**一.需求**

1.游戏界面应该有一个固定大小的游戏区域，可以在其中移动贪吃蛇。

2.贪吃蛇由一个头部和若干个身体组成，初始时应该有一个头部和一个身体。

3.游戏开始时，贪吃蛇应该向右移动，玩家可以使用方向键控制贪吃蛇的行动方向（上、下、左、右）。

4.当贪吃蛇头部碰到游戏区域里的石头或自身的身体时，游戏结束。

5.在游戏区域中随机生成食物，贪吃蛇可以通过吃食物来增长身体长度。

6.当贪吃蛇头部碰到食物时，应该吃掉食物并在尾部增加一个身体。

7.游戏运行时，可以暂停和继续游戏。

8.可以设置游戏各部位的颜色

9.有多种游戏地图可供选择

10.游戏中的大部分参数可通过配置文件设置

11.支持选择贪吃蛇头部图标，如设置为龙头。

12.支持设置择贪吃蛇尾部图标，如设置为龙尾。

13.支持设置择贪吃蛇中间图标

14.支持在每吃几个食物，能随机掉出奖品，每个奖品可设置不同的积分。完成后可得出本次游玩的总积分

**二.设计**

1.游戏界面：使用Java Swing创建游戏窗口，游戏区域生成固定的石头，穿过边界会从另一边出来

2.贪吃蛇：用一个链表（LinkedList）数据结构来表示贪吃蛇，每个节点代表贪吃蛇的一个身体部分。初始化时，只有2个节点表示贪吃蛇的头部和一个身体，每当吃到食物后，就在链表尾部新增一个节点。

3.食物：由地形对象生成空的坐标点（Point）用于投放食物，初始投放一个食物，当蛇吃掉食物后再丢一个食物

4.地图：用二维数组保存布尔值记录对应位置是否为石块

5.游戏流程：游戏开始时，贪吃蛇会自动向右移动，玩家可以通过键盘方向键来改变贪吃蛇的移动方向。每当贪吃蛇移动一格后，需要判断是否与石头或自身撞击，如果撞击则游戏结束。如果贪吃蛇的头部碰到了食物，就吃掉食物并增加贪吃蛇的长度。

