

---

# 电子科技大学实验报告

课程名称: 数学类综合实验 2

实验地点: 科 A229

指导教师: 张 勇

评 分:

完成实验学生信息:

	姓名	学号	共享百分比/%	备注 (主要工作)
学 生 1	张惠东	2017100101 021		
学 生 2	樊思明	2017100101 013		
学 生 3	罗秋林	2017100101 012		

## 桌游设计—卡牌游戏 UNO 的仿真与模拟

### 1 实验内容

UNO 是一种牌类游戏, 于 1971 年由 Merle Robbins 发明, 本实验基于 UNO 的一些玩法与规则, 通过 MATLAB 编程实现这款桌游的仿真与模拟, 并重现了一局游戏的整个过程 (计算机与计算机之间进行游戏, 游戏过程没有与玩家交互).

UNO 部分卡牌展示如图 1 所示.



图 1 UNO 卡牌展示

## 2 实验目的

由于计算机仿真模拟是随机的过程，通过多次模拟实验，我们可以通过游戏最终的结果在一定程度上评定游戏的合理性、科学性和趣味性。

并且在桌游的仿真模拟过程中，我们可以对桌游有一个更加深入地了解，对 MATLAB 图形交互界面的开发以及数据的可视化有一个初步的了解。

## 3 实验过程

### 3.1 UNO 游戏规则及编程思路

#### 3.1.1 卡牌种类

一副 UNO 卡牌共有 108 张，可供 2~4 人同时参加，其中 76 张普通牌（红蓝黄绿 1~9 各有两张，0 各有一张），24 张功能牌（红蓝黄绿的“跳过”，“翻转”，“加 2”各有两张），8 张万能牌（黑色的“变色牌”，“变色加 4”各有 4 张）

---

### 3.1.2 数据结构存储

定义卡牌的颜色属性集 `{'yellow','red','green','blue','black','noColor'}`，功能属性集 `{'reverse','skip','plus2','plus4','changeColor','none'}`，标号集 `{1,2,...,108}`，这三个集合就能刻画整副卡牌的信息. 定义初始化牌库函数 `initialize`，建立 `108*3` 的数组就能将卡牌库的信息存储

### 3.1.3 游戏规则

游戏开始时，每人分派 7 张牌，然后按逆时针顺序一位接一位出牌，出牌时必须和上家所出牌的颜色，数字或者图案相同.

举个例子容易理解:假设上家打出的牌为红色的 3，那你可以出的牌有:红色的所有牌(颜色相同)，所有颜色的 3(数字相同);假设上家出的牌为黄色的加 2，你可以出的牌有:黄色的所有牌(颜色相同)，其他颜色的加 2(图案相同).

若玩家无牌可出，则须从牌库再摸一张，若这张牌可出，就可即时打出，否则就跳过，让下家出牌;玩者也可假装手上没有牌而抽牌，但若要出牌也必须出刚抽的牌. 最快出完手中牌的人获胜. 其他的人继续进行游戏直到最后一个人输. 若牌库中的牌用完，而尚未有人获胜时，则将废牌堆中的牌(最上一张除外)重洗过，再为新的牌库继续玩.

### 3.1.4 特殊牌效果

**跳过**，打出“跳过”牌后，你的下家将停止出牌，轮到再下家出牌.

该牌只能在上一张是相同颜色的牌或者也是跳过牌的情况下使用.

**翻转**，打出“翻转”牌后，整个出牌顺序将翻转，如逆时针顺序变为顺时针顺序(即假设有 A, B, C, D 四个玩家，本来的出牌顺序应该为  $A \rightarrow B \rightarrow C \rightarrow D$ ，在打出此牌之后，顺序变为  $D \rightarrow C \rightarrow B \rightarrow A$ ).

该牌只能在上一张是相同颜色的牌或者也是翻转牌的情况下使用.

**加 2**，打出“加 2”后，下家将被罚摸 2 张牌，并不能再出牌. 该牌只能在上一张是相同颜色的牌或者也是加 2 牌的情况下使用.

**变色牌**，“变色牌”的功能为不论上一张牌是什么颜色，都可以随意指定下家出牌的颜色. 该牌即使在你手中有其它可出的牌时也可以打出.

**加 4**，“加 4”的功能是不论上一张出牌者打出的是什么颜色，而可以随意指定下家出牌的颜色，同时下家需从牌堆中罚摸 4 张牌，且不能再出牌.

### 3.1.5 洗牌、发牌、出牌实现思路

对初始化的牌库(即 `108*3` 的矩阵)，我们随机生成其中两列的下标，对他们进行交换，对此操作进行 `5*108` 次，便可以模拟洗牌的效果.

