**软件开发实习报告**

没

**学 号**

**姓 名 米迦**

**班 级**

**2021 年 11 月 30 日**

# 贪吃蛇

## 摘要

“贪吃蛇”游戏是一个经典的游戏，它因操作简单，娱乐性强而广受欢迎。本文基于Java技术和IDEA开发环境，开发了一个操作简单、界面简洁的""贪吃蛇""游戏。实现了游戏的开始，暂停，结束。通过本游戏的开发，达到学习Java技术和熟悉软件开发流程的目的

整个游戏分为3个类，1个static文件存放各种静态资源。分别是StartGame ，Data和GamePanel类。其中 StartGame类中有main方法，主要是设置了frame框架，窗口可见性，可关闭性。Data类中是导入static中的图片资源，并new对应的ImageIcon对象。GamePanel类继承JPanel类直接实现了KeyListener，ActionListener。

为了保证程序代码的正确性和功能实现的可靠性，本文章还介绍了游戏软件的功能测试结果。

**关键词**：**Java IDEA 贪吃蛇**

Java基础

画笔

事件监听

键盘监听

定时器Timer（帧，如果时间片足够小，就是动画，一秒30帧。连起来是动画，拆开就是静态的图片！）

# 引言

“贪吃蛇”游戏是一个经典的游戏，它因简单的操作、很强的娱乐性而广受欢迎。对于程序来说贪吃蛇制作相对简单，而且逻辑性强，通常是要步入游戏开发这个队伍前必练的项目。贪吃蛇这个游戏的制作方法对于以前的我而言是很神秘的，我希望通过我自己所学的知识和自己的努力把它剖析起来。正是由于这个想法，本文基于java技术和IDEA开发环境，设计并实现了一个操作简单、界面美观的“贪吃蛇”游戏。

文章游戏优化给出了个人对游戏的一些想法优化，已经实现部分，未实现的我会在今后深入的学习中将整个项目补充完整并不断优化。由于时间的紧迫和自己的能力有限，编写的贪吃蛇游戏还是相对简单，但就整个游戏来说，目的是通过本游戏的设计与实现，达到学习java技术和熟悉软件开发流程的目的。我相信在以后通过自己不断的努力我会在编程这条路上越走越远。

# 一、系统开发环境

(1)操作系统： Windows10

(2). 应用软件：jdk1.5

# 二、系统需求分析

## 2.1需求分析

本系统主要是完成贪吃蛇游戏的基本操作。用户可以自己练习和娱乐。本系统需要满足以下几点要求： (1) 利用方向键来改变蛇的运行方向。 (2) 空格键暂停或继续游戏，并在随机的地方产生食物。 (3) 吃到食物就变成新的蛇体（有增长、变短和变形），碰到自身或者特殊的食物则游戏结束，否则正常运行。

## 2.2可行性分析

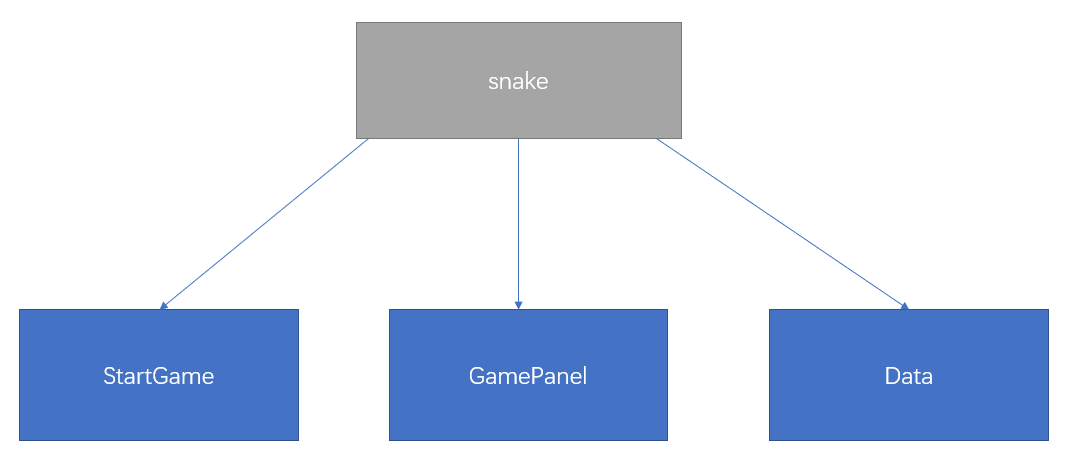
贪吃蛇游戏是一种简单的大众的游戏，自从计算机实现以来，深受广大电脑玩家的喜爱，做一个简单的贪吃蛇小游戏。 贪吃蛇的核心算法时如何实现移动和吃掉食物，把蛇头和蛇身体放入到一个数组。没有碰到食物的时候，蛇头坐标向前移动一个，身体的数组通过循环i—来实现一节一节向前移动。当按上下左右键的时候