6.用户交互：玩家可以通过键盘方向键来控制贪吃蛇的移动方向。在游戏结束后，玩家可以点击控制栏重新开始游戏，且可通过控制栏选择颜色地图贪吃蛇图片等

7.图片方向：头部图片根据前进方向，身体根据前一截与后一节身体位置判断是直是弯，对应方向。

1. **编码**

主要类：

Controller:控制器，负责游戏的启停计分等

Food：食物的实现

Gift：奖励的实现

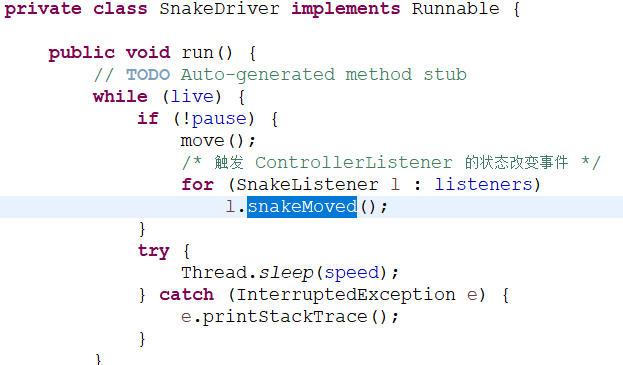
Ground：场地地图的实现

Snake:蛇的实现

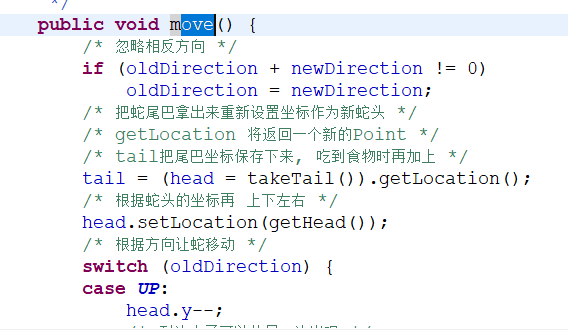
MainFrame:游戏的主窗口，启动类

核心功能：

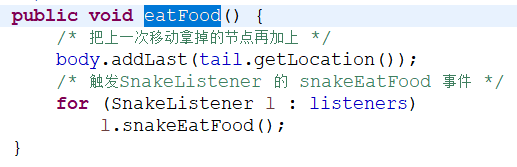
1. 贪吃蛇的行动：通过实现Runnable接口创建线程，使用while循环在循环内判断游戏是否存活和是否暂停的状态来执行move方法，在try-catch 中，通过调用 Thread.sleep(speed) 方法使线程休眠一段时间，以控制游戏速度。



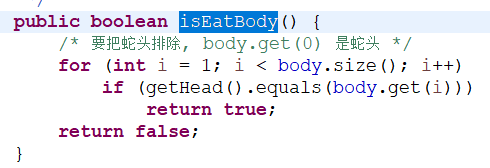
1. 蛇的move方法：通过设置上下方向为+1 -1，左右方向为+2 -2，同时记录正在前进的方向和输入的方向，通过对应的值相加是否为0的判断来忽略不合理的反向移动，去掉尾部并根据方向增加头部



1. 吃食物：通过将蛇尾的位置加到蛇的末尾增加蛇的长度，然后通过遍历 listeners 集合，触发每个 SnakeListener 对象的 snakeEatFood() 方法。

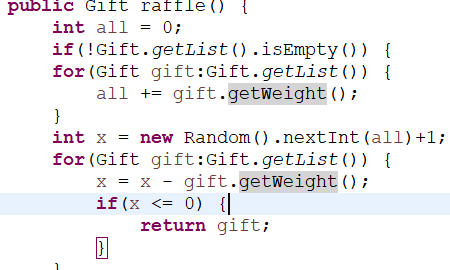


1. 碰撞检测：通过rocks[snake.getHead().x][snake.getHead().y] 获取贪吃蛇头部所在位置是否有石头。通过头部位置和身体进行遍历判断是否吃到自己的身体



4.地图：通过二维数组保存布尔值来记录对应点位是否为石块

5.随机奖励抽取：增加奖励会将该奖励加入list中，在总权重范围内生成随机数，之后遍历list减去对应权重，若减完则抽中该项



**四.使用**

1.运行软件

点击开始新游戏后游戏开始，通过方向键控制方向（不可反向移动），回车键可暂停或继续游戏，page up ，page down加速或减速，贪吃蛇吃掉食物身体变长，撞到石头或蛇身游戏结束，点击停止游戏可直接游戏结束，结束后弹出本局得分



2.背景网格颜色设置

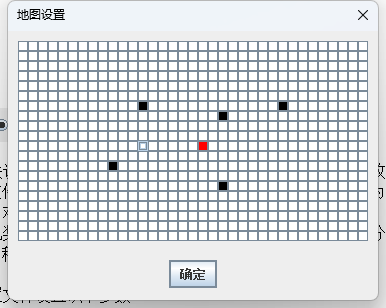
点击背景颜色设置可设置背景颜色，勾选显示网格可显示地图网格，再点击颜色可进行网格颜色设置，设置后继续游戏可直接生效



3地图选择

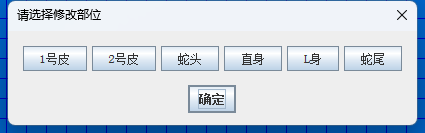
有俩种地图布局可供选择，也可点击自设布局自由设置（红色为出生点不可设置石块）地图选择后不会直接生效，需开始新游戏生效





