

基于液晶显示屏的逻辑推理游戏



——死兔

心於至善

凌昕彤

lim3300@163.com

严谨 求实 团结 奋进



主要功能

玩家扮演一只陷入迷宫的兔子。开始仅能看到周围的9格方格。方格上标有数字，代表的是该方格直接相连的四格中可以通行的方格数量。玩家需要通过对数字的判断，找到通往“生”——即目的地——的路。而如果走到不能通行的格子，即“死”，则游戏结束。

在上述核心的基础上，添加了开机、过关、重玩和通关画面，增加了游戏的完整性。



游戏界面





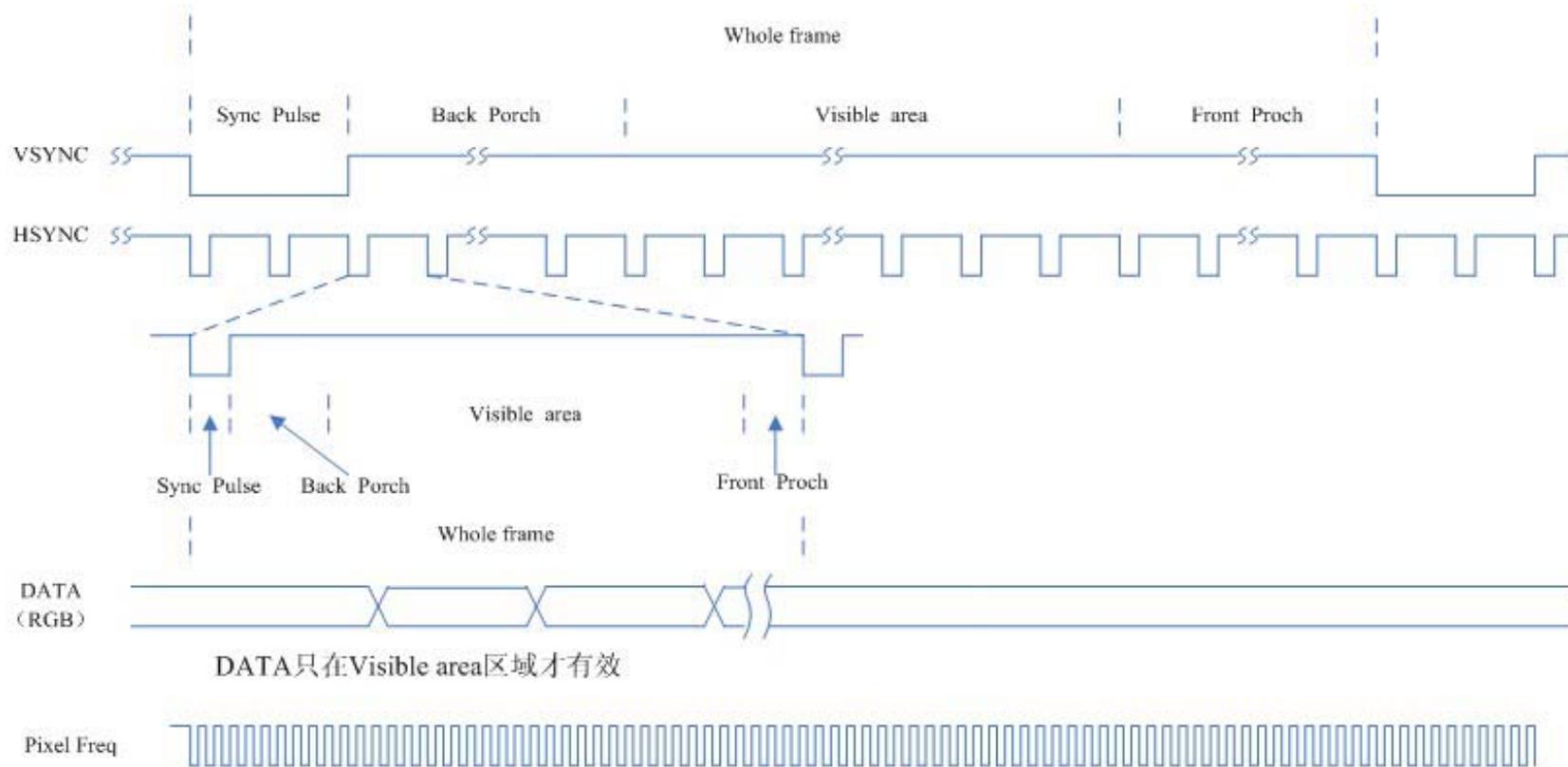
核心点：驱动液晶屏显示

优势：

- 1、接线方便，驱动VGA仅需8根钉子线
- 2、画面美观，相比与点阵显示的颜色、内容和精度都有优势