在发牌阶段，我们先读入玩家个数 `players_num`，每个玩家有属性 `players_cards` 来记录手牌状态，牌堆状态由 `paidui_cards`, `paidui_num` 记录，每次完成发牌都要更新上述三个参数。

在玩家出牌阶段，记录上一位玩家的玩家编号、出牌信息，根据游戏的出牌顺序调用 `next` 函数找到下一位玩家的手牌情况，根据游戏规则检索手牌，检索到第一个符合条件的牌就进行出牌操作，更新桌面上的牌（即上一位玩家打出的牌），检索不到合适的牌就进行抽牌操作，更新牌库信息。

### 3.1.6 特殊牌的处理

设置控制变量 `skipF`, `sdSkip`, `sdReserve`，用于控制加 2 加 4 牌、跳过牌、翻转牌对其他玩家的影响，如跳过牌只对下一个玩家有影响. 设置 `plus` 变量用于计算加 2 加 4 牌叠加的情况，与控制变量 `skipF` 一起实现保证抽牌效果只对一个玩家有效。

### 3.1.7 游戏反馈情况

若选择观看游戏过程，每次出牌之后，命令行都会更新游戏状态，包括桌上牌数（已经打出的牌数），牌堆剩余牌数，玩家的手牌情况，玩家的出牌情况，如图 2

```
Game Start!  
玩家数:4.  
玩家1先出牌.  
当前是第0次出牌  
桌面上有0张牌,牌堆剩余88张牌  
Player 1 手牌: yellow3 green0 blueskip redplus2 red7  
Player 2 手牌: yellow8 greenreverse red1 red4 blue6  
Player 3 手牌: red2 blue6 yellowskip red2 yellow2  
Player 4 手牌: blue0 blue4 green5 yellow1 green7  
Player1有 5张牌  
Player2有 5张牌  
Player3有 5张牌  
Player4有 5张牌  
  
player 1's turn:  
Player 1 打出 :yellow3 请按任意键继续. . .
```

```
当前是第1次出牌  
桌面上有1张牌,牌堆剩余88张牌  
Player 1 手牌: red7 green0 blueskip redplus2  
Player 2 手牌: yellow8 greenreverse red1 red4 blue6  
Player 3 手牌: red2 blue6 yellowskip red2 yellow2  
Player 4 手牌: blue0 blue4 green5 yellow1 green7  
Player1有 4张牌  
Player2有 5张牌  
Player3有 5张牌  
Player4有 5张牌  
当前牌:yellow3  
player 2's turn:  
Player 2 打出 :yellow8 请按任意键继续. . . |
```

所示。

(a)

(b)

```
当前是第9次出牌
桌面上有9张牌, 牌堆剩余88张牌
Player 1 手牌: red7
Player 2 手牌: blue6 red4 red1
Player 3 手牌: red2 red2 yellow2
Player 4 手牌: green7 blue4 green5 yellow1
Player1有 1张牌
Player2有 3张牌
Player3有 3张牌
Player4有 4张牌
当前牌:redplus2
player 4's turn:

Player 4 拿了 2 张牌, 分别是
yellow9 green4
请按任意键继续. . . |
```

(c)

```
当前是第13次出牌
桌面上有12张牌, 牌堆剩余86张牌
Player 1 手牌:
Player 2 手牌: blue6 red1
Player 3 手牌: yellow2 red2
Player 4 手牌: green7 blue4 green5 yellow1 blue1 blue3
Player1已经赢了
Player2有 2张牌
Player3有 2张牌
Player4有 6张牌
当前牌:red7
player 4's turn:
Player 4 打出 :green7 请按任意键继续. . . |
```

(d)

```
当前是第53次出牌
桌面上有37张牌, 牌堆剩余68张牌
Player 1 手牌:
Player 2 手牌:
Player 3 手牌: red3
Player 4 手牌: blue4 yellow7
Player1已经赢了
Player2已经赢了
Player3有 1张牌
Player4有 2张牌
当前牌:blackplus4 修改颜色为:red
修改数字为:8

player 4's turn:

Player 4 拿了 4 张牌, 分别是
red9 yellow5 green3 yellow9
请按任意键继续. . . |
```

