**深大漂流记**

--LabVIEW游戏程序说明

**作者1： 陈耀昱 1810294007@email.szu.edu.cn**

**作者2： 赖伟鉴 1810294033@email.szu.edu.cn**

**作者3： 杨一帆 1810294048@email.szu.edu.cn**

**作者4： 叶文圳 1810294049@email.szu.edu.cn**

**目录**

[程序设计说明书 1](#_Toc12348503)

[1.引言 1](#_Toc12348504)

[1.1编写目的 1](#_Toc12348505)

[1.2游戏背景 1](#_Toc12348506)

[1.3游戏设计亮点 1](#_Toc12348507)

[1.4 人员及分工 2](#_Toc12348508)

[2使用说明 3](#_Toc12348509)

[2.1 游戏主界面 3](#_Toc12348510)

[2.2 进入游戏 4](#_Toc12348511)

[2.3状态与功能按键 5](#_Toc12348512)

[3.总体设计 13](#_Toc12348513)

[3.1需求概述 13](#_Toc12348514)

[3.2程序结构 13](#_Toc12348515)

[4．程序描述 18](#_Toc12348516)

[4.1 游戏开始 18](#_Toc12348517)

[4.2 游戏进行 19](#_Toc12348518)

[4.3 游戏结束 33](#_Toc12348519)

# 程序设计说明书

# 1.引言

## 1.1 编写目的

本游戏为一款以深圳大学为背景的经营类小游戏，旨在帮助人们在繁忙的生活中缓解疲劳。同时本游戏还具有一定的教育作用，如“健康值过低将死亡”提醒玩家健康第一、“欠债会因复利效应越来越多”警示玩家远离校园贷。

## 1.2 游戏背景

游戏主人公是一名深圳大学机电学院大四的学生，在临近毕业之际，想利用校园里的最后的30天校园生活，通过在不同的地方购买和贩卖不同的商品来挣取人生的第一桶金。

游戏设有四个交易地点，包括：机电学院、饭堂、图书馆、演会中心。点击新的交易地点，会进入新的一天，市面上商品的种类和价格刷新。

游戏设有四个特殊地点，包括：银行、信贷、医院、放弃游戏。银行可进行存款/取款；信贷可偿还欠债；医院可恢复健康值。

游戏过程中，玩家也会受特殊事件影响，如：生病、深大购物节、供不应求、通货膨胀。

## 1.3 游戏设计亮点

1. 以深圳大学为游戏背景，题材新颖，贴近生活；
2. 经营类游戏，可玩性强，简单易上手；
3. 注重用户体验，采用友好的游戏界面，独立设计扁平化风格的按钮UI以及游戏壁纸；
4. 多样性强，包含多种游戏事件，如生病、通货膨胀、高利贷、存款利息、购物节、商品供不应求等；
5. 能有效监测出违规输入并提示用户。

# 2使用说明

## 2.1 游戏主界面



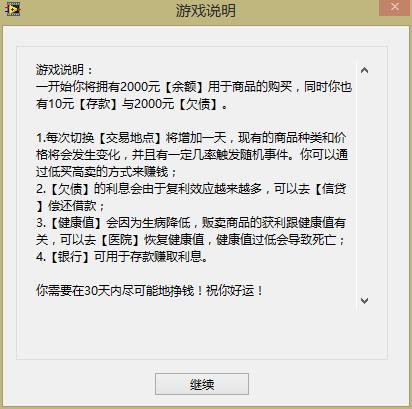
图1 游戏主界面

在本游戏中，**玩家需要在30天的时间内通过商品交易赚取尽可能多的钱财**。图1展现了本游戏的运行主界面，由图可知，玩家状态众多，也可执行许多不同的动作，每个部分的功能为：

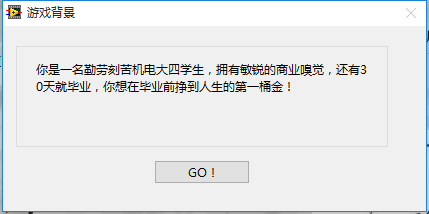
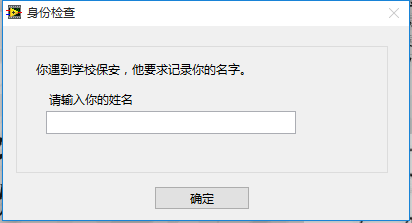
* **余额、存款、欠债、健康值与天数**为玩家的角色**状态**，其中，余额的功能包括商品买入、存款、还钱以及看病；玩家可把余额放入银行成为存款，存款会产生一定倍率的利息；欠债值会随着天数增加，玩家需尽早还完债款；每天玩家的健康值都会有一定概率减少，健康值不满会使玩家卖出商品时的卖出价格下降；玩家一共有30天的游戏时间，每按下一个交易地点，天数变会加一，而其他按键将不会改变游戏天数。在上述状态的**初始值设定**中，余额初始值为2000，存款为10，欠款2000，健康值100，天数为1；
* **商品价格**展示了当天玩家可以选择买入或卖出的商品及其价格，这些商品会随机出现，价格也会随机变动，**库存**则会随着玩家对物品的买入或卖出操作而变化，作为买卖交易游戏的核心，玩家需要根据市面与库存的差价选择买入或卖出；
* **机电学院、饭堂、图书馆与演会中心为交易地点**，玩家每点击一次交易地点，游戏天数会加一，同时市面商品、存款数、欠债数、健康值都会随之改变；
* **特殊地点**：除买卖商品外，玩家还可在特殊地点中执行其他特殊操作。其中，玩家可前往**银行**中通过存取款转换余额与存款的数额；玩家可通过**信贷系统**还掉目前的欠债；**医院**用于恢复玩家的健康值，但同时花费玩家的余额；若玩家选择**放弃游戏**，则可直接结束游戏进入结算画面；
* **其它功能**：右上方的**游戏说明**对本游戏进行了简要说明；玩家显示控件可显示玩家姓名；点击**查看富豪榜**可查看游戏分数最高的玩家，点开后还可选择**关闭富豪榜**；最下方的**提示**显示控件则会给出游戏的一些运行规则与建议。

## 2.2 进入游戏

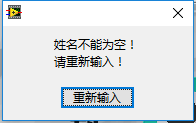
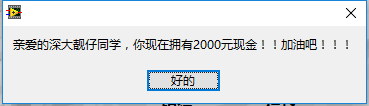
在玩家点击运行程序后，游戏即开始运行，最开始将弹出窗口进行游戏背景介绍与游戏说明，同时要求玩家输入姓名，而若玩家为输入用户名，则游戏将进行提示，直至输入完成后开始进入游戏。以上内容的前面板显示如图2所示：



(a)游戏说明

(b)游戏背景介绍 (c)玩家姓名输入

(d)玩家姓名为空时提示 (e)开始进入游戏

图2 程序运行后的玩家输入、游戏介绍等

## 2.3 状态与功能按键

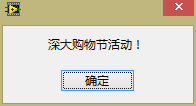
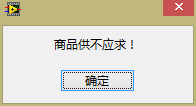
接下来则将正式开始游戏，本文将对每种玩家状态与功能按键逐个进行介绍：

**（1）商品价格显示栏**

上文已介绍，对于市面的商品及价格都会随机变化，商品价格随着交易地点的按下（同时天数加一）而发生种类和数量上的变换。图3给出了游戏运行时某一天的市面商品与点击交易地点后的市面商品变化情况，而游戏会有一定几率触发两类事件，其中“**深大购物节活动**”意味着市面商品价格减半，而“**商品供不应求**”意味着市面商品价格翻倍：

(a)第n天的市面商品 (b)第n+1天的市面商品

(c)商品价格减半提示 (d)商品价格翻倍提示

图3 商品价格显示栏随天数变化的变换示意图

**（2）物品买入**

图4为玩家的物品买入示意图在买入物品过程中，玩家需要首先在商品价格显示栏中选定目标物品，选择买入数量后，点击“买入”按钮，即可完成对该物品的购买。由图4可知，在买入选定物品后，库存中该物品的数量会增加，同时买入价格则会是该物品多次买入的均值。

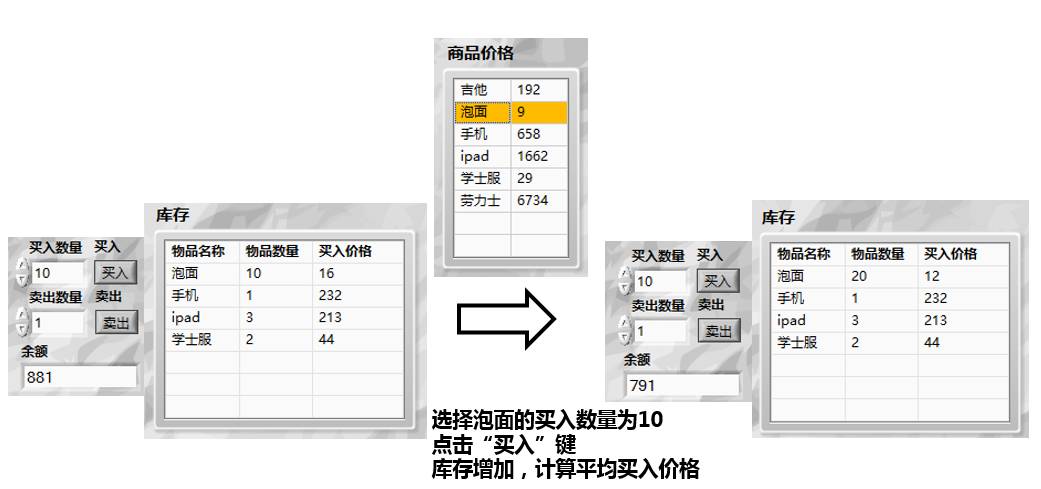
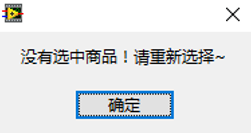
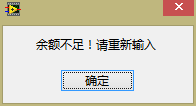
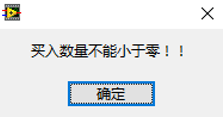


图4 物品买入示意图

在物品买入过程中，若玩家的输入有误，将会触发错误提示。在买入过程中，涉及到的错误输入包括：**玩家未选择商品、余额不足、商品数量为负**。买入错误触发错误提示如图5所示：

(a)未选中商品 (b)余额不足 (c)买入数量为负

图5 买入物品时错误输入产生触发提示

**（3）物品卖出**

物品卖出的方式与上文所述的物品买入方式类似，其示意图见图6。当玩家希望卖出库存物品时，需要首先在库存栏中点击希望卖出的物品，输入卖出数量后，点击“卖出”按键，即可实现对库存物品的卖出。相比物品买入，物品卖出需要多考虑的一个条件是：目前的市面中**是否存在希望卖出的商品**，若存在，则可成功卖出，若不存在，则卖出失败，如图6(b)所示。在卖出系统中还需要判定的错误输入情况如图7所示，其中包括**卖出数量是否大于零**，**是否选中卖出商品**，以及**卖出数量是否超过库存量**。

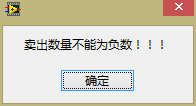
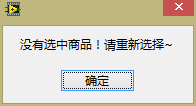
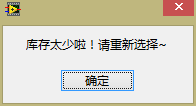


(a)物品成功卖出示意图



(b)因市面中并不存在该物品而卖出失败示意图

图6 物品卖出示意图

(a)卖出数量为负 (b)未选中商品 (c)卖出数量大于库存量

图7 卖出物品时错误输入触发提示

**（4）交易地点**

本游戏的交易地点包括机电学院、饭堂、图书馆与演会中心，图8展示了每个按键的按钮形状。当玩家按下交易地点后，游戏天数会加一，而天数的增加会触发多种事件，包括市面商品随机生成，存款与欠债数额上涨，以及健康值以一定几率下降。

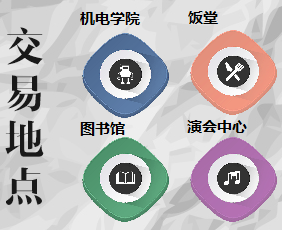


图8 游戏交易地点

**（5）特殊地点：银行**

银行系统提供用户进行存取款操作，影响的状态为玩家余额与存款。当目前天数玩家不再进行其他操作时，可将余额转为存款生成利息（利率为5%）；而当玩家在银行存有存款，又希望执行商品购买等操作时，则可将存款转为余额。关于银行存取款的操作示意图见图9，而存取款的错误操作触发提示见图10：