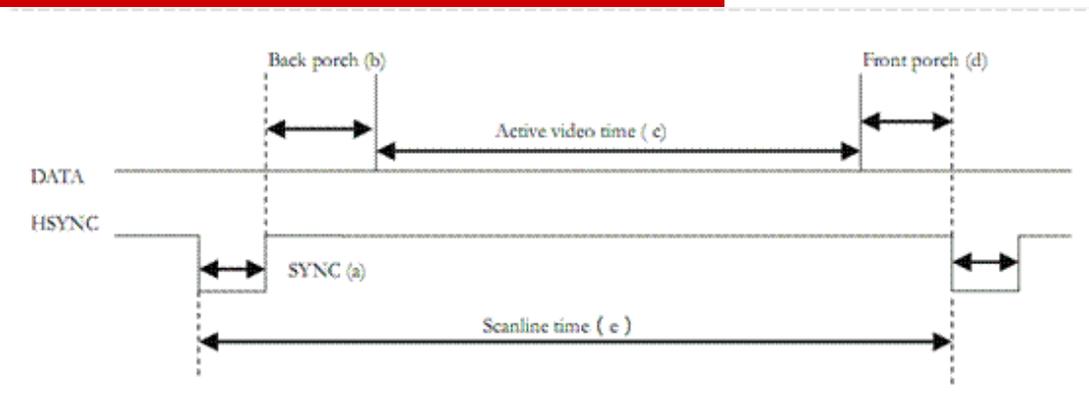
核心点：驱动液晶屏显示



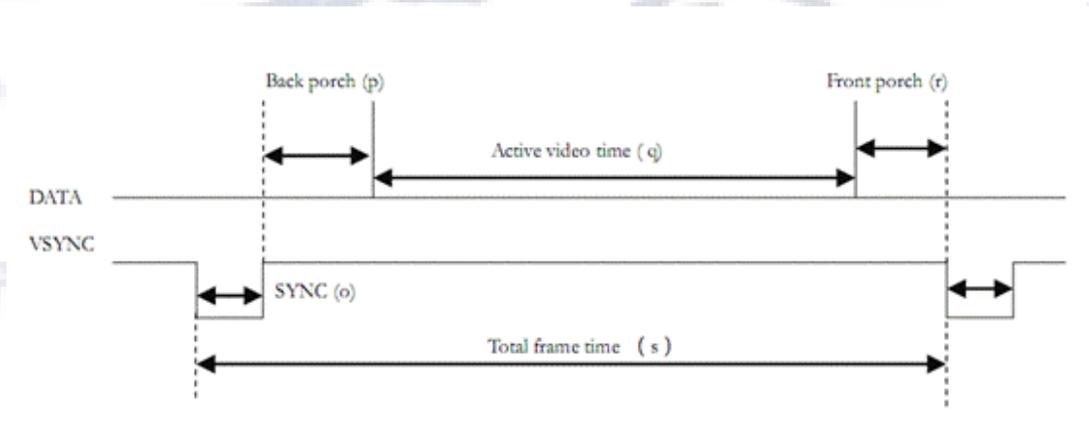
VGA的信号时序



核心点：驱动液晶屏显示



行信号时序



场信号时序



核心点：驱动液晶屏显示

800X600 X50HZ	a段 (同步段)	b段 (后肩 段)	c 段 (激活 段)	d 段 (前肩 段)	e 段总共 n个列像 素
HSYNC Signal 列像素	128	88	800	40	1056
800X600 X50HZ	o 段 (同步段)	p 段 (后肩 段)	q 段 (激活 段)	r 段 (前肩 段)	s段总共n 个行像素
VSYNC Signal 行像素	4	23	600	1	628



