

# 项目成果和总结

## 项目成果

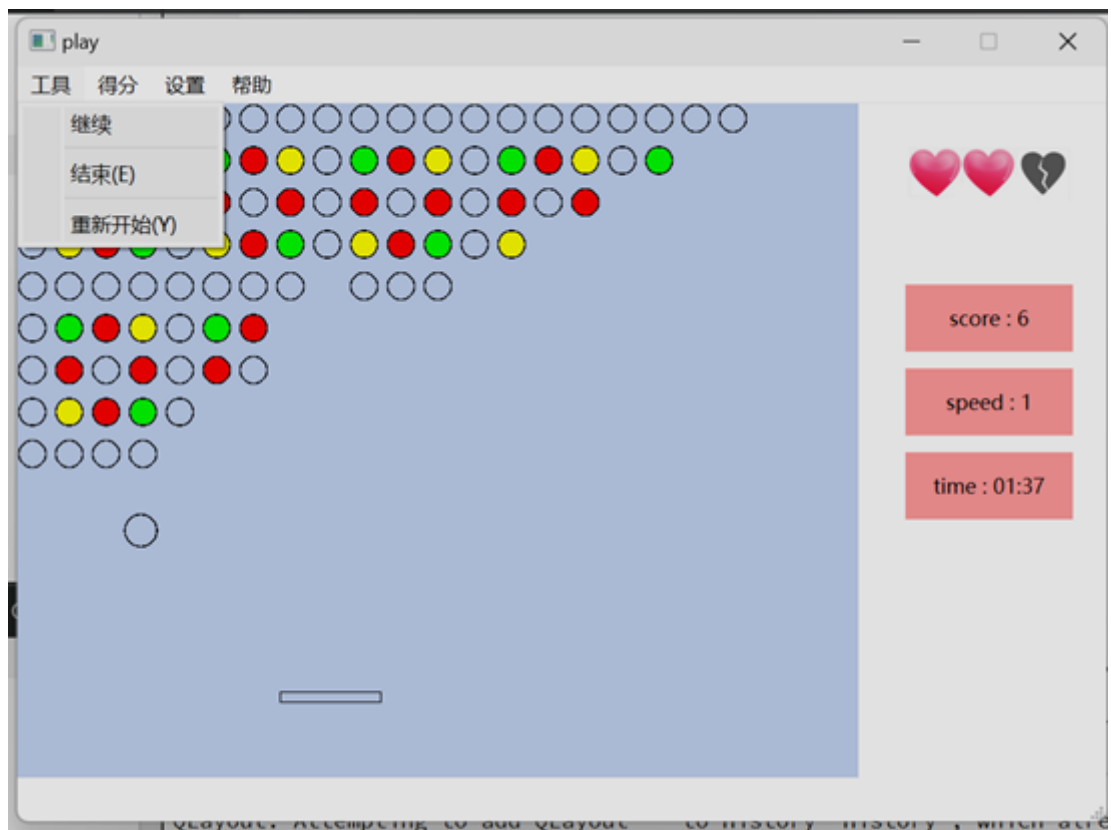
本次项目中，我们完成的是一个反弹球的游戏，游戏中一个小球将以一定速度直线前进，遇到阻碍时将反弹，玩家需要做的就是通过鼠标或键盘操控挡板，使小球尽可能不掉下去并击中尽可能多的砖块。砖块分为普通块和功能块，每个砖块击中均得一分，击中普通块无特殊效果，功能块则会产生对应效果：若击中绿色砖块，则小球加速运动；若击中红色砖块，则小球减速运动；若击中黄色砖块，则小球变大。若小球未被挡板接住下落到边界则视为一次死亡，每局游戏开始提供三条生命，生命耗光则游戏失败。基本的游戏规则如上。



