

实验报告：科斯魔

沈祯晖 2010195

1. 设计思路

本次实验的实现中，我希望的是制作一个大鱼吃小鱼形式的游戏，底层的逻辑其实和大鱼吃小鱼是一致的，即为体积的比较，体积大的鱼吃掉体积小的鱼。但是在素材选择上，不像大鱼吃小鱼那样都是相似的鱼类，而是改成了热门游戏“原神”里的生物的截图，以此避免审美疲劳，增加游戏的趣味性。将吃鱼变成击杀怪物。由于大鱼吃小鱼的机制广为人知，因此这里就不再进行过多的赘述，下面主要进行游戏的设计思路的讲解。

这个游戏的整体思路就是分为己方人物，敌方人物和击败或是被击败的判定三个关键进行实现。

Fish 文件：

己方人物的实现是通过 Fish 类来实现的，Fish 类中有五个属性：x，y，Dir，panel，group。

x 和 y 用于确认己方人物的位置，横向位置和纵向位置。Dir 用于识别键盘的输入，也就是确认玩家摁下的是哪个键，同时 Dir 自己也作为一个类存在。

在 Fish 文件里还有 move 类，这一类也调用了 Dir 函数，识别玩家的按键以后，控制己方人物的移动，向各个方向移动距离，再调用 eat 函数判断路径上是否有和敌方人物发生碰撞，然后执行消灭敌方人物或者跳出死亡。

eat 类是用于判断己方人物是否能够击杀敌方人物，如果能则输出击败，并消除敌方人物，如果不能则输出死亡，并结束游戏，然后弹出重新开始的界面。

BadFish 文件：

BadFish 类与 Fish 文件对应，是用来实现敌方人物的，各种属性和 Fish 一致，这里不再做一遍叙述了。

Init 函数用于初始化敌方人物，利用随机数 random 实现随机产生 36 种敌人的其中一种，每次产生敌方单位的数量多少。

Move 函数用于控制敌方单位以固定的速度固定的方向前进。

Hit 类用于判断是否发生碰撞，以及碰撞后比较己方单位和敌方单位的体积大小，如果胜利，己方单位变大并且将敌方单位移除，输出“吸收能量。。。”；如果失败则游戏结束，弹出失败界面，输出“被敌人击败。。。”

Paint 函数就是将各个单位绘制在 panel 上，以达到可视化的效果。

Remove 函数用于将被击败的敌方单位移除。由于己方单位失败则直接游戏失败，所以没有设计己方单位的移除。

GamePanel 文件：

GamePanel 类用于构造面板，构造初始化的相关参数。

生成菜单，创建己方单位，创建敌方单位。在初始化完成之后，启动主线程，开始游戏，并添加键盘事件监听“createKeyListener”

CreatKeyListener 函数用于识别玩家在键盘上按下的按键，本游戏主要有两种操作方式“up, down, left, right”和“w, s, a, d”。分别对应上下左右，