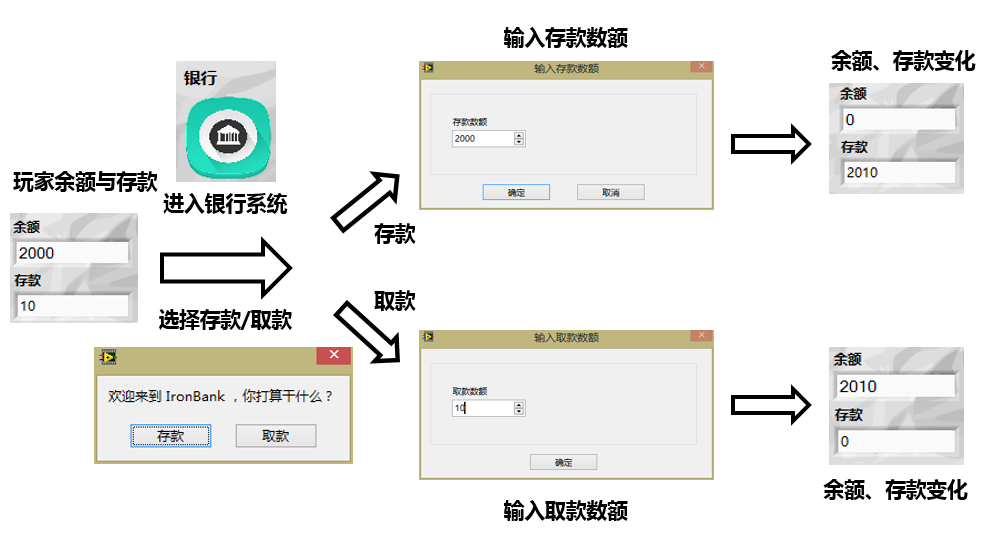
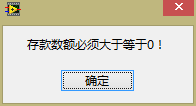
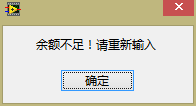
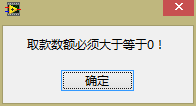
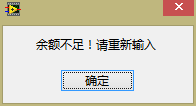


图9 银行存取款操作示意图

(a)存款输入数额小于0 (b)存款输入数额大于余额

(c)取款输入数额小于0 (d)取款输入数额大于存款数额

图10 存取款错误输入引发错误提示

**（6）特殊地点：信贷**

由于玩家存在欠债，而该欠债数额会随着天数的变化急剧增大（欠债利率10%），因此需要玩家尽早前往信贷系统偿还欠债数额。图11给出了信贷系统的操作示意图，在本游戏中，若玩家输入还款数额大于欠债值，系统将在扣取足够欠债值后，将多余的输入值放回至余额中。在信贷系统中的错误输入提示如图12所示：

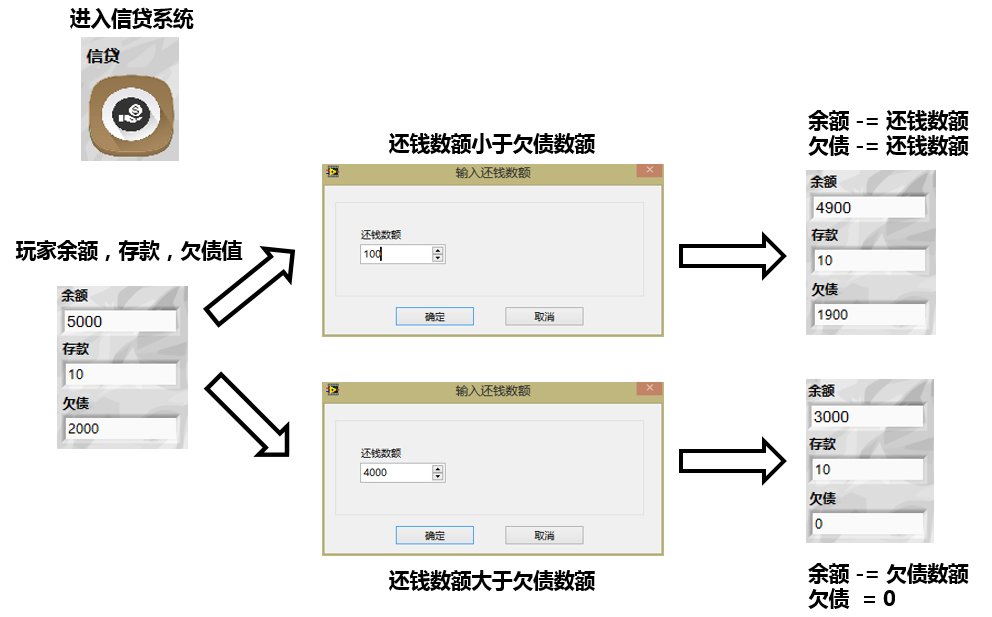
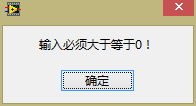
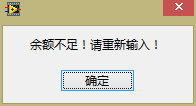


图11 信贷系统操作示意图

(a)输入数额为负数 (b)输入数额大于余额

图12 信贷系统错误输入触发提示

**（7）特殊地点：医院**

在游戏过程中，玩家的健康值有一定几率减少，而若健康值不足100，不仅会造成**卖出物品价格以一定幅度减少**，同时也会在最终**总分计算中扣去一定分数**，更有可能在健康值小于0时因**死亡而结束游戏**。医院系统可恢复玩家健康值，但会扣除健康值恢复大小\*100的余额，因此玩家需考虑什么时候是前往医院的较优选择。关于健康值与医院系统的示意图见图13，而前往医院的错误操作包括玩家在健康值已满时进入医院，以及余额不足以支付医疗费，此处不再展现错误输入所触发的提示。

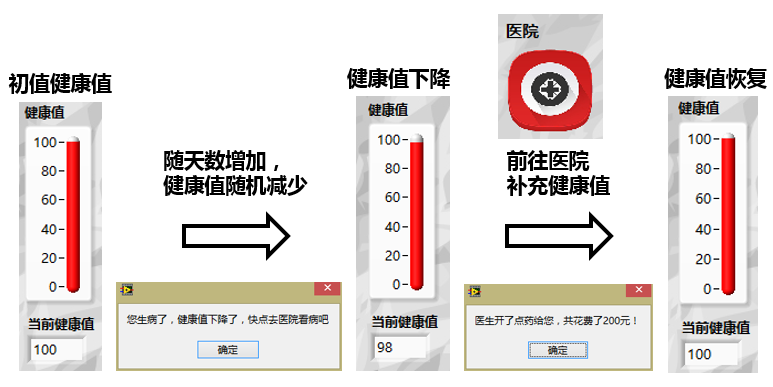


图13 健康值与医院

**（8）查看富豪榜**

在游戏过程中，玩家可随时选择查看富豪榜。富豪榜为此游戏在历史上分数最高的一众玩家，其中不乏有马云、马化腾、我校校长李清泉等一众杰出人才。他们绝非泛泛之辈，能达到如今层次，无不受天降大任，苦其心志，劳其筋骨，肝其体肤。若在打开富豪榜后，希望关闭富豪榜显示，则可点击“关闭富豪榜”按钮，则富豪榜与“关闭富豪榜”按钮将同时不再显示，如图14所示。



(a)点击查看富豪榜（红框部分），则可显示富豪榜



(b)右侧红框即富豪榜，点击“关闭富豪榜”则可回到(a)界面

图14 富豪榜查看与关闭

**（9）小提示**

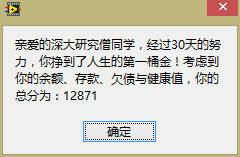
提示功能位于游戏界面的最下方，其显示内容每秒都会更新，用于向玩家介绍游戏的一些运行规则，同时对游戏策略给出一些建议。表1给出了部分提示信息，玩家在游戏过程中可细心留意一下其中的内容哦！

表1 提示信息

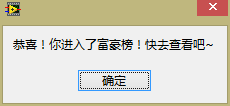
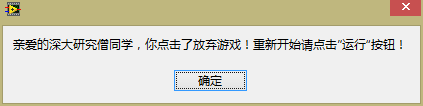
|  |  |
| --- | --- |
| 信息编号 | 提示信息内容 |
| 0 | 库存中买入价格为平均值 |
| 1 | 医院恢复1点健康值消耗100元！ |
| 2 | 风险越大！回报越大 |
| 3 | 每天只能去一个交易地点噢！ |
| 4 | 欠债越早还清越好噢！ |
| 5 | 健康值对利润也会造成影响！ |
| 6 | 可以把钱存入银行收取利息！ |
| 7 | 抓住随机事件的机会可以改变人生！ |
| 8 | 鸡蛋不要放在同一个篮子里！ |

**（10）游戏结束**

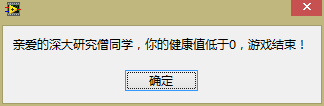
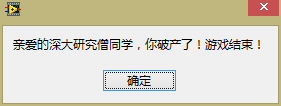
在本游戏中，存在三种情况造成游戏结束：1）30天期限已至，此时将弹出对话框显示玩家总分，而若总分足够高则将将提示玩家已进入富豪榜；2）玩家点击“放弃游戏”按键，将弹出放弃游戏对话框；3）由于健康值低于零，玩家死亡，游戏结束，此时将弹出对话框显示玩家健康值过低。在游戏结算时，若玩家总分低于零，则判定玩家破产，同样将给出对话框提示。关于游戏结束的各种情况见图15所示：



