

Plane::War 报告

一、背景

飞机类的游戏经常会遇到,从雷霆战机到 B 计划等等,飞机的形式多种多样,其中的子弹, boss 等也各具特色。但是,在我们玩这些游戏背后,它有着怎样的实现原理,通过怎样的手段来达到我们所看到的效果?因此,我希望通过编程实现这类游戏的简单实现,来了解其背后的原理。

二、分析问题

在飞机类游戏中,主要的规则是,玩家通过控制飞机移动来躲避敌机和敌机的子弹,同时玩家可以发射子弹来射击敌机。玩家可以通过吃到屏幕上随机出现的 buff 来增强自己的飞机,比如可以发射两发子弹,可以为飞机添加护盾等,当玩家到达一定阶段后会有最终 boss 出现,有更高的血量,更强的攻击力。

根据这些规则,选择实现其中的部分功能,采用面向对象的原则,抽象出要实现这些功能所需要的类。Plane 类(封装飞机的相关属性), Bullets 类(用来生成飞机类所产生的子弹对象), Enemy 类(敌机类,用来敌机的相关属性), EnemyBullet 类(用来生成敌机类所生成的子弹对象), checkEvent 类(用来检测各类事件), enhanceElement 类(增强道具类), gamePanel 类(游戏面板类,用于在上面添加游戏元素), gameFrame 类(游戏框架类,添加菜单和游戏面板), GamePlay 类(游戏主循环)。

三、解决问题

要实现相关的功能,需要先找到飞机的素材,适合的背景音乐。依据分析问题阶段所抽象出来的游戏所需要的类,构建出每个类之间的关系,考虑到每个类中的许多属性在其他类中都需要调用,将相关属性设置为 public,以方便调用,(有些属性在最初实现时采用了 private)用 xmind 画出思维导图如下。



XMind:ZEN | 试用模式

1. 如何实现动态添加敌机和子弹等元素, 同时还要对其能够删除

采用动态数组的方式来实现，动态数组可以实现对对象的动态添加，可以删除指定位置的元素

2. 如何检测飞机和子弹的碰撞

使用基于规则图形的 `Rectangle` 中的方法：

`public Rectangle intersection(Rectangle r)`，此方法由于是基于矩形，与图片的形状符合，但是在检测时会出现检测不准的情况，因为游戏元素不是规则的矩形，这里采用此方法来简单的实现检测功能

3. 如何读取图片和将其绘制在面板上

通过面板中的 `paint` 方法来实现对游戏元素的绘制

四、设计及编码实现

游戏说明：玩家的飞机最初出现在底部中央，通过方向键控制飞机移动，通过空格键发射子弹。当玩家的飞机被击中时，飞机将重新回到底部中央，当玩家被击中三次时，游戏结束。在此期间，玩家可以通过碰到界面出现的增强道具来强化玩家飞机能力。同时，敌机飞机共有两组，若期间玩家的飞机都未被击中，则玩家获得胜利。

GamePlay 类：

控制游戏的循环和面板的刷新，当在 `checkEvent` 中达到一定条件时更改 `stop_flag` 标志，结束游戏的循环。包括两个主循环，外部循环控制游戏重新开始，内部循环控制游戏结束

gameFrame 类：

作为一个游戏框架容器，用来添加菜单栏，相应的菜单和菜单项和游戏面板类。其中，菜单栏提供一些游戏属性的设置，游戏的说明以及选择飞机等内容

gamePanel 类：

作为游戏的主面板，用于添加飞机，敌机，飞机子弹，敌机子弹等游戏元素，同时完成对这些元素在指定位置的绘制，这个类同时创建好两个标签 `game_stop` 和 `game_win`，并在其中添加图片。当玩家被击中三次时，将 `game_stop` 标签加入面板，当玩家获胜时将 `game_win` 标签加入面板。同时包含一个处理键盘事件的内部类，用于控制玩家飞机的移动。

Plane 类：

用于创建玩家飞机的类，主要包含坐标，飞机图像，飞机速度，图像的宽度和高度等属性。根据面向对象的原则，在 `Plane` 类中还包括对飞机位置更新的方法。

Bullets 类：

用于创建玩家飞机发射子弹的对象，也是属于 `Plane` 类的属性，不过作为一个独立的类会更容易操作和设置。同时，由于子弹对象的创建是由于 `gamePanel` 中处理键盘事件来创建。所以，将其作为一个独立类。主要包括坐标，子弹的宽度和高度，速度等属性。

Enemy 类：

用于创建敌机对象，主要包括坐标，敌机宽度高度，速度，`move_count`（记录敌机移动次数，到达一定次数后发射子弹，每个对象都有这个属性），`enemy_group`（用于记录敌机编组，依据此属性来选择加载敌机的图像，飞机位置的设置及飞机运动模式的选择）。在此类中有 `updatePosition` 方法，它依据 `enemy_group` 属性来更新敌机编组的位置。同时，由于敌机子弹是基于 `move_count` 属性来设置，所以在此类中包含子弹这一属性，表现为用动态数组来存储 `EnemyBullet` 对象。

EnemyBullet 类：

用于创建敌机子弹对象，主要包含坐标，宽度高度，速度等属性，包含更新子弹位置的方法。

checkEvent 类：

用于做各种碰撞检测，因此需要包括玩家飞机对象，玩家子弹对象，敌机对象，敌机子弹对象，同时还要有游戏主面板对象（在做检测时，若玩家飞机被击中，需要通过面板来重新生成飞机对象）（这里不包括边界检测，边界检查在对应类的 `updatePosition` 类中来实现）。

enhanceElement 类：

增强道具类，实现两种增强道具（子弹变为双发，子弹大小加倍）主要包括增强道具图像，速度，移动次数等属性。其中 `move_count` 达到一定次数时，随机选择下一次的移动方向。这里不是现对玩家子弹的更改，而是在处理空格键的事件中依据 `elementCatch` 属性来设置。

最后，在游戏时会出现一些异常，但是不影响游戏的正常运行，所以对其进行捕获，不做处理。

五、编程中遇到的一些问题

1. 对图像宽度高度的获取

最初采用 `Toolkit.getDefaultToolkit().getImage(path)` 方法来获取 `Image` 对象，但是在获取后并不能获得图像的宽度和高度。因为不同敌机对象，不同飞机对象的这些属性有差异，这就给设置他们的初始位置，对他们进行边界检测和碰撞检测带来困难，需要手动每次进行设置。查找相关问题后发现是因为这种方法获得图像后并未将其加载至内存，所以获得的属性为 0。所以，替换为用 `ImageIO.read(new File(path))` 方法来获取 `Image` 对象。

2. 在对飞机移动时刚开始移动会卡顿

刚开始用键盘事件来调用 `Plane` 类中的移动方法来实现移动，这会使得事件的响应比较慢。所以，改为将飞机移动放入 `updatePosition` 方法中，在 `gamePanel` 的 `paint` 方法中将其加入线程，通过键盘事件来改变移动标志为 `false` 或 `true`，以此实现在 `updatePosition` 中对飞机位置的更新。

六、参考文献

1. [爱给（专注免费素材）](#)（游戏素材来源）
2. [创建 Image 图像的几种方法](#)
3. [用 Java 读取图片的三种方式](#)
4. [Java 游戏之碰撞检测](#)
5. [Java 简单游戏开发之碰撞检测](#)
6. [Java API 说明文档](#)
7. [Java 怎么实现音乐播放](#)
8. [关于 Java 中 paint, repaint, update 三个方法的关系](#)