

# Forest Witch 说明文档

## 游戏背景设定

(由于工作量巨大，剧情没来得及制作，因此只有游戏关卡)

大致文案：

一个森林中的小女巫前往人类的国度进行历练，途中经历重重困难得到成长。在她经历过第一个关卡（新手引导）后，来到了一个人类村落。村民们热情地接待了她并治疗她的伤口，在听说村子里有好多女孩子被魔王掳走后，她不顾村民们的劝阻，毅然决然的决定前往魔王的城堡解救姑娘们。

在森林中经历重重关卡和危险后，终于到了魔王城堡。小女巫却发现村里的女孩们并未受到伤害，并且是自愿留在这里的。她从女孩们口中得知，魔王将她们掳来后为她们提供食宿，并为她们提供学习魔法的环境，在她们能力足够后，会被送往首都的魔法学院学习。反之，村民们才是愚昧的，他们认为只有将少女献祭给神灵才能得到庇护，因此每年都会献祭大量女孩。村民们热情对待小女巫，其实都是为了将她作为替代品牺牲。

知道真相后的小女巫受到了震撼，心境上似乎有所成熟。在她见到魔王后，发现魔王竟然是当初她来村子时替她治疗的医师。魔王无奈地表示自己没有杀人掳人的喜好，并且为了养活一城堡的女孩，不得不白天在村子当医师打工，晚上偷摸救人、采购魔法书、处理事务等等，天天007，身边的魔将都累走一批又一批。

小女巫想留下来帮助魔王，却被他拒绝了。魔王看她也是个可怜孩子，让她也去首都魔法学院学习，以提高自己的能力。如果学完还能记得他，再回来帮助他。于是小女巫便踏上了前去首都的新旅程。

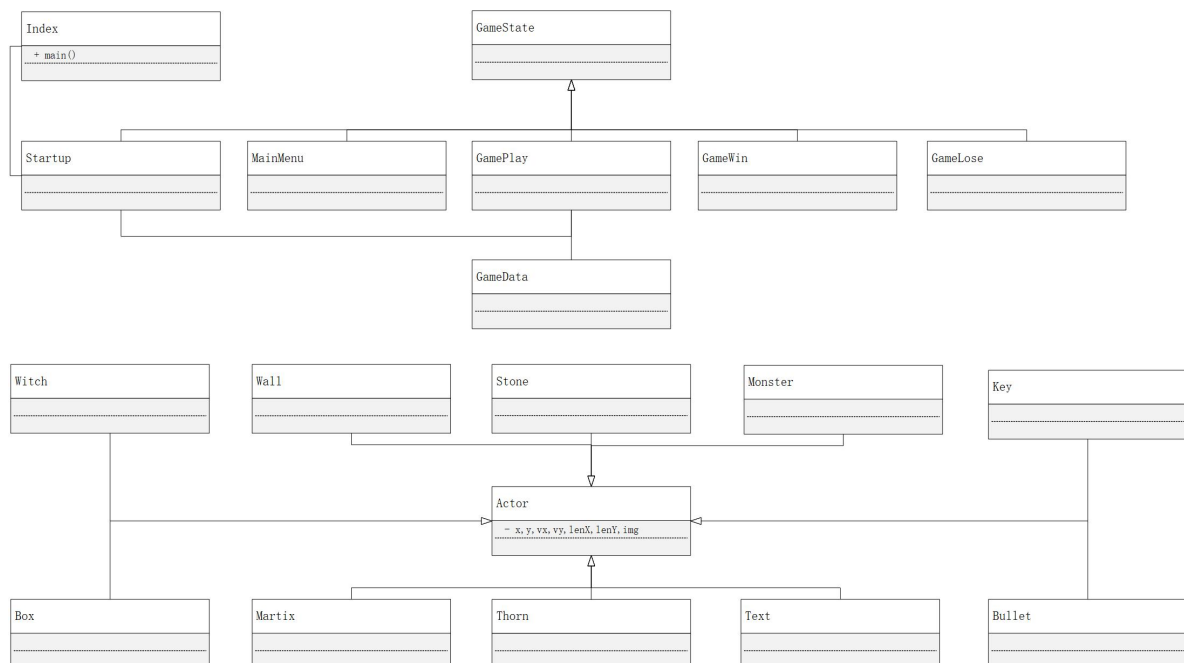
## 游戏玩法说明

- 通过 w、a、s、d 键对角色进行控制
- 通过点击鼠标发射子弹，可攻击怪物
- 按ESC键可以从游戏关卡返回菜单页
- 遇到石头可以进行移动
- 在规定步数内走到终点即可通关(传送阵)

## 游戏实现功能

1. 点击任意位置发射子弹，子弹可消灭怪物
2. 角色碰撞石头，若前方无遮挡物可进行推动
3. 得到钥匙后可以打开宝箱
4. 地图周围被强围住，角色不可越过
5. 地刺随移动步骤间隔出现，碰到则会减少步数