(a)总分展示

(b)玩家刷榜 (c)放弃游戏

(d)健康值过低致游戏结束 (e)总分过低致破产

图15 游戏结束画面

以上便是【深大漂流记】的游戏使用说明，祝您游戏愉快！

# 3.总体设计

## 3.1 需求概述

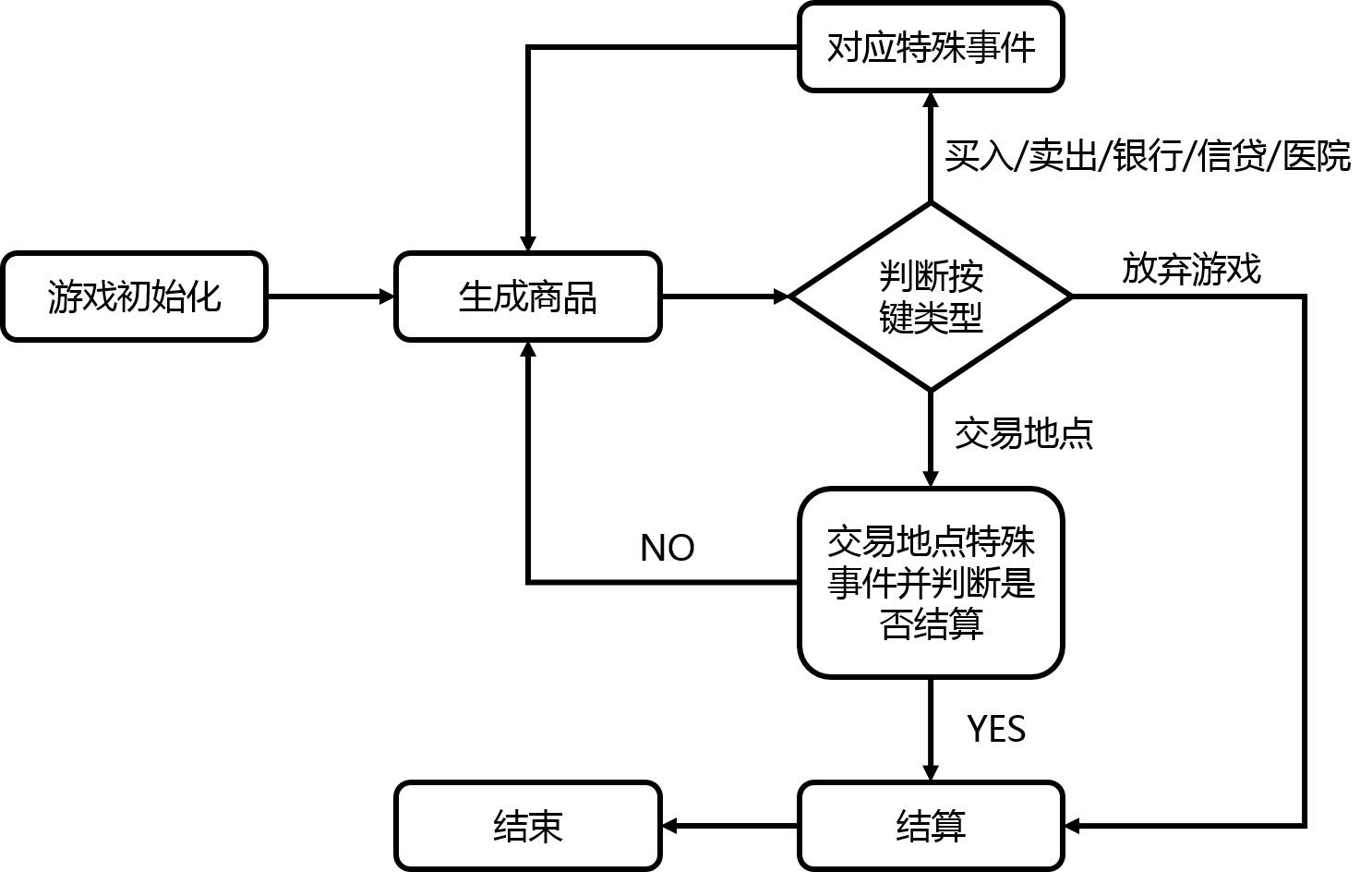
根据设计需要，本游戏应满足如下的需要：

1. 能实现对指定商品的买入与卖出，且若市面上无指定商品，则无法卖出；
2. 点击交易地点能进入新的一天，30天后游戏结束，且游戏过程中玩家可随时放弃游戏进入结算；
3. 每一天商品的价格与种类都会发生变化，并且随着时间的推移，整体物价上涨；
4. 每天有一定概率产生随机事件（价格翻倍与价格减半），且每一天有一定概率生病随机降低健康值；
5. 为提高用户友好度，库存内显示买入商品的平均买入价格；
6. 银行能够存款与取款，并且存款每天可以产生利息收益；
7. 信贷能够偿还借款，同时若未能还清欠债，欠债的利息会越来愈多；
8. 医院能够恢复健康值，同时会收取一定的医药费。若健康值为满，则不扣钱；
9. 健康值将影响卖出价格，健康值越低卖出所得利润越少，若健康值低于零则死亡；
10. 提示框给予玩家游戏提示；
11. 游戏过程中能随时查看/关闭富豪榜，富豪榜显示总分前十的玩家；
12. 有四种结局、死亡、破产、未破产但未上富豪榜、上富豪榜；
13. 游戏结束后，判断玩家得分是否上榜，若是则记录玩家姓名与得分，并提示玩家刷榜成功；
14. 能有效检测出违规输入并提示用户。

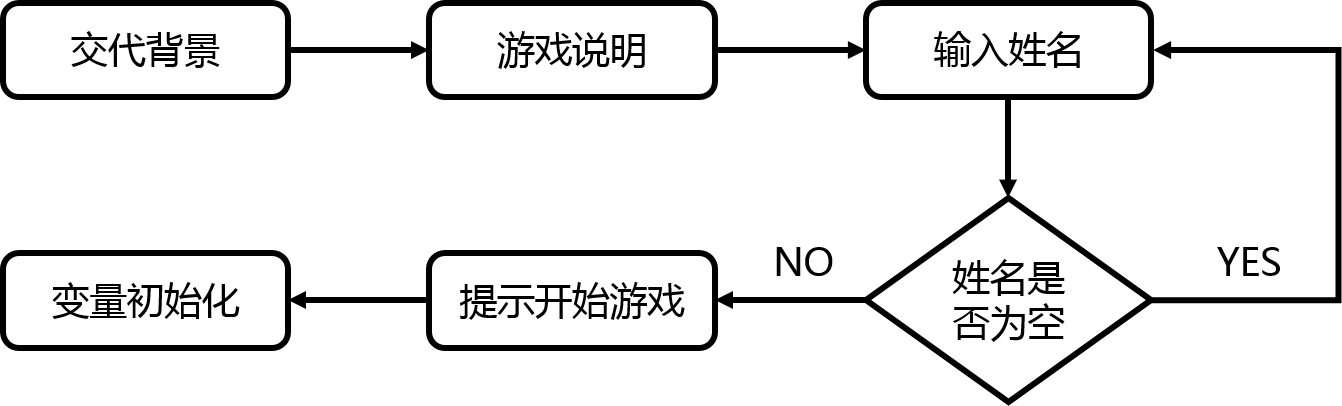
## 3.2 程序结构

本小节共包含十一张程序框图，其中第一张为总体游戏框图，剩余十张为总体游戏框图中的各子模块的具体框图。

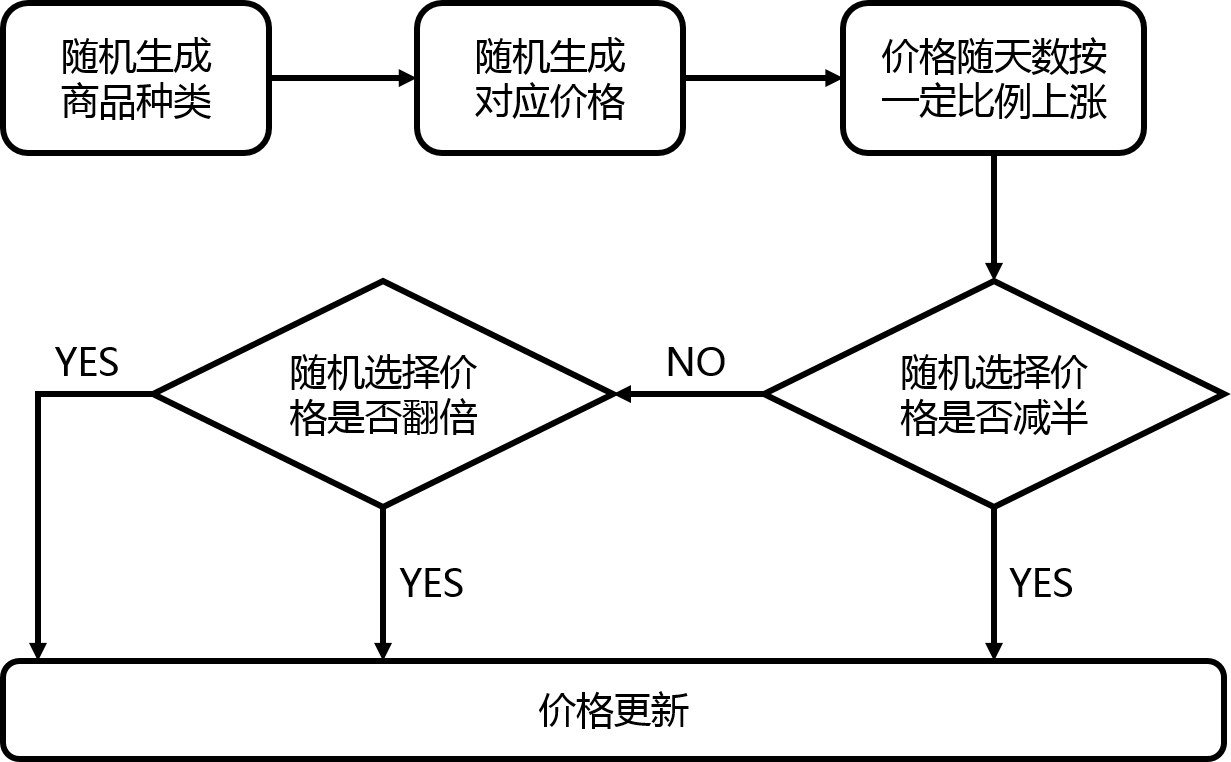
（1）**总体游戏框图**



**（2）游戏初始化**



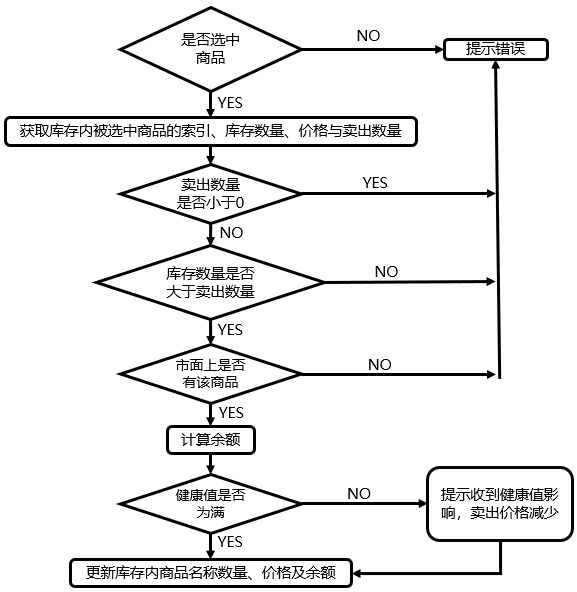
**（3）生成商品**



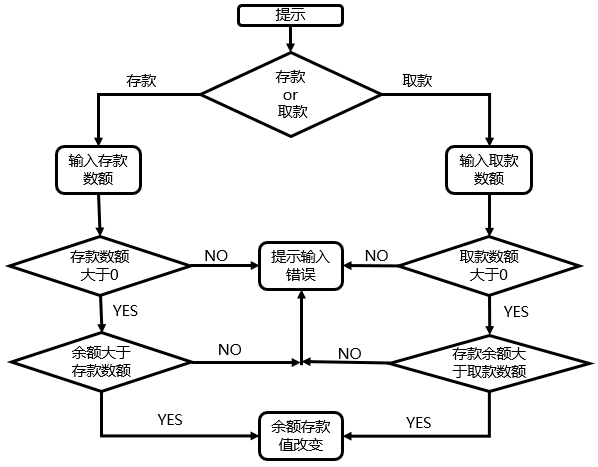
**（4）物品买入**



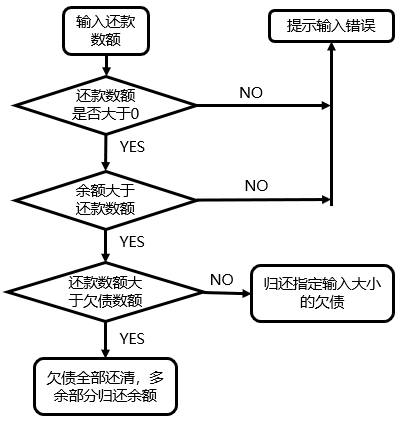
**（5）物品卖出**



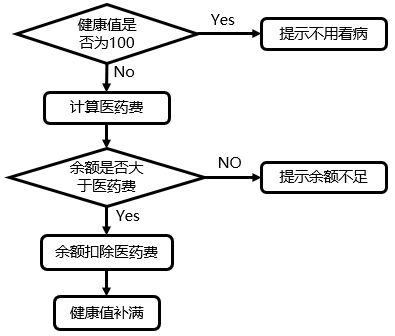
**（6）银行**



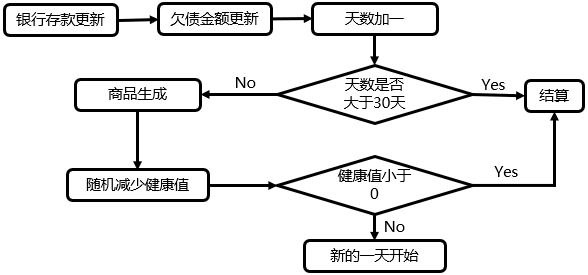
**（7）信贷**



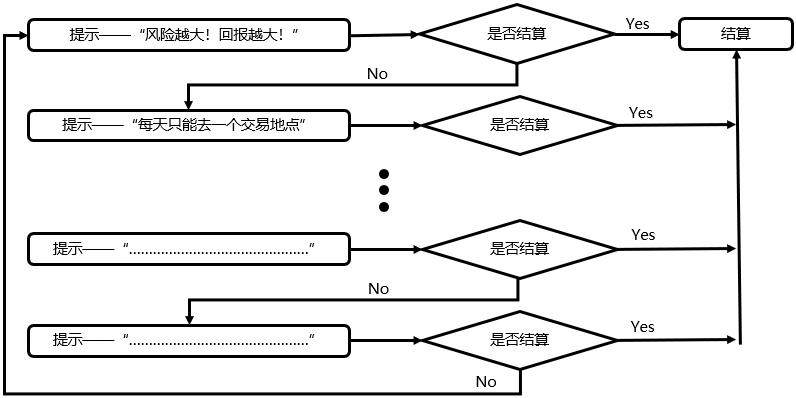
**（8）医院**



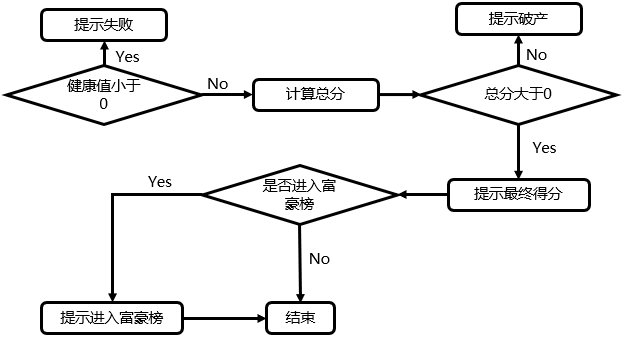
**（9）交易地点**



**（10）提示**



**（11）结算**



# 4．程序描述

为保证游戏按目标顺序依次进行，在程序设计时将最外层设置为3层顺序结构，如图16所示。顺序结构的第一层为**游戏准备**，其功能包含游戏介绍、玩家姓名输入、富豪排行榜查看等；顺序结构第二层为**游戏进行**，即进行为期30天内的物品交易，试图赚取尽可能多的钱财；顺序结构第三层为**游戏结束**，此时程序需要计算当前玩家的最终健康、余额等情况，并判断该玩家的余额是否进入富豪榜，从而进一步判断是否需要将当前游戏信息写入存储富豪榜的excel中。

