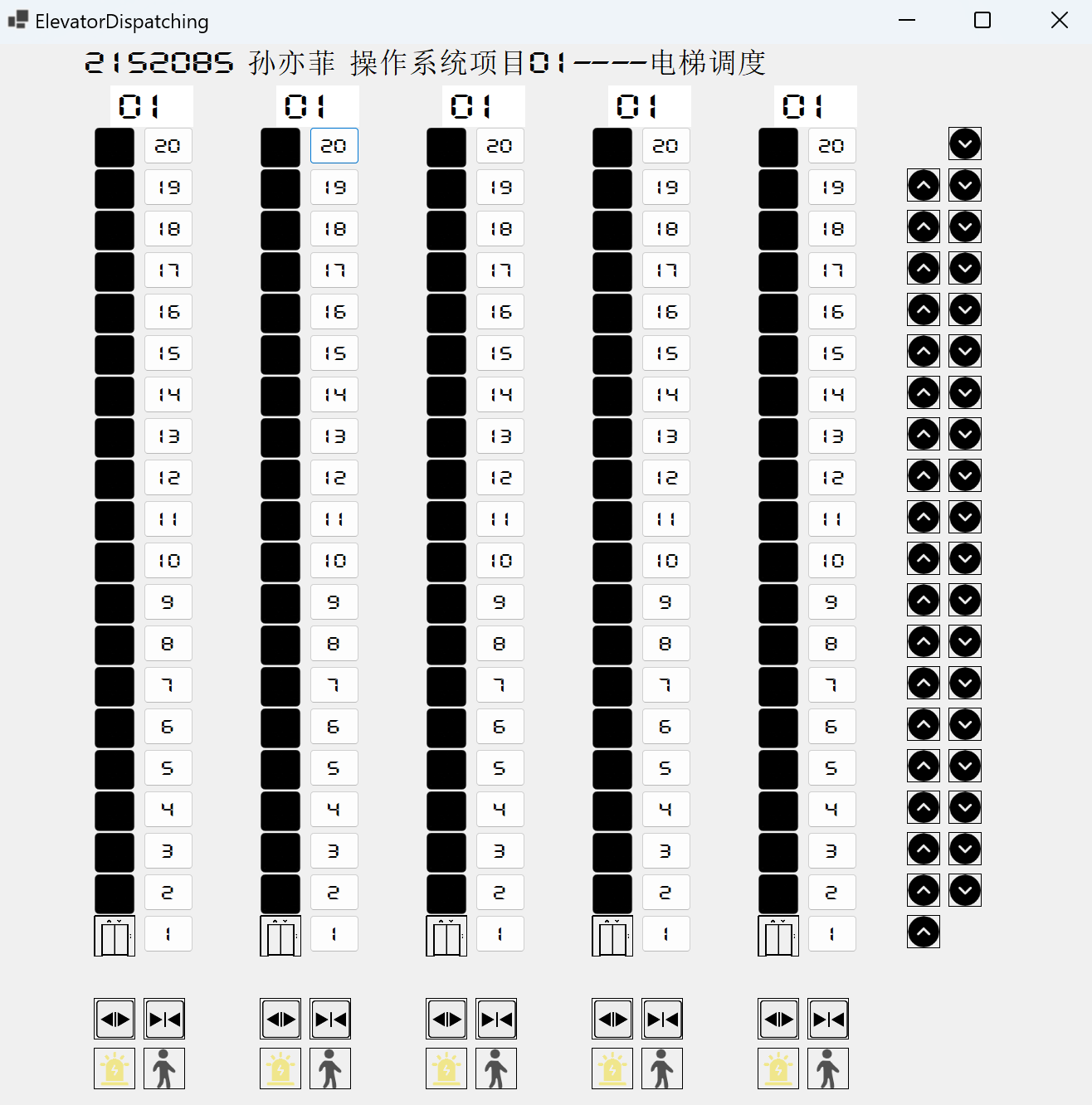
ElevatorDispatching设计方案

2152085孙亦菲



# 一、项目简介

**本项目实现了以下功能**：

* **多楼层多电梯的调度**

项目实现了五部电梯在二十个楼层之间的调度，并且可以通过电梯内部按钮（每部电梯内部都有1-20个按钮）和外部按钮（每一层都有上行↑、下行↓）进行楼层之间的调度来控制电梯的状态。

