

《系统仿真与 matlab》综合试题

题 目： 电影院售票与咨询系统仿真

编 号： 21

姓 名 曾恒宇

班 级 人工智能 2003 班

学 号 U202015108

联系方式 15897758209

得分项	创新性	工作量	代码可读性	报告写作	总分
分数					

目录

《系统仿真与 MATLAB》试题报告	3
一、试题具体内容.....	3
电影院售票与咨询系统仿真.....	3
二、试题建模过程.....	3
1. 关于题目的个人理解.....	3
2. 模型参数假设.....	4
3. 系统自动产生服务.....	4
只考虑现场售票的人与服务员交互过程.....	4
考虑现场售票和现场咨询的人与服务员交互过程	4
考虑现场售票、现场咨询与电话咨询的人与服务员交互过程	5
4. 非系统自动产生服务.....	5
5. 综合上述考虑在不同服务选择方式下电影院售票与咨询系统模型的输出情况	6
三、系统仿真流程图.....	7
四、试题实现中的关键难点.....	8
1. 电影院售票与咨询系统仿真页面设计	8
2. 人员的动态显示过程.....	8
3. 泊松分布与负指数分布的实现.....	9
4. 系统自动产生服务的实现.....	10
5. 键盘产生服务的实现.....	10
6. 鼠标产生服务的实现.....	11
7. 按钮产生服务的实现.....	12
8. 模型输出的计算.....	12
9. 模型输入的约束.....	12
10. 停止仿真的处理.....	13
11. 模型输入输出的总表格记录.....	13
五、程序运行指南.....	14
1. 文件介绍.....	14
2. 电影院售票与咨询仿真系统介绍	14
六、程序运行实例分析.....	15
七、新增功能及亮点.....	24
八、系统仿真心得体会.....	24

《系统仿真与 MATLAB》试题报告

一、 试题具体内容

电影院售票与咨询系统仿真

模型假设

电影院售票窗口假设只有一名服务员，负责现场售票与咨询、以及电话咨询，其中现场售票与咨询的优先级高于电话咨询的优先级。假设现场售票人员的到达、现场咨询人员的到达、电话咨询的到达都服从泊松分布，每种服务时间的长短服从负指数分布。

模型输入

现场售票人员的到达率、现场咨询人员的到达率、电话咨询的到达率，每种服务时间的长短的参数。

模型输出

每类服务的平均等待时间、服务员在工作时间内的占用率。要求有输入输出界面、人员的动态显示过程。每种服务的产生可以由系统自动产生，也可以定义键盘上的三个按键来分别表示三种服务的产生，也可以定义鼠标的三个按键来分别表示三种服务的产生。

二、 试题建模过程

1. 关于题目的个人理解

该选题的主要工作应该在于可视化图形界面，根据题目相关说明以及日常生活常识很容易能抓住题目的主要要求，以及要实现的相关功能情况。我们需要建立一个电影院售票与咨询系统进行仿真，其关键点在于人与服务员之间的交互性，根据模型输入在二者的不断交互过程中获取题目要求的模型输出。初步思路步骤可定为如下：确定问题为一个服务员与多个人的交互问题，其中人可以视为一种资源，每单位时间内服从泊松分布产生一定数量的资源（人），服务员则视为消耗资源者，而现场售票、现场咨询、电话咨询分别代表着三种消耗资源的方式，且这三种方式服从负指数分布。服务的产生即代表着消耗资源者（服务员）选择哪种消耗资源的方式，即题目要求的系统自动产生、定义键盘按键产生、定义鼠标按键产生。

综合实际情况，本人认为原应该考虑当人处理完相关事务后的对应去向，去观影或去出口离开，但结合题目要求，发现此方面对于仿真结果及其要求没有意义，故不予考虑。

2. 模型参数假设

设 T_{sale} 、 T_{ask} 、 T_{call} 分别为售票处、现场咨询、电话咨询的平均等待时间， O 为服务员的工作时间占用率， λ_{sale} 、 λ_{ask} 、 λ_{call} 分别为产生人数泊松分布的参数， θ_{sale} 、 θ_{ask} 、 θ_{call} 分别为服务员服务时间负指数分布的参数。

3. 系统自动产生服务

只考虑现场售票的人与服务员交互过程

根据界面初始条件输入的参数确定泊松分布，在入口处产生对应人数，运动到售票处后服务员得到响应，根据负指数分布产生处理时间进行处理，处理完毕后若售票处仍存在等待人数，则服务员重复上述步骤，直至最后无人，服务员回到初始位置。易知第一个单位时间内，第 n 个人的等待时间为前 $n-1$ 个人的处理时间之和(忽略人和服务员的运动更新时间)即有：

$$T_{sale}^n = t_{sale}^{n-1} + t_{sale}^{n-2} + \dots + t_{sale}^1, \text{ 且 } t_{sale}^n \sim \text{Exp}(\theta_{sale}), n \sim P(\lambda_{sale})$$

$$\text{故一个单位时间内 } T_{sale} = \frac{\sum_{i=1}^n T_{sale}^i}{n}$$

考虑 N 个单位时间，前 N 个单位时间结束后仍然在等待的人数为 m_N ，则有：

$$T_{sale(N)}^i = t_{sale(N)}^{i-1} + t_{sale(N)}^{i-2} + \dots + t_{sale(N)}^1 + t_{sale(N-1)}^{n_{N-1}-1} + t_{sale(N-1)}^{n_{N-1}-2} + \dots + t_{sale(N-1)}^{n_{N-1}-m_{N-1}} (i = 1, 2, \dots, n_N)$$

$$\text{故前 } N \text{ 个单位时间内 } T_{sale} = \frac{\sum_{j=1}^N \sum_{i=1}^{n_j-m_j} T_{sale(j)}^i}{\sum_{j=1}^N n_j - m_N}$$

总人数 $A = \sum_j n_j$ ，工作时间 $time$ ，服务员的工作时间占用率有：

$$O = \frac{t_{sale}^1 + t_{sale}^2 + \dots + t_{sale}^{A-m_N}}{time}$$

考虑现场售票和现场咨询的人与服务员交互过程

再将现场咨询处的人考虑进去，由于题目没有说明售票处与现场咨询的优先级，本系统设置为在二者均有人等待时服务员进行等概率随机选择。故 $T_{sale(j)}^i$ 和

$T_{ask(j)}^i$ 应进行修正为: $T_{sale(j)}^i = T_{sale(j)}^i + bias_{sale}$, $T_{ask(j)}^i = T_{ask(j)}^i + bias_{ask}$,

其中 $bias_{sale}$ 是由于当售票处和现场咨询处均有人存在时而选择现场咨询处, 从而让售票处的人们多等待时间而产生的偏差, $bias_{ask}$ 同理。服务员的工作时间占用率:

$$O = \frac{t_{sale}^1 + t_{sale}^2 + \dots + t_{sale}^{A_{sale}-m_{sale}(N)} + t_{ask}^1 + t_{ask}^2 + \dots + t_{ask}^{A_{ask}-m_{ask}(N)}}{time}$$

其中 $t_{sale}^n \sim Exp(\theta_{sale})$, $t_{ask}^n \sim Exp(\theta_{ask})$, $n_{sale} \sim P(\lambda_{sale})$, $n_{ask} \sim P(\lambda_{ask})$

$$A_{sale} = \sum_j^N n_{sale(j)}, A_{ask} = \sum_j^N n_{ask(j)}$$

考虑现场售票、现场咨询与电话咨询的人与服务员交互过程

将电话咨询的人考虑进去, 由于售票和现场咨询的优先级大于电话咨询, 故将电话咨询考虑进来对 $bias_{sale}$ 和 $bias_{ask}$ 几乎没有影响, 而因此导致的 $bias_{call}$ 将会大概率远远大于 $bias_{sale}$ 和 $bias_{ask}$ 的数值。 $T_{call(j)}^i$ 进行修正: $T_{call(j)}^i = T_{call(j)}^i + bias_{call}(bias_{call} > bias_{sale}, bias_{ask})$, 服务员的工作时间占用率为:

$$O = \frac{\sum_{i=1}^{A_{sale}-m_{sale}(N)} t_{sale}^i + \sum_{i=1}^{A_{ask}-m_{ask}(N)} t_{ask}^i + \sum_{i=1}^{A_{call}-m_{call}(N)} t_{call}^i}{time}$$

其中有:

$t_{sale}^n \sim Exp(\theta_{sale})$, $t_{ask}^n \sim Exp(\theta_{ask})$, $t_{call}^n \sim Exp(\theta_{call})$, $n_{sale} \sim P(\lambda_{sale})$, $n_{ask} \sim P(\lambda_{ask})$, $n_{call} \sim P(\lambda_{call})$

$$A_{sale} = \sum_j^N n_{sale(j)}, A_{ask} = \sum_j^N n_{ask(j)}, A_{call} = \sum_j^N n_{call(j)}$$

4. 非系统自动产生服务

非系统自动产生服务中, 不管是定义键盘按键产生服务、鼠标按键产生服务, 还是通过点击对应按钮产生服务, 都是人为选择的, 相较于系统自动产生服务, 考虑到人的选择需要一定的反应时间, 以及选择的随机性, 在最后的模型输出中需要新添由于人的主观因素而产生的多余时间偏量 $bias_{sale}^{people}$ 、 $bias_{ask}^{people}$ 、

$bias_{call}^{people}$, 同时由于在人的主观选择下, 会使得在同一人流量情况下, 相同时间内处理的效率更低, 因为需要考虑到总工作时间内需要腾出一部分时间给予人思考选择服务方式, 故服务员的工作时间占用率在相同情况下相较于系统自动产生服务会偏低, 或者可以理解为占用率的期望偏低。

5. 综合上述考虑在不同服务选择方式下电影院售票与咨询系统模型的输出情况

初始条件定为默认值，此时系统模型运行不会因为人数过多或者服务时间过长产生拥堵，如下：

初始条件设置	
售票人员到达率	0.5
咨询人员到达率	0.8
电话人员到达率	1
现场售票服务时长参数	1
现场咨询服务时长参数	1
电话咨询服务时长参数	1

图 1. 默认初始条件值

设定仿真时间为 1min，仿真结束后，系统自动产生服务、键盘产生服务、按钮产生服务、鼠标产生服务的模型输出分别如下：

现场售票服务平均等待时间
0.6795 s
现场咨询服务平均等待时间
1.753 s
电话咨询服务平均等待时间
2.5144 s
服务员在工作时间的占用率
53.6485 %

图 2. 系统自动服务的模型输出

现场售票服务平均等待时间
3.3301 s
现场咨询服务平均等待时间
0.63999 s
电话咨询服务平均等待时间
7.5481 s
服务员在工作时间的占用率
34.6479 %

图 3. 键盘产生服务的模型输出



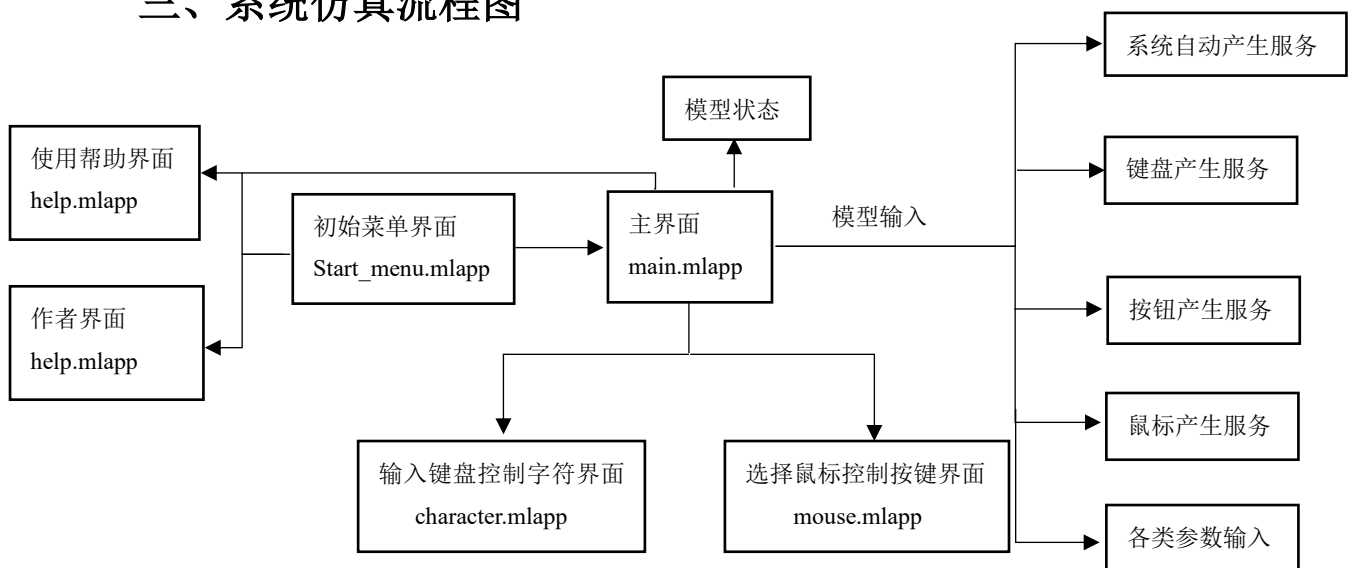
图 4. 按钮产生服务的模型输出



图 5. 鼠标产生服务的模型输出

由以上仿真结果可知，非系统自动产生服务的时间和占用率普遍低于系统自动产生服务，并且等待时间大体趋势上也小于系统自动产生服务，而对于三类服务来说，电话咨询服务的优先级小于另外两类服务，从结果上看其平均等待时间大于现场售票和现场咨询，均符合先前的模型条件要求和分析。

三、系统仿真流程图



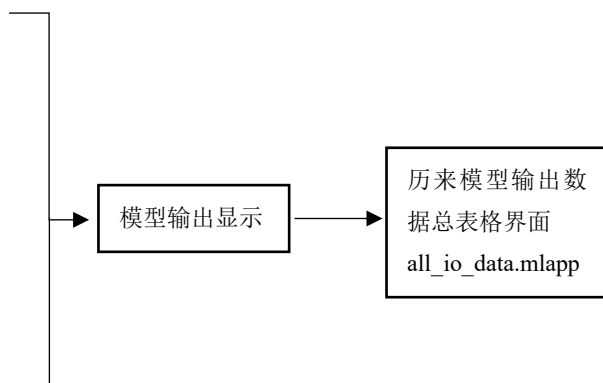


图 6.系统仿真流程图

四、试题实现中的关键难点

1. 电影院售票与咨询系统仿真页面设计

本次仿真作业采用 `appdesigner` 工具进行界面设计，默认布局下在组件库中有许多功能的小组件可直接进行拖拽来实现界面布局，同时系统会生成相对应代码。而且大部分组件都有与之对应的多个回调函数供我们使用，例如按钮组件，可通过添加 `ButtonPushed` 函数进行回调，在函数中键入执行内容，当按下按钮时便会执行该函数，实现对应功能。`appdesigner` 与设计者之间有很好的交互性，我们也可以直接在组件浏览器中修改组件属性，如位置、内容、字体、背景颜色等等。在熟悉所有组件的功能后，开始进行试题要求及内容的界面设计安排。首先考虑到可能需要多次改变条件进行仿真，而本系统的主要部分在于动画显示与模型数据输出，故将模型输入即初始条件设置安排在整体界面的左方，模型输出安排在整体界面的右方，而中间部分则是主要部分即人员的动态显示过程。模型主要采用了“编辑字段（数值）”、“面板”、“标签”、“按钮”、“单选按钮组”、“信号灯”、“工具栏”，“选项卡组”等组件，通过对部分组件回调函数的调用以及界面 `app` 属性全局变量和不同 `app` 之间变量的数据传递，实现人员交互和模型仿真。

2. 人员的动态显示过程

为了实现人员的动态显示，首先需要弄清楚动画是怎么实现的，实际上动画视频等等也是由许多帧画面的不断更新而实现的视觉上的动态效果。要实现动态显示，我们需要在坐标区上进行画图不断更新以实现动画。而对于本仿真系统来说我们需要模拟人从影院入口到售票处或者现场咨询处位置的运动，可以通过获取坐标区的 `children` 属性，清除掉刚刚画出的人，再更新位置将其重新画出，只需要画出人运动路线上的每一帧即可实现运动。考虑到模型的简化性以及现实情况，所有人从入口进入向售票处或咨询处直线前进，那么只需要确定起点和终点，再在二者之间取一定数量的点，通过 `for` 循环便可以具体实现。但鉴于程序的运