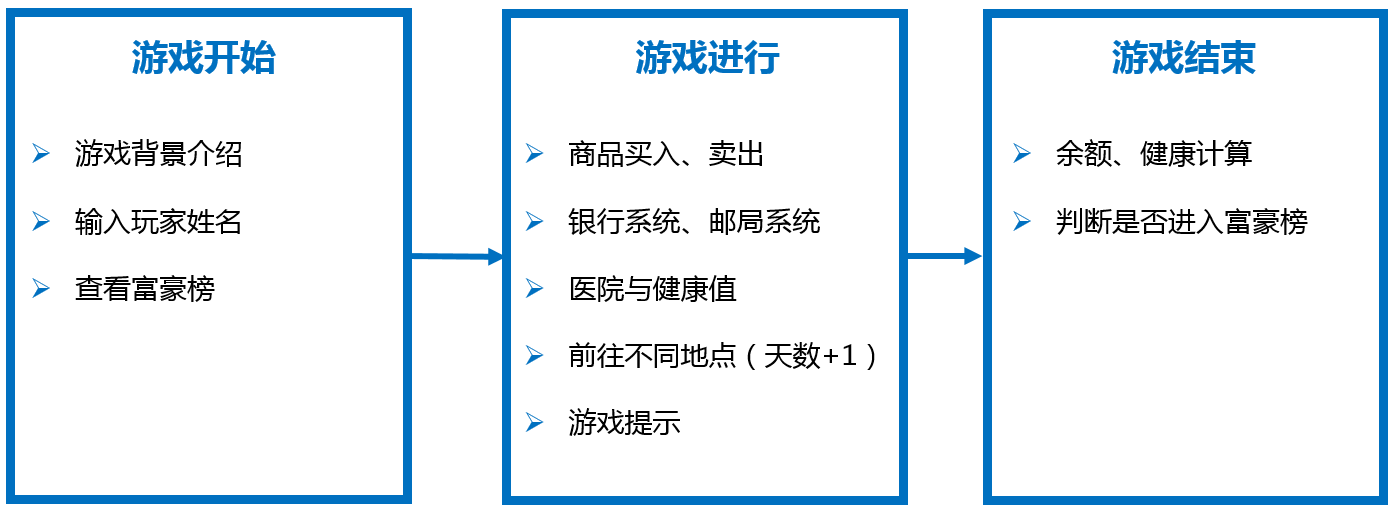


图16 游戏整体顺序结构

## 4.1 游戏开始

游戏开始阶段将按顺序进行必要的背景说明，并且需要玩家输入姓名，其后面板程序如图17所示。其中，图17下面的将关闭富豪榜按键与富豪榜设为隐藏，是为了对两者可见性设置初值，使玩家仅在按下“查看富豪榜”按钮时，这两个按键才会显示。此外，关于游戏背景与身份检查的提示用户说明控件内容则见图18所示。

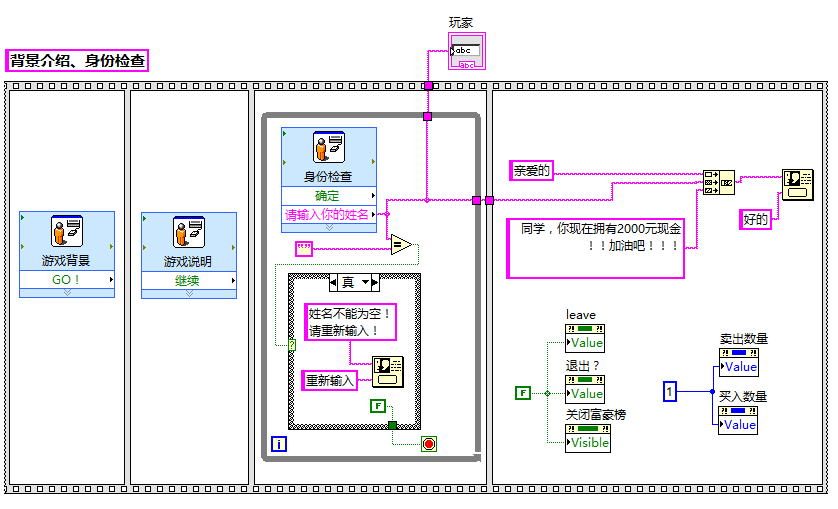
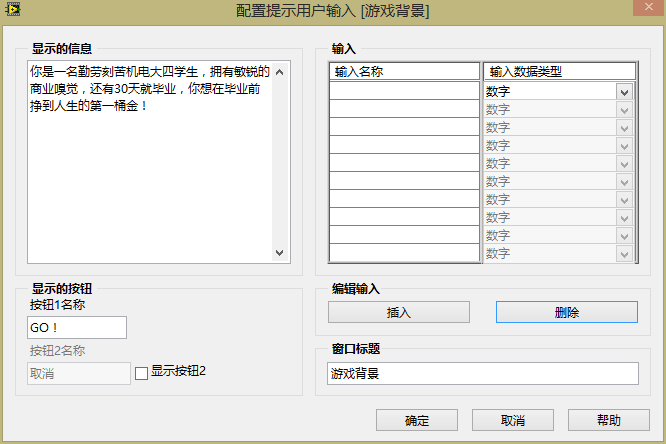
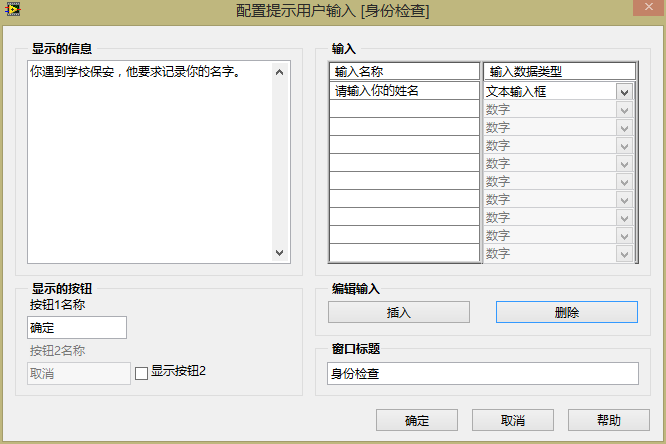


图17 游戏开始执行命令

(a)游戏背景控件 (b)身份检查控件

图18 提示用户输入控件内容

## 4.2 游戏进行

由第二章已知，本程序的游戏规则为30天的商品交易，故在游戏进行程序中，使用while结构不断进行循环，而当游戏天数到达30时，程序将终止该while循环，如图19所示。在游戏进行时，需要在while循环中添加移位寄存器记录游戏状态，这些状态包括：**余额、银行存款、欠债、健康值、游戏天数以及仓库存储物品**。上述寄存器的作用与初始状态见表2。

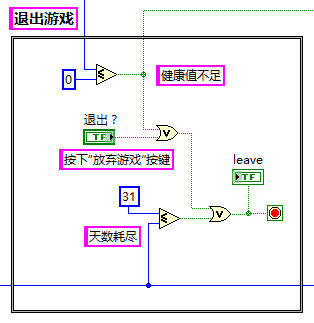


图19 到指定日期时退出游戏

表2 游戏进行时移位寄存器储存状态

|  |  |  |
| --- | --- | --- |
| 移位寄存器 | 初值特定 | 说明 |
| 余额 | 2000 | 创业不易，玩家的初始金额相对较少，需要通过不断的商品买卖增加余额 |
| 银行存款 | 10 | 放入银行的钱每天都会以固定的倍率增加 |
| 欠债 | 2000 | 在此次买卖之行之前，你已欠下高额的高利贷，为了防止欠债的累加，快尽快还完高利贷吧！ |
| 健康值 | 100 | 每天你都会有一定几率生病，生病会影响到你卖出物品时的价格与最终的余额分数，所以需要取医院治疗哈！ |
| 天数 | 1 | 每前往一个地点，游戏就会增加一天，30天游戏就结束了！度日如金！抓紧赚钱！ |
| 仓库存储物品 | 0 | 仓库存储物品包括你所买入的物品名称、物品价格以及买入均价，要抓紧商机及时将商品买入和卖出！！！ |

游戏进行时，玩家可选策略众多，因此本程序会包含需要众多按键，关于按键的名称及说明如表3所示，而所有按键都放置于程序最外部，如图20所示。而由于希望按钮控件上显示想要的图像，因此选择自制按钮控件，这些按键形状如表4与表5所示。针对按下的不同按键，程序会通过事件结构执行相应指令。本文在接下来将介绍物品栏的生成原理，以及每个事件结构处理相应任务的程序框图。

表3 程序前面板按键介绍

|  |  |  |  |
| --- | --- | --- | --- |
| 按键 | 按键说明 | 按键 | 按键说明 |
| 买入 | 买入市面中的选定商品 | 卖出 | 卖出库存中市面存在的选定商品 |
| 银行 | 前往银行系统存取款 | 信贷中心 | 前往信贷系统偿还高利贷 |
| 交易地点 | 前往指定地点买卖物品，天数+1 | 医院 | 恢复健康值 |

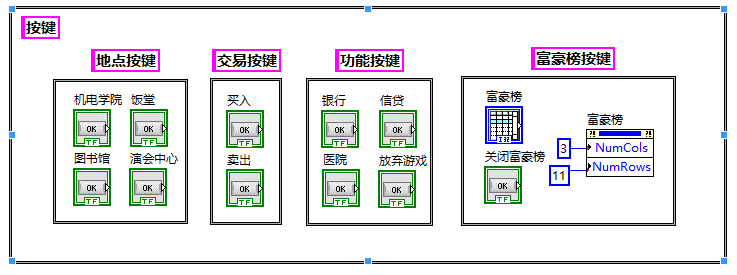


图20 后面板所有按键

表4 交易地点按钮控件设计（真假状态差异性体现在按键高度上）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地点 | 饭堂.ctl | 图书馆.ctl | 演会中心.ctl | 机电学院.ctl |
| True状态 | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\eat1.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\book1.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\mus1.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\bot1.png |
| False状态 | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\eat2.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\book2.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\mus2.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\bot2.png |

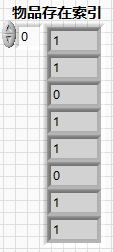
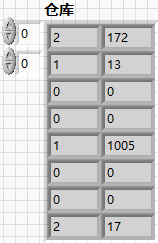
表5 功能按钮控件设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能 | 银行.ctl | 信贷.ctl | 医院.ctl | 放弃游戏.ctl |
| True状态 | bank1 | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\dk1.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\yy1.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\leave1.png |
| False状态 | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\bank2.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\dk2.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\yy2.png | C:\Users\Jenkins\Desktop\控件相关\控件UI及背景\leave2.png |

**（1）物品栏生成**

物品栏的生成是本程序**设计难度最大，制作时间最长的部分**，其中涉及到仓库物品存储数组（未显示），市面物品价格随机生成（未显示）、市面物品存在随机生成（未显示）、市面物品存在索引（未显示）、市面多列表框物品展示（显示）、市面多列表框鼠标选择位置索引（未显示）、库存多列表框物品展示（显示）、库存多列表框鼠标选择位置索引（未显示）。可以说，虽然前面板仅显示出随机生成的市面商品和库存物品，但为保证游戏按目标逻辑进行，在其背后需要进行大量设置。由于上述的部分变量与鼠标事件紧密相关，故这些内容会在后面的触发事件中说明，而在此不再展开。

