# 贪吃蛇游戏

#### 游戏设计思路：

1. 定义墙体长宽高、蛇头(H)、蛇身（xxx）、食物(F)、以及上述参数位置
2. 定义函数将游戏界面显示出来：墙体、初始蛇、食物
3. 读用户操作：w-上、a-左、s-下、d-右、q-退出游戏
4. 判断读入该操作后，蛇的下一步是什么字符：

空白位置（space）/ 食物（F）/ 墙（wall）/ 蛇身

1. 根据上述判断进行移动：

空白位置——移动、食物——移动并将蛇身加长一格、墙/蛇身——游戏结束

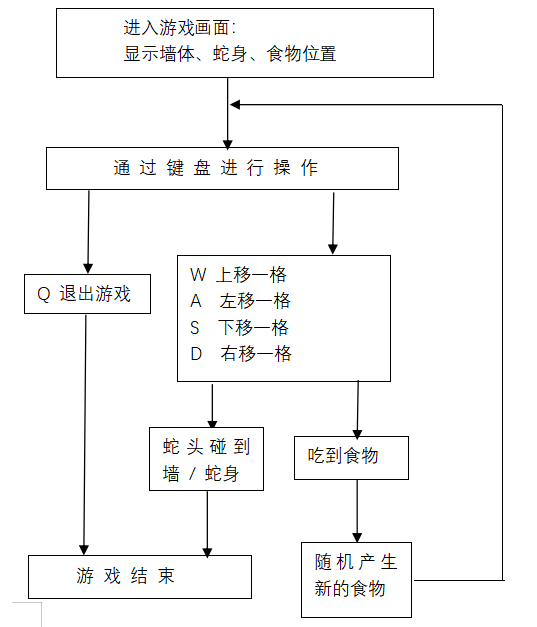
1. 其他函数：（1） 刷新游戏界面

（2） 随机产生食物

1. 每一步移动后，调用函数刷新游戏界面，显示移动后的情况
2. 当食物被吃掉后，调用函数产生新的食物 （食物位置随机）
3. 统计分数：当吃到一个食物的时候，score+1；

当撞墙或撞到蛇身的时候，游戏结束并输出出最终分数

#### 流程图



#### 代码实现

#! /bin/bash

readonly snakeHead='H' #定义蛇头

readonly snakeBody='x' #定义蛇身

readonly wall='I' #定义墙体

readonly spac='.' #定义空隙

readonly snakeFood='F' #定义食物

map\_length=20 #墙长

map\_width=20 #墙宽

snake\_begin=(28 29 30) #蛇的初始位置

snake\_length=3 #蛇的初始长度

judge\_return=0 #蛇下一步的情况的返回

snake\_return=0 #蛇走下一步后的返回

var=0 #一个变量

source\_x=0 #蛇头的x坐标位置

source\_y=0 #蛇头的y坐标位置

food\_pos=51 #食物的初始位置

**#显示墙体**

map=($wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $spac $wall

$wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall $wall

)

printMap(){

echo

for((i=0;i<map\_length;i++)); do

for((j=0;j<map\_width;j++));do

echo -e ${map[i\*$map\_length+j]} "\c"

done

echo

done

}

**#读入用户下一步操作，并定义该操作对应的x，y坐标改变值**

readOption(){

read next

case $next in

'q')

snake\_return=0

;;

'w')

snakeMove -1 0

;;

's')

snakeMove 1 0

;;

'a')

snakeMove 0 -1

;;

'd')

snakeMove 0 1

;;

\*)

echo ''please input w,a,s,d to operate the snake and then press enter!! \n Or you can press q to exit '

snake\_return=-1;

;;

esac

}

**#判断用户想让蛇走的下一步是什么字符——食物or墙体/蛇身or空白**

judge(){

x=$1;

y=$2;

pos=$(($y\*$map\_length+$x));

if test ${map[$pos]} = $snakeFood

then

judge\_return=1

elif test ${map[$pos]} = $wall -o ${map[$pos]} = $snakeBody

then

judge\_return=0

else

judge\_return=-1

fi

}

**#根据判断结果进行蛇的移动：**

**空白位置——移动、食物——移动并将蛇身加长一格、墙/蛇身——游戏结束**

snakeMove(){

ver\_move=$1;

lev\_move=$2;

go\_x=$(($source\_x+$lev\_move)); #蛇头的x坐标变化

go\_y=$(($source\_y+$ver\_move)); #蛇头的y坐标变化

judge $go\_x $go\_y

res=$judge\_return #调用judge返回值

if test $res == 0 #若为0，则说明碰到墙体or蛇身，游戏结束

then

snake\_return=0;

elif test $res == 1 #若为1，则说明吃到食物，蛇身加长

Then

snake\_begin[snake\_length]=$(($go\_y\*$map\_length+$go\_x));

snake\_length=$snake\_length+1;

snake\_return=1;

else #若为-1，则说明空白位，蛇头蛇身移动

for((i=0;i<$snake\_length-1;i++));do

mid=${snake\_begin[i]};

if test $i -eq 0

then

map[$mid]=$spac;

fi

snake\_begin[i]=${snake\_begin[i+1]};

snake\_begin[i+1]=$mid;

done

snake\_begin[snake\_length-1]=$(($go\_y\*$map\_length+$go\_x));

snake\_return=-1;

fi

}

**# 刷新游戏界面**

mapGenerator(){

mid=$(($snake\_length-1))

for((i=0;i<$snake\_length;i++));do

if [[ $i == $mid ]];then

map[${snake\_begin[$i]}]=$snakeHead;

else

map[${snake\_begin[$i]}]=$snakeBody;

fi

done

map[$food\_pos]=$snakeFood

}

**#产生新的食物**

foodGenerator(){

while [[ 1 ]]; do

num=$((RANDOM% ($map\_length \* $map\_width) ))

echo $num ${map[$num]} + $spac

if test ${map[$num]} = $spac

then

food\_pos=$num

break

fi

done

}

**#主函数**

score=0

while [[ 1 ]]; do

var=$(($snake\_length-1));

source\_y=$((${snake\_begin[$var]} / $map\_length));

source\_x=$((${snake\_begin[$var]} % $map\_length));

snake\_return=0

mapGenerator

echo -e "score: $score \n w -up s -down a -left d -right q -exit\n "

printMap

readOption

res=$snake\_return

echo $res

if test $res == 0

then

echo "你的最终得分是" $score "!下次继续努力! "

break;

elif test $res == 1

then

score=$(($score+1))

foodGenerator

fi

clear

done

#### 调试运行