行速度，实际上这个过程以极小时间便可以完成，故需要在循环内加入 `pause()` 函数(输入参数单位为 s)进行暂停，本系统设置为 0.1ms。至此我们已经实现了一个人从一个地点到另一个地点的运动，但实际情况是同一时间内会有多个人从入口进入，而作为终点的售票处和咨询处均是固定的，只需考虑因为人数的不同而有多个人的进入造成起点不同，在这里我将其设置为均匀分散在入口处上下两侧。具体实现如下：

```
%运动更新到对应位置
x = linspace(0,dest_x,25);
if ~isempty(dest_y) %判断此时是否无人
    k = (dest_y-original_y)/dest_x;
    b = dest_y-k*dest_x;
    y = k'.*x+b';
    for i=1:length(x)
        h = get(app.draw, 'children');
        for j=1:app.enter_people_num
            delete(h(2*j-1));
            delete(h(2*j));
        end
        for j=1:app.enter_people_num
            draw_people(app,x(i),y(j,i));
        end
        pause(0.0001);
    end
end
```

图 7.人员动态显示的部分代码

对于服务员的动态显示，由于服务员的运动动画独立于人员显示，故采用另外一个坐标系进行单独作画，服务员的运动过程与进来的顾客相同，均为直线更新运动，服务员的关键点主要在于其服务的动态选择上。但是与顾客人员运动不同的是，存在着服务员处理完一种服务后需要紧接着另一种服务，那么这个时候再回到初始位置运动到服务处是不可取的，这个时候需要判断服务员所处的状态，根据对应状态来实现运动，比如若服务员处理完售票后前往现场咨询处则直接沿着 y 轴向下运动即可。具体实现如下：

```
%运动更新到对应位置
x = linspace(server_x,dest_x,update_rate);
k = (dest_y-server_y)/(dest_x-server_x);
b = dest_y- k * dest_x;
y = k*x+b;
for i=1:length(x)
    h = get(app.draw_2, 'children');
    delete(h);
    draw_server(app,x(i),y(i));
    pause(pausetime);
end

elseif app.server_laststate == 1 %服务员上一状态
%运动更新到对应位置
y = linspace(sale_y,call_y,update_rate);
for i=1:length(y)
    h = get(app.draw_2, 'children');
    delete(h);
    draw_server(app,dest_x,y(i));
    pause(pausetime);
end
```

图 8.服务员动态显示的部分代码

3. 泊松分布与负指数分布的实现

泊松分布的概率函数为： $P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, \dots$ λ 是单位时间内随机事件的平均发生次数。

负指数分布又称指数分布，其概率密度函数为： $f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$ ，其中 $\lambda > 0$ 被称为率参数，即每单位时间内发生该事件的次数。

在 matlab 中可以通过 `random('poisson', λ , 1)`，`exprnd(λ)`函数产生服从 λ 参数的泊松分布和负指数分布随机数。那么我们还需要确定一个单位时间，每经过一个单位时间后产生相应分布随机数，即进来的顾客人数和服务员的服务时长，则此时需要启用定时器。本系统中每 5s 中产生进来一批人数服从泊松分布的顾客，即单位时间设置为 5s。

```
%生成三种人数，服从泊松分布
app.sale_people_num = random('poisson',app.sale_rate.Value,1);
app.ask_people_num = random('poisson',app.ask_rate.Value,1);
app.call_people_num = random('poisson',app.call_rate.Value,1);
```

图 9.顾客人数的泊松分布实现部分代码

```
pausetime = exprnd(app.sale_time_rate.Value); %负指数分布处理时间
```

图 10.服务员服务时长的负指数分布实现部分代码

```
function timer2_init(app) %定时器2初始化
    app.datetimer2 = timer;
    app.datetimer2.StartDelay = 0;
    app.datetimer2.Period = 5; %设置单位时间为5s
    app.datetimer2.ExecutionMode = 'fixedRate';
    app.datetimer2.BusyMode = 'queue';
    app.datetimer2.TimerFcn = @(~,~) generate_people_num(app);
    start(app.datetimer2);
end
```

图 11.定时器 2 初始化

4. 系统自动产生服务的实现

由模型假设有现场售票与现场咨询的优先级大于电话咨询的优先级，定义现场售票与现场咨询的优先级相同，故在现场售票与现场咨询和电话咨询都有人时候应优先考虑现场售票和现场咨询，再在这两者之间进行等概率随机选择，等处理完现场售票与现场咨询的人之后，再去处理电话咨询。而且还需考虑在三个服务处服务员的移动情况，比如从售票处移动到现场咨询处时应该是售票处此时无人而现场咨询处有人的情况，从售票处移动到电话咨询处时应该是服务员刚处理完售票处的顾客，而且此时现场咨询无人而电话咨询处有人的情况等等。当处理完所有人之后，服务员还要回到初始位置。由于此部分代码过多，不放入报告中，可直接查看文件。

5. 键盘产生服务的实现

本系统默认'a','s','d'字符分别对应售票、现场咨询和电话咨询，可通过“输入控制字符”按钮，调出界面进行改变。仿真开始后，键入字符将会启用 `UIfigureKeyPress(app,event)`函数，键入的字符在 `event` 的 `key` 属性中保存。根据键入字符对服务员的服务方式进行改变。

```

% Key press function: UIFigure
function UIFigureKeyPress(app, event)
    key = event.Key; %储存按键状态
    if app.keyboard_select.Value %如果选中'键盘选择'
        switch key
            case app.key_sale
                if app.sale_people_allnum > 0
                    %如果售票处等待人数大于0，则改变服务员状态为售票服务
                    app.server_state = 1;
                end
            case app.key_ask
                if app.ask_people_allnum > 0
                    %如果现场咨询处等待人数大于0，则改变服务员状态为现场咨询服务
                    app.server_state = 2;
                end
            case app.key_call
                if app.call_people_allnum > 0 && ((app.sale_people_allnum + app.ask_people_allnum) == 0)
                    %如果电话咨询处等待人数大于0并且售票处和现场咨询处均无人，则改变服务员状态为电话咨询服务
                    app.server_state = 3;
                end
            end
        end
    end
end
end

```

图 12.键盘产生服务的部分代码

6. 鼠标产生服务的实现

本系统默认左键、滚轮、右键分别对应售票、现场咨询、电话咨询服务，可以通过点击“输入控制按键”按钮，弹出界面进行修改。当鼠标点击后将会调用 `UIFigureWindowButtonDown(app,event)` 函数，鼠标按键信息保存在 `app.UIFigure` 的 `SelectionType` 属性中，左键对应'normal'，右键对应'alt'，滚轮对应'extend'，当鼠标点击在图中的灰黑色区域即电影院区域即可产生对应服务。

```

% Window button down function: UIFigure
function UIFigureWindowButtonDown(app, event)
    %选中'鼠标选择'并且鼠标点击在灰黑色区域内
    if app.mouse_select.Value && app.UIFigure.CurrentPoint(1)>210 &&...
        app.UIFigure.CurrentPoint(1)<910 && app.UIFigure.CurrentPoint(2)>5 &&...
        app.UIFigure.CurrentPoint(2)<770
        origin_s = app.server_state; %保存原来的状态
        switch app.UIFigure.SelectionType
            case 'normal' %左键
                app.server_state = app.mouse_dir.left;
            case 'alt' %右键
                app.server_state = app.mouse_dir.right;
            case 'extend' %滚轮
                app.server_state = app.mouse_dir.middle;
        end
        switch app.server_state
            case 1
                if ~app.sale_people_allnum
                    %如果鼠标点击售票按键，但是售票无人等待，则继续保持原状态
                    app.server_state = origin_s;
                end
            case 2
                if ~app.ask_people_allnum
                    %如果鼠标点击现场咨询按键，但是现场咨询无人等待，则继续保持原状态
                    app.server_state = origin_s;
                end
            case 3
                if (app.sale_people_allnum + app.ask_people_allnum) > 0 || app.call_people_allnum == 0
                    %如果鼠标点击电话咨询按键吗，但是售票或现场咨询有人或者电话咨询无人，则继续保持原状态
                    app.server_state = origin_s;
                end
            end
        end
    end
end
end

```

图 13.鼠标产生服务的部分代码

7. 按钮产生服务的实现

按钮产生服务与键盘、鼠标产生服务均类似，通过改变服务员此时的服务状态即服务方式来实现。当选择服务的方式勾选按钮选择后，点击售票处、现场咨询处或者电话咨询处的按钮，调用 ButtonPushed 函数即可产生相应服务。

```
% Button pushed function: sale_label_button
function sale_label_buttonButtonPushed(app, event)
    if app.button_select.Value && app.sale_people_allnum > 0
        app.server_state = 1;
    end
end

% Button pushed function: ask_label_button
function ask_label_buttonButtonPushed(app, event)
    if app.button_select.Value && app.ask_people_allnum > 0
        app.server_state = 2;
    end
end

% Button pushed function: call_label_button
function call_label_buttonButtonPushed(app, event)
    if app.button_select.Value && app.call_people_allnum > 0 && ((app.sale_people_allnum + app.ask_people_allnum) == 0)
        app.server_state = 3;
    end
end
```