创新点1：以矩阵形式输出图文

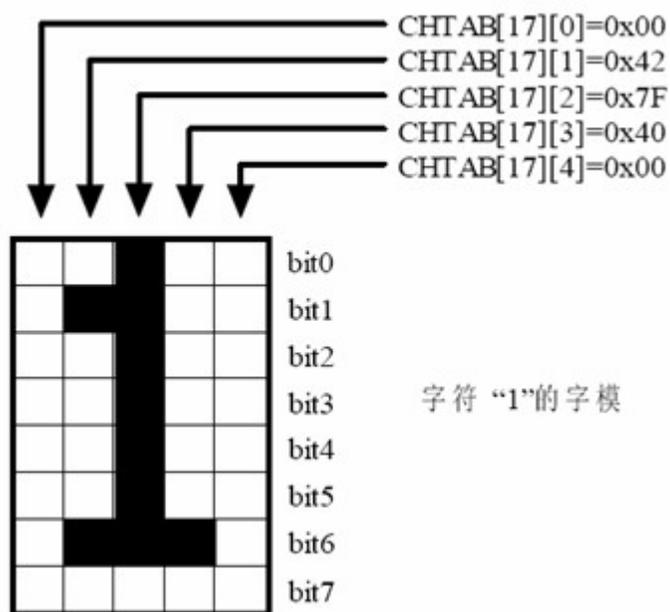
```
else if ((h_cnt>xx2-21&&h_cnt<xx2-14&&v_cnt>y-16&&v_cnt<y+36) //"\n"
|| (h_cnt>xx2-21&&h_cnt<xx2-18&&v_cnt>y-19&&v_cnt<y-14)
|| (h_cnt>xx2-20&&h_cnt<xx2-17&&v_cnt>y-21&&v_cnt<y-14)
|| (h_cnt>xx2-19&&h_cnt<xx2-16&&v_cnt>y-23&&v_cnt<y-14)
|| (h_cnt>xx2-18&&h_cnt<xx2-15&&v_cnt>y-25&&v_cnt<y-14)
|| (h_cnt>xx2-17&&h_cnt<xx2-14&&v_cnt>y-26&&v_cnt<y-14)
|| (h_cnt>xx2-16&&h_cnt<xx2-13&&v_cnt>y-28&&v_cnt<y-16)
|| (h_cnt>xx2-14&&h_cnt<xx2-11&&v_cnt>y-30&&v_cnt<y-18)
|| (h_cnt>xx2-13&&h_cnt<xx2-9&&v_cnt>y-31&&v_cnt<y-22)
|| (h_cnt>xx2-11&&h_cnt<xx2-7&&v_cnt>y-33&&v_cnt<y-25)
|| (h_cnt>xx2-9&&h_cnt<xx2-5&&v_cnt>y-34&&v_cnt<y-27)
|| (h_cnt>xx2-7&&h_cnt<xx2-1&&v_cnt>y-35&&v_cnt<y-28)
|| (h_cnt>xx2-3&&h_cnt<xx2+3&&v_cnt>y-36&&v_cnt<y-29)

|| (h_cnt>xx2+14&&h_cnt<xx2+21&&v_cnt>y-16&&v_cnt<y+36)
|| (h_cnt>xx2+18&&h_cnt<xx2+21&&v_cnt>y-19&&v_cnt<y-14)
|| (h_cnt>xx2+17&&h_cnt<xx2+20&&v_cnt>y-21&&v_cnt<y-14)
|| (h_cnt>xx2+16&&h_cnt<xx2+19&&v_cnt>y-23&&v_cnt<y-14)
|| (h_cnt>xx2+15&&h_cnt<xx2+18&&v_cnt>y-25&&v_cnt<y-14)
|| (h_cnt>xx2+14&&h_cnt<xx2+17&&v_cnt>y-26&&v_cnt<y-14)
|| (h_cnt>xx2+13&&h_cnt<xx2+16&&v_cnt>y-28&&v_cnt<y-16)
|| (h_cnt>xx2+11&&h_cnt<xx2+14&&v_cnt>y-30&&v_cnt<y-18)
|| (h_cnt>xx2+9&&h_cnt<xx2+13&&v_cnt>y-31&&v_cnt<y-22)
|| (h_cnt>xx2+7&&h_cnt<xx2+11&&v_cnt>y-33&&v_cnt<y-25)
|| (h_cnt>xx2+5&&h_cnt<xx2+9&&v_cnt>y-34&&v_cnt<y-27)
|| (h_cnt>xx2+1&&h_cnt<xx2+7&&v_cnt>y-35&&v_cnt<y-28))
rgb <= 3'b011;
.. .. .. .. ..
```