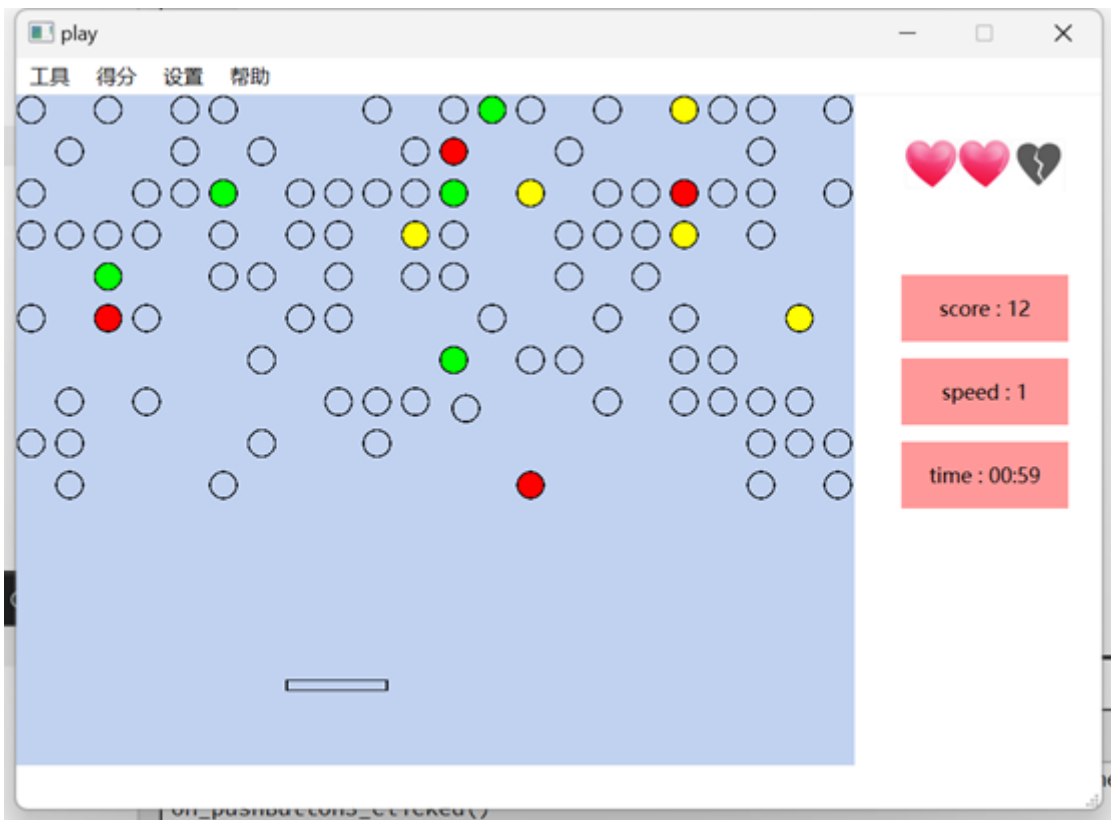
为了丰富游戏的可玩性，我们加入了三个模式，再开始界面点击开始游戏后将进入游戏选择页面，按游戏的难度，模式分为简单、普通、困难三档。



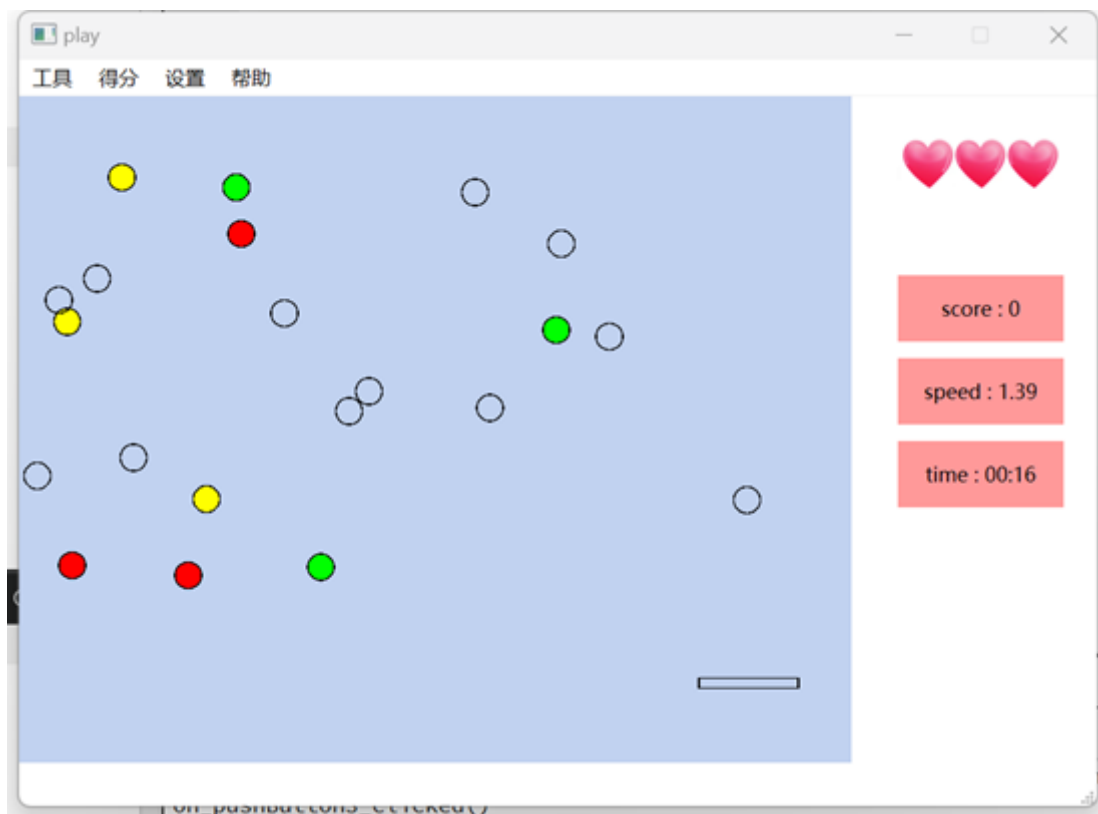
简单模式中，砖块以固定位置摆放，较易击中，剩余生命将显示在界面右上角。菜单栏中我们设置了工具，可以控制游戏的暂停、结束，或开始一盘新游戏。