图 14. 按钮产生服务实现的部分代码

8. 模型输出的计算

根据试题建模过程的分析可从理论上得出三类服务的平均等待时间与服务员的工作时间内的占用率。在程序中主要通过仿真开始的时候计时 tic，当每个人从入口产生时会记录一个属于自己的进入时刻，当他被处理后再记录一个处理完的时刻，二者相减即为等待时间，将该等待时间存储进列表内，最后对列表求和平均，即可得到对应服务的平均等待时间。当顾客服务完成后还可记录一下此时所用的服务时长，当仿真结束后，将服务员的三类服务总时长相加，并除以总工作时间即从仿真开始到仿真结束的时间，得到服务员工作时间内的占用率。

```
app.process_sale_num = app.process_sale_num + 1;
app.server_sale_time(app.process_sale_num) = toc - app.server_sale_time(app.process_sale_num); %获取等待时间
pausetime = exprnd(app.sale_time_rate.Value); %负指数分布处理时间
pause(pausetime);
app.sale_people_allnum = app.sale_people_allnum - 1; %处理完人数减一
app.sale_people_allnum_text.Text = string(app.sale_people_allnum); %更新可视化数量
app.server_sale_pause_list = [app.server_sale_pause_list pausetime]; %处理时间列表

rate = (sum(app.server_sale_pause_list)+sum(app.server_ask_pause_list)+sum(app.server_call_pause_list))/app.server_time; %计算服务员工作时间的占用率

%计算各类服务的平均等待时间
if app.process_sale_num
    app.server_sale_time = sum(app.server_sale_time(1:app.process_sale_num))/app.process_sale_num;
end
```

图 15.模型输出计算的部分代码

9. 模型输入的约束

考虑到实际情况现场售票人员、现场咨询人员、电话咨询人员的到达率不能过高，否则服务员处理不及将会造成人数拥堵，同理服务员的服务时长参数也不

能过低（负指数分布中服务时长参数越大，服务时间越短）

```
elseif app.sale_rate.Value<0||app.sale_rate.Value>5||...
    app.ask_rate.Value<0||app.ask_rate.Value>5||...
    app.call_rate.Value<0||app.call_rate.Value>5||...
    app.sale_time_rate.Value<0.2||app.ask_time_rate.Value<0.2||app.call_time_rate.Value<0.2
run errortip5.mlapp;
```

图 16.模型输入约束的部分代码

10. 停止仿真的处理

当仿真开始以后，整个程序除以一个 while(1)循环中，要终止这个循环就需要设定一个变量，当变量改变以后 break 出循环，设定的终止变量为 app 的属性 stop_simulate，当按下仿真结束按钮后 stop_simulate 由 0 变为 1，通过 if 语句 break 跳出循环。但经过实际尝试，发现需要在 while(1)循环中放置 pause 函数进行一小段时间的暂停，让程序有时间来得及反应才能进入 if 语句跳出循环。设置的 pause 时间为 1 μ s。

```
while(1)
    pause(0.000001); %暂停极小一段时间留以程序反应，使得能够终止仿真
    if app.stop_simulate == 1
        stop(app.datetimer2); %暂停定时器2
        delete(app.datetimer2); %删除定时器2
        app.work_state = 0;
        break
    end
```

图 17.停止仿真处理的部分代码

11. 模型输入输出的总表格记录

当每次仿真结束后此次仿真的输入参数、输出参数将被记录在 app.subtable 中。点击“模型总输入输出参数表格”按钮后，弹出表格，里面记录着从运行系统后的所有仿真结果包括输入和输出。

```
% Button pushed function: all_io_data_button
function all_io_data_buttonButtonPushed(app, event)
    app.subtable = all_io_data();
    app.subtable.UITable.Data = [app.subtable.UITable.Data;app.input_output];
end
```

图 18.模型输入输出表格的部分代码

五、程序运行指南

1. 文件介绍

本电影院售票与咨询仿真系统一共包括 12 个.m 文件: start_menu.m、help.m、author.m、main.m、character.m、mouse.m、all_io_data.m、errortip.m、errortip2.m、errortip3.m、errortip4.m、errortip5.m 和 11 张图片 (.png 和.jpg 格式)

start_menu.m 文件: 欢迎菜单界面文件, 可以通过该界面打开使用帮助、作者介绍和进入系统

help.m 文件: 使用帮助文件, 通过该界面可以查看本系统的具体使用步骤以及所有组件的功能

author.m 文件: 作者介绍界面

main.m 文件: 电影院售票与咨询仿真系统主文件, 有模型输入、仿真动态显示、模型状态、模型输出等模块

character.m 文件: 键盘输入控制字符文件, 可以通过该界面选择产生对应服务的字符

mouse.m 文件: 输入鼠标控制按键文件, 可以通过该界面选择产生对应服务的鼠标按键

all_io_data.m 文件: 模型总输入输出表格文件, 包括自进入系统以来的所有仿真运行结果

errortip.m 文件: 错误提示文件, 在键盘输入控制字符界面中, 若输入字符为空或者重复将运行该文件

errortip2.m 文件: 错误提示文件, 在输入鼠标控制按键文件中, 若选择按键重复则运行该文件

errortip3.m 文件: 错误提示文件, 当程序正在运行的时候再次点击了仿真开始则会运行该文件

errortip4.m 文件: 错误提示文件, 当输入参数中到达率不在 0-5 之间或者服务时长参数小于 0.2 则会运行该文件

errortip5.m 文件: 错误提示文件, 当程序没有开始的时候点击了仿真结束则会运行该文件

2. 电影院售票与咨询仿真系统介绍

在命令行窗口中输入 start_menu 运行文件, 或者打开文件 start_menu.m 在 appdesigner 中进行运行, 进入欢迎系统菜单界面, 可以选择进入系统或者退出系统, 同时在界面上方有工具栏, 可以打开使用帮助界面和作者介绍界面。当进入系统后可以看到左侧是模型输入, 中间为具体动态仿真过程展示, 右侧是模型输出和模型状态。在模型输入中, 依次是人员到达率、服务时长参数、服务方式的选择方式的输入, 勾选键盘选择或者鼠标选择则可看到对应的红灯亮起, 代表选中该方式, 对应面板中展示了控制字符或控制鼠标按键。点击确认仿真开始即可进行仿真。同时在界面的左下角也有使用帮助按钮, 便于不熟悉系统者进行

参阅。中间的动态仿真展示界面中左侧是入口，正中间从上到下依次是售票处、现场咨询处、电话咨询处，当顾客从入口进来以后达到指定位置后人数加 1，服务员位于展示界面右侧，通过系统自动或者人为选择到达服务处进行处理。模型输出面板的上方实时显示着时间，再从上到下依次是售票服务的平均等待时间、现场咨询的平均等待时间、电话咨询的平均等待时间以及服务员在工作时间的占用率。点击仿真结束确认即可结束仿真，此时上方模型输出将被清空。点击模型总输入输出参数表格将弹出表格界面，可以看到自进入系统以来所有仿真的输入输出结果。模型状态面板中显示着进影院的总人数和此单位时间时刻内进入影院的人数。