通过网络查找到的代码，均使用了if-else语句以及大量的与-或逻辑关系，来“绘制”图片和文字。相当于是通过一个一个矩形叠加构成图片和文字。对于规则的图形，似乎还可以编写理解；但是对于不规则的非直线的，工作量让人望而生畏。



创新点1：以矩阵形式输出图文



在本数字系统中，由于使用大量字母和数字，且局部显示均以静态为主，所以使用了矩阵形式来显示图文。

严谨 求实 团结 奋进



创新点1：以矩阵形式输出图文

步骤1：输入矩阵

通过Matlab的imread函数和C++的文件输入输出流，可以方便的得到字母库和bmp格式转换来的8色图片矩阵

```
271 chara[16][0]<=8'h3E;      chara[16][1]<=8'h41;      chara[16][2]<=8'h41;      chara[16][3]<=8'h49;      chara[16][4]<=8'h7A;
272 chara[17][0]<=8'h7F;      chara[17][1]<=8'h08;      chara[17][2]<=8'h08;      chara[17][3]<=8'h08;      chara[17][4]<=8'h7F;
273 chara[18][0]<=8'h00;      chara[18][1]<=8'h41;      chara[18][2]<=8'h7F;      chara[18][3]<=8'h41;      chara[18][4]<=8'h00;
274 chara[19][0]<=8'h20;      chara[19][1]<=8'h40;      chara[19][2]<=8'h41;      chara[19][3]<=8'h3F;      chara[19][4]<=8'h01;
275 chara[20][0]<=8'h7F;      chara[20][1]<=8'h08;      chara[20][2]<=8'h14;      chara[20][3]<=8'h22;      chara[20][4]<=8'h41;
276 chara[21][0]<=8'h7F;      chara[21][1]<=8'h40;      chara[21][2]<=8'h40;      chara[21][3]<=8'h40;      chara[21][4]<=8'h40;
277 chara[22][0]<=8'h7F;      chara[22][1]<=8'h02;      chara[22][2]<=8'h0C;      chara[22][3]<=8'h02;      chara[22][4]<=8'h7F;
278 chara[23][0]<=8'h7F;      chara[23][1]<=8'h06;      chara[23][2]<=8'h08;      chara[23][3]<=8'h30;      chara[23][4]<=8'h7F;
279 chara[24][0]<=8'h3E;      chara[24][1]<=8'h41;      chara[24][2]<=8'h41;      chara[24][3]<=8'h41;      chara[24][4]<=8'h3E;
280 chara[25][0]<=8'h7F;      chara[25][1]<=8'h09;      chara[25][2]<=8'h09;      chara[25][3]<=8'h09;      chara[25][4]<=8'h06;
281 chara[26][0]<=8'h3E;      chara[26][1]<=8'h41;      chara[26][2]<=8'h51;      chara[26][3]<=8'h21;      chara[26][4]<=8'h5E;
282 chara[27][0]<=8'h7F;      chara[27][1]<=8'h09;      chara[27][2]<=8'h19;      chara[27][3]<=8'h29;      chara[27][4]<=8'h46;
283 chara[28][0]<=8'h26;      chara[28][1]<=8'h49;      chara[28][2]<=8'h49;      chara[28][3]<=8'h49;      chara[28][4]<=8'h32;