## 类图



# 类说明

## 1. Index

- 游戏主类，main 函数所在位置
- 实现了键盘监听、鼠标监听、双缓冲技术、多线程

## 2. GameState

- 游戏状态接口

## 3. StartUp

- 继承自 GameState
- 游戏启动页，文字从底部向上滚动
- 文字上升到固定位置 / 按任意键 / 点击鼠标可以切换至菜单页面

## 4. MainMenu

- 继承自 GameState
- 游戏菜单页，显示帮助信息以及按钮
- 菜单上下浮动：
  - 定义一个存放移动方向的数组，来记录移动规律：

```
int deltaMove[] = {1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,
-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,0,0};
```

- 反复遍历该数组，来改变起始
- 按钮：
  - 点击开始切换到 GamePlay 状态，点击退出则关闭游戏
  - 出现的问题：Button 组件无法自定义图片，查到一般使用 JButton 组件来美化按钮。但是刷新页面时，JButton 组件会不断闪烁，无法保持在画面最上层。

- 猜测的可能原因：用户鼠标悬在 `JButton` 上方时，组件会移到上层，但随后又被其他图片刷新给覆盖住了。从而导致的闪烁。
- 解决方法：在 `JButton` 组件的上层覆盖一张一模一样的图片，以欺骗用户的眼睛。

## 5. GameWin

- 继承自 `GameState`
- 游戏全部关卡通关后跳转，"游戏成功"字样自下而上滚动，滚动到一定位置后切换到 `Startup` 状态

## 6.GameLose

- 继承自 `GameState`
- 任意一关步数  $< 0$  则切换到此状态，"游戏失败"字样自下而上滚动，滚动到一定位置后切换到 `MainMenu` 状态

## 7.GamePlay

- 继承自 `GameState`
- 游戏进行类，实现了关卡切换、碰撞后的逻辑处理、子弹发射方向计算等核心功能

## 8. GameData

- 游戏静态数据类

## 9. Actor

- 可碰撞物体类
- 实现了判断碰撞功能

## 10. Witch

- 游戏角色类
- 对其他游戏角色进行碰撞
- 对地刺的碰撞需要与其他角色的碰撞分离，这是因为石头会和地刺同时出现，需要分别做判断

## 11. Wall

- 女巫不可通过的“墙”，继承自 `Actor` 类
- 通过鼠标点击查看控制台输出的坐标，来确定墙应该在的位置及大小，并将数据存放在 `GameData` 类中：

```
// 墙
static int wallInfo0[][]={
    {540,110,200,100},
    {270,190,270,100},
    {170,270,100,200},
    {80,520,100,200},
    {95,780,800,100},
    {825,560,100,200},
    {740,210,100,450},
    {635,420,100,150},
    {370,490,250,90}};
static int wallInfo1[][]={
    {85,455,100,300},
```

```
{160,265,100,200},
{270,180,400,100},
{650,270,200,100},
{835,340,100,500},
{545,835,300,100},
{180,750,200,100},
{370,470,150,400},
{370,380,80,90}};
static int wallInfo2[][]={
    {27,650,100,200},
    {130,535,100,100},
    {210,380,90,360},
    {300,200,80,180},
    {390,100,400,80},
    {390,300,270,80},
    {770,200,100,180},
    {855,380,100,300},
    {765,570,100,300},
    {130,830,700,100},
    {405,480,80,80},
    {585,480,80,80},
    {405,660,80,80},
    {585,660,80,80}};
```