六、程序运行实例分析

以下为仿真程序的运行实例。在命令行中输入 `start_menu` 运行仿真程序，进入欢迎菜单界面。



点击使用帮助，将弹出使用帮助界面。

使用帮助

123

使用帮助

使用步骤：1.输入初始条件 2.确认仿真开始 3.待一定时间后结束仿真 4.得到模型输出 5.重新回到1步骤进行仿真

输入初始条件

1.输入相关参数

售票人员到达率0.5

咨询人员到达率0.8

电话人员到达率1

现场售票服务时长参数1

现场咨询服务时长参数1

电话咨询服务时长参数1

2.选择服务方式

服务员服务方式的选择

系统选择

键盘选择输入控制字符

按钮选择

鼠标选择输入控制按键

(1) 键盘控制字符输入

键盘输入确定字符控制

处理现场售票

处理现场咨询

处理电话咨询

确认

提示：需输入对应控制字符（一个），注意不要重复

本模型设定单位时间为5s

其中人员到达均服从泊松分布，输入参数为 λ ，即单位时间内产生的平均人数建议0-5为宜，否则人数过多造成拥堵。

其中服务时长均服从负指数分布，输入参数为单位时间内平均发生该事件的次数建议不小于0.2为宜，否则等待人数过多造成拥堵

一共设置了四种服务员的服务方式，分别为系统自动服务，键盘控制服务，按钮点击服务，鼠标左键、滚轮、右键控制服务

系统自动服务：服务员根据优先级（现场售票=现场咨询>电话咨询）进行服务其中现场售票和现场咨询同时有等待时随机选择

按键点击服务：由我们鼠标点击相应售票处或咨询处进行服务，服务的过程中也严格遵循优先级，当售票和现场咨询有人时不能服务电话咨询

输入现场售票服务、现场咨询服务、电话咨询服务对应控制字符
注意提示：仅输入一个字符，且不要重复，否则会提示相应错误信息，进行重新输入

点击上方 1、2、3 即可切换相应页面

使用帮助

123

2.鼠标控制按键选择

鼠标控制按键选择

现场售票left

现场咨询middle

电话咨询right

确认

根据下拉菜单选择对应服务的鼠标控制按键，注意不要重复选择，否则会提示相应错误信息

键盘选择对应字符

现场售票：a

现场咨询：s

电话咨询：d

鼠标选择对应按键

现场售票：left

现场咨询：middle

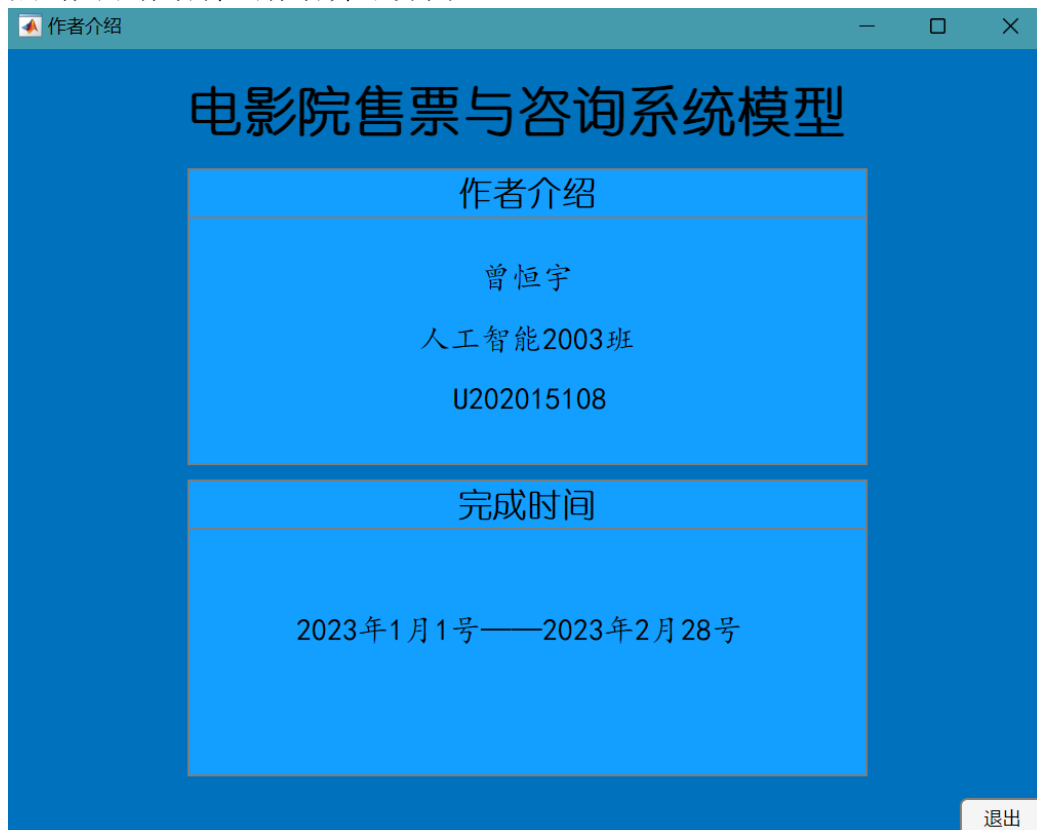
电话咨询：right

在页面左下方显示了对应的控制信息，以键盘选择对应字符为例，如果绿灯亮起则代表未选中键盘选择，红灯亮起则代表选中键盘选择。

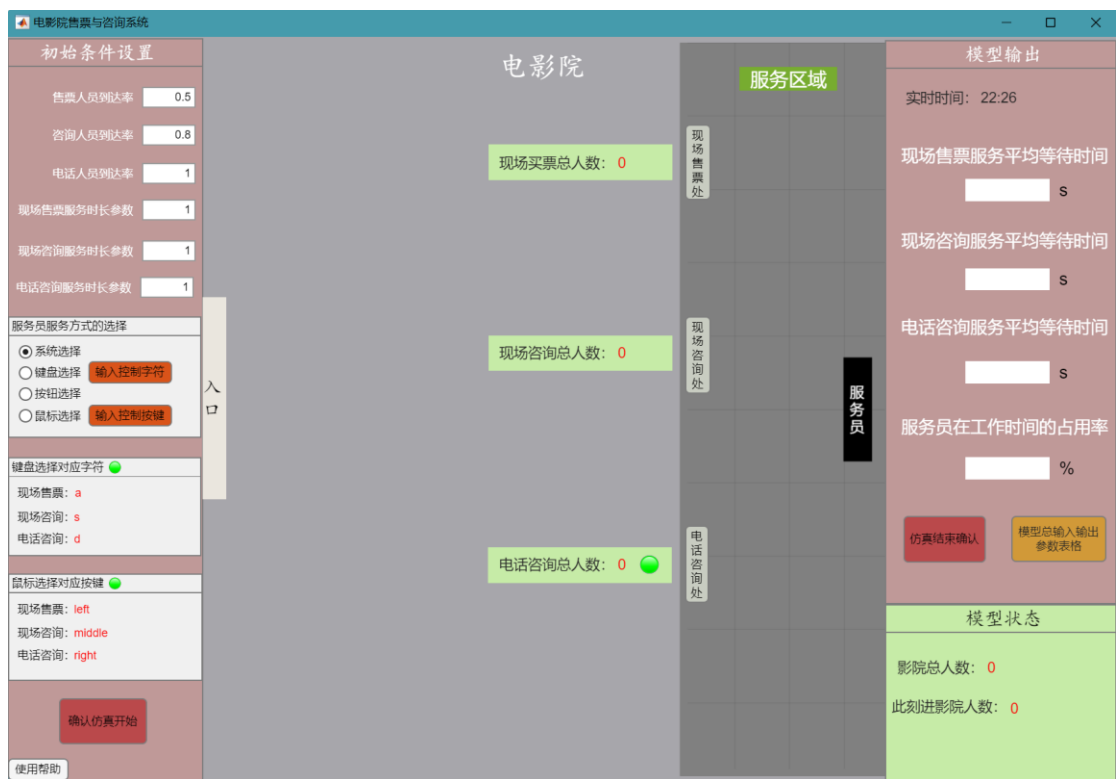
最后点击'确认仿真开始'按钮进行仿真



点击关于作者弹出作者介绍界面



点击“进入系统”按钮，进入仿真主界面，自动关闭欢迎菜单界面。



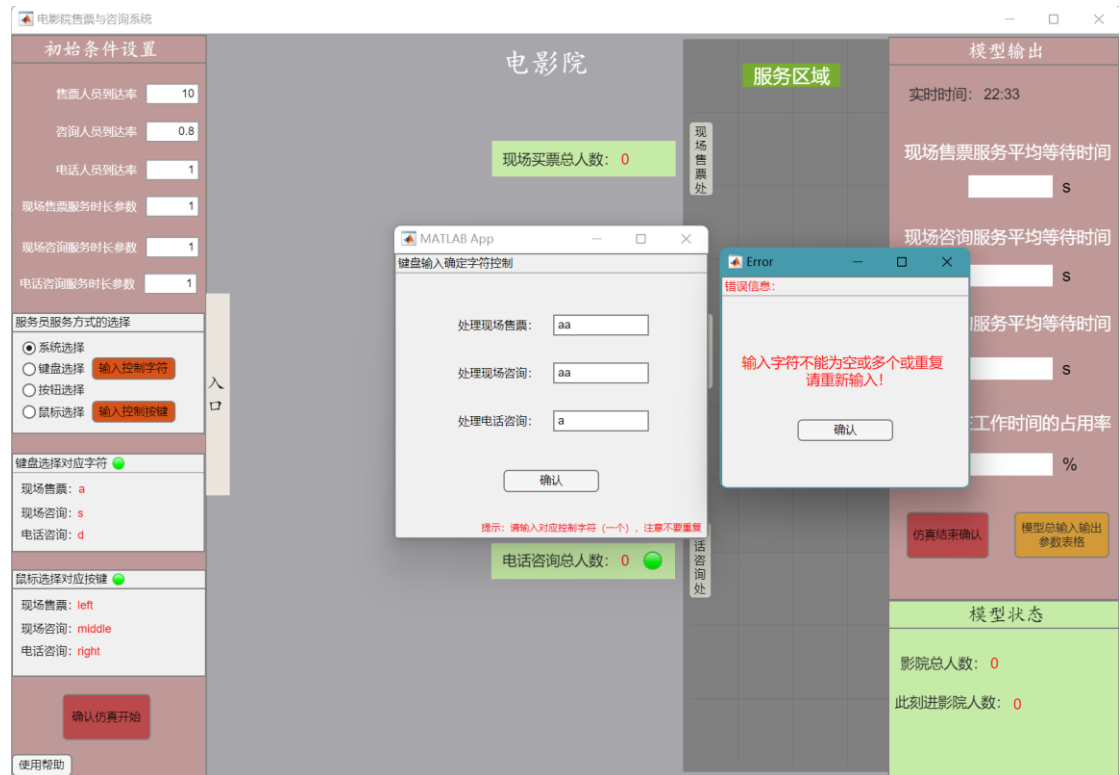
输入到达率参数应在 0-5 之间，时长参数应大于 0.2，如果参数不符合要求则会弹出错误提示窗口！例子如下：