284 chara[29][0]<=8'h01;      chara[29][1]<=8'h01;      chara[29][2]<=8'h7F;      chara[29][3]<=8'h01;      chara[29][4]<=8'h01;
285 chara[30][0]<=8'h3F;      chara[30][1]<=8'h40;      chara[30][2]<=8'h40;      chara[30][3]<=8'h40;      chara[30][4]<=8'h3F;
286 chara[31][0]<=8'h1F;      chara[31][1]<=8'h20;      chara[31][2]<=8'h40;      chara[31][3]<=8'h20;      chara[31][4]<=8'h1F;
287 chara[32][0]<=8'h7F;      chara[32][1]<=8'h20;      chara[32][2]<=8'h18;      chara[32][3]<=8'h20;      chara[32][4]<=8'h7F;
288 chara[33][0]<=8'h63;      chara[33][1]<=8'h14;      chara[33][2]<=8'h08;      chara[33][3]<=8'h14;      chara[33][4]<=8'h63;
289 chara[34][0]<=8'h07;      chara[34][1]<=8'h08;      chara[34][2]<=8'h70;      chara[34][3]<=8'h08;      chara[34][4]<=8'h07;
290 chara[35][0]<=8'h61;      chara[35][1]<=8'h51;      chara[35][2]<=8'h49;      chara[35][3]<=8'h45;      chara[35][4]<=8'h43;
291 chara[36][0]<=8'h20;      chara[36][1]<=8'h54;      chara[36][2]<=8'h54;      chara[36][3]<=8'h54;      chara[36][4]<=8'h78;
292 chara[37][0]<=8'h7F;      chara[37][1]<=8'h48;      chara[37][2]<=8'h44;      chara[37][3]<=8'h44;      chara[37][4]<=8'h38;
293 chara[38][0]<=8'h38;      chara[38][1]<=8'h44;      chara[38][2]<=8'h44;      chara[38][3]<=8'h44;      chara[38][4]<=8'h28;
294 chara[39][0]<=8'h38;      chara[39][1]<=8'h44;      chara[39][2]<=8'h44;      chara[39][3]<=8'h48;      chara[39][4]<=8'h7F;
295 chara[40][0]<=8'h38;      chara[40][1]<=8'h54;      chara[40][2]<=8'h54;      chara[40][3]<=8'h54;      chara[40][4]<=8'h18;
296 chara[41][0]<=8'h00;      chara[41][1]<=8'h08;      chara[41][2]<=8'h7E;      chara[41][3]<=8'h09;      chara[41][4]<=8'h02;
297 chara[42][0]<=8'h0C;      chara[42][1]<=8'h52;      chara[42][2]<=8'h52;      chara[42][3]<=8'h4C;      chara[42][4]<=8'h3E;
298 chara[43][0]<=8'h7F;      chara[43][1]<=8'h08;      chara[43][2]<=8'h04;      chara[43][3]<=8'h04;      chara[43][4]<=8'h78;
```



