

# 108-作业报告——Splatank

## 一、程序功能报告：

本小组设计了一个名为 Splatank 的双人坦克对战游戏，灵感来源于任天堂游戏 Splatoon 和 4399 小游戏坦克动荡，游戏中双方坦克生成于地图两侧，在限定的时间内，通过发射子弹填涂地面，互相攻击，同时积攒能量，攒满能量后，可以发出威力巨大的技能，游戏最终以填涂地面面积大小作为获胜的标准。此外游戏还有丰富的音乐音效，独特的美术风格，以及数个精致的动画 CG。

主要实现的功能列举如下：

- 游戏开场动画
- 根据不同场景切换的不同背景音乐
- 坦克射击时的音效
- 主界面：包含开始游戏，规则介绍（游戏指南）以及退出游戏三个选项
- 游戏界面：对键盘事件的相应，刷新游戏状态，记录游戏内物体信息
- 坦克：坦克的移动与发射，与敌方坦克、墙体的碰撞，能量的增加，生命值的减少及回复，脱战状态，复活后无敌时间，大招
- 子弹：飞行，爆炸，爆炸对地图环境的影响，爆炸对周围坦克造成伤害
- 墙体：对坦克行进的影响，对子弹飞行及爆炸的影响
- 计时器：显示本局游戏剩余时间
- 地图：记录地图染色状态，根据地图状态判断最终胜负，地图状态对坦克移动速度的影响，地图改变对坦克充能的影响

## 二、项目各模块与类设计细节

### （1）模块设计

项目总体而言可以分为三个模块，分别是 ui 界面模块，游戏机制模块，游戏场景构造模块。

1、在 ui 界面模块中，主要设计了 introduce.ui 和 widget.ui 两个 ui 界面，分别为游戏介绍和游戏开始菜单，通过信号-槽机制实现了界面间的平滑切换，introduce 界面中给出了游戏操作的详尽介绍和说明，主界面中设计了“开始游戏”，“游戏指南”和“退出游戏”三个跳转按钮，ui 界面设计主要着重于提高游戏的上手性。

2、在游戏机制模块中，程序以 MyScene 类作为核心中枢，接受信号并向各物体发送指令，最后更新游戏内物体状态显示在屏幕上。游戏核心机制借鉴了任天堂游戏“喷射战士”，与传统设计类游戏以击杀作为胜负判断的标准不同，本游戏胜负的评判标准为双方对地图的染色面积，子弹爆炸时既会对周围敌方坦克造成伤害，同时也会对地图进行染色。直观上可认为本游戏中发射的子弹为墨水弹，且敌方墨水会对坦克造成伤害。

在此核心机制基础上，为了增加游戏的趣味性，我们还为坦克设计了能量条机制，每当坦克成功对地图进行染色后，能量条会相应增加，能量满后可发射威力极强的

大招。同时，作为玩家对地图积极染色的正反馈，当敌方坦克处于己方墨水之上时，其移动速度和旋转速度均会降低，利于我方对其进行击杀。同时为了防止“堵家行为”的发生，促进游戏公平有序进行，我们在坦克复活时增设了三秒的无敌时间，期间不会收到任何伤害。同时，在坦克进入脱战状态后，每秒自动恢复一定血量。

为了增加游戏真实性，使游戏符合玩家在现实生活中的直观感受，本游戏中的物体均为实体，继承自 `QGraphicsPixmapItem`。其优势包括但不限于：坦克，子弹，墙之间的碰撞更为精确真实，子弹爆炸喷溅造成的伤害和染色均会被墙体遮挡。

总而言之，游戏为玩家提供了真实有趣的双人坦克对战平台，玩家需要在游戏中寻找染色与杀敌之间的平衡，或迂回牵制，或正面迎击，通过恰当的策略与灵敏的反应取得游戏胜利。

3、在游戏场景构建模块中，本小组设计继承自 `QGraphicsScene` 的 `myscene`、继承自 `QGraphicsView` 的 `myview` 组合来完成游戏场景的构建，在 `myscene` 类中本小组设计了 `keypressevent` 函数，保证了键盘的平滑交互，同时设计了简洁有力的一系列地图判定函数，来检查坦克、子弹和障碍物之间的相互位置，从而保证了游戏在物理模型上的合理性；而在游戏地图的设计方面，本小组通过 `board` 和 `myscene` 两个类来实现，在前者中本小组设计了地图的基本模式和基本函数，而在后者中本小组设计了地图的绘制函数；最后，在 `timeboard` 类中，本小组设计了游戏的记时系统和结算界面系统。