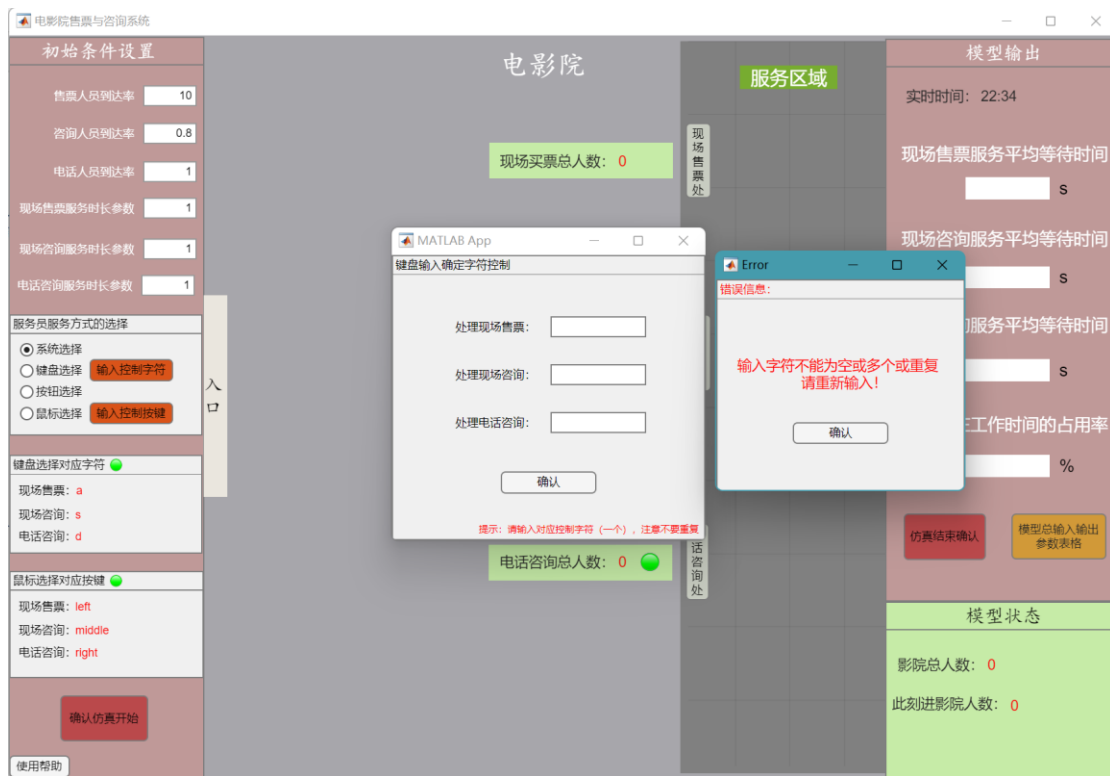


同时进行服务方式的选择。键盘控制默认字符和鼠标控制默认按键如面板所示。若要重新选择控制字符，则点击“输入控制字符”按钮，如下：



根据下方红色提示信息输入，若控制字符为空或多个或重复，则会弹出以下错误信息

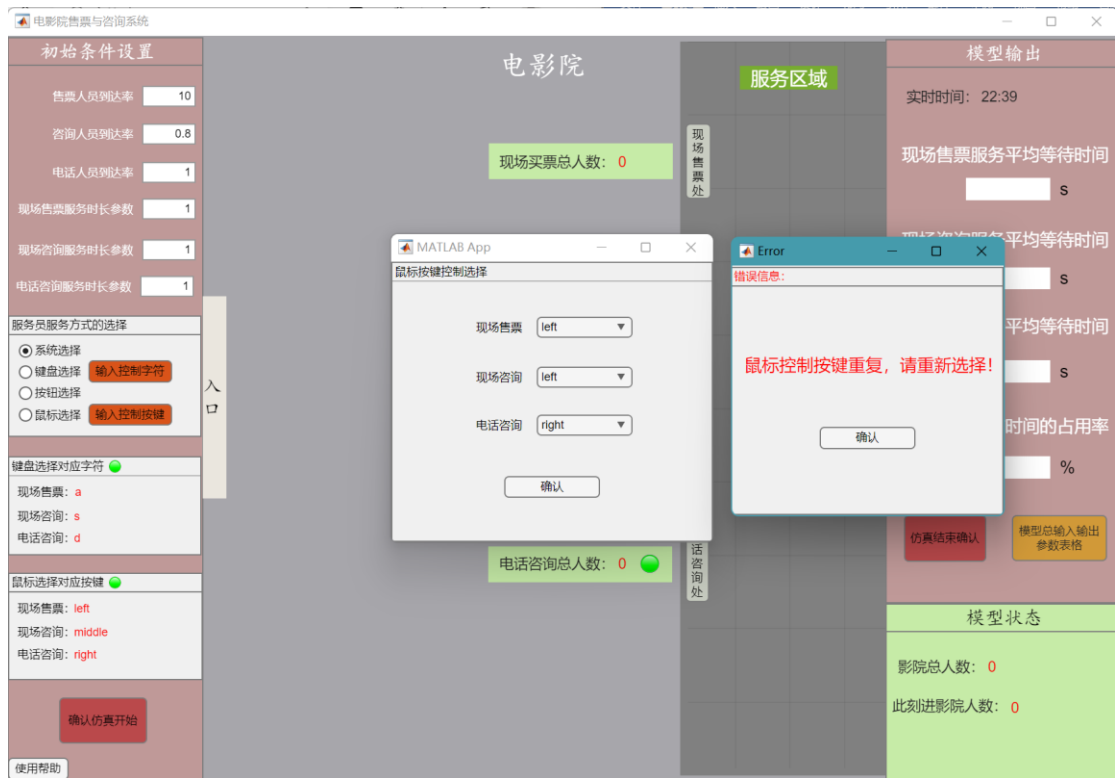




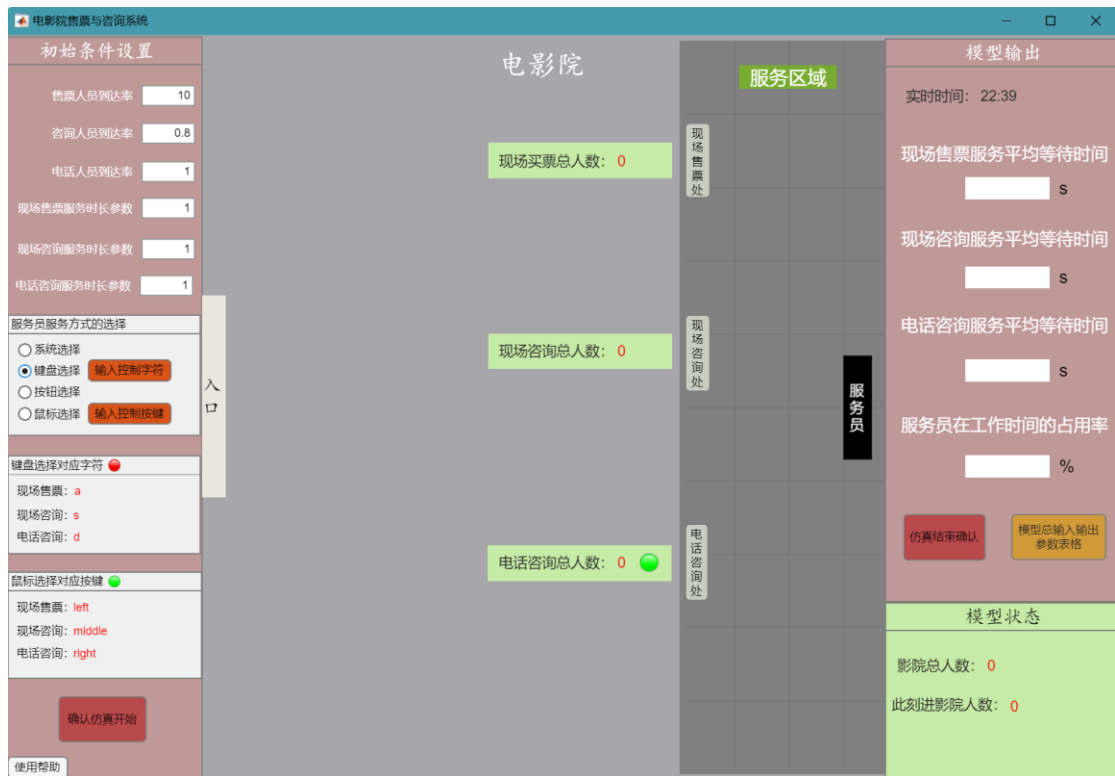
若要重新定义鼠标控制按键，则点击“输入控制按键”按钮

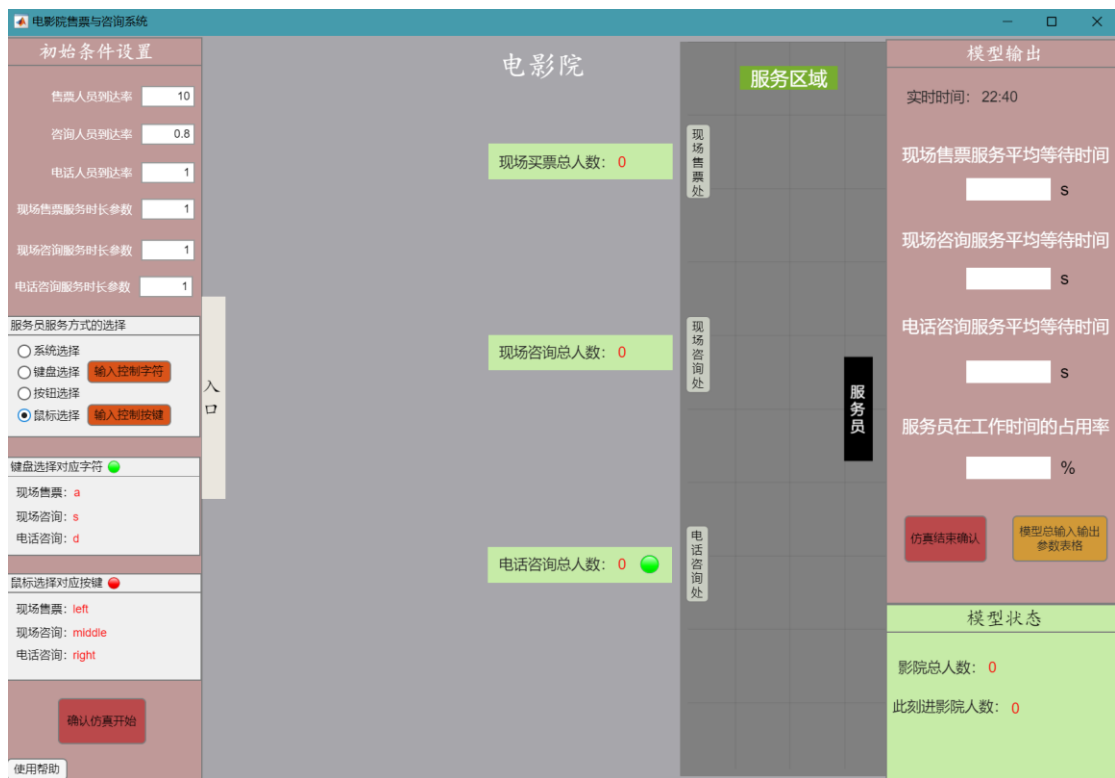


注意不要选择重复，若选择重复也会提示错误信息！

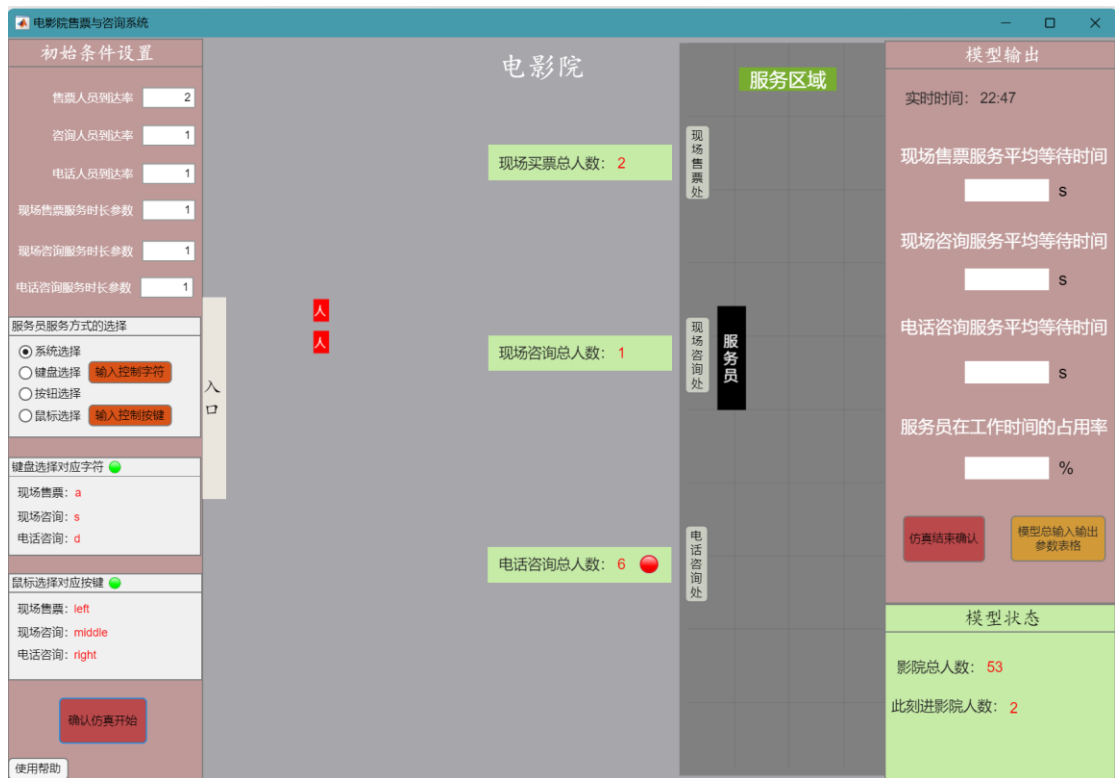


勾选“键盘选择”或“鼠标选择”则对应面板的红灯将会亮起，代表被选中

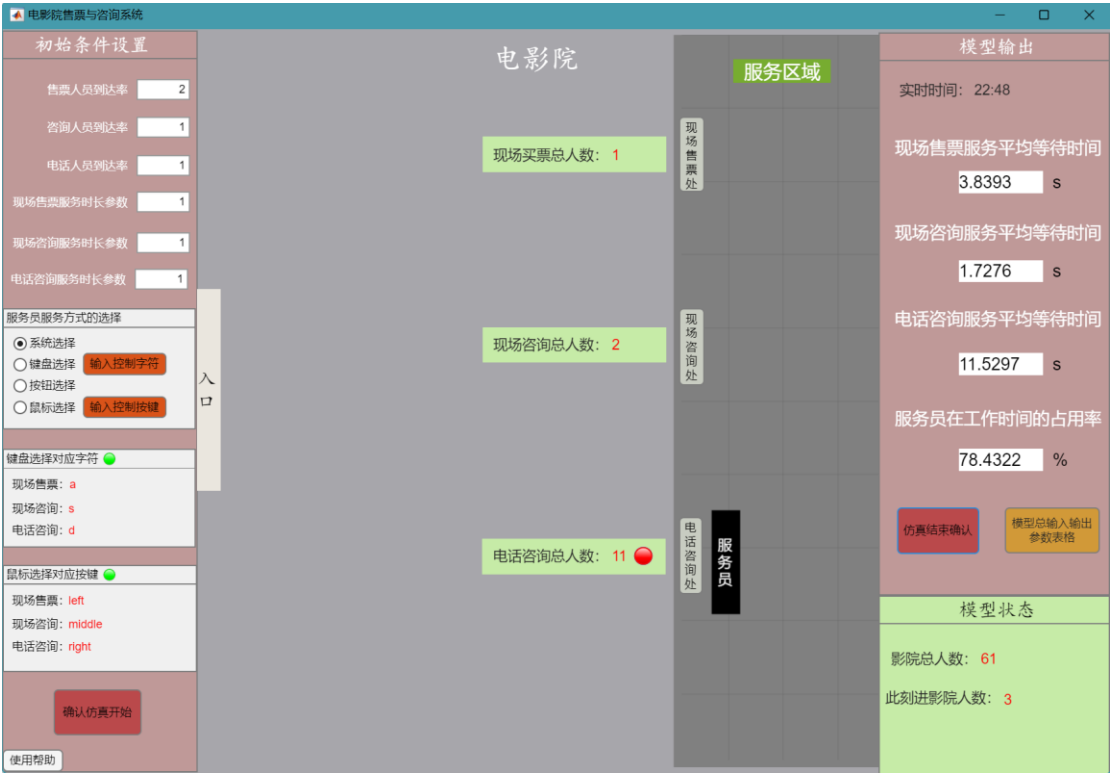




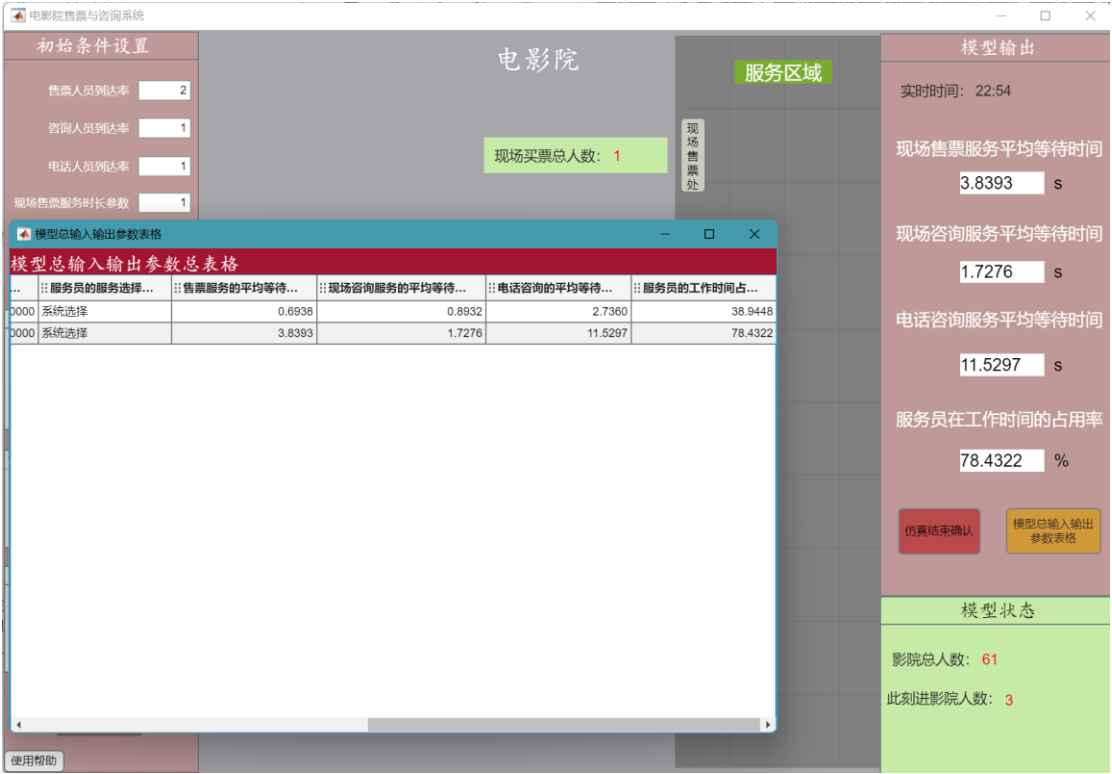
以系统自动选择仿真为例，运行过程一截图如下：



结束仿真后，如图



点击“模型总输入输出参数表格”按钮，弹出表格界面



七、新增功能及亮点

1.题目要求三种选择服务方式：系统自动产生服务、键盘产生服务和鼠标产生服务，我还定义了一种选择服务方式——按钮产生服务，点击服务处的按钮即可产生对应服务。