* **警报（停止运行）及恢复**

同时每部电梯还有警报和通行按钮（画面底部），来控制电梯是否可以运行。

* **电梯信息的动画显示和显示器显示**

每部电梯都会配备一个显示器来显示电梯当前的楼层，底部的警报器和通行图都会随着电梯状态而改变，同时每一步电梯的运行都进行了动画可视化，能够更为直观的观测到电梯当前的运行位置，运行方向等。

* **电梯的开关门**

每部电梯都配备了开门关门按钮（画面底部），来控制电梯开关门动作。

# 二、项目环境

* **开发环境**

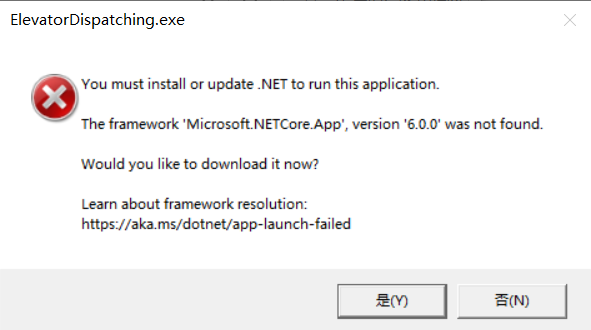
C#.NET Windows窗体

Win11

Visual Studio 2022

目标框架：.NET 6.0

* **运行环境**
* 需要安装.NET的6.0版本，否则无法运行

****

* 显示屏分辨率：2880x1800，该分辨率下运行较佳，其他分辨率可能会出现窗口超界。
* **运行方法**
* 直接运行

打开ElevatorDispatching.exe即可运行本项目。

1.点击对应电梯的楼层按钮即可使对应电梯开始移动，同时点击电梯外部对应的的↑或↓按钮即可让电梯移动到该楼层并开门**。**

2.点击对应电梯的警报按钮会让电梯立刻停止在当前楼层，点击通行按钮恢复最后的运行状态。

3.点击开门可实现开门后自动关闭效果（若保持开门需一直点击），点击关门可实现开门时（任何非完全关闭状态）关门。

* 源码运行

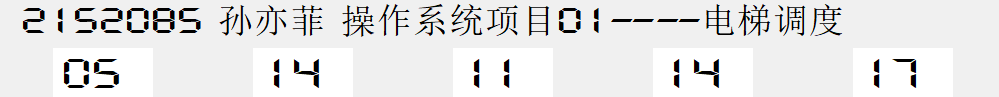
若要运行源代码，需要将img文件夹放入对应项目的debug/net6.0/或release/net6.0/文件夹内，否者无法加载图片文件。

* 注意：

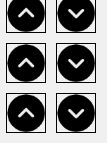
若点击过快，可能会出现可视化电梯错位现象。

# 三、界面及功能介绍

* 此界面显示了电梯当前的楼层。

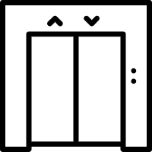
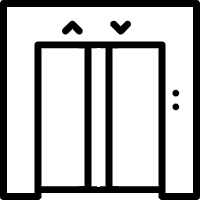
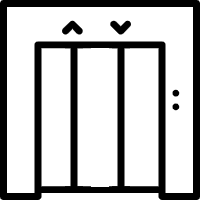
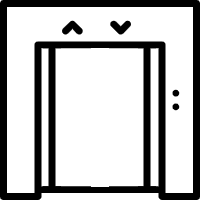
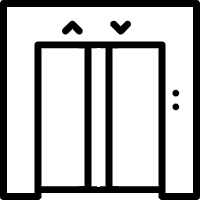
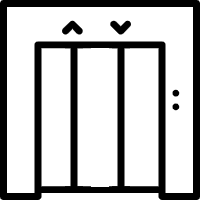
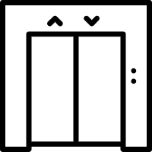


