

## 程序功能介绍

WoWG 是一款海战回合制游戏，本游戏有设计较为精良的角色，玩家可对战斗编队进行调整。

## 模块与类设计细节

**worship 类：** 本程序最基本的类，用于角色初始化。

该类包含 QString 变量 kind 与 name，表示角色的舰种和名称，int 变量 hp（血量）、atk（攻击力）、speed（装填，表示舰船在某一回合的攻击顺序）、double 变量 shield，表示舰船装甲类型，bool 变量 ifsunk（判断舰船是否沉没）、extratk（是否有鱼雷攻击），equipment 类指针\*equip，指向舰船当前装备，QMovie 类指针 AttackMovie、MoveMovie、StandMovie、VictoryMovie、DeadMovie、TuozhuaiMovie，分别表示舰船各个动作的 Gif，QLabel 类指针 label，用于角色动画显示，以及 Attack、Extratk、Hurt、Victory、Dead、performStand 函数，用于实现舰船的各个动作。

**mypushbutton 类：** 该类为主界面使用的 button 类，初始化时可指定 button 的图片，并能实现按压动画效果。

其中，zoom1() 与 zoom2() 函数实现按钮的按压动画，两函数通过创建 QPropertyAnimation 对象，设置时间间隔和开始位置、结束位置实现按钮的移动，达到按压动画的效果。

**w\_battle 类：** 战斗窗口及战斗算法类，包含了战斗的主要实现部分。

战斗算法实现了交战双方根据 speed 决定出击顺序，根据编队顺序决定受击顺序。界面的内容包含双方舰队的显示、战斗中的攻击和舰船死亡的过程、攻击和受击的效果、当前进行的轮数和战斗类型、暂停按钮等。

**MyAnimation 类：** 动画播放类，用于战斗过程中攻击和死亡动画的播放。

基本逻辑是先构造循环，并用 connect 设定循环的终止条件，再开启循环，以实现 gif 动画只播放一次的效果。

**pause 类：** 暂停界面类，在点击暂停按钮时使用，包括转到结束界面和继续游戏的按钮，并通过重载 `closeEvent()` 函数实现关闭时转向游戏界面。

**end 类：** 结束界面类，在战斗结束或暂停退出游戏时使用，包含 `QLabel` 显示文字和返回到主界面的按钮，并通过重载 `closeEvent()` 函数实现关闭时转向主界面

**mymenu 类：** 该类为程序主菜单，包含界面与 `mypushbutton` 类，其中开始游戏按钮有指向编队界面的槽函数。

**Button\_Ship 类：** 该类包含成员变量 `int n` (按键序号)、`QLabel` 类 `*label` (按键动画)、`QMovie` 类 `*movie` 和 `*mov` (分别为舰船待机 Gif 和选中 Gif)。

舰船编队通过 `Button_Ship` 类实现舰船顺序调换，该顺序会由 `fleets` 这一 `int` 类型数组传给战斗界面 `w_battle` 并实现战斗编队。通过用于判断点击次数为奇偶的 `ifs` 变量，点击按钮能够实现按钮的 gif 动画切换及按钮间的 gif 动画切换。具体细节为：`upload()` 函数会将点击按钮的序号传入 `si`、`sj` 变量，`change()` 函数会将 `ifs*-1`，`Gif_change()` 函数用于改变当前按钮的 Gif；该类通过手动关联信号与槽，将 `Button_Ship::clicked()` 信号与 `upload()`、`change()`、`Gif_change()` 三个槽连接，实现了点击按钮的 Gif 切换及点击两个按钮角色动画互换的效果。同时，在 `Gif_change()` 中有通过 `ifs` 的 `if` 判断，当偶数次点击按钮时会调用 `Swap()` 函数，完成对与按键序号相同的 `fleets` 数组内部元素的交换，从而实现点击按键调整编队顺序的效果。

**levelselection 类：** 该类为编队界面及关卡选择界面，有四个 `Button_Ship` 类按键实现编队 4 个角色的顺序调整，并有一个 `QPushButton` 按键进入 `w_battle` 界面。

## 小组成员分工情况

江杲坤：程序整体思路，`worship` 类设计，舰船编队界面、算法设计，角色素材提供

杨智卓：战斗算法 (`w_battle`、`pause`)、暂停界面、关卡选择

袁梓乔：主界面

## 总结与反思

本次作业存在初期任务量过重、任务分配不合理、类封装不够简洁等问题。初期计划的船坞界面与装备选择功能已被取消；初期预想的小组多线程工作模式未能完全实现，因为编队界面与战斗界面关联过多，部分细节必须先后实现；又如编队界面的交互设计，初期计划是使用 `MouseEvent` 实现 gif 跟随鼠标移动达成拖拽效果，但该方法实践后发现效果不佳且有小组无法解决的 bug，于是换为使用 `Button` 实现，效果较好，但仍存在 bug。

本项目亮点主要有两个，一是战斗算法，实现了较为复杂的机制与延时效果，同时角色在各种动作时的 gif 使得战斗画面较为美观；二是编队界面，能够实现编队顺序的交换，使回合制游戏有了一定的交互性。

本次大作业也反映出小组能力的不足。战斗界面多线程应用失败，在拖动界面时战斗会停止；使用 `Button` 的编队界面无法实现复杂交互，如计划的点击与长按拖拽；界面美化方面也略显不足。