2.可以由使用者自定义键盘按键和鼠标按键控制服务，实现了服务的灵活性。

3.在仿真主界面中显示了控制字符或控制按键的具体信息，便于使用者在进行仿真的时候查看。

4.人员动态显示考虑比较周全，对于同一批同一时间从入口进来的顾客来讲，并不是从一个固定点进入，而是在入口原点两侧均匀分布的，可以明显区分不同顾客进入。而且服务员在服务的运动过程中存在着许多种运动情况，比如从初始位置到三个服务处，在其中一个服务处处理完后运动到另一服务处，当售票处和现场咨询仍然有人时要优先留在这两个地方，不能去往电话咨询处理以及全部顾客处理完后需要回到初始位置等等情况。

5.本系统包含着多个界面，不同界面之间存在着参数传递以及函数调用。

6.界面设计友好，整体界面的左侧为输入，右侧输出，而且实时显示着模型状态——各类服务处的人数、影院总人数、此刻进人数以及时间等等，并且电话咨询有人以信号灯来展示，当有人电话等待时亮红灯，无人时亮绿灯。

7.考虑到可能进行多次仿真，故记录了从进入系统以来的所有仿真数据包括每次的输入输出，存放在表格中便于查看。

八、系统仿真心得体会

这是我第一次对 GUI 界面设计进行这么深入的了解，虽然在之前的计算机网络实验中也进行过一个简单的 GUI 设计来实现海明编码，但当时对其也是一片空白，也不知道应该怎么去实现我脑海中想要做的更多功能。