* 此界面是每一层楼表达上行下行需求的外部按钮，且一次点击后，在电梯没到之前，外部按钮会被红色标记且不可用。电梯到后会恢复。

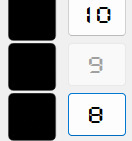
 

* 此界面是开关门界面，当按下开关门按钮或者电梯到达指定楼层后会播放以下开关门动画。运行时为关门状态。



* 此界面是控制每一部电梯内的楼层按钮，以及可视化电梯，展示电梯的运行过程。且一次点击后，在电梯没到之前，内部按钮会被标记不可用。电梯到后会恢复。

* 此界面是报警和通行界面。

警报器为黄色及人物走动，代表此时电梯正常运行，乘客通行，此时警报器按钮可用而通行按钮不可用。



按下黄色警报器，警报器变为红色及人物坐下等待，代表此时电梯停止运行，乘客不可通行，此时警报器不可用而通行按钮可用。



# 四、调度算法

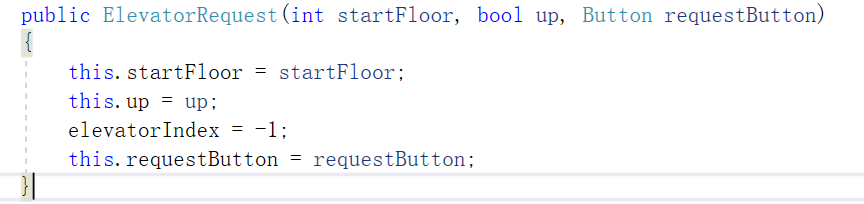
1. 本项目中的核心调度算法采用了LOOK算法的思想：

LOOK算法是扫描算法（SCAN）的一种改进，扫描算法（SCAN） 是一种按照楼层顺序依次服务请求，它让电梯在最底层和最顶层之间连续往返运行，在运行过程中响应处在于电梯运行方向相同的各楼层上的请求。但当 LOOK 算法发现电梯所移动的方向上不再有请求时立即改变运行方向或暂停，而扫描算法则需要移动到最底层或者最顶层时才改变运行方向。

1. 本项目在每部电梯内部，实现以下关键变量

|  |  |  |
| --- | --- | --- |
| **变量名** | **变量类型** | **功能** |
| elevatorMovingState | enum ELEVATORMOVINGSTATE  { UP, DOWN, WAIT} | 对应该电梯当前状态，  上行/下行/等待 |
| bool[] stopFloor | bool[floorMax+1] | 该电梯的某楼层是否需要停靠 |
| outsideButton | List<Button> | 绑定该电梯目前响应的外部按钮 |
| insideButton | List<Button> | 绑定该电梯目前响应的内部按钮 |

本项目在具体逻辑中，实现关键ElevatorRequest结构体，代表外部按钮的需求，记录发起楼层，是否上行，分配的电梯编号及对应外部按钮。

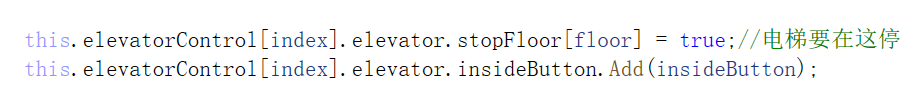


其余变量如下

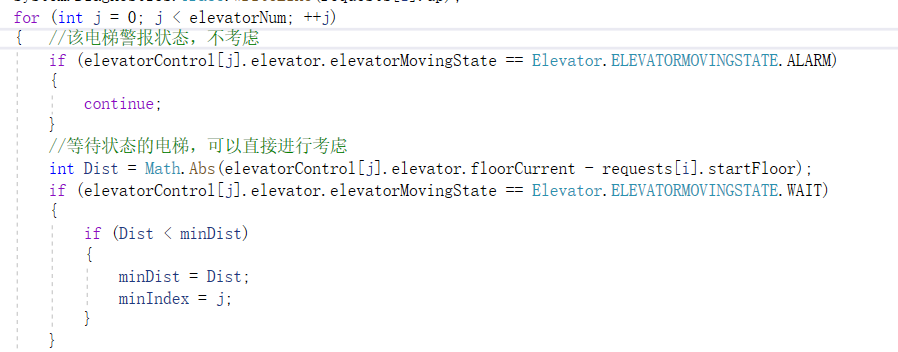
|  |  |  |
| --- | --- | --- |
| **变量名** | **变量类型** | **功能** |
| requests | List<ElevatorRequest> | 记录电梯请求列表 |