在物品栏生成中，仓库物品数组、市面物品展示数组、市面物品存在索引数组与库存展示数组是非常重要的四个数组，它们将涉及到物品生成、前面板物品展示、买入与卖出等各个功能，它们的结构如图21所示，其中(a)为仓库物品数组，用于存放所有物品的买入数量与买入均价（关于数量与均价的运算见后文事件1：买入与事件2：卖出），其中包含数量为0（未买入）的物品；市面物品(b)则在每天随机生成，每个物品都会以一定概率出现，其价格也会围绕着相应的均值随机波动，对于未出现的物品，市面物品多列表框将不会展示；市面存在索引(c)则记录哪些物品在市面中出现，出现时索引对应数值置1，否则置0；库存物品则将仓库物品中未买入的物品去除，并在第0列前插入物品名称用于前面板显示。在以上所有数组中，关于物品的索引排序已经固定，其排序顺序为：吉他、泡面、手机、ipad、笔记本、学士服、劳力士、老鼠药。

(a)市面物品 (b)市面存在索引 (c)仓库物品 (d)库存物品展示

图21 物品存储关键数组

下面将介绍商品的生成原理。当天数改变时，程序将运行子VI（商品生成.vi），如图22所示，子VI输出为市面商品数组和物品存在索引数组。子VI内部结构如图23所示，其中包括随机生成物品、随机生成价格、特殊事件、商品显示共4个模块。

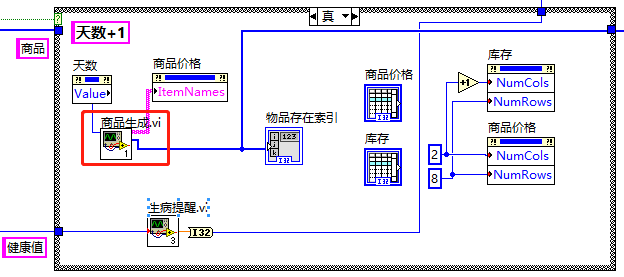


图22 天数改变后运行商品生成.vi（红框部分）

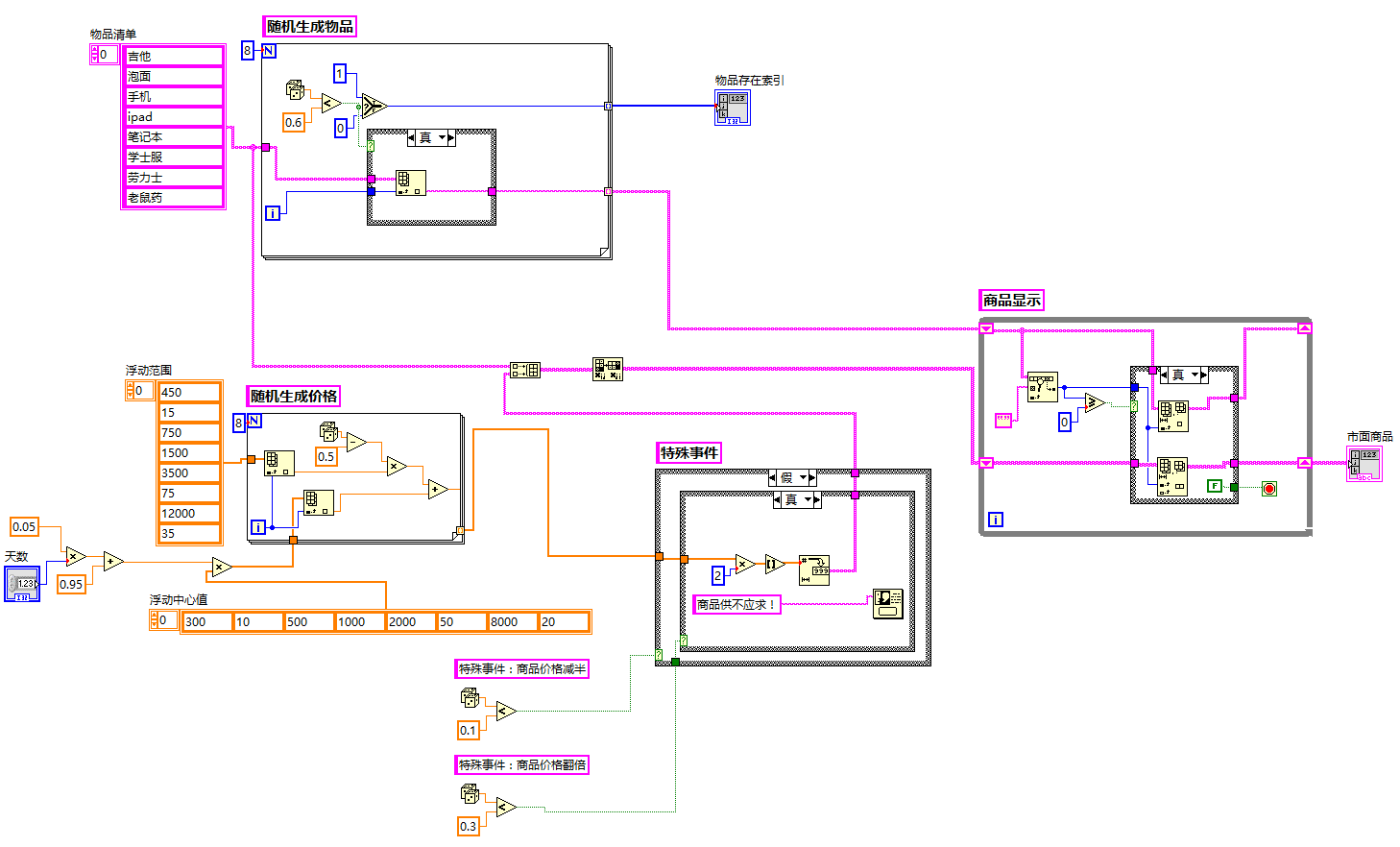


图23 商品生成VI结构

在商品生成.vi中，随机生成物品模块将以一定概率决定市面是否存在该物品，通过8次循环生成0/1组成的物品存在索引数组，而物品存在索引数组也会作为子VI输出，方便总程序在物品交易时进行处理；在随机生成价格模块中，每个物品将会存在随天数变化的价值浮动中心值（计算公式：浮动中心 = 初始值\*((天数\*0.05)+0.95)），以设定的浮动范围生成价格；特殊事件模块包含两个特殊事件，分别是商品价格减半与商品价格翻倍，特殊事件发生时将会使用单按钮对话框提示用户（快抓紧机会买入卖出~）；商品显示模块则用于处理拼接后的市面物品，目标是删除存在索引数组中0元素对应的物品。

讲述完物品栏后，下面将介绍各鼠标按键按下所引起的事件：

**（2）事件0：超时**

超时事件是为处理当前循环时无鼠标按下的情况，其结构如图24所示。在设计程序时，设置超时事件是为了不断更新变量参数，如不设置该事件，以商品买入为例，商品价格的更新经常发生滞后，由此会造成余额与库存等变量的数据出错。

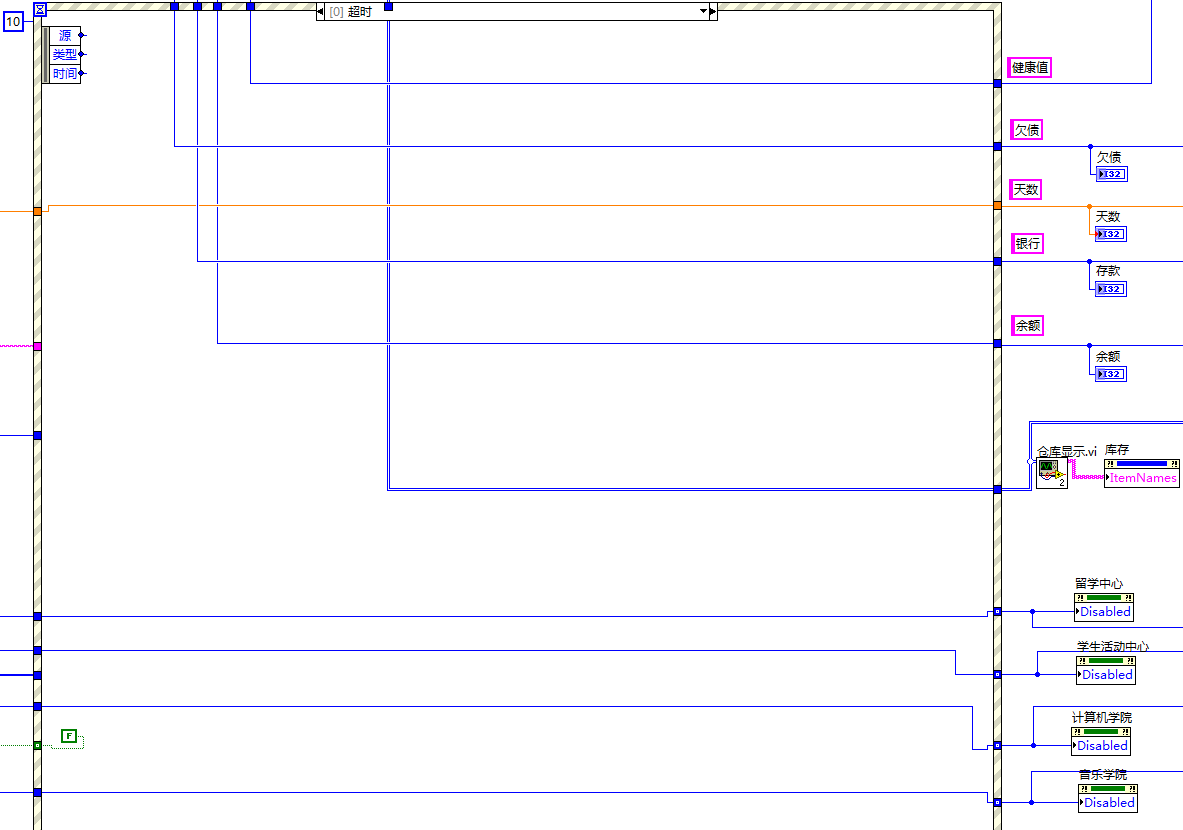


图24 超时事件结构（参数无变化传递）

**（3）事件1：“买入”鼠标按下**

当鼠标按下“买入”控件时，说明用户希望买入选中的市面商品，因此需将前面板的“买入数量”输入控件设定为非负整数，如图25所示，此外，在后面板中，也存在买入数量为负数或小数时的处理方法（“卖出数量”输入控件采用相同设定，因此后文不再介绍）。鼠标按下后，程序将首先读取前面板输入的买入数量以及索引商品列表框，确保上述数值保持最新，接着则将其传给顺序结构的下一帧，如图26所示。

图25 “买入数量”输入控件的数据设置

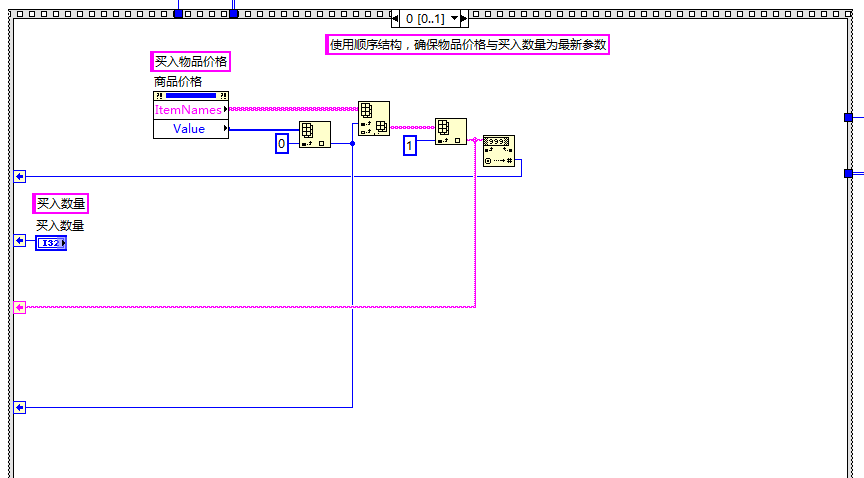


图26 “买入”按下后获取最新参数

而在下一帧，程序将根据买入信息调整变量数值。图27给出了该帧结构，在该事件中，需要首先判断手上的余额是否足够完成本次交易，若余额不足，将保持参数不变，并弹出对话框提醒用户余额不足。