上课时系统地学习了系统仿真与 MATLAB 的相关知识，本次题目的主要难点在于不同界面之间的数据交互以及人员之间的交互动态显示，在这方面我花费了大量时间去了解不同界面之间参数是如何传递的，动画是如何显示的，以及动画中不同人员运动的逻辑关系。本次作业让我们利用 MATLAB 去解决了一个在生活中存在的实际问题，让我们了解到如何运用课程知识去解决实际问题，以及在解决问题的过程中自我学习能力的重要性。

同时在界面设计中颜色的搭配对于自己来说也是一大难题，总是选不好恰当的颜色而使得不同模块或组件之间显示非常违和。于是我特意查阅了许多色彩搭配、色卡搭配的资料，才完成了这个颜色布局。

还记得上一次用 MATLAB 完成具有一定大工作量的作业还是在学习数据科学基础的时候，当时我们需要用 MATLAB 解决一个实际物理问题，仿真出运动曲线以及运动数据等等。在完成本次试题后，我对 MATLAB 实现系统仿真有了更深层次的理解，更好地掌握了相关的仿真技巧。感慨实现仿真系统是可能需要花非常多时间和精力完成的事情，也让我领悟到什么事情都不是一蹴而就的，我们都需要沉下心来完成每一件事情。在进行课程设计的过程中，不可避免的会多

次出现 bug 以及一些逻辑漏洞，每次我不理解的时候就会查阅资料或沉思问题，直至找到答案才豁然开朗，原来找到问题答案的那一刻才是最令人感到柳暗花明又一村的，这也告诉我，人生中跟课程设计也是相似的，我们遇到困难就要寻求帮助或者努力克服，当打破前方的巨石后，又是一片光景。