4.图片设置

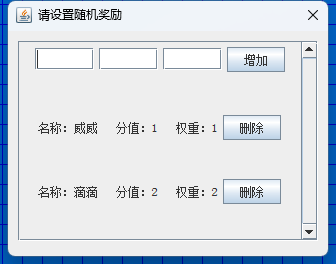
点击蛇皮肤设置可设置皮肤图片，有俩套默认，以及四部位的自由修改（L身为直角蛇身），也可自找图片放入img文件夹（若无可在游戏目录下自行创建，需注意图片方向与默认图片方向一致），食物图片同样



5.随机奖励

点击随机奖励可设置随机奖励的抽取池，权重代表所占比例，分值与权重需为整数，奖励名称可为字符串，每吃到5个食物进行抽取（提示文字仅初次打开显示）

权重默认非负数，输入负数自动取绝对值，设置完成后点击X关闭窗口



1. 得分显示

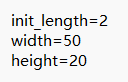
游戏顶部显示当前速度以及得分信息（食物得分+奖励得分）



7.配置文件

可通过配置文件设置以下参数

配置文件为游戏目录下snake.ini若无可自行创建,格式如图



单格宽度/像素 : cell\_width, 或 cell\_size （1~100）默认23

地图宽度/格 ： width （10~80）默认30

地图高度/格 ：height （10~60）默认20

蛇初始长度/格 ：init\_length （2~最大宽度）默认2

初始速度 ：speed （10~无限）默认200

加减速幅度 ：speed\_step （1~无限）默认25

E=[LOC乘B的0.333次方除以P]的三次方乘(t的4次方分之一)

E=effort in person-months or person-years

B=special skills factor

P=productivity factor

t=project duration (months or years)

**以下是一个贪吃蛇的项目，请结合这个项目细节，给出一个可能的The Software Equation**

**The Software Equation是E=[LOC乘B的0.333次方除以P]的三次方乘(t的4次方分之一)**

**E=effort in person-months or person-years**

**B=special skills factor**

**P=productivity factor**

**t=project duration (months or years)**

**项目细节：**

**• Divide the project into 4 little parts. 1. Project Management 2.coding 3.testing 4. summarizing• Develop each part in 4 weeks.• During those 4 weeks have a meeting almost every day.**

**一.需求**

1.游戏界面应该有一个固定大小的游戏区域，可以在其中移动贪吃蛇。

2.贪吃蛇由一个头部和若干个身体组成，初始时应该有一个头部和一个身体。

3.游戏开始时，贪吃蛇应该向右移动，玩家可以使用方向键控制贪吃蛇的行动方向（上、下、左、右）。

4.当贪吃蛇头部碰到游戏区域里的石头或自身的身体时，游戏结束。

5.在游戏区域中随机生成食物，贪吃蛇可以通过吃食物来增长身体长度。

6.当贪吃蛇头部碰到食物时，应该吃掉食物并在尾部增加一个身体。

7.游戏运行时，可以暂停和继续游戏。

8.可以设置游戏各部位的颜色

9.有多种游戏地图可供选择

10.游戏中的大部分参数可通过配置文件设置

11.支持选择贪吃蛇头部图标，如设置为龙头。

12.支持设置择贪吃蛇尾部图标，如设置为龙尾。

13.支持设置择贪吃蛇中间图标

14.支持在每吃几个食物，能随机掉出奖品，每个奖品可设置不同的积分。完成后可得出本次游玩的总积分

**二.设计**

1.游戏界面：使用Java Swing创建游戏窗口，游戏区域生成固定的石头，穿过边界会从另一边出来

2.贪吃蛇：用一个链表（LinkedList）数据结构来表示贪吃蛇，每个节点代表贪吃蛇的一个身体部分。初始化时，只有2个节点表示贪吃蛇的头部和一个身体，每当吃到食物后，就在链表尾部新增一个节点。