普通模式下，砖块数量较多，且随机排列，小球速度适中，游戏难度适中。当前分数，小球速度及游戏时长将在界面右侧实时更新。



困难模式下，砖块完全随机分布，且小球速度会随时间变快，砖块剩余一定数量时，将有新的砖块产生，随游戏时间增加，难度变大。




我们还设置了历史分数界面，分数由高到低排列，玩家可以在此查看自己的游戏记录

	Date	Model	Score
1.	2023-07-15	普通	106
2.	2023-07-15	困难	35
3.	2023-07-15	简单	30
4.	2023-07-15	普通	29
5.	2023-07-15	简单	25
6.	2023-07-15	简单	22
7.	2023-07-19	普通	21
8.	2023-07-15	普通	21
9.	2023-07-15	困难	20
10.	2023-07-15	困难	18
11.	2023-07-15	困难	17
12.	2023-07-15	简单	15
13.	2023-07-14	普通	13
14.	2023-07-14	普通	13
15.	2023-07-15	困难	13
16.	2023-07-14	困难	13
17.	2023-07-15	困难	12
18.	2023-07-15	普通	10

## 协作、开发和集成工具的使用

我们使用了 git 进行版本管理，GitHub 进行代码托管，部分协作过程截图如下：

#### Update res.qrc

 Briscky committed 3 days ago


#### Merge branch 'main' of <https://github.com/Briscky/bouncing-ball>

 minini03 committed 3 days ago

