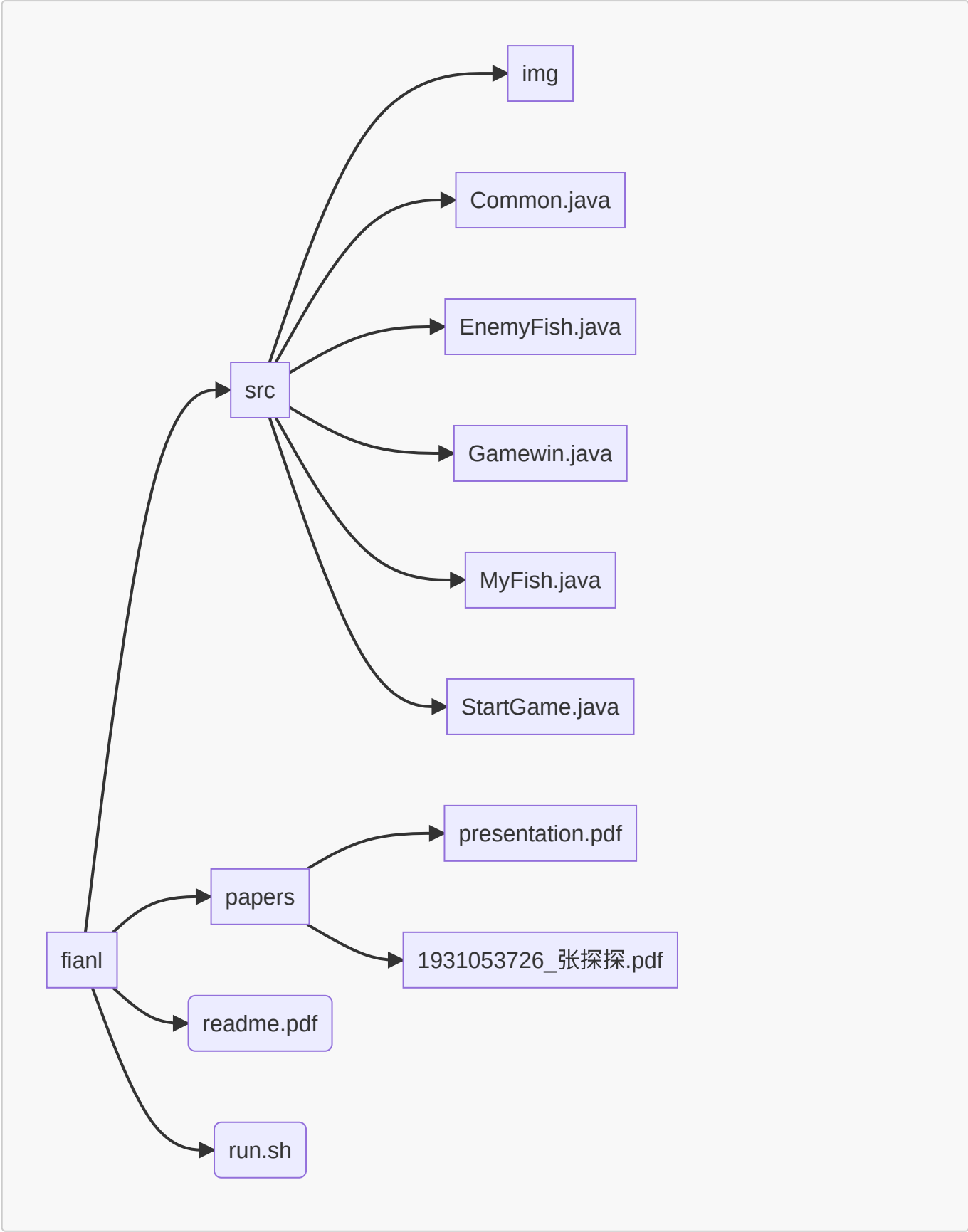


文档目录



运行方法

进入FINAL目录下，使用shell命令：

```
./run.sh
```

来运行整个程序，由于脚本执行需要可执行权限，所以在shell内部添加了chmod命令：以下是run.sh的部分内容：

```
echo "脚本需要管理员密码来获取可执行权限, 若已获得请忽略!"  
sudo chmod a+x run.sh
```

项目依赖环境

- 所需包

```
import java.awt.Color;  
import java.awt.Graphics;  
import java.awt.event.*;  
import java.awt.*;  
import java.util.List;  
import java.util.ArrayList;  
import java.util.Random;  
import javax.swing.JFrame;  
import javax.swing.WindowConstants;  
import javax.swing.event.MouseInputAdapter;
```