当玩家在键盘上按下这几个按钮的时候，赋予 myFish.move 函数新的 dir 的值，以用于在 Fish 文件中执行。

createBadFish 用于创建敌方单位，而 initBadFish 用于确定初始创建的敌方单位的数量，初始值设定为 12 个。

在这个文件里的 paint 函数用于绘制面板上的所有图形，先是背景图，然后己方单位，再是敌方单位。然后是初始化各个按钮，菜单和菜单选项。

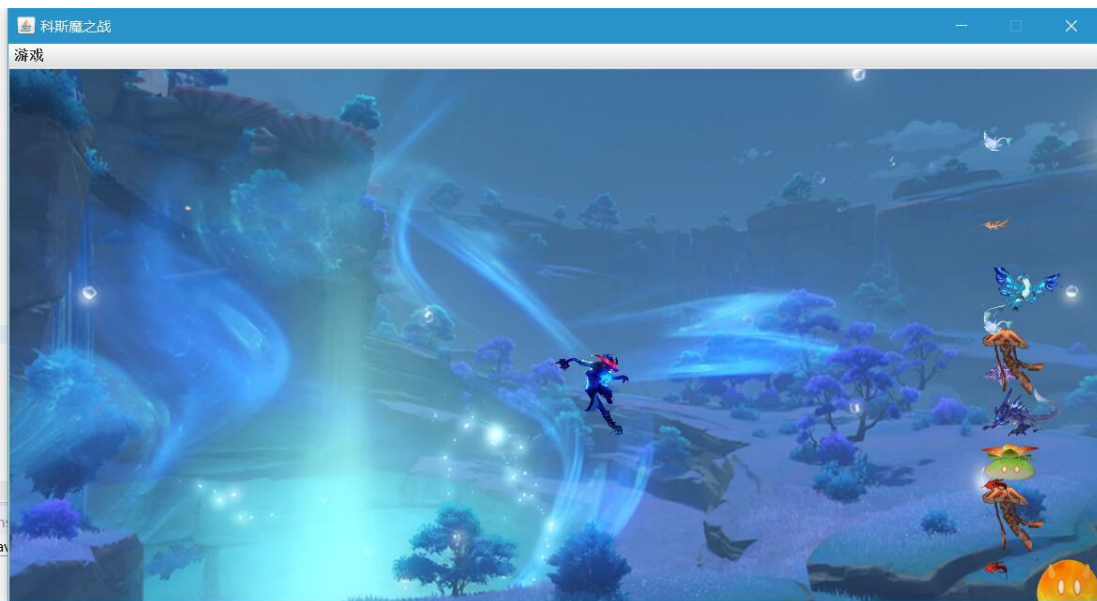
之后在面板上调用前两个文件内的函数，以在游戏中实现功能。

Play0 文件：

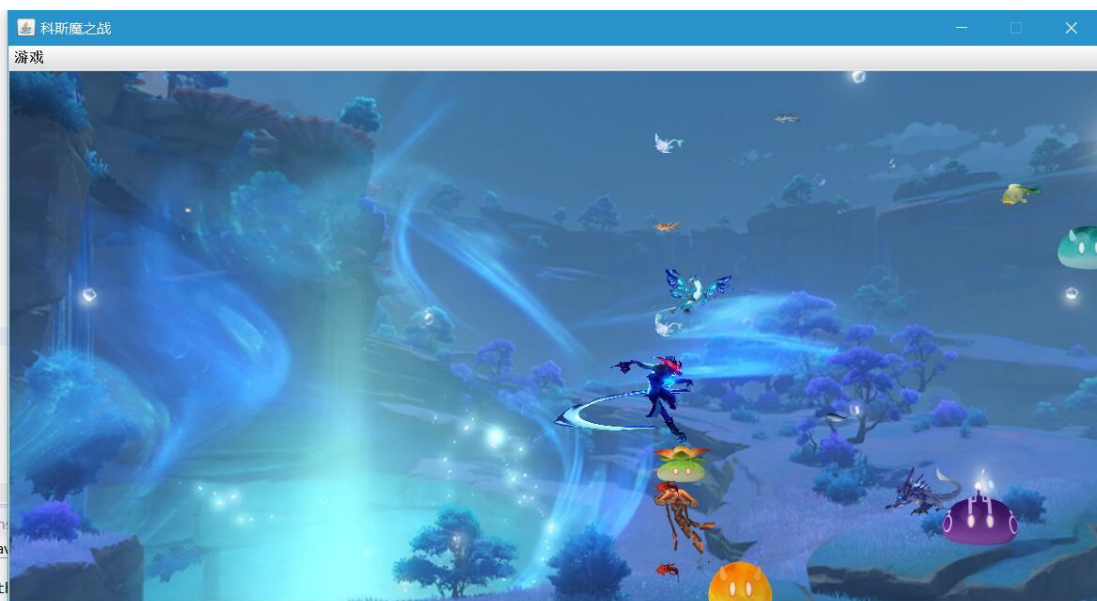
这个文件是用于在开启游戏时播放存放在素材中的音乐，也就是 bgm。

2. 测试成果截图

接触到敌方单位之前，己方单位呈现原形态；

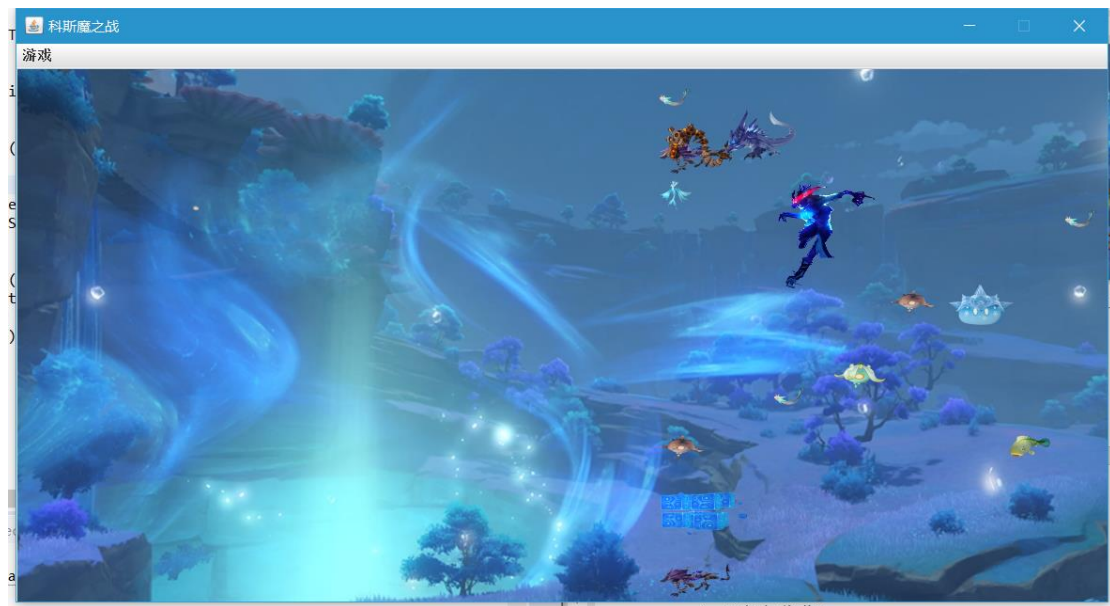


在接触到敌方单位之后会产生一圈光环，以告知玩家击败的发生。



在短暂的出现后光环会消失，再下一次接触到敌方单位之前，己方单位恢复并保

持原形态。



3. 心得与收获