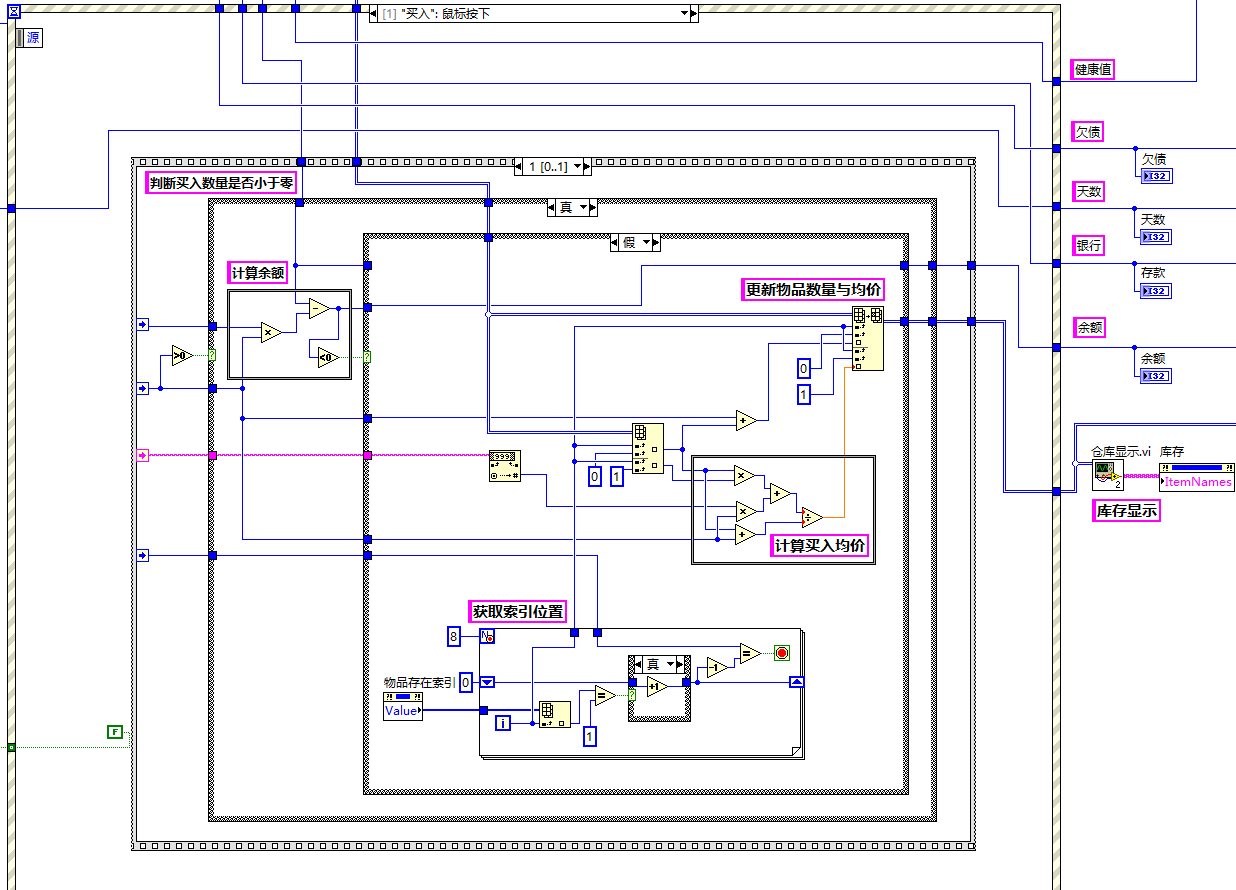


图27 “买入”程序结构

而当“买入”时余额足够时，需要首先获取买入的具体是什么物品。处理该问题的流程见图28，其难度在于需要将多列表框选择的索引位置转换为其具体对应的物品，关于该问题的具体实现可见图27的获取索引位置模块。拿到物品的索引位置后，就很容易更新库存物品的数量及买入均价了。

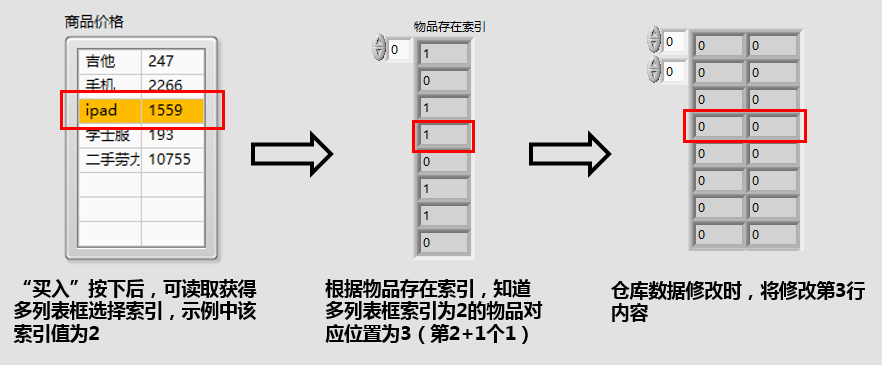
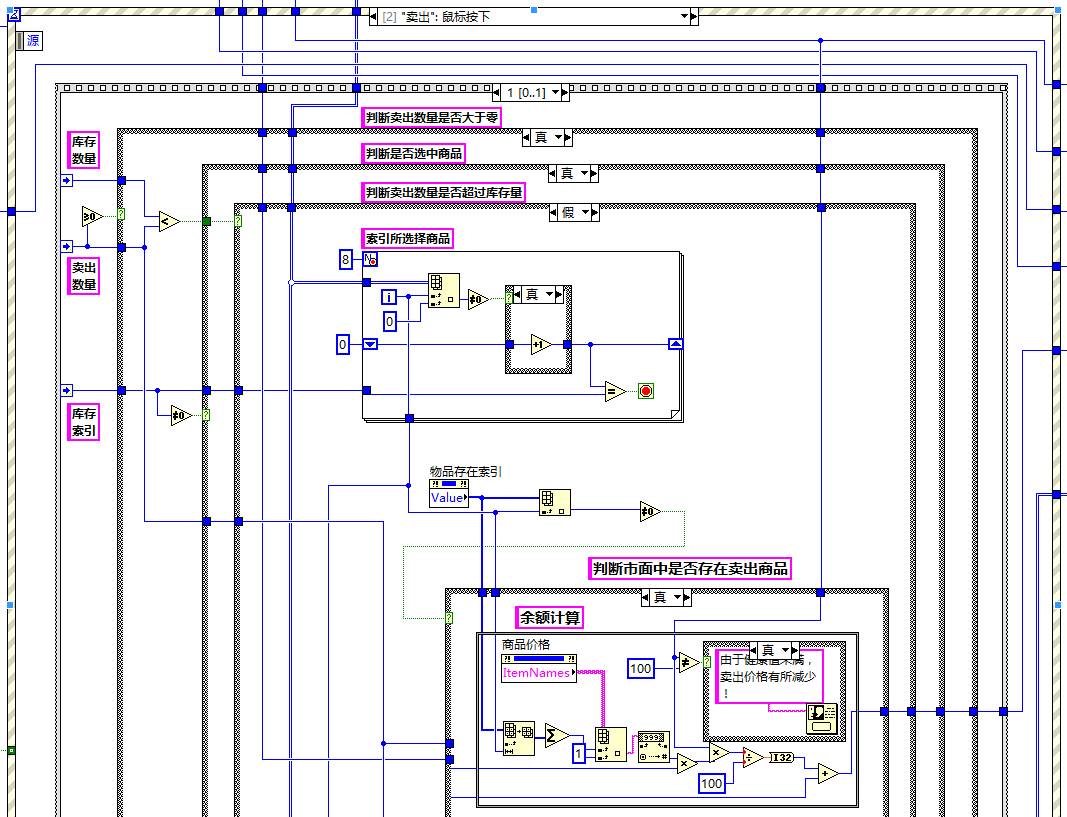


图28 索引买入物品位置

**（4）事件2：“卖出”鼠标按下**

该事件发生时，与商品买入类似，此时程序需要将商品卖出。由于商品卖出包含更多条件约束，故该事件具有更多的条件结构，如图29所示。与买入时相同，在商品卖出时需要获取最新市面物品及其价格，故需要使用顺序结构首先读取这些信息。接下来，则需要判断用户是否在库存多列表框中选中商品，若没有，则参数不变动，提示用户未选中商品，并退出该事件；若用户已选中商品，则需要判断选择的卖出数量是否超出库存量。



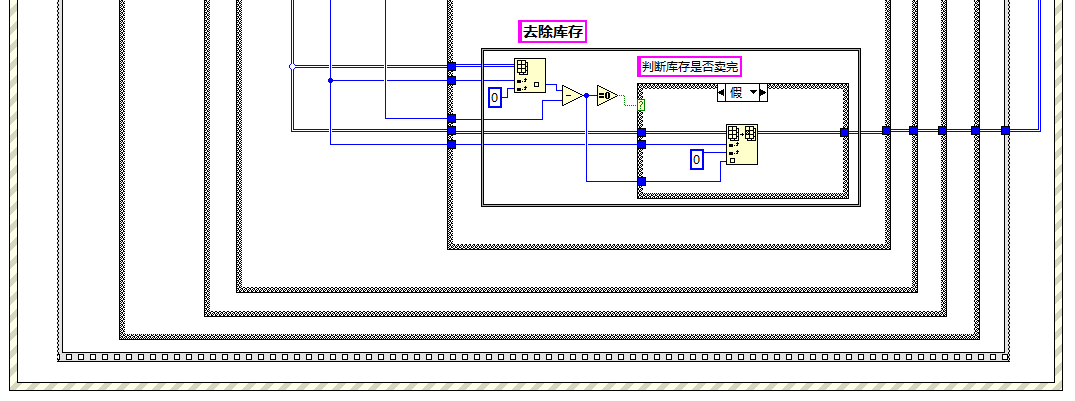
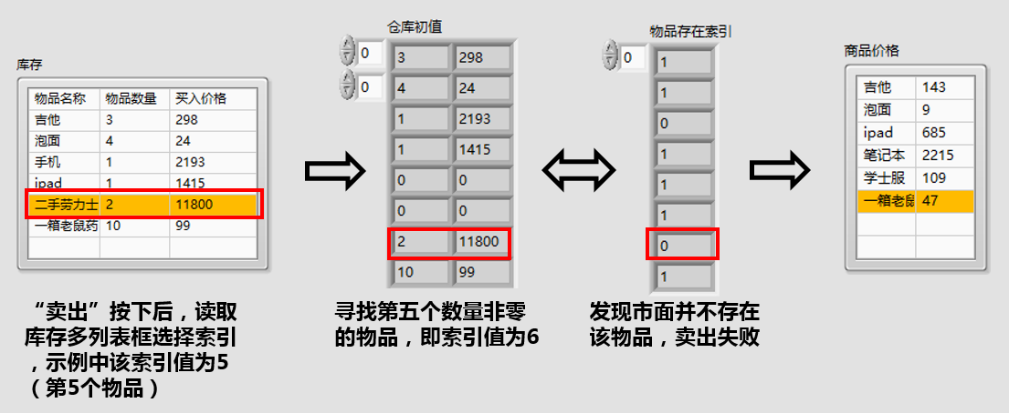
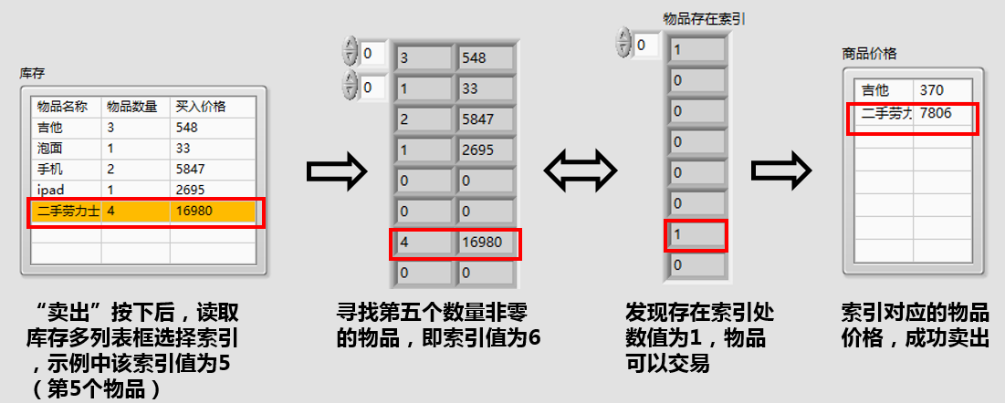