(e)

```
当前是第54次出牌
桌面上有37张牌, 牌堆剩余64张牌
Player 1 手牌:
Player 2 手牌:
Player 3 手牌: red3
Player 4 手牌: blue4 yellow7 redplus2 redskip yellow2 red5
Player1已经赢了
Player2已经赢了
Player3有 1张牌
Player4有 6张牌
当前牌:blackplus4 修改颜色为:red
修改数字为:8

player 3's turn:
Player 3 打出 :red3
第1个获胜的人是:Player 1.
第2个获胜的人是:Player 2.
第3个获胜的人是:Player 3.
输掉比赛的人是:Player 4.
输入任意键以退出
请按任意键继续. . . |
```

(f)

图 2 游戏反馈情况

(a) 比赛开始; (b) 玩家 2 打出相同颜色的牌; (c) 玩家 4 由于上一个玩家出的 plus2 牌而抽取 2 张牌; (d) 玩家 1 由于出完所有的牌而获胜，其他人继续进行游戏; (e) 上一个玩家出了王牌，并修改颜色为红色; (f) 剩余 1 个玩家有手牌，该玩家输掉比赛，显示获胜顺序。

若不选择观看游戏过程，可直接看到游戏结果如图 3 所示。

```
第1个获胜的人是:Player 2.  
第2个获胜的人是:Player 1.  
第3个获胜的人是:Player 3.  
输掉比赛的人是:Player 4.  
输入任意键以退出  
请按任意键继续. . . >>
```

图 3 游戏结果展示

## 3.2 GUI 设计

考虑到游戏确实有部分可变动参数，如玩家个数，玩家手牌上限数，于是我们便尝试做了一个简单的图形交互界面，让用户选择玩家个数，如图 3 所示.



图 4 简单的图形交互界面

## 4 结果分析

假设游戏共四位玩家，每局游戏中第一个出完牌获胜，模拟 1000 局比赛中，四

---

人的获胜次数分别为 226, 235, 264, 275, 可见游戏具有一定的公平性.

## 5 程序

```
function UNOCardGame
    init();
end

function init
    figHandle = figure('Name','UNOCard_Game','MenuBar','none',...    % 生成
主窗口
    'ToolBar','none','NumberTitle','off',...
    'Units','normalized','Position',[0.39 0.30 0.3 0.6],...
    'KeyPressFcn',@keyevent);
    uicontrol(figHandle,'Style','text','Units','normalized',...    % 生成 UNO 标志
    'Position',[0.01,0.81,0.3 0.15],'String','UNO@',...
    'FontSize',25,'FontName','微软雅黑');
    uicontrol(figHandle,'Style','text','Units','normalized',...    % 生成 UNO 标志
    'Position',[0.01,0.50,0.4 0.30],'String','TIPS:选择人数并决定是否看游戏过程
才能开始游戏，游戏终止后才能开始下一轮',...
    'FontSize',15,'FontName','微软雅黑');
    uicontrol(figHandle,'Style','pushbutton','Units','normalized',...
    'Position',[0.33 0.81 0.245 0.09],'String','选择人数',...
    'FontSize',13,'Callback',@player_choose,'KeyPressFcn',@keyevent) % 生成选
择人数按钮
    uicontrol(figHandle,'Style','pushbutton','Units','normalized',...
    'Position',[0.60 0.81 0.40 0.09],'String','是否观看游戏过程',...
    'FontSize',13,'Callback',@process_choose,'KeyPressFcn',@keyevent) % 生成
选择人数按钮
    uicontrol(figHandle,'Style','pushbutton','Units','normalized',...
    'Position',[0.75 0.70 0.245 0.09],'String','开始游戏',...
    'FontSize',13,'Callback',@game_start,'KeyPressFcn',@keyevent) % 生成选择
```