此外，游戏的多媒体系统并没有通过单独设计类来实现，而是通过合理的设计嵌入在各个类之中，以保证代码的条理性和多媒体的及时性。

## （2）类的实现细节

### 1、MyScene 类：继承自 `QGraphicsScene`

- 通过 `keyPressEvent` 和 `keyReleaseEvent` 两个函数及每个按键对应的布尔型变量，记录在某一时间步内按键的状态，并据此向游戏内物体发送信号
- 为了精确表示地图染色状态并比较大小，维护一个二维数组 `map` 记录地图上每一个像素点的染色状态，并据此判断双方染色面积
- 重写 `drawbackground` 函数，令其根据 `map` 中记录的数据绘制背景颜色即地图颜色
- 在 `bombAt` 函数中传入子弹爆炸中心，自动计算其对周围地图和坦克的影响，判断是否会被墙体遮挡，并据此改变游戏内各物体状态
- `myUpdate` 函数，每 10ms 调用一次，其中更新坦克及子弹的位置，绘制坦克新的能量条及血量条

### 2、tank 类：继承自 `QGraphicsPixmapItem`

- 坦克移动：按左右键进行旋转，前后键按当前方向前进或后退
- 碰撞效果的实现：采用先前进再后退的方式，例如当坦克在墙边按前进时，先将坦克向前移动，若判定发生了碰撞，则向后再退一步回到原位，旋转和后退同理。
- 死亡：血量为零时将坦克移出场景三秒，期间血量为零，能量减 50，并禁止移动、发射操作。复活后有三秒无敌时间
- 脱战状态：当坦克在最近 5 秒内既未发射子弹又没受到伤害时，进入脱战状态，每秒恢复一定血量

- 受伤及能量增加：每当己方染色面积增加时，能量都会相应提升。程序记录坦克在每个时间步内收到的伤害和增加能量总和，在 MyUpdate 中结算更新
- 大招：能量攒满后，可发生一枚范围极大的炮弹，但其飞行速度稍有降低
- 速度变化：当坦克处于己方墨水上时速度增加 0.5，处于敌方墨水上时减少 0.5

3.bullet 类：继承自 QGraphicsPixmapItem

- 移动：记录发射角度，每个时间步移动一次，并检测是否碰撞到物体，若发生碰撞则爆炸
- 对地图和坦克影响：爆炸后会对爆炸中心周围一定范围染色，且每个像素点是否被染色为概率事件，概率与该点与爆炸中心距离成反比。同理，对坦克的伤害亦与二者距离成反比，击中时则会一击毙命。
- 受墙体遮挡：染色和造成伤害前，会首先判断两点之间是否有墙体遮挡，若有，则不会对该点产生影响

### 三、小组成员分工情况

王奕丁：

主要负责后端部分内容，完成了游戏地图的设计与绘制实现，设计实现了核心涂色算法（判定是否能染色），协助完成了子弹爆炸染色地图的功能。独立实现了血量和蓝量的全部函数（脱战恢复，被打中扣血），并实现了在游戏页面中动态更新血条和蓝条。为游戏功能开发和系统级优化提出建议，完成一些 debug 工作。

李长烨：

主要负责前端工作，实现了主界面和游戏指南界面的设计和各个界面的平滑跳转，设计了游戏中绝大多数的美术元素，独立设计了游戏的启动动画和所有的音乐和音效，参与了游戏功能开发的讨论，为游戏设计的风格提出了建议，负责了工程的整合工作（包括多次的版本同步和 github 仓库管理）。

周燕居：

领导小组内成员进行开发，主要负责后端工作及游戏核心机制的设计，确立了游戏美术风格，绘制了主菜单背景图概念图，协调组内工作。独立设计了游戏的核心机制，完成了游戏早起框架的搭建，完成了游戏中信号处理核心及显示核心 MyScene 的搭建，实现了键盘控制，物体显示，画面刷新等功能。实现了游戏重要物体坦克类和子弹类中绝大部分功能，包括但不限于坦克的移动、发射、碰撞、死亡、复活、大招，子弹的飞行、爆炸、染色。还实现了计时器 timeboard 类，编写了游戏结束时的结算画面。对组内其他成员的开放工作提出建议，对已完成的代码进行了优化整改，bug 修复，确保游戏能够稳定流畅运行。

### 四、项目总结与反思

本项目至今基本完成了开发前的项目企划，实现并超越了计划书中提到的全部功能。做出了一个具有游戏性，能令玩家获得乐趣和全新游戏体验的双人坦克对战游戏。巧妙利用了 qt 中所提供的功能，没有拘泥于传统图形界面软件，打造出了别具一格的游戏。

项目中仍存在许多值得继续打磨的地方，如物体贴图的美化，游戏机制的完善，进一步优化代码增加游戏的流畅性等。反思未能某些方面未能达到预期的原因，本人认为问题在于立项初期没有很好的认识到 qt 的优势功能，项目没有很好的利用 qt 优势的图形界面设计功能，反而涉足了 qt 所不擅长处理的 fps 类游戏，这导致后期开发时收到 qt 不能满足全部开发要求，程序流畅性差强人意。本项目作为开发成员的第一部游戏作品，是全组人员呕心沥血之作，受限于专业技能的不足，当下不能完美实现预想效果，但是未来随着专业技能的完善，也将持续打磨这款游戏以达到最好的效果，今后在项目开发前也会更加审慎的对待，认清开发软件的优势所在，合理利用软件功能已完成更高水平的项目。