图29 商品卖出总结构图

与商品买入类似，在商品卖出时也需要将库存多列表框的选择商品与库存数组中的商品相对应，其索引过程如图30所示。若该物品在市面上不存在，则卖出失败，同时弹出窗口提醒用户；若市面上存在希望卖出的物品，则按市面价格\*健康值/100卖出该物品。此时，需要调整手上的余额数值与库存数组，若库存制定物品全部卖出，则将该行清零；若只卖出部分，则仅将该物品的库存数量-卖出数量即可。



(a)市面上不存在该物品，卖出失败



(b)市面上存在该物品，卖出成功

图30 卖出商品位置索引过程

**（5）事件3 ：地点鼠标释放（天数+1）**

当玩家点击某个地点后，将会使天数发生变化（+1）。在上述事件中，除天数加一，银行存款与高利贷数额按比例上升，其它移位寄存器参数正常传递。该事件中会生成一个真常量控制图31所示的条件结构发生。其它的控件则是显示在前面板的两个多列表框，即商品价格显示与库存显示。

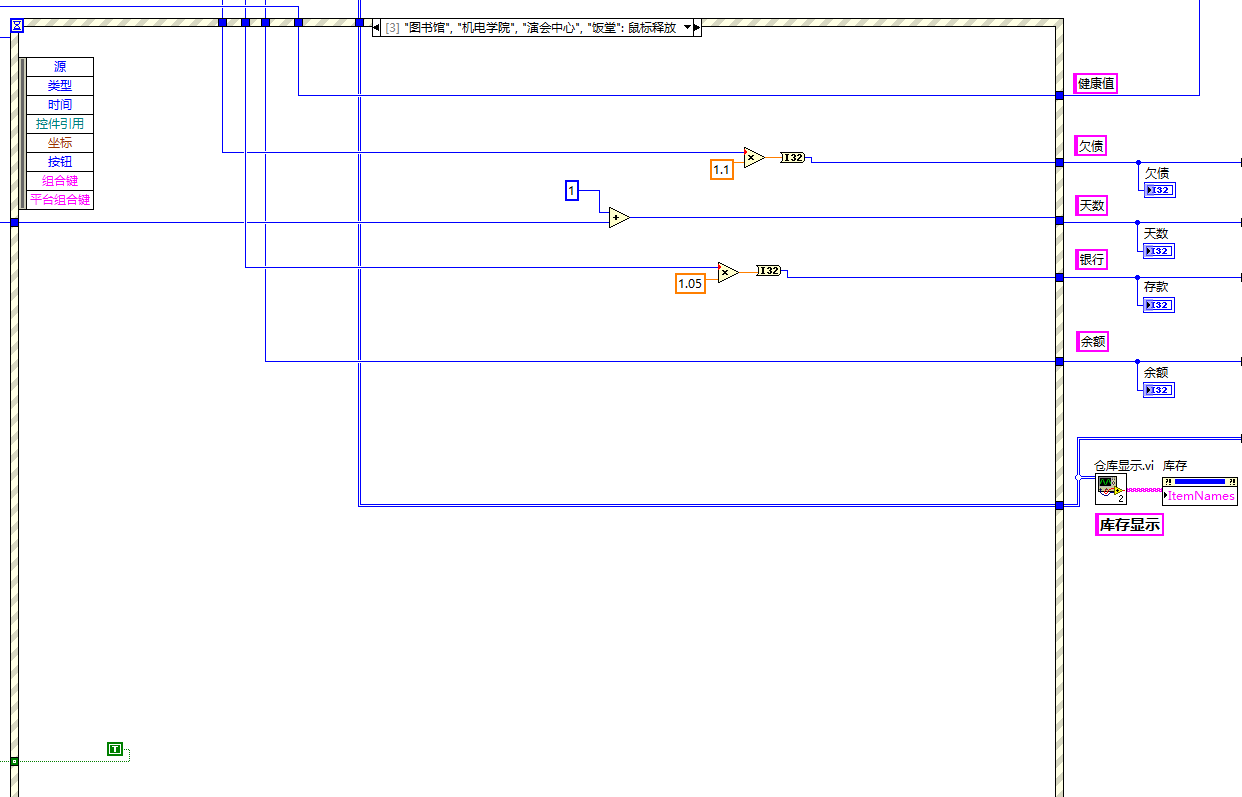


图31 地点鼠标释放后程序运行示例

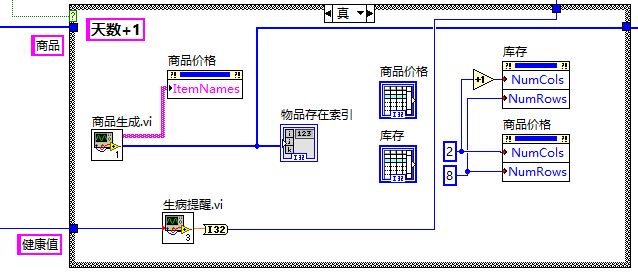
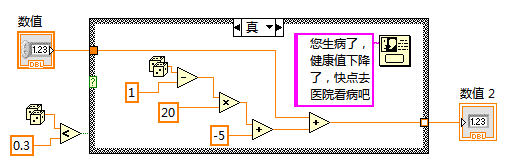
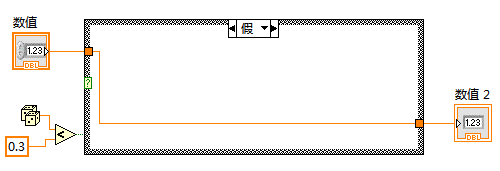


图32 天数+1执行操作

在图32所示的条件结构中，商品生成.vi已在上文介绍，而另一个子VI，即生病提醒.vi则用于随机减少玩家健康值，其VI内部结构图如图33所示。该VI会影响到健康值的变化，生病时，程序将随机减少一定健康值，而未生病时健康值讲直接传递。



(a)生病时执行框图

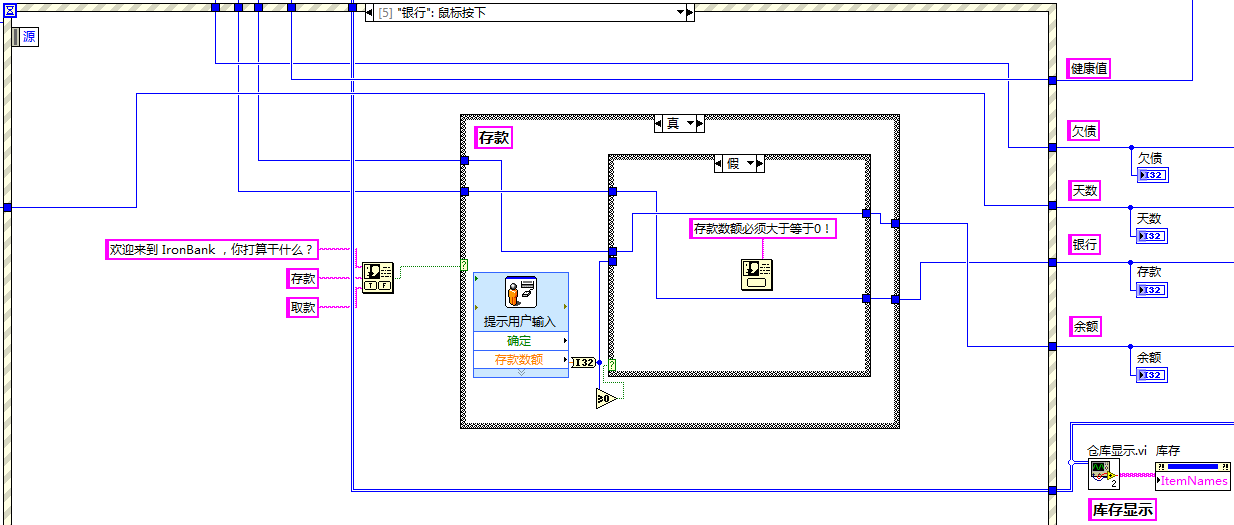


(b)未生病时执行框图

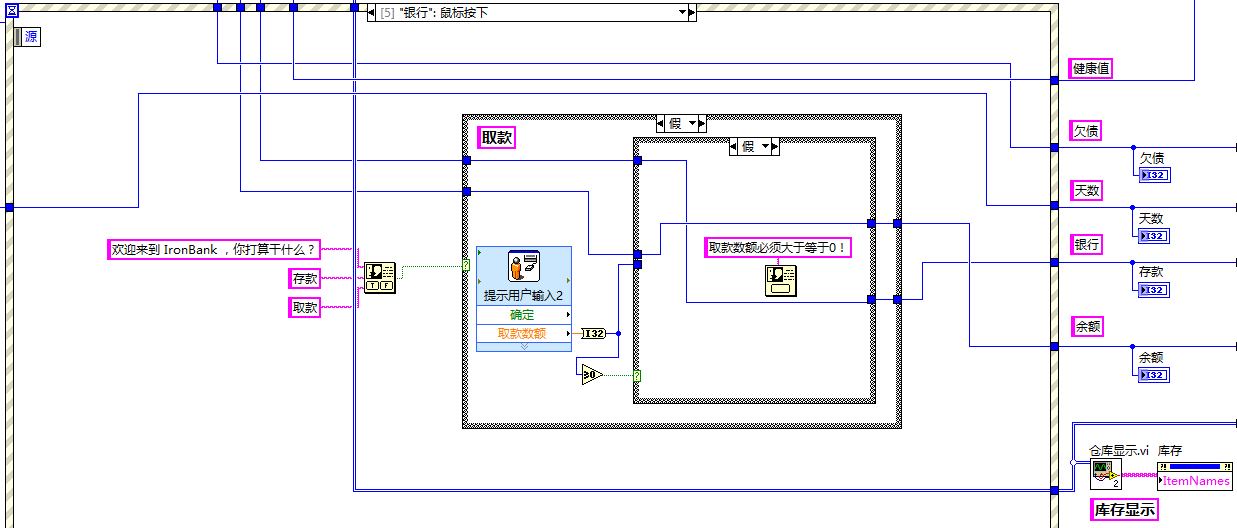
图33 生病提醒.vi结构图

**（6）事件5：“银行”鼠标按下**

触发该事件后，将进入银行系统，放入银行中的钱每过一天都会产生利息，因此建议玩家将每天未花完的钱放入银行中。银行系统涉及到余额与存款的互换，所涉及操作包括存款与取款，如图34所示。点击存款按键，需要减少余额，增加存款；点击取款按键，需要减小存款，增加余额。在该事件中，需要进行**判断的情况**包括：**用户选择存款还是取款、存取款额数是否大于0，以及余额或存款数额是否足够本次操作。**



(a)存款程序框图



(b)取款程序框图

图34 “银行”按下后存取款框图

**（7）事件6：“信贷”鼠标按下**

信贷系统的设置用于提供给用户偿还高利贷，其程序框图见图35。在“信贷”按键有鼠标按下后，将弹出对话框让用户选择还款数额，系统接下来将把欠债的钱数减去用户所选还款数。若用户输入还款数大于欠债数，系统会把欠债还清，并把多余的钱放回余额中。在信贷系统中，程序还需要判定**输入还款数是否大于零、用户余额是否大于还款数**。

