

107 组 QT 作业报告

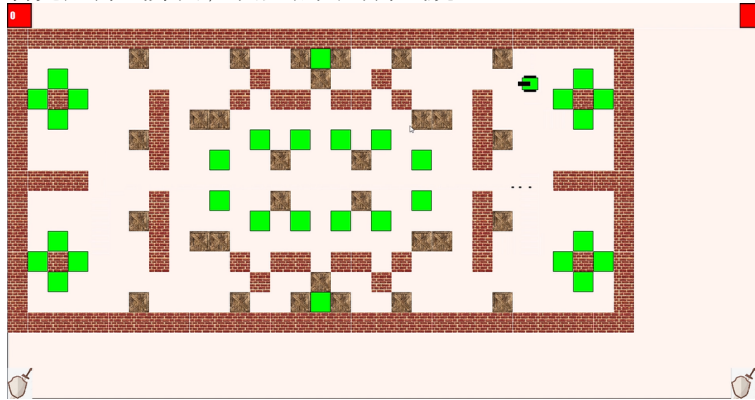
王阳，何润明，杨新叶

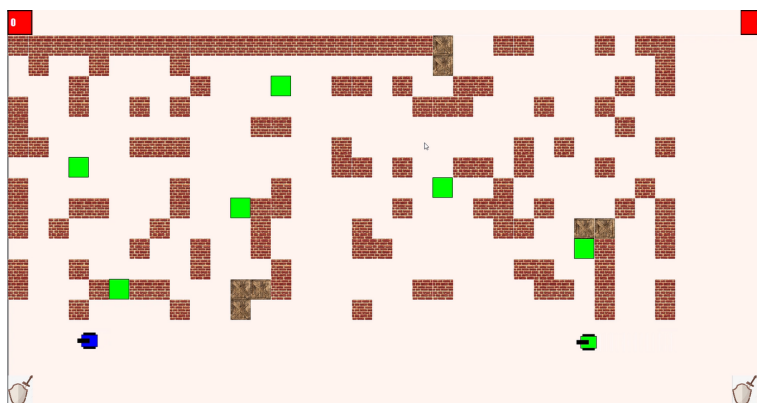
July 2023

1 程序功能介绍

本游戏是一个双人设计游戏，由两位玩家各自控制一台坦克，可进行移动、射击、使用道具、使用技能等功能。为了模拟战场上无法看清敌方位置的现实情况，我们决定将坦克在大部分时候都设置为隐形，只有在射击后的一秒内、以及敌方使用显性道具时才变为显形。该设置增加了可玩性，且增添了游戏过程中的戏剧性。以下是本游戏在不同页面上的功能介绍。

首先是游戏界面，以及相关的游戏规则。





游戏开始五秒内，双方的坦克会处于显性状态，使得玩家了解自己坦克的位置，随后立即进入隐形。

1.1 坦克

有蓝绿两个坦克，一号玩家蓝色坦克可以使用 wasd 移动，按下 c 射击，二号玩家使用上下左右移动，按下 k 射击。



坦克拥有技能，可以给自己套一个保护罩，在三秒内使自己处于无敌状态（可以在观察到子弹快射中自己/两个坦克狭路相逢的时候使用技能）。蓝坦克按 c 发动技能，绿坦克按 l 发动技能。