---

难度按钮

end

```
function player_choose(hObject,handles)
    global numOption numPop numPush
    numOption = dialog('Name',' 请 选 择 游 戏 人 数',
    'Units','normalized','Position',[0.45 0.50 0.20 0.20]); % 难度选择
    numPop = uicontrol(numOption,'Style','popupmenu','Units','normalized',... %
弹出式菜单
    'Position',[0.31 0.5 0.4 0.4],'String',{'1','2','3','4'},...
    'FontSize',10);
    numPush = uicontrol(numOption,'Style','pushbutton','Units','normalized',... %
确定按钮
    'position',[0.3      0.05      0.4      0.32],'String','      确      定',
    'FontSize',20,'Callback',@pushCallback);
end
```

```
function process_choose(hObject,handles)
    global proOption proPop proPush
    proOption = dialog('Name',' 请 选 择 游 戏 人 数',
    'Units','normalized','Position',[0.45 0.50 0.20 0.20]); % 难度选择
    proPop = uicontrol(proOption,'Style','popupmenu','Units','normalized',... %
弹出式菜单
    'Position',[0.31 0.5 0.4 0.4],'String',{'是','否'},...
    'FontSize',10);
    proPush = uicontrol(proOption,'Style','pushbutton','Units','normalized',... %
确定按钮
    'position',[0.3 0.05 0.4 0.32],'String','确定','FontSize',20,'Callback',@retrospect);
end
```

```
function pushCallback(hObject,handles) % 人数选择 确定按钮 回调函数
    global numPop numOption num_player
    num_player = get(numPop,'Value');
```



---

```

        close(numOption)
    end

function retrospect(hObject,handles)
    global proOption proPop pause
    pause = get(proPop,'Value');
    close(proOption)
end

function game_start(hobject,handles)
    global num_player pause
    global C %用字典存储了游戏的控制参数
    C = py.dict(pyargs(...
        'clc',1, ...%控制画面刷新
        'pause',pause, ...%控制每次出牌后暂停
        'N',108, ...%牌数
        'P',num_player, ...%玩家数，可修改 2~8
        'maxP',8, ...%最大玩家数
        'maxPC',50, ...%最大手牌数
        'C2C',1, ...%1 则电脑 vs 电脑，0 则人机，暂时不可修改
        'color',py.list({'yellow','red','green','blue','black','noColor'}),...
        'function',py.list({'reverse','skip','plus2','plus4','changeColor','none'}))...
    ));
    global color2num
    color2num = py.dict(pyargs( ...
        C{'color'}{1},1, ...
        C{'color'}{2},2, ...
        C{'color'}{3},3, ...
        C{'color'}{4},4, ...
        C{'color'}{5},5, ...
        C{'color'}{6},6 ...
    ));
    global function2num

```

---

```
function2num = py.dict(pyargs( ...
    C{'function'}{1},1, ...
    C{'function'}{2},2, ...
    C{'function'}{3},3, ...
    C{'function'}{4},4, ...
    C{'function'}{5},5, ...
    C{'function'}{6},6 ...
));
global cards
cards = zeros(C{'N'},3);%3*108 的矩阵， 储存卡牌库的信息
color,function,num
global players_num
players_num = zeros(1,C{'maxP'});
global players_cards
players_cards = zeros(C{'maxP'},C{'maxPC'},3);
global paidui_num
paidui_num = 0;
global used_num
used_num = 0;
global used_cards
used_cards = zeros(C{'N'},3);
global paidui_cards
paidui_cards = zeros(C{'N'},3);

global skipF
skipF = 0;%用于控制加 2 加 4 牌只对后面一位玩家有效
global sdSkip
sdSkip = 0;%用于控制 skip 牌只对后面一位玩家有效
global sdReverse
sdReverse = 0;%用于控制 reverse 牌只对后面一位玩家有效
global plus
plus=0;%累计加 2 加 4 牌加的值
global paihang
```

---

```
paihang = zeros(1,C{'maxP'});
global colorNow
colorNow = 1;
global numNow
numNow = 1;
global using
using = zeros(1,3);
%num2clr = py.list({'yellow','red','green','blue','black','noColor'});

%    setGlobal;
%    global C;
%    global players_num;
%    global sdSkip;
%    global sdReverse;
%    global plus;
%    global paihang;
%    global cards;
%players_num = zeros(1,C{'maxP'});
disp('Game Start!')
i=1;%玩家序号
j=0;
k=1;%reverse 后*=-1
q=0;%获胜玩家数
lost=0;%剩下的玩家的序号
plus=0;

i=randi(C{'P'});%随机决定谁先出牌
initialize;
fprintf('玩家数:%d.\n 玩家%d 先出牌.\n',C{'P'},i)
sheffle_init;
%    for ii=1:108
%        printCard(cards(ii,:));
%        if mod(ii,8)==0
```