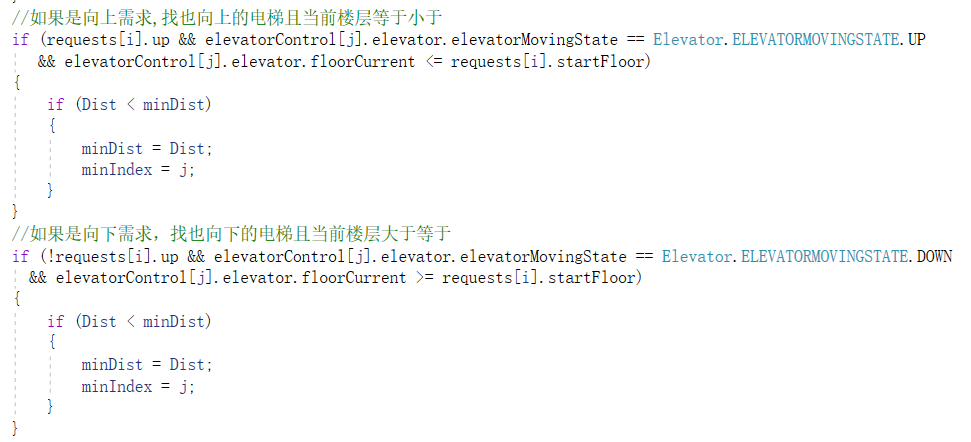
1. 实现逻辑如下：

* 当某部电梯内部按钮按下，则将该电梯在该楼层需要停靠改为真，且将内部按钮加入表单；

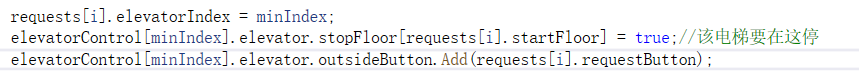


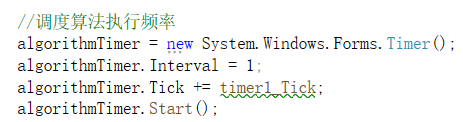
* 当某楼层外部按钮按下，遍历五部电梯找最合适调度的电梯，在符合条件的情况下选取相距楼层最少的电梯。





确定调度电梯后，则将该电梯在该楼层需要停靠改为真，且将外部按钮加入表单；



该方法即为调度算法中的扫描算法，执行频率1ms 1次。

总结一下，外部调度算法包含以下几点：

1. 就近原则：优选选择距离最近的的电梯。
2. 上升请求可以挑选空闲电梯（WAIT）和上升电梯（UP），下降请求可以挑选空闲电梯（WAIT）和下降电梯（DOWN）。
3. 上升电梯不能响应比电梯当前楼层低的上升请求，下降电梯不能响应比电梯当前楼层高的下降请求。
4. 正在响应下降请求的上升电梯不能响应任何上升请求，正在响应上升请求的下降电梯不能响应任何下降请求。
5. 定时检查外部请求按钮已经被恢复的请求，移出请求队列（requests）。

# 五、其他功能

1.开关门功能

电梯在运行时不支持开关门，在电梯到达某楼层自动开关门，或等待状态时，点击当前楼层的开门按钮，或者在电梯外点↑或↓时可打开电梯门。

2.警报功能

无论电梯处于什么状态，当电梯内部按下警报按钮后，电梯立刻停在当前当前位置，当前电梯内的所有按钮均为不可用状态。但之前被分配的任务仍然存在。

按下通行按钮，则电梯运行恢复，继续执行停止之前的任务。

# 六、项目亮点

* 实现了多楼层电梯之间的调度。
* 显示完整，有展示当前电梯楼层、运行状态的动画、及警报和通行的图标。
* 采用C# 的System.Windows.Forms.Timer控件控制进程，节省系统资源。

# 七、项目改进方向

* 由于框架的使用，仅限于在Windows平台运行。在未来项目开发过程中，可以考虑采取其他开发工具，使得项目可移植性更好。
* 当电梯按下警报按钮，之前被分配的任务仍然存在，合理情况应为分配给其他电梯，可以进一步考虑改进调度算法。