3.食物：由地形对象生成空的坐标点（Point）用于投放食物，初始投放一个食物，当蛇吃掉食物后再丢一个食物

4.地图：用二维数组保存布尔值记录对应位置是否为石块

5.游戏流程：游戏开始时，贪吃蛇会自动向右移动，玩家可以通过键盘方向键来改变贪吃蛇的移动方向。每当贪吃蛇移动一格后，需要判断是否与石头或自身撞击，如果撞击则游戏结束。如果贪吃蛇的头部碰到了食物，就吃掉食物并增加贪吃蛇的长度。

6.用户交互：玩家可以通过键盘方向键来控制贪吃蛇的移动方向。在游戏结束后，玩家可以点击控制栏重新开始游戏，且可通过控制栏选择颜色地图贪吃蛇图片等

7.图片方向：头部图片根据前进方向，身体根据前一截与后一节身体位置判断是直是弯，对应方向。

1. **编码**

主要类：

Controller:控制器，负责游戏的启停计分等

Food：食物的实现

Gift：奖励的实现

Ground：场地地图的实现

Snake:蛇的实现

MainFrame:游戏的主窗口，启动类

核心功能：

1. 贪吃蛇的行动：通过实现Runnable接口创建线程，使用while循环在循环内判断游戏是否存活和是否暂停的状态来执行move方法，在try-catch 中，通过调用 Thread.sleep(speed) 方法使线程休眠一段时间，以控制游戏速度。
2. 蛇的move方法：通过设置上下方向为+1 -1，左右方向为+2 -2，同时记录正在前进的方向和输入的方向，通过对应的值相加是否为0的判断来忽略不合理的反向移动，去掉尾部并根据方向增加头部
3. 吃食物：通过将蛇尾的位置加到蛇的末尾增加蛇的长度，然后通过遍历 listeners 集合，触发每个 SnakeListener 对象的 snakeEatFood() 方法。
4. 碰撞检测：通过rocks[snake.getHead().x][snake.getHead().y] 获取贪吃蛇头部所在位置是否有石头。通过头部位置和身体进行遍历判断是否吃到自己的身体

4.地图：通过二维数组保存布尔值来记录对应点位是否为石块

5.随机奖励抽取：增加奖励会将该奖励加入list中，在总权重范围内生成随机数，之后遍历list减去对应权重，若减完则抽中该项

**四.使用**

1.运行软件

点击开始新游戏后游戏开始，通过方向键控制方向（不可反向移动），回车键可暂停或继续游戏，page up ，page down加速或减速，贪吃蛇吃掉食物身体变长，撞到石头或蛇身游戏结束，点击停止游戏可直接游戏结束，结束后弹出本局得分

2.背景网格颜色设置

点击背景颜色设置可设置背景颜色，勾选显示网格可显示地图网格，再点击颜色可进行网格颜色设置，设置后继续游戏可直接生效

3地图选择

有俩种地图布局可供选择，也可点击自设布局自由设置（红色为出生点不可设置石块）地图选择后不会直接生效，需开始新游戏生效

4.图片设置

点击蛇皮肤设置可设置皮肤图片，有俩套默认，以及四部位的自由修改（L身为直角蛇身），也可自找图片放入img文件夹（若无可在游戏目录下自行创建，需注意图片方向与默认图片方向一致），食物图片同样

5.随机奖励

点击随机奖励可设置随机奖励的抽取池，权重代表所占比例，分值与权重需为整数，奖励名称可为字符串，每吃到5个食物进行抽取（提示文字仅初次打开显示）

权重默认非负数，输入负数自动取绝对值，设置完成后点击X关闭窗口

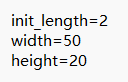
得分显示

游戏顶部显示当前速度以及得分信息（食物得分+奖励得分）

7.配置文件

可通过配置文件设置以下参数

配置文件为游戏目录下snake.ini若无可自行创建,格式如图



单格宽度/像素 : cell\_width, 或 cell\_size （1~100）默认23

地图宽度/格 ： width （10~80）默认30

地图高度/格 ：height （10~60）默认20

蛇初始长度/格 ：init\_length （2~最大宽度）默认2

初始速度 ：speed （10~无限）默认200

加减速幅度 ：speed\_step （1~无限）默认25

在小组的组成方式上，我们采用了一个固定的负责人,协调特定的任务及负责子任务的二级负责人关系的控制分权式，固定的负责人由pm担任。对于我们来说控制分权式主要有以下两个优点