本次实验大体采用了之前有过经验的“大鱼吃小鱼”游戏，但是在此基础上添加了一些新的想法并实现。例如背景音乐的加入，在之前的游戏设计时，并没有考虑到，但是在完成游戏之后，朋友建议我加入一点背景音乐。因此产生了这种想法，搜集了资料以后，发现不仅需要代码，还需要导入 jar 包来实现背景音乐的播放。在游戏中插入了《Seething Animosity 腾溢的敌意》，这首歌是游戏“原神”的原声，由于这次制作的游戏图片素材大多来自于“原神”，因此在背景音乐上也同样借用了他们的原声（致谢米哈游）。不过这次的背景音乐应该是单曲播放，后续如果有可能，我会继续加入暂停和循环播放的功能。

除了背景音乐的导入，在己方单位和敌方单位接触时产生的变化也是新的想法，在完成游戏制作之后，感觉到游戏似乎有些死板，不过是图片的移动与消除似乎太过单调，于是想到制作一些“特效”来增加可玩性。起初是想把特效作为技能来使用，后面发现如果添加新的按键识别并替换图片，很容易对原有的功能产生影响（比如特效出现以后无法消失，或者失去转向功能，永远朝向前方或者后方）。于是想到将这个作为接触特效，也就是在判定接触的函数中添加一个参数来判断胜利与否，并在其之后添加一条根据这个参数判断并进行绘制特效，接触完成以后再重新绘制原形态，以此达到想要的效果。

本次实验是首次利用 java 语言进行游戏的汇编，无论是面板还是逻辑都是一种挑战，完成这次的实验让我对 java 语言的理解更加的深刻并且体会到了编写游戏以及其他 java 程序的乐趣。有很大的收获。