"/>	BreakfastCom...	BigMacCoke
2	<input type="checkbox"/>	DessertCombo	CroutonsChocol...

下单成功后，程序将给出提示
且这一订单将显示在订单状态窗口当中

订单状态

	订单ID	下单时间	状态
1	1	07:00:03	07:00:20

若用户在点餐时没有勾选任何单品或套餐，程序将拒绝下单，并做出提示

来自店员的疑惑

 您还什么都没有点呢

OK

对订单的显示关于两方面：下单时间及订单状态 下单时间是用户点击“下单!”时的程序时间


而状态有两种可能：若订单可以完成，则状态显示订单预计完成时间；若订单下入时队列已关闭，则会 输出“Fail”

订单状态

	订单ID	下单时间	状态
1	1	07:00:07	07:04:50
2	2	07:12:12	07:16:04
3	3	07:14:38	07:20:54
4	4	07:17:00	07:25:44
5	5	07:19:44	07:30:34
6	6	07:21:52	Fail
7	7	07:38:53	07:42:45
8	8	07:42:52	07:47:35
9	9	07:46:12	07:52:25
10	10	07:54:12	07:58:04
11	11	07:54:14	08:02:54
12	12	07:54:17	08:07:44
13	13	07:54:20	Fail

若程序运行到内置计时器时间的 22: 00: 00，计时器将停止，且程序将会做出提示

提示

 今天已经下班了，明天再来吧！

OK

若点击返回上一级菜单，则程序将退回原界面，且需要重新读取菜单

2. 项目结构设计

项目核心分为五部分，分别是数据结构模块(Data_Structure)，输入模块(input)，工作模块(work)，开始界面(mainwindow)，点餐界面(order_window)

2.1 数据结构模块(Data_Structure.h)

详情参见《OJ版概要设计书-刘南府》-【2. 高层数据结构设计】

2.2 输入模块(input.h)

在单文件 OJ 版本的基础上，对一些函数进行了对 QT 框架适配的重新处理，如

```
int Qfood_id_find(struct Node1 *food_array,const char *food_name,int foodnum);
int Qcombo_id_find(struct Node2 *combo_array,const char *combo_name,int combonum);
void Qorder_list_insert(List_order L,order_Position aim_pos);
```

其余参见《OJ版概要设计书-刘南府》-【3.2.1 输入模块(Input.h)】

2.3 工作模块(work.h)

在单文件 OJ 版本的基础上，对时间转换逻辑与工作逻辑进行了重新处理,使得工作模块的时间逻辑将以程序内计时器时间为参照标准

```
int QStringToSeconds(const QString& timeStr);
void Qworking_simulation(struct Node1 *food_information,List_order L_order,int foodnum,int time,int foodStore[]);
```

其余参见《OJ版概要设计书-刘南府》-【3.2.3 模拟工作模块(Working.h)】

2.4 开始页面(mainwindow.h)

开始界面总体基于 QT 的 Widget 类进行开发

```
void on_read_data_clicked();
void on_begin_order_clicked();
```

上述两个槽函数完成开始界面的核心功能——读取菜单与跳转点餐页面

```
Ui::BUPT_McDonald *ui;
order_window *orderpage;
int food_kinds_num;
int combo_kinds_num;
Node1* food_information;
Node2* combo_information;
List_order L_order;
List_combo_food L;
```

上述代码分别用于导入 UI 设计，创建点餐界面对应的类，存储菜单信息

2.5 点餐界面(order_window.h)

点餐界面总体基于 QT 的 Widget 类进行开发

```
void displayFoodInformation(Node1 foodInformation[], int foodCount);
void displayComboInformation(Node1 foodInformation[],Node2 comboInformation[], int comboCount);
void setFoodInformation(struct Node1 *foodInfo);
void setComboInformation(struct Node2 *comboInfo);
void setOrderListHead(order_Position orderListHead);
void on_pushButton_clicked();
void updateClock();
void on_order_button_clicked();
void on_speedUpButton_clicked();
void on_slowDownButton_clicked();
void updateFoodStorage();
```

前两个函数分别用于在单品窗口 (foodtableWidget) 与套餐窗口 (combotableWidget) 中展示菜单内容，并创建商品前的单选框；中间三个函数将把数据结构传入 order_window 类中以供读取和存储；后六个函数构成页面的核心交互功能，他们分别实现返回上一级菜单、实现并更新程序内计时器、控制计时器速度、下单、实时更新食物储量这三大功能

```
Ui::order_window *ui;  
QTimer *timer;  
QTime currentTime;  
int speedFactor;  
int timerInterval;  
int foodnum;  
int combonum;  
struct Node1 *food_information;  
struct Node2 *combo_information;  
order Position order list head;
```

上述代码分别用于导入 UI 设计，设定计时器变量，引入数据结构以供操作

3. 核心算法设计

在单文件 0J 版本的基础上，我们在原有的算法逻辑上对订单的接收逻辑进行了改写，我们将 0J 版原有的读取所有预下单订单的逻辑改写为以订单按钮的槽函数激活订单接收响应，使其处理后的输出结果能够符合实时下单的标准，同时我们对程序对订单队列的处理过程进行动态追踪，以此实时输出备餐数量，并且图形化程序支持单品/套餐混合多样选择，不必像单文件 0J 版的程序一样只能选择一种食物或套餐。

此外，图形化程序保留了文件输出方式，你可以在程序运行结束后于可执行程序同级目录下得到 output.txt，其中记录了订单处理结果

其余详情参见《0J 版概要设计书-刘南府》-【3.3 核心算法设计】