①高效的决策和灵活性：允许团队成员在决策和问题解决方案上拥有更大的自主权。这样可以加快决策过程，减少决策层级，提高项目响应速度和灵活性。

②激发创新和创造力：鼓励团队成员发挥创新和创造力，提出新的想法和解决方案。团队成员可以根据自身的专业知识和经验，自主决策和探索更好的方法来完成项目任务。

根据项目细节和Scrum模型的要求，可以将该项目划分为以下四个小部分：

1. 项目管理：

- 确定项目需求和目标。

- 制定项目计划和安排。

- 管理项目进度和资源分配。

- 组织每日会议，以确保团队协作和沟通。

2. 编码：

- 创建游戏界面，并使用Java Swing实现。

- 实现贪吃蛇的移动和控制逻辑。

- 实现食物的生成和贪吃蛇的吃食物逻辑。

- 实现游戏的暂停和继续功能。

- 实现游戏地图的生成和选择功能。

3. 测试：

- 编写单元测试，验证各个功能模块的正确性。

- 进行集成测试，确保各个模块之间的协作正常。

- 进行系统测试，测试整个游戏的功能和性能。

- 修复和调试出现的问题和错误。

4. 总结：

- 对项目进行总结和评估，确认项目是否满足需求和目标。

- 收集用户反馈和意见，进行改进和优化。

- 归档项目文档和代码，准备项目交付。

根据每个小部分的开发时间为4周，可以按照Scrum的迭代方式进行开发。每个迭代周期为4周，期间每天都有会议进行团队协作和沟通。在每个迭代周期结束后，进行总结和评估，并根据需要进行调整和改进。

在具体实施Scrum模型时，可以使用产品待办清单、迭代计划、每日站会、迭代评审和迭代回顾等Scrum活动来管理项目。团队成员可以根据每个小部分的任务和需求，自行分解任务并进行开发，同时保持每日站会的沟通和协作。在迭代结束后进行评审和回顾，根据反馈和意见进行改进和优化。

在项目目前的阶段，我们一共发现了7个错误，其中7个全部解决了，在接下来的过程中，我们预计可能出现19个错误，其中的16个将可以被解决，根据公式，DRE为84.21%

根据项目细节和给定的The Software Equation，我们可以将项目的各个部分与软件开发中的工作量、技能和生产力因素对应起来，从而得出一个可能的The Software Equation。

根据项目细节，我们将项目划分为以下几个部分：

项目管理（Project Management）

编码（Coding）

测试（Testing）

总结（Summarizing）

对应The Software Equation的各个参数如下：

LOC (Lines of Code): 这里我们可以将LOC定义为整个项目的代码行数。

B (Special Skills Factor): 这个因素表示开发人员在项目中所需的特殊技能和经验。贪吃蛇项目相对来说比较简单，不需要太多特殊技能，因此可以将B设为较小的值，比如1。

P (Productivity Factor): 这个因素表示开发人员的生产力水平。由于没有具体提到开发人员的生产力水平，我们可以将P设为1，表示标准生产力水平。

t (Project Duration): 这个因素表示项目的持续时间，可以根据项目细节中的信息来确定。在这里，我们将t设为4个月，因为每个部分的开发都需要4周时间。

根据以上参数，我们可以得到一个可能的The Software Equation：

E = (LOC \* B^(0.333)) / P^3 \* (t^0.25)