#### Update help.cpp

 Briscky committed 3 days ago

#### final version

 Briscky committed 3 days ago

#### Merge branch 'main' of github.com:Briscky/bouncing-ball

 Briscky committed 3 days ago

#### Update help.cpp

 Briscky committed 3 days ago

#### final version

 Briscky committed 3 days ago


#### final

 minini03 committed 3 days ago


#### add game rules

 xxxxueice committed 3 days ago

#### add heart

 Briscky committed 3 days ago

#### Update brick.h

 xxxxueice committed 3 days ago

#### Update ball.h

 xxxxueice committed 3 days ago

#### Update ball.cpp

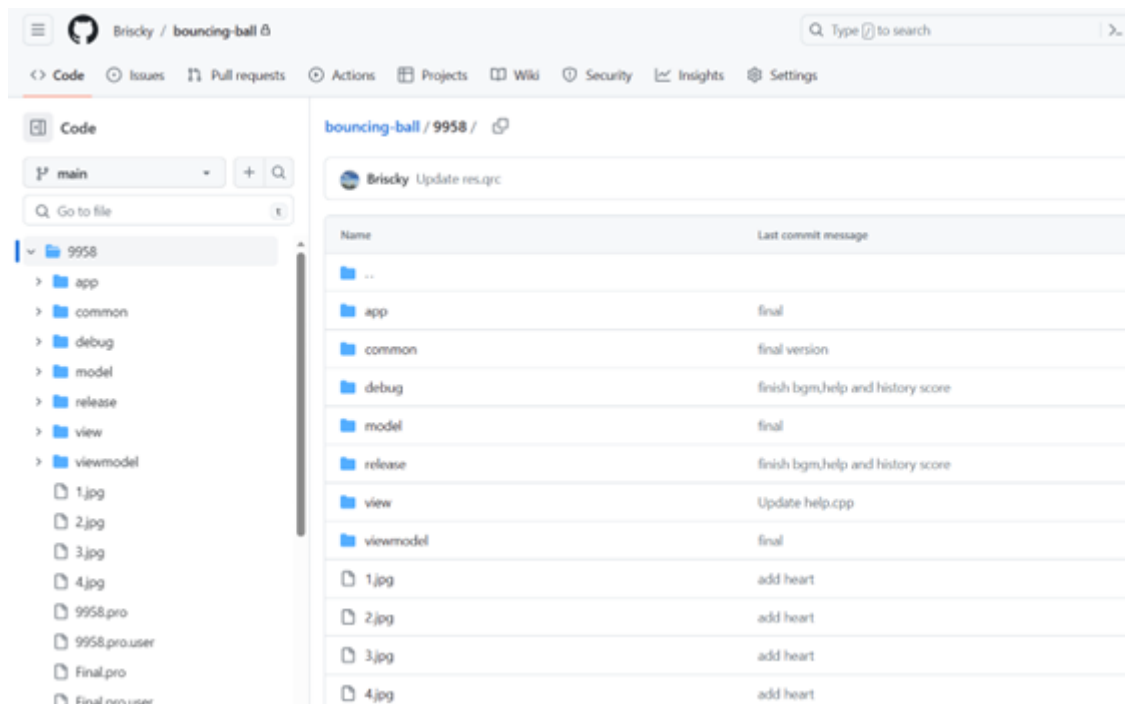
 xxxxueice committed 3 days ago

#### change brick

 xxxxueice committed 3 days ago

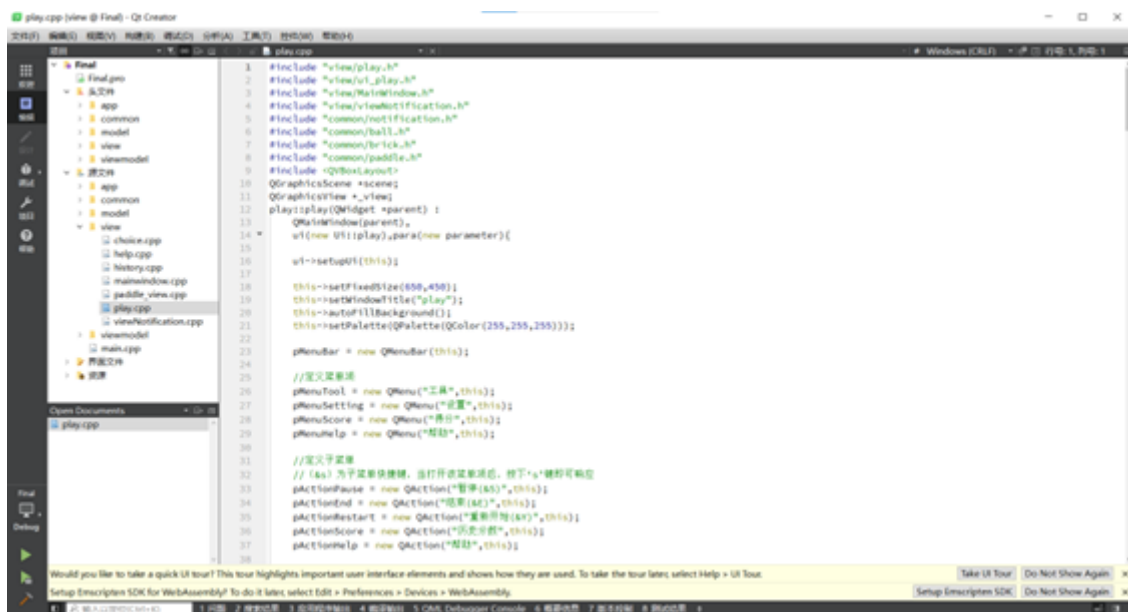