创新点1：以矩阵形式输出图文

步骤2：通过一句if-else语句限定显示区域

步骤3：将矩阵的值赋给相应的显示位置，输出

```
334         if(state==3'b001)           //s_init
335             if(h_cnt>ini_1_x&&h_cnt<ini_1_x+ini*5&&v_cnt>ini_y1&&v_cnt<ini_y1+ini*8)
336                 rgb0 <= chara[13][ (h_cnt-ini_1_x)/ini ][ (v_cnt-ini_y1)/ini ];
337             else if(h_cnt>ini_2_x&&h_cnt<ini_2_x+ini*5&&v_cnt>ini_y1&&v_cnt<ini_y1+ini*8)
338                 rgb0 <= chara[40][ (h_cnt-ini_2_x)/ini ][ (v_cnt-ini_y1)/ini ];
339             else if(h_cnt>ini_3_x&&h_cnt<ini_3_x+ini*5&&v_cnt>ini_y1&&v_cnt<ini_y1+ini*8)
340                 rgb0 <= chara[36][ (h_cnt-ini_3_x)/ini ][ (v_cnt-ini_y1)/ini ];
341             else if(h_cnt>ini_4_x&&h_cnt<ini_4_x+ini*5&&v_cnt>ini_y1&&v_cnt<ini_y1+ini*8)
342                 rgb0 <= chara[39][ (h_cnt-ini_4_x)/ini ][ (v_cnt-ini_y1)/ini ];
343             else if(h_cnt>ini_1_x&&h_cnt<ini_1_x+ini*5&&v_cnt>ini_y2&&v_cnt<ini_y2+ini*8)
344                 rgb0 <= chara[27][ (h_cnt-ini_1_x)/ini ][ (v_cnt-ini_y2)/ini ];
345             else if(h_cnt>ini_2_x&&h_cnt<ini_2_x+ini*5&&v_cnt>ini_y2&&v_cnt<ini_y2+ini*8)
346                 rgb0 <= chara[36][ (h_cnt-ini_2_x)/ini ][ (v_cnt-ini_y2)/ini ];
347             else if(h_cnt>ini_3_x&&h_cnt<ini_3_x+ini*5&&v_cnt>ini_y2&&v_cnt<ini_y2+ini*8)
348                 rgb0 <= chara[37][ (h_cnt-ini_3_x)/ini ][ (v_cnt-ini_y2)/ini ];
349             else if(h_cnt>ini_4_x&&h_cnt<ini_4_x+ini*5&&v_cnt>ini_y2&&v_cnt<ini_y2+ini*8)
350                 rgb0 <= chara[37][ (h_cnt-ini_4_x)/ini ][ (v_cnt-ini_y2)/ini ];
351             else if(h_cnt>ini_5_x&&h_cnt<ini_5_x+ini*5&&v_cnt>ini_y2&&v_cnt<ini_y2+ini*8)
352                 rgb0 <= chara[44][ (h_cnt-ini_5_x)/ini ][ (v_cnt-ini_y2)/ini ];
353             else if(h_cnt>ini_6_x&&h_cnt<ini_6_x+ini*5&&v_cnt>ini_y2&&v_cnt<ini_y2+ini*8)
354                 rgb0 <= chara[55][ (h_cnt-ini_6_x)/ini ][ (v_cnt-ini_y2)/ini ];
355
356         else
357             rgb0<=0;
```



创新点2：以0-1建立迷宫



由于屏幕上显示的3*3的格子，每个方格都需要显示一个0-4的数字，同时迷宫又有横向和纵向两个方向。按照一般的思路，可能需要建立一个 $a*b*3$ 的三维矩阵，而且输入量会进一步增加。

但是在此，使用了0-1矩阵。迷宫通过二维的0-1矩阵就可以建立。在显示到屏幕上时，只需要累加每一个方格周围相邻四格的数字即可。在建立迷宫上，减小了很大的工作量，同时也避免了人工的计算错误。

```
71 -----level1-----  
72     area1(0) <="0000000000";  
73     area1(1) <="0000000000";  
74     area1(2) <="0011111100";  
75     area1(3) <="0000000000";  
76     area1(4) <="0000000000";
```



不足之处

1. 由于时间和器件的限制，最终没有显示图片，但是个人认为已经没有技术难题
2. 兔子的移动以格为单位，不流畅，做进一步改进可以实现微移，使游戏更加精致
3. 未对走过路径作标记，对玩家而已增加了难度



收获

1. 学习了两门标准硬件描述语言
2. 了解了液晶屏的工作方式
3. 增加了对数字电路构架的系统的理解
4. 快速学习与查阅资料
5. 独立思考与合作交流



主要功能

