捕鱼达人

1. 资源

Ⅰ 图片资源

①．鱼资源 命名 fish%02d\_%02d.png

②．炮弹资源

分为1~7级普通效果子弹 命名 cannon%02d\_%02d.png

对应7种不同的炮台形态，可有加减按钮所操作(子弹图片 炮台图片 渔网)

③．特殊炮弹资源 强烈弹 特殊效果。会有很大几率穿透鱼而继续穿透，持续捕捉几率会越来越小(特殊炮弹图片 )

④．特效资源

1. 炮弹

**炮弹**分为普通炮弹和特殊炮弹。

**特殊炮弹** 会有很大几率穿透鱼而继续穿透，持续捕捉几率会越来越小

关于炮弹的碰撞，炮台会对应成功几率，子弹会对应特殊效果

**激光炮** 这个的实现是没有渔网的生成的

炮台层 CannonLayer 负责炮台的选择

子弹层 BulletLayer 负责子弹的选择 碰撞并生成对应渔网

渔网层 FishingnetLayer 负责渔网的生成

游戏层

GameScene

碰撞处理，

外部文件

一．动画文件

动画包含在animation.plist文件中。规则

1. fish%d\_run 鱼的运行动画
2. fish%d\_dead 鱼的死亡动画
3. fish%d\_attacked 鱼被攻击后的反映动画(只有少部分的鱼有 作扩展使用)

二．Csv配置文件

Fish.csv配置文件 规则

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Name | Description | Gold | Success | rare |
| 鱼的唯一ID | 鱼的名称 | 关于鱼的描述 | 鱼的价值 | 捕到鱼的成功率 | 鱼的稀有率 |

三．图片文件

1.fish命名

Fish%d\_%02d.png 运动图片

Fish%d\_d%02d.png 死亡图片

Fish%d\_a%02d.png 被攻击后的图片

**四．代码实现思路**

Class Fish

//根据当前的鱼的状态和id获取Animation

Auto animationName = StringUtils::format(“fish\_%s%d”,state,id)

Class Fish

{

virtual void collided()=0;//被碰撞到的回调函数

virtual void caught()=0;//被抓住的回调函数

}