---

```
%          fprintf('\n');
%          end
%      end

      fapai();
%牌堆所有牌
%      for ii=1:10
%          printCard(paidui_cards(ii,:));
%          if mod(ii,8)==0
%              fprintf('\n');
%          end
%      end
winning = 0;
while 1
    while 1%此循环用于在出牌时跳过已经获胜的玩家
        if players_num(i)>0
            break;
        end
        i = next(i,k);
        j = j + 1;
        if j >= C{'P'}
            j = 0;
            winning = 1;
            break;
        end
    end
    if winning == 0
        j = 0;
        output(i);%显示信息
    end
    q = winnerProcess(q);
    if alldone(q) == 1,
        break;
    end
end
```

---

```
end

if sdReverse == 1
    sdReverse = 0;
    if C{'P'}-q ~= 2
        fprintf('\nReverse\n');
        k = -k;
    else
        sdSkip = 1;%当只剩 2 人时，把 reverse 当 skip 用
    end
end

if sdSkip == 1
    sdSkip = 0;
    i = next(i,k);
    while players_num(i)<=0
        i = next(i,k);
    end
    fprintf('\nSkip player %d\n',i);
end

i = next(i,k);
collect;
disp('输入任意键以继续');
if(C{'pause'}==1)
    system('pause');
end
if(C{'clc'}==1)
    clc;
end

end
fprintf('\n');
clc;
```

---

```

    for ii=1:C{'P'}
        if ii~=C{'P'}
            fprintf('第%d 个获胜的人是:Player %d.\n',ii,paihang(ii));
        end
        lost = lost + ii;
    end
    for ii=1:C{'P'}-1
        lost = lost - paihang(ii);
    end
    fprintf('输掉比赛的人是:Player %d.\n',lost);
    disp('输入任意键以退出');
%     fprintf('%d',num_player);
    system('pause');
end

```

```

function r = next(i,k)
    global C;
    i = i + k;
    if i>C{'P'}
        i=1;
    elseif i<1
        i = C{'P'};
    end
    r = i;
end

```

```

function r=winnerProcess(q)
    global players_num;
    global C;
    global paihang;
    for i=1:C{'P'}
        if players_num(i)==0
            paihang(q+1)=i;

```

---

```

        q = q + 1;
        players_num(i) = -1;
    end
end
r = q;
end
function r=alldone(q)
    global C;
    if(q==C{'P'}-1)
        r=1;
    else
        r=0;
    end
end
function collect()
    global used_num;
    global used_cards;
    global paidui_num;
    global paidui_cards;
    if used_num>=60
        %system('pause')
        for i = 1:60
            paidui_num = paidui_num + 1;
            paidui_cards(paidui_num,:) = used_cards(used_num,:);
            used_num = used_num - 1;
        end
    end
    sheffle;
end
function initialize()
    global color2num;
    global function2num;
    global cards;
```

---

```
global paidui_num;
global used_num;
for i = 1:19
    cards(i,1) = color2num{'yellow'};
    cards(i,2) = function2num{'none'};
    if i == 1
        cards(i,3) = 0;
    else
        cards(i,3) = mod(i-1,10)+floor((i-1)/10);%只有一个 0，有两个 1~9
    end
end
for i = 20:38
    cards(i,1) = color2num{'red'};
    cards(i,2) = function2num{'none'};
    if i == 20
        cards(i,3) = 0;
    else
        cards(i,3) = mod(i-20,10)+floor((i-20)/10);
    end
end
for i = 39:57
    cards(i,1) = color2num{'green'};
    cards(i,2) = function2num{'none'};
    if i == 39
        cards(i,3) = 0;
    else
        cards(i,3) = mod(i-39,10)+floor((i-39)/10);
    end
end
for i = 58:76
    cards(i,1) = color2num{'blue'};
    cards(i,2) = function2num{'none'};
    if i == 58
```