# 三、系统概要设计

## 3.1设计目标

本软件是针对贪吃蛇小游戏的JAVA程序，利用方向键来改变蛇的运行方向，空格键暂停或继续游戏，并在随机的地方产生食物，吃到食物就变成新的蛇体，碰到壁或自身则游戏结束，否则正常运行。

## 3.2系统功能模块



定时器是使小蛇动起来的关键，通过定时器刷新小蛇的位置！







通过吃食物升级，你可以变成不同蛇\龙。文案写的有些中二。



碰到自身，游戏结束。



这里设定吃樱桃同样会死亡



吃芒果会缩短长度。

# 四、游戏优化

* 积分 等级 等级越高速度越快 (定时器ms设置越小)
* 游戏撞墙的判定
* 小蛇回头问题 （如果fx是left就不可以按right，fx是up就不可以按down 等等......)
* 超级果实（吃一个 length+3 +5等）
* 毒果实（吃了就死， 或者长度减小）
* 也可以设置鼠标控制移动（例如：如果鼠标的y坐标大于头坐标，就向着下移动）
* 数据库 续盘，可以暂停过几天接着玩
* 网络编程 联机对战
* 登陆界面 绑定用户 (使用弹窗dialog)

# 五、关键代码展示

//画小蛇

if(fx.equals("R")){

Data.right.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("L")){

Data.left.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("U")){

Data.up.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("D")){

Data.down.paintIcon(this,g,snakeX[0],snakeY[0]);

}

for (int i=1;i<length;i++)

{

Data.body.paintIcon(this,g,snakeX[i],snakeY[i]);

}

switch(score)

{

case -20:

g.setColor(Color.cyan);

g.setFont(new Font("微软雅黑",Font.BOLD,30));//设置字体

g.drawString("你还贪吃！！",250,30);

g.drawString("只剩一个头了吧=、=。",250,65);

if(fx.equals("R")){

Data.fenlongright.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("L")){

Data.fenlongleft.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("U")){

Data.fenlongup.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("D")){

Data.fenlong.paintIcon(this,g,snakeX[0],snakeY[0]);

}

for (int i=1;i<length;i++)

{

Data.fenbody.paintIcon(this,g,snakeX[i],snakeY[i]);

}

break;

case -10:

g.setColor(Color.cyan);

g.setFont(new Font("微软雅黑",Font.BOLD,30));//设置字体

g.drawString("你贪吃了什么不该垂涎的东西！",250,30);

g.drawString("你快只剩一个头了。",250,65);

break;

case 10:

g.setColor(Color.cyan);

g.setFont(new Font("微软雅黑",Font.BOLD,30));//设置字体

g.drawString("暗影龙",250,30);

g.drawString("你现在只是一个灵体，飘荡在虚无中。",250,65);

if(fx.equals("R")){

Data.fenlongright.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("L")){

Data.fenlongleft.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("U")){

Data.fenlongup.paintIcon(this,g,snakeX[0],snakeY[0]);

}else if(fx.equals("D")){

Data.fenlong.paintIcon(this,g,snakeX[0],snakeY[0]);