技能有二十秒的冷却时间，在冷却好后会在屏幕上亮起，否则不会在屏幕上出现。

1.2 障碍物

图片中砖块纹路的方形障碍物是墙壁，会阻止坦克移动且无法被摧毁。

木箱纹路的障碍物则可以被发射子弹摧毁（虽然不知道是怎么摧毁的，可能这个子弹威力比较大），未被摧毁时影响坦克移动。

绿色的障碍物是感应器。在有坦克与之碰撞的时候会改变颜色，变为黄色。如果想隐藏自己的位置，请小心不要撞到感应器。

1.3 道具

道具隔一段时间会出现在地图的随机位置，到达道具位置的坦克即可触发道具的效果。



治疗道具，可以使生命值 +1。



加速道具，使坦克变快为 1.5 倍。



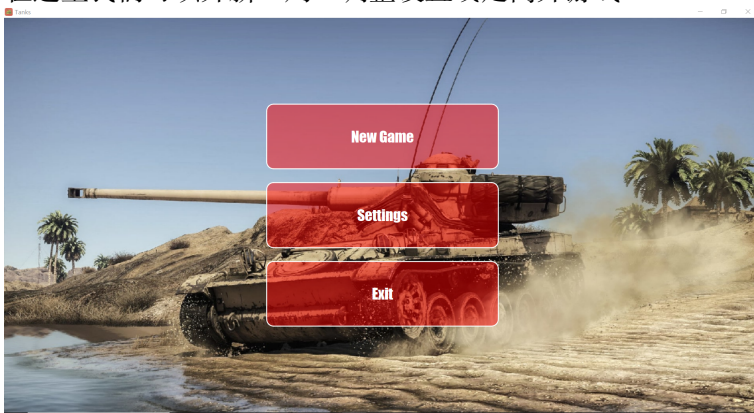
在拾起该道具之后可以穿过任何墙体（有点太强了）。



可以使敌人强制显性。

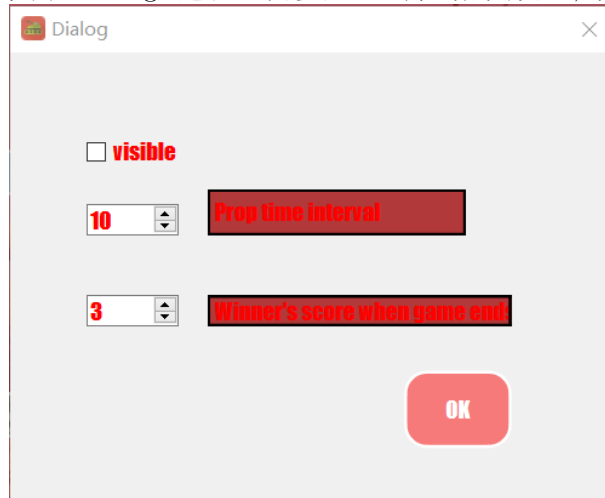
1.4 初始界面

在游戏开始的时候，以及游戏中途按下暂停键，我们都会进入初始界面。在这里我们可以开新一局、调整设置或是离开游戏。



该界面有三个选项：new game, settings, exit。

其中 settings 选项可以修改一些游戏相关设置，如图。



visible 选项可强制使得（建议在第一局游戏前先开 visible 熟悉坦克移动速度）

prop time interval 选项可以更改道具出现的时间间隔。

winner's score before game end 可以改变每局胜利者需要的分数（每次使得敌人生命值归零后胜利者分数 +1，随后立刻开始新的一局）

2 项目各模块与类设计细节

2.1 class tank

两个坦克通过 wasd 和上下左右移动，分别按下 c 和 k 键发射子弹，这两个功能都通过在 mainscene 添加一个 timer 实时更新坦克位置和子弹位置实现。

tank 拥有：生命、移速、方向、显隐 (bool visible)、子弹发射冷却时间、技能冷却时间。

为了实现两个冷却时间，设置了一个随 timer 增加的变量，每次超过冷却时间 T 后允许开枪/使用技能，开枪/使用后归零。

2.2 class bullet

子弹拥有：方向、目标

通过检测子弹与目标的距离判定子弹是否击中目标，击中目标后子弹触发 signal，使得目标受伤（生命值-1）

2.3 class wall

坦克与墙体会发生碰撞判定，该判定阻碍坦克移动，实现坦克不能穿墙的功能（取消该判定则可以实现穿墙道具的功能）

绿色的感应器在发生碰撞判定时变色。

2.4 class prop

道具的生成由两个随机数决定，并且需要避开墙体的位置（如果正好位于墙体位置，则重新生成道具）。

四种道具，在于坦克距离充分近的时候被坦克拾取，立即生效，该功能也通过信号与槽实现。

2.5 其他

class gamemenu 初始界面的显示

class mainscene 游戏界面的显示

class option 设置界面

class skill 技能图标显示

class score 玩家得分

3 小组成员分工情况

王阳：游戏界面（包括设置界面、游戏界面布局以及其上功能的实现）

何润明：道具和障碍物（包括道具的生成、道具功能的实现、障碍物的设计和实现）

杨新叶：坦克和子弹（包括坦克移动和使用技能、发射子弹的实现）

4 项目总结与反思

我们成功地实现了最初设想的坦克对战游戏，并在过程中遇到了一些意想不到的问题。通过反复探索和研究，我们最终通过使用全局变量和槽-信

号连接的方式几乎实现了所有功能。这个过程让我们对 C++ 的特性有了更深入的理解，也增加了我们对 QT 函数和功能的深刻认识。

我们在解决意外的错误时也学到了很多。例如，在实现墙体时，我们尝试将坦克的位置分配给墙体，但墙体始终只处理一个玩家的坦克。这促使我们去了解更多有关槽函数和 C++ 机制的知识，并成功解决了这个问题。另外，我们还遇到了一些无法解决的困难。当我们尝试通过槽函数发送指针时，出现了异常和不明原因的报错。最后，我们采用了将所有信息发送给 `mainscene` 并进行分发，然后通过反向传播和位置比较的方式来替代传递指针的操作。

通过这个项目，我们团队充分体验到了软件开发中的挑战和乐趣。在实现游戏的过程中，我们不仅锻炼了自己的编程技能，还学到了很多关于问题解决和团队合作的重要经验。尽管遇到了一些困难，但我们通过持续努力和合作克服了它们，并最终取得了满意的结果。这个项目让我们更加深入地了解了 C++ 和 QT，并对软件开发的流程有了更清晰的认识。