## 12. Stone

- 石头，继承自 Actor 类
- 女巫碰撞石头，若当前女巫的移动方向上，前方没有阻挡物，则石头可以被推动

## 13. Monster

- 怪物，继承自 Actor 类
- 女巫不可通过，但是可以使用子弹进行射击

## 14. Key

- 钥匙，继承自 Actor 类
- 在 GameData 中存储了钥匙是否获得

## 15. Box

- 没有放奖励的箱子，继承自 Actor 类
- 用来堵路，强迫玩家去拿钥匙才能通过

## 16. Martix

- 传送阵，继承自 Actor 类
- 碰撞可传送至下一关卡

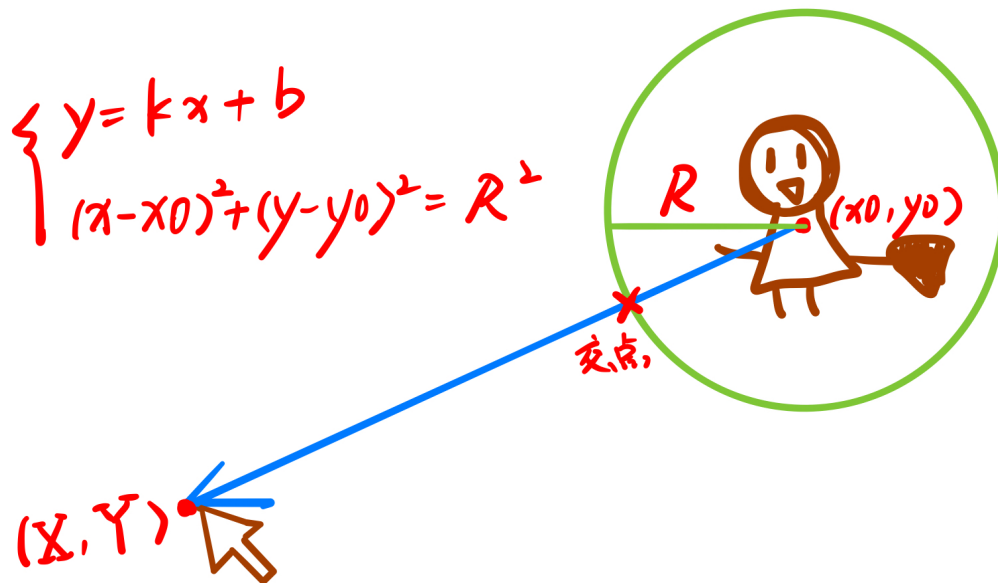
## 17. Thorn

- 地刺，继承自 Actor 类
- 随着第奇/偶次移动的变化间隔出现
- 地刺出现时，路过/站在上方均会减少剩余步数

- 因此，女巫和地刺的碰撞分为两部分：女巫判断的下一步是否会撞到地刺，而地刺需要判断女巫在原地不动时是否会被地刺扎到

## 18. Bullet

- 子弹，继承自 Actor 类
- 碰撞到任意物体/移动距离超过一格都会被销毁
- 可以攻击怪物
- 遇到的问题：
  - 点击任意位置发射子弹，必须使子弹朝向点击位置的同时，移动速度也一致
  - 问题分析：将女巫所在坐标到点击位置看成一条射线，将规定的步数看做是以女巫为起点、步长为半径的圆，则求得他们的交点就是子弹所要发射的方向向量。如下图：



- 解一元二次方程可以得到直线和圆的两个交点，通过判断鼠标点击的方位可舍弃一个交点
  - 将交点坐标与女巫坐标相减即可得到子弹的方向向量
  - 由于只能用 int 进行计算（图片的坐标只支持int类型设置），因此精度上不太精准，子弹方向可能会有一点点偏差
- 关键代码：

```
// 新建子弹
int nowX = 0, nowY = 0;
for (Actor a : allActors) {
    if (a instanceof witch) {
        nowX = ((Witch) a).getX() + a.lenX / 2;
        nowY = ((Witch) a).getY() + a.lenY / 2;
    }
}

// 子弹方向向量()
int dirX = me.getX() - nowX;
int dirY = me.getY() - nowY;
int vx = 0, vy = 0;
// System.out.println("方向向量"+ "("+dirX+","+dirY+")");
// 计算子弹轨迹直线，从而获得其速度

// 直线垂直于x轴的情况
```

```

if (dirX == 0) {
    vx = 0;
    vy = dirY > 0 ? GameData.BulletV : -GameData.BulletV;
}

// 求子弹的【单位】方向向量，就要求：
// 【以人物为原点、单位速度(BulletV)为半径】的圆，与【人物到目标点击处所在的】直线的交点
// 这样才能保证在改变方向的同时，使速度保持一致
// 1. 圆的方程是(x-nowX)^2+(y-nowY)^2=ConstValue.BulletV^2
// 2. 直线的方程是y=Kx+B
// 3.
// 整理得：[x^2-2nowX*x+nowX^2]+[(Kx)^2+B^2+nowY^2+2KBx-2K*nowY*x-2B*nowY]=ConstValue.BulletV^2
// 4.
// 整理得：(1+K^2)x^2+(-2nowX+2KB-2K*nowY)x+(nowX^2+B^2+nowY^2-2B*nowY-ConstValue.BulletV^2)=0
// 5. 通过一元二次方程的系数求X和Y坐标
else {
    double K = (double) dirY / (double) dirX;
    double B = nowY - K * nowX;

    // 一元二次方程的三个系数
    double a = 1 + K * K;
    double b = -2 * nowX + 2 * K * B - 2 * K * nowY;
    double c = nowX * nowX + B * B + nowY * nowY - 2 * B * nowY - GameData.BulletV * GameData.BulletV;

    // tmpX和tmpY是求得的圆和射线的交点
    double tmpX = new GameData().getAnswer(a, b, c, dirX);
    double tmpY = K * tmpX + B;

    // 减去起点坐标才是方向向量
    vx = (int) Math.round(tmpX - nowX);
    vy = (int) Math.round(tmpY - nowY);
}

// 更新速度
Bullet bullet=new Bullet(nowX, nowY, 10, 10, 40, 40, new
ImageIcon(getClass().getResource("../imgs/star.png")).getImage());
bullet.setV(vx, vy);
allActors.add(bullet);

```

## 19. Text

- 文字，继承自 Actor 类
- 显示剩余步数、当前关卡

## 游戏素材来源

- 女巫立绘、菜单、按钮均为本人绘制
- 游戏菜单背景来源于网络图片
- 游戏进行时的背景图及游戏数据均改编自 steam 游戏《Helltaker》，并非本人策划
- 音乐：

- 游戏状态音乐——《Get Your Wish》（作者：Crazy Bucket）
- 其他状态音乐——《Malibu Nights》（作者：LANY），剪辑后的10s前奏循环播放
- 字体：由 <https://izihun.com/?from=sem&fk=1241> 在线网站制作后截图，并抠图后导出 png