项目实施细节

整体使用图片的方式进行显示，众所周知，所有的视频皆是由一张一张图片的快速切换所构成。

- 窗口使用的是JFrame建立，图像使用Image来创建，使用Graphics来创建画笔，来创建图像。
- 不断遍历各个资源借助Graphics添加在Image上，然后将Image添加在JFrame，当我们不断构图，则静态的图片达到了动态的效果。
- 使用JFrame内置的按键监听的内置方法addKeyListener，然后重写了keyPressed方法。来进行按键监听。
- 使用内置addMouseListener方法进行鼠标监听。然后重写了mouseClicked方法，进行鼠标监听。
- 所有的资源数据均存放在Common类中，左右移动的鱼是头朝向左右的两张图片，用户控制的鱼则有上下左右四张照片。
- 敌方鱼的生成，获取左右随机数，屏幕高度随机数，以及鱼种类随机数。来生成敌方鱼。
- 我方鱼采用按键监听来获取鱼的移动方向。然后加减移动速度来使得鱼的图片移动。
- 碰撞检测：遍历敌方鱼，然后检测敌方鱼与我方鱼所在矩阵是否有相交的部分，若是相交则进行判断。
 - 若敌方鱼等级大于我方鱼： 游戏结束
 - 若敌方鱼等级小于我方鱼： 吃掉

- 敌方鱼的死亡，当被我方鱼吃掉或者自动移动出范围，鱼从列表中被删除。
- 我方鱼的转向：通过改变Myfish类的Image的图片来实现鱼的转向。
- 然后构建不同的底图来实现游戏开始或者暂停以及游戏结束的图片。

效果展示

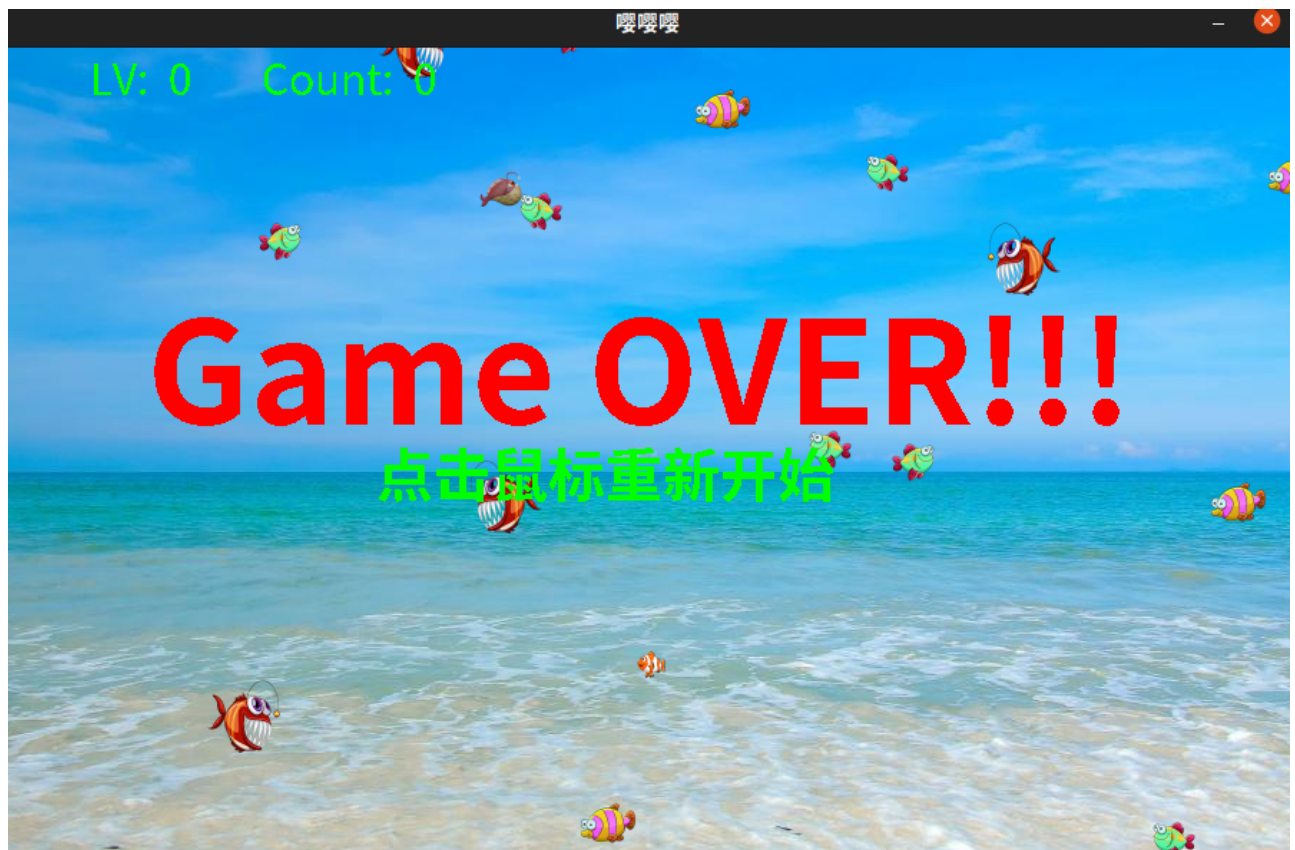
- 开始界面



- 游戏暂停



- 游戏结束



- 游戏开始1



- 游戏运行2

