

## 代码类参考题目——贪吃蛇

### 1. 项目说明

用 MATLAB 实现较为简单、基础的贪吃蛇。

### 2. 实现方案

实现的要点包括墙壁与蛇身碰撞的检测、吃到食物的判定与蛇身增长、食物与障碍物的随机刷新、蛇的移动、分数的记录等。实现思路为在坐标系中利用矩阵储存蛇头、蛇身以及食物的位置坐标，通过绘制散点图的方式绘制出蛇与食物，再利用定时器逐帧刷新图像，通过图像的变化呈现出蛇的移动同时实时记录分数，游戏暂停或结束后会给出该局游戏的分数。

游戏的操作通过  $\uparrow$   $\downarrow$   $\leftarrow$   $\rightarrow$  调整蛇的前进方向，吃到食物分数+1，另外每吃到食物后会随机刷新三个障碍物，蛇头碰到障碍物、墙壁以及自身都会被判定为结束，结束时显示最终分数。

### 3. 设计说明

代码较为简单，主要包括以下几个部分：

首先是相关参数的设定。首先设定相关参数，包括坐标系的形状、颜色、坐标范围、蛇头、蛇身的初始位置、长度、颜色、食物与障碍物的初始位置与颜色、定时器的各项参数以及分数显示

```
function snake
    axis equal %设置x,y刻度相同
    axis([0.5, 20.5, 0.5, 20.5]) %坐标轴范围
    set(gca,'xtick',[],'ytick',[],'xcolor','w','ycolor','w') %设置坐标轴刻度范围与颜色
    set(gca,'color','g') %背景颜色设置为绿色
    hold on %每次更新图像保持原有图像
    Head = [7,7]; %蛇头初始位置
    Direct = [1,0]; %蛇头初始方向
    Body = [7,7;6,7;5,7]; %蛇身初始位置数组
    Length=3; %蛇的初始长度
    food = [12,14]; %食物的初始位置
    obstacle = randi(20,[3,2]);
    Snake = scatter(gca,Body(:,1),Body(:,2),150,'ro','filled'); %绘制蛇身为红色实心圆
    Food = scatter(gca,food(1),food(2),150,'bh','filled'); %绘制食物为蓝色六角形
    Obstacle = scatter(gca,obstacle(:,1),obstacle(:,2),200,'ro'); %绘制障碍物
    set(gcf,'KeyPressFcn',@key) %读入用户键盘信息
    fps=5; %设置帧数，即蛇的移动速度
    Gaming=timer('ExecutionMode','FixedRate','Period',1/fps,'TimerFcn',@game); %通过定时器完成图像更新，调用ga
    start(Gaming) %启动定时器，开始游戏
    h1=text(20.5,20.5,'当前分数:'); %显示游戏开始时的分数
    h2=text(20.5,19.5,'0');
```

其次是蛇的移动。蛇的移动通过蛇头与方向确定，蛇头坐标与方向实现蛇头的移动，移动后的蛇头加移动前的蛇身再去掉移动前的蛇尾便完成了整条蛇的移动。

```

function game(~,~) %实现蛇的移动
    Head = Head + Direct; %实现蛇头的移动
    Body = [Head;Body];%蛇身移动第一步
    while length(Body)>Length
        Body(end,:)=[]; %将剩余的蛇尾变为空的
    end

```

之后是蛇身碰撞、碰壁以及障碍物的判定。大体思路相同，即通过坐标表示是否重合或有交集完成判断，判断后利用 `questdlg` 问题对话框显示最终分数以及下一步的操作。

```

if intersect(Body(2:end,:), Head, 'rows') %判断蛇头是否撞到自己的身体，蛇头与蛇身数组交集非空则为撞上
    a='最后分数: ';%显示最后分数
    b=num2str(Length-3);%将分数转化为字符串显示
    c=[a,b];
    Button1 = questdlg(c,'您撞到了自己','重新开始','关闭游戏','Yes');%弹出选项对话框，失败之后选择下一步
    if Button1 == '重新开始'
        clf;%清屏
        snake();%重新开始
    else
        close;%关闭游戏
    end
end
if ismember(Head,obstacle,'rows')
    a='最后分数: ';%显示最后分数
    b=num2str(Length-3);%将分数转化为字符串显示
    c=[a,b];
    Button1 = questdlg(c,'您撞到了障碍','重新开始','关闭游戏','Yes');%弹出选项对话框，失败之后选择下一步
    if Button1 == '重新开始'
        clf;%清屏
        snake();%重新开始
    else
        close;%关闭游戏
    end
end

```

最后是输入键位的判断。采用 `switch-case` 结构，首先通过输入  $\uparrow \downarrow \leftarrow \rightarrow$  操纵蛇的移动方向

```

function key(~,event) %判断用户操纵的方向
    switch event.Key
        case 'uparrow'
            direct = [0, 1];
        case 'downarrow'
            direct = [0, -1];
        case 'leftarrow'
            direct = [-1, 0];
        case 'rightarrow'
            direct = [1, 0];
    end

```

```

        direct = 11, 0],
    case 'space' %空格使游戏暂停
        stop(Gaming);
        direct = Direct;
        a='当前分数:';
        b=num2str(Lengh-3);
        c=[a,b];
        Button3 = questdlg(c, '游戏暂停', '重新开始', '关闭游戏', '继续游戏', '关闭游戏');%暂停对话?
        if Button3 == '重新开始'
            clf;
            snake();
        elseif Button3 == '关闭游戏'
            close;
        else
            start(Gaming); %继续游戏启动定时器
        end
    otherwise
        direct = nan; %其它按键无意义
    end
    if any(Direct + direct) %蛇不能180度转弯
        Direct = direct;
    end
end
end
...1

```

输入空格键暂停游戏，暂停时弹出问题对话框显示当前分数与下一步操作。输入其他键位无变化。另外蛇不能 180 度转弯，蛇的运动方向不变。

#### 4. 难点与解决

在设计过程中我认为整体的逻辑并不复杂，设计思路是主要的难点。与设计思路密切相关的一个点是分数的显示。贪吃蛇的移动是通过图像的叠加与更新实现的，但数字叠加在一起并不能完成分数的显示。解决方法是每次定时器调用 game 函数时首先清除上一次显示的分数，但将 text 的定义置于 text 消除后又无法实现。因此首先在游戏开始时显示 text 文本，之后每次调用 game 时首先势上一次的文本内容不可见，之后重新定义，从而解决了分数显示的问题。

#### 5. 总结与参考资料

总的来说我认为这段代码的逻辑比较简单，对我来说可以说是在学习 MATLAB 过程中的一次实际运用。整体的思路设计参考了一些资料，在这些资料的基础上我对代码进行了简化、详注与拓展，添加了分数显示以及障碍等功能。虽然难度比较低，但是我深刻体会到了 MATLAB 强大的计算功能。

以下是主要的参考资料：

[https://blog.csdn.net/qq\\_37688204/article/details/102709090](https://blog.csdn.net/qq_37688204/article/details/102709090)