---

```
        cards(i,3) = 0;
    else
        cards(i,3) = mod(i-58,10)+floor((i-58)/10);
    end
end
for i = 77:84
    cards(i,1) = mod(i-1,4)+1;
    cards(i,2) = function2num{'reverse'};
    cards(i,3) = -1;
end
for i = 85:92
    cards(i,1) = mod(i-1,4)+1;
    cards(i,2) = function2num{'skip'};
    cards(i,3) = -1;
end
for i = 93:100
    cards(i,1) = mod(i-1,4)+1;
    cards(i,2) = function2num{'plus2'};
    cards(i,3) = -1;
end
for i = 101:104
    cards(i,1) = color2num{'black'};
    cards(i,2) = function2num{'changeColor'};
    cards(i,3) = -2;
end
for i = 105:108
    cards(i,1) = color2num{'black'};
    cards(i,2) = function2num{'plus4'};
    cards(i,3) = -2;
end

paidui_num = 0;
used_num = 0;
```

---

end

function sheffle %洗牌

global paidui\_cards;

global paidui\_num;

global C;

n=paidui\_num;

for i = 1:5\*C{'N'}

    ran1 = randi(n);

    ran2 = randi(n);

    %switch

    temp = paidui\_cards(ran1,:);

    paidui\_cards(ran1,:) = paidui\_cards(ran2,:);

    paidui\_cards(ran2,:) = temp;

end

end

function sheffle\_init

global C;

global cards;

for i = 1:5\*C{'N'}

    ran1 = randi(C{'N'});

    ran2 = randi(C{'N'});

    %switch

    temp = cards(ran1,:);

    cards(ran1,:) = cards(ran2,:);

    cards(ran2,:) = temp;

end

end

function fapai

global C;

global players\_cards;

global players\_num;%players\_card = zeros(C{'maxP'},C{'maxPC'},3);

global cards;

---

```
    global paidui_cards;
    global paidui_num;
    for i=1:C{'P'}
        for j=1:5
            players_cards(i,j,:) = cards(5*(i-1)+j,:);
        end
        players_num(i) = players_num(i) + 5;
    end
    paidui_num = C{'N'}-5*C{'P'};
    for i=1:paidui_num
        paidui_cards(i,:) = cards(i+5*C{'P'},:);
    end
end
function printCards(who)
    global players_num;
    global players_cards;
    fprintf("Player %d 手牌: ",who)
    for i=1:players_num(who),
        printCard(players_cards(who,i,:));
    end
    fprintf('\n')
end
function output(who)
    global C;
    global players_num;
    global used_num;
    global color2num;
    global colorNow;
    global numNow;
    global using;
    global paidui_num;
    fprintf('桌面上有%d 张牌,',used_num);
    fprintf('牌堆剩余%d 张牌\n',paidui_num);
```

---

```

    printCards(1);
    printCards(2);
    printCards(3);
    printCards(4);
    for i=1:C{'P'},
        if players_num(i)>0,
            fprintf('Player%d 有%2d 张牌\n',i,players_num(i))
        else
            fprintf('Player%d 已经赢了\n',i)
        end
    end
end
if used_num ~=0,
    fprintf('当前牌:')
    printCard(using)
    if using(1)==color2num{'black'},
        fprintf('修改颜色为:')
        fprintf('%s',C{'color'}{colorNow})
        fprintf('\n 修改数字为:%d\n',numNow)
    end
end
end
if C{'C2C'} == 1
    fprintf('\nplayer %d"s turn:\n',who);
end
comInput(who)
end
function printCard(x)
    global C;
    fprintf('%s',C{'color'}{x(1)});
    if x(3)>=0
        fprintf('%d ',x(3));
    else
        fprintf('%s',C{'function'}{x(2)})
        fprintf(' ');
    end
end

```

---

```
end
end
function useCard(who,which)
    global players_cards;
    global players_num;
    global used_num;
    global used_cards;
    global colorNow;
    global numNow;
    global plus;
    global sdSkip;
    global sdReverse;
    global function2num;
    global color2num;
    global using;
    using = players_cards(who,which,:);
    used_cards(used_num+1,:) = players_cards(who,which,:);
    players_cards(who,which,:) = players_cards(who,players_num(who),:);
    players_num(who) = players_num(who) - 1;
    used_num = used_num + 1;
    fprintf('Player %d 打出 :',who);
    printCard(using);
    if using(1)~=color2num{'black'},%不是黑色牌，则更新当前颜色
        colorNow = using(1);
    else,%如果是黑色牌，则选择颜色
        colorNow = randi(4);
        while hasColor(who,colorNow)==0,
            colorNow = randi(4);
        end
    end
end
if using(2)==function2num{'plus4'}
    plus = plus + 4;
elseif using(2)==function2num{'plus2'}
```