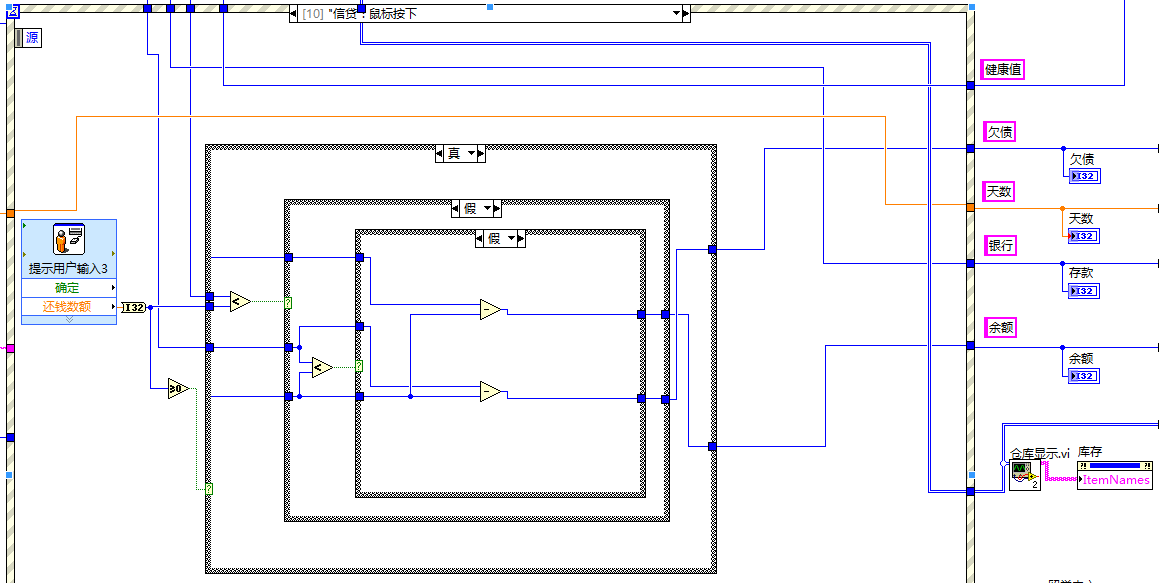


图35 “信贷”鼠标按下程序框图

**（8）事件7：“医院”鼠标按下**

医院系统用于恢复玩家的健康值，恢复健康值的花费价格为恢复健康点数\*100，其程序框图如图36所示。在医院系统中需要判断的情况为：**在健康值已满时点击医院、所剩余额是否足够支付本次花费**。

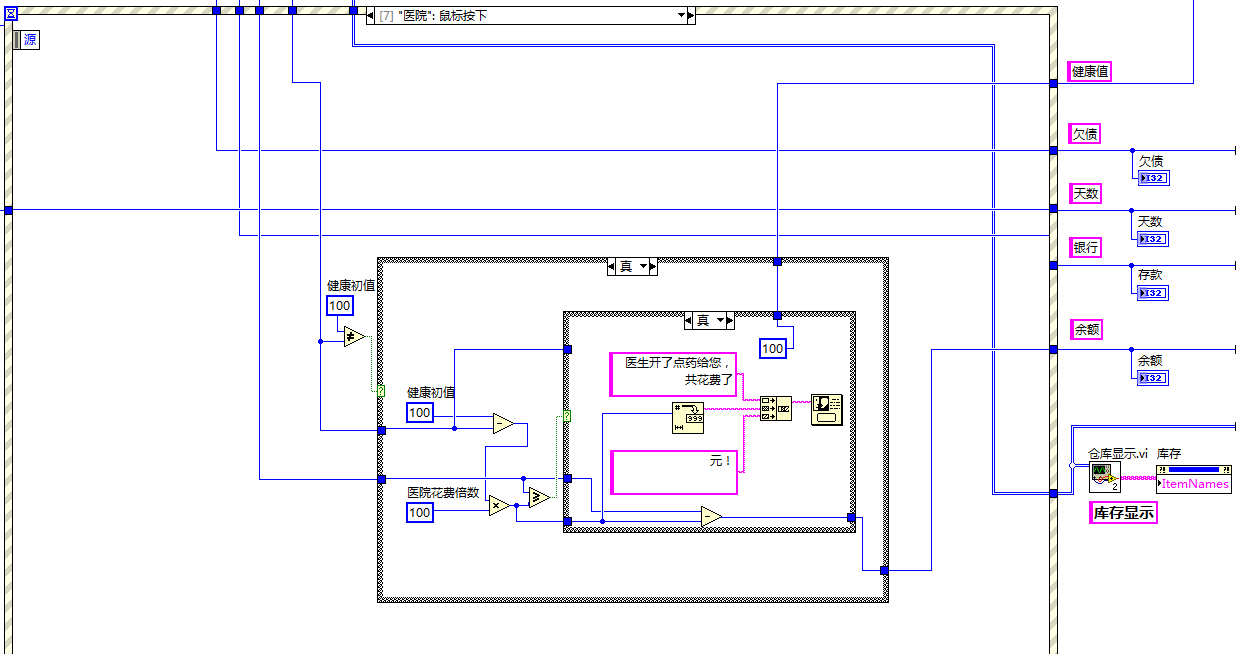


图36 “医院”鼠标按下程序框图

**（9）小提示**

在程序前面板存在一行提示显示控件，该控件将每隔一秒向用户提示一个消息，提醒玩家一些游戏的运行规则与建议，其程序框图如图37所示。在该程序框图中使用离开游戏键的真值作为判断结构的输入，是为了快速在游戏结束后停止提示循环。



图37 小提示循环框图

**（10）富豪榜查看**

在未点击“查看富豪榜”按钮时，富豪榜与“关闭富豪榜”按钮处于隐藏状态。而单击“查看富豪榜”按钮后，程序将读取存储富豪榜的score.csv文件。需要注意的是，由于LabVIEW的文件读取需要使用**绝对路径**，因此需要使用“当前VI路径”功能读取主程序的路径，并将csv文件放置在与主程序同样的路径中。若在查看完富豪榜，点击“关闭富豪榜”后，该按键与富豪榜将隐藏不可见。富豪榜的程序框图见图38。

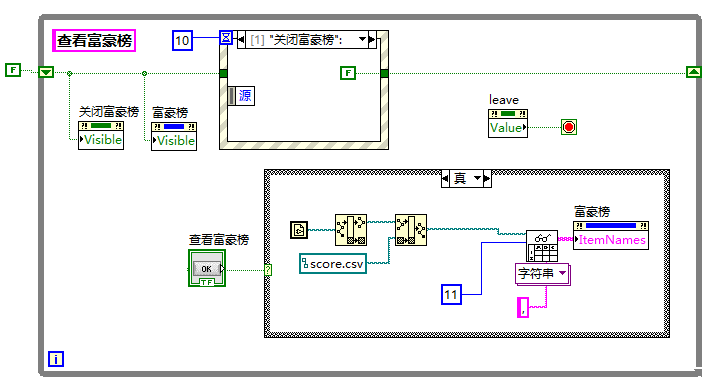


图38 富豪榜查看程序框图

**（11）特殊事件造成游戏结束**

在本程序中，造成游戏结束的特殊情况包括：天数达到设置值，用户点击“离开”按钮，以及健康值小于零，如图39所示。遇到上述情况后，系统将结束游戏进程，进入最后游戏结束的结算程序框图。

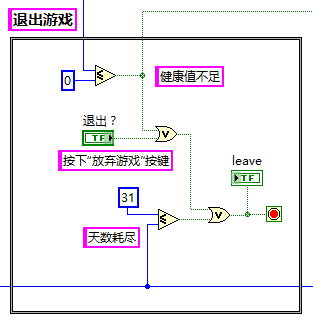


图39 造成游戏退出的特殊情况

## 4.3 游戏结束

在游戏结束后，程序将进入第三部分，即总分计算，并判断当前分数是否足以进入富豪榜，其程序框图如图40所示。在该程序框图中，程序首先将判断玩家是因健康值不足而死亡还是其他原因，对于前者，系统将弹出对话框告诉玩家游戏结束原因是健康值不足，见图41(a)，而对于后者，也会存在玩家欠债过多导致总分小于0的情况，此时游戏判定玩家破产，其结束提示见图41(b)。

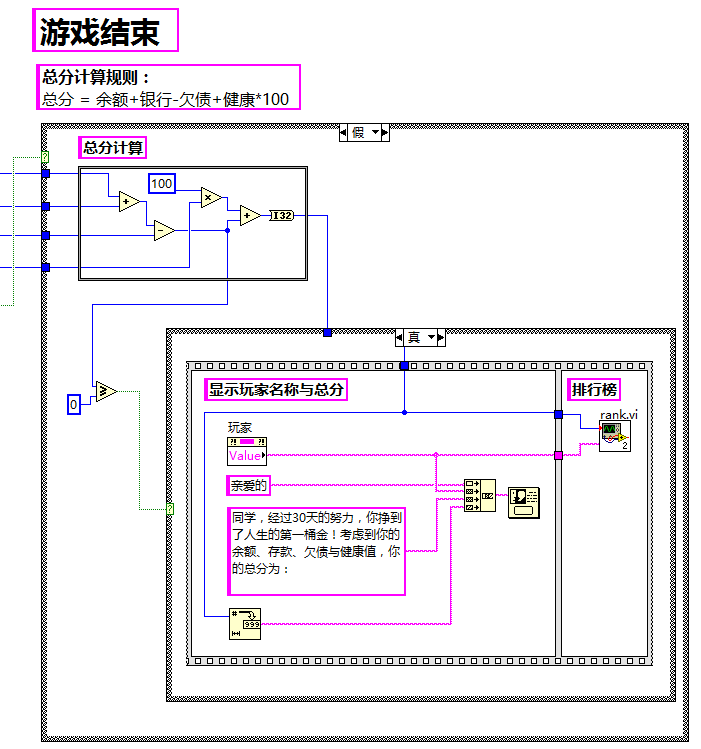
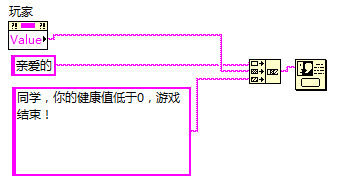
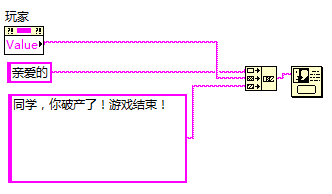
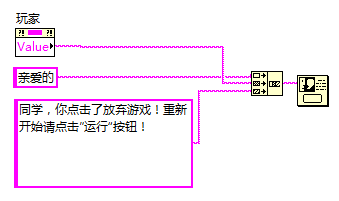


图40 游戏结束后程序框图

(a)健康值过低致游戏结束提示 (b)总分小于零的游戏结束提示



(c) 玩家按下“放弃游戏”按键

图41 特殊情况的游戏结束提示

而正常游戏结束提示则是图40所示的程序框图。该程序框图中涉及到判定玩家分数是否进入富豪榜的rank.vi，该子VI程序框图见图42。该子VI的程序框图主要流程为：1）score.csv文件数据读取；2）拆分富豪榜中玩家姓名与玩家得分；3）通过富豪榜从最高分开始，使用for循环判断该次玩家得分是否大于富豪榜得分；4）若在某个循环i中玩家得分大于富豪i的分数，则将本玩家姓名与得分插值至i处，同时停止for循环；5）重组富豪榜存储数组，并将数组数值写入score.csv文件。

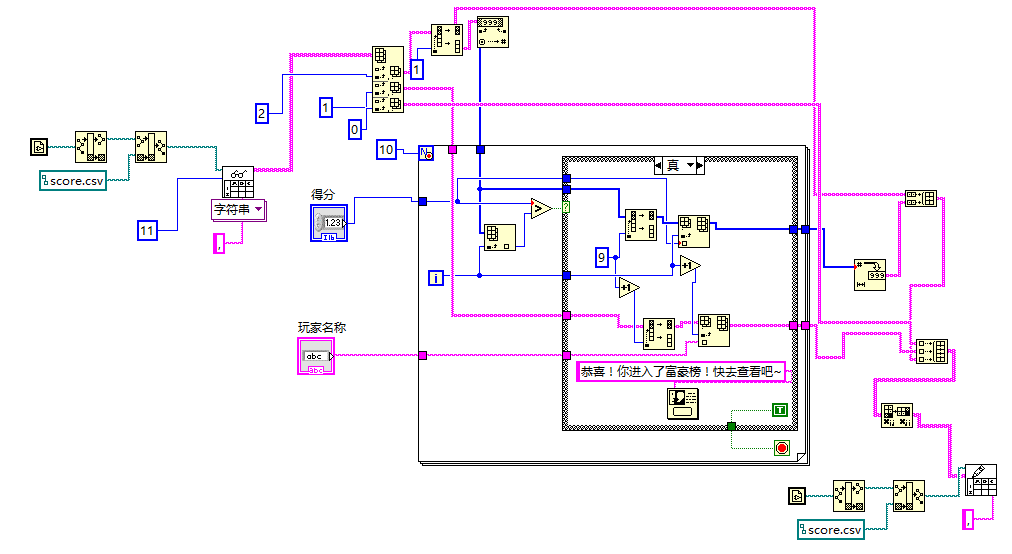


图42 rank.vi程序框图

以上便是深大漂流记游戏的后面板程序框图介绍，希望能帮助到您对程序的理解，同时愿您享受玩游戏的过程~！