#### add a timer

 Briscky committed 3 days ago

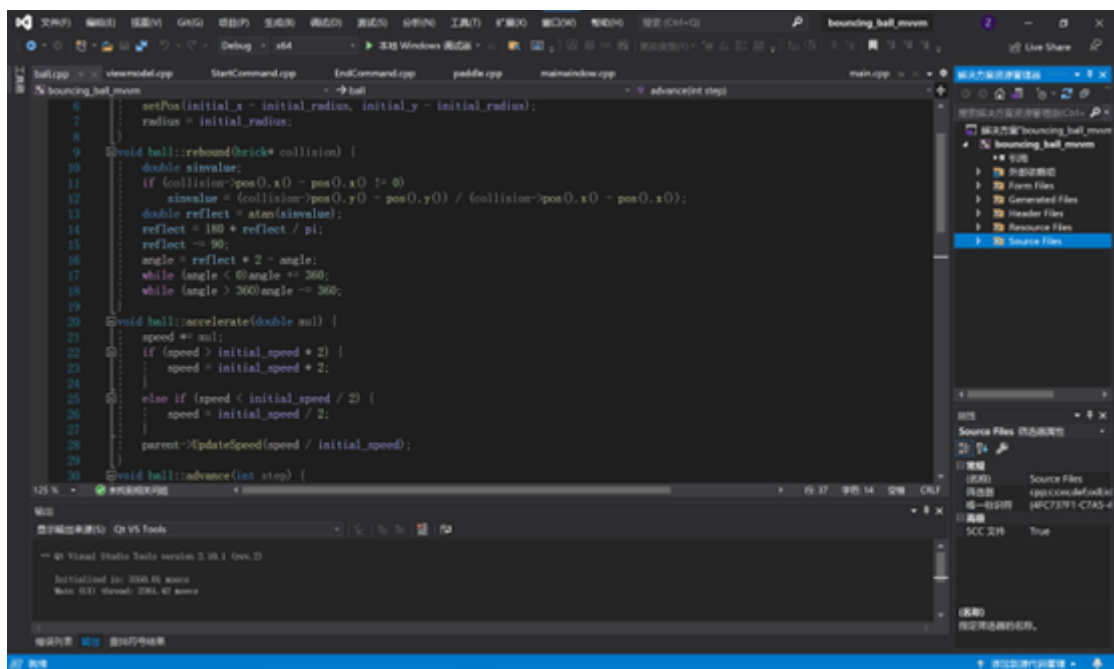


同时我们使用 Qt 和 Visual Studio 进行集成开发：

Qt:



VS:



## 课程总结

本次短学期的课程，我们用工程协作的方式完成了C++的项目管理与实践。在团队合作开发过程中，我们能够高效协作，这得益于：1.开发框架完整清晰。尽管在第一轮迭代之时，我们耗费了大量的时间去搭建和修改框架，但这为后续地开发奠定了坚实的基础。在多轮迭代中，团队成员能够更加清晰精准地优化相应模块，同时进行不同层级间代码的修改，大大提高了项目开发及优化的效率；2.灵活使用多种工具。我们使用了git并依托github进行了项目管理，清楚记录了团队协作开发的过程。而在开发工具上，我们使用了Qt图形库以达到界面设计的需求，同时尝试Visual Stdio与Qt联合开发，不仅提升了编码能力，还进一步掌握了多种开发工具的使用。3.成员分工明确，沟通及时。我们根据MVVM框架的不同层级进行了开发分工，并在此基础上进一步优化，增加了程序的可读性和趣味性。在项目进行过程中，我们会及时交流进度，共同攻克难关，成员们的团队协作能力都得到了提升。

当然我们还有很多需要改进的地方，比如最初时尚未了解框架就匆忙上手编码；程序设计中发现一些原本的想法难以实现，或实现后的效果并不好，欠缺考量；即使是最终版本依然有很多可以增强的功能.....但总的来说，本次的课程体验与团队协作都让我们收获良多，提升了多方面的能力。

## 个人心得

刘姝宁：

在本次课程中我学会了更多项目开发使用的工具，增强了自己的代码水平，也锻炼了团队协作的能力。在本次开