}

for (int i=1;i<length;i++)

{

Data.fenbody.paintIcon(this,g,snakeX[i],snakeY[i]);

}

break;

不同分数对应不同的蛇的图片和蛇的文字描述信息！！！

……后面重复性代码 略

//键盘监听事件

@Override

public void keyTyped(KeyEvent e) {

}

@Override

public void keyPressed(KeyEvent e) {

int keyCode = e.getKeyCode();//获得键盘的按键是哪一个

if(keyCode==KeyEvent.VK\_SPACE){

//如果按下的是空格键

if (isFail){

//重新开始

//初始化

isFail=false;

init();

}else {

isStart = !isStart; //取反

}

repaint();//重新绘制

}

//小蛇移动

if (keyCode==KeyEvent.VK\_UP){

fx="U";

}else if (keyCode==KeyEvent.VK\_DOWN){

fx="D";

}else if (keyCode==KeyEvent.VK\_LEFT){

fx="L";

}else if (keyCode==KeyEvent.VK\_RIGHT){

fx="R";

}

}

@Override

public void keyReleased(KeyEvent e) {

}

这里是对键盘按压监听事件的重写，上下左后控制小蛇移动，空格控制暂停或开始。

//事件监听---需要通过固定

@Override

public void actionPerformed(ActionEvent e) {

if (isStart &&isFail==false){

//吃食物

if (snakeX[0]==foodx & snakeY[0]==foody||snakeX[0]==foodx2&snakeY[0]==foody2){

//长度+1

length++;

//分数加10

score=score+10;

//再次随机食物

foodx=25+25\*random.nextInt(34); // 850/25=34

foody=75+25\*random.nextInt(24); // 600/25=24

foodx2=25+25\*random.nextInt(34); // 850/25=34

foody2=75+25\*random.nextInt(24); // 600/25=24

ytx=25+25\*random.nextInt(34); // 850/25=34

yty=75+25\*random.nextInt(24); // 600/25=24

mgx=25+25\*random.nextInt(34); // 850/25=34

mgy=75+25\*random.nextInt(24); // 600/25=24

}else if(snakeX[0]==mgx&snakeY[0]==mgy){

length--;

score=score-10;

//再次随机食物

foodx=25+25\*random.nextInt(34); // 850/25=34

foody=75+25\*random.nextInt(24); // 600/25=24

foodx2=25+25\*random.nextInt(34); // 850/25=34

foody2=75+25\*random.nextInt(24); // 600/25=24

ytx=25+25\*random.nextInt(34); // 850/25=34

yty=75+25\*random.nextInt(24); // 600/25=24

mgx=25+25\*random.nextInt(34); // 850/25=34

mgy=75+25\*random.nextInt(24); // 600/25=24

}else if(snakeX[0]==ytx&snakeY[0]==yty){

isFail=true;

}

//如果游戏是开始状态就让小蛇动起来！

//移动

for (int i=length-1;i>0;i--){

//后面一节移动到前面一节的位置！！！

snakeX[i]=snakeX[i-1];

snakeY[i]=snakeY[i-1];

}

// snakeX[0]=snakeX[0]+25;

//走向,边界

if(fx.equals("R")){

snakeX[0]=snakeX[0]+25;

if(snakeX[0]>850){

snakeX[0]=25;//碰到边界

}

}else if(fx.equals("L")){

snakeX[0]=snakeX[0]-25;

if(snakeX[0]<25){

snakeX[0]=850;//碰到边界

}

}else if(fx.equals("U")){

snakeY[0]=snakeY[0]-25;

if(snakeY[0]<75){

snakeY[0]=650;//碰到边界

}

}else if(fx.equals("D")){

snakeY[0]=snakeY[0]+25;

if(snakeY[0]>650){

snakeY[0]=75;//碰到边界

}

}

//失败判定条件,碰到自己就算失败

for (int i =1;i<length;i++){

if (snakeX[0]==snakeX[i]&&snakeY[0]==snakeY[i])

isFail=true;

}

这里是对事件监听的重写。