---

```

        plus = plus + 2;
    end
    if using(3)>=0%有数字则更新数字
        numNow = using(3);
    end
    if using(2)==function2num{'skip'}
        sdSkip = 1;
    end
    if using(2)==function2num{'reverse'}
        sdReverse = 1;
    end
end

```

```

function r = check(who,which)
    global players_cards;
    global function2num;
    global color2num;
    global used_num;
    global skipF;
    global using;
    global colorNow;
    global numNow;
    global players_num;
    if
        players_num(who)==1
        players_cards(who,which,2)~=function2num{'none'}
        r = 0;%最后一张牌不能出功能牌
        return;
    end
    if used_num==0%第一张牌可以随便出
        r = 1;
        return;
    end
    if skipF==1%如果上一位玩家由于加 2 加 4 牌被罚，则现在这位玩家可以
    &&

```

---

无视现在的加 2 加 4 牌的功能，只需要有适当的颜色\数字

```
if (players_cards(who,which,1)==colorNow) ||
(players_cards(who,which,1)==color2num{'black'}) || ...
    (players_cards(who,which,3)==numNow)
    r = 1;%颜色或数字一样可以出
    return;
else
    r = 0;
    return;
end
else
    if players_cards(who,which,2)==function2num{'plus4'}%有+4 则什么都
能出
        r = 1;
        return;
    elseif using(2)==function2num{'plus4'}%没+4 不能出在+4 后面
        r = 0;
        return;
    end
    if players_cards(who,which,2)==function2num{'plus2'}%有+2 则什么都
能出(除+4)
        r = 1;
        return;
    elseif using(2)==function2num{'plus2'}%没+2 不能出在+2 后面(除+4)
        r = 0;
        return;
    end
    if players_cards(who,which,1)==color2num{'black'}%有黑牌什么都能出
(除+2+4)
        r = 1;
        return;
    elseif using(1)==color2num{'black'},
        if
```

---

```

(players_cards(who,which,1)==colorNow)||(players_cards(who,which,3)==numNow)
    r = 1;%颜色或数字一样可以出
    return;
else
    r = 0;
    return;
end
end

if using(3)==-1,%如果当前是功能牌
    if (players_cards(who,which,1)==colorNow) ||
(players_cards(who,which,2)==using(2))
        r = 1;
        return;
    else
        r = 0;
        return;
    end
end

if
(players_cards(who,which,1)==colorNow)||(players_cards(who,which,1)==color2num{'b
lack'})|| ...
        (players_cards(who,which,3)==numNow)
    r = 1;%颜色或数字一样可以出
    return;
else
    r = 0;
    return;
end
end
end
function comInput(who)

```



---

```
global using;
global plus;
global skipF;
global function2num;
global players_num;
for i=1:players_num(who),%遍历手牌
    if check(who,i)==1%如果可用，就出牌
        useCard(who,i)
        skipF = 0;
        return;
    end
end
%如果没有可以出的牌，则抽牌
if using(2)==function2num{'plus4'},
    takeCard(who,max(1,plus));
    skipF = 1;
    plus = 0;
elseif using(2)==function2num{'plus2'},
    takeCard(who,max(1,plus));
    skipF = 1;
    plus = 0;
else
    takeCard(who,1);%没牌出，摸牌
end
end
function r = hasColor(who,c)
    global players_num;
    global players_cards;
    for i=1:players_num(who),
        if c==players_cards(who,i,1)
            r=1;
            return;
        end
    end
```

---

```
    end
    r = 0;
end
function takeCard(who,x)
    global players_cards;
    global paidui_num;
    global players_num;
    global paidui_cards;

    %fprintf('who:%d,players_num(who)+1==%d\n',who,players_num(who)+1)f
    %fprintf('paidui_num:%d\n,used_num:%d\n',paidui_num,used_num);
    for i=1:x,
        players_cards(who,players_num(who)+1,:)= paidui_cards(paidui_num,:);
        paidui_num = paidui_num - 1;
        players_num(who) = players_num(who) + 1;
    end
    fprintf('\nPlayer %d 拿了 %d 张牌,分别是\n',who,x);
    for i=x-1:-1:0,
        printCard(paidui_cards(paidui_num-i,:))
    end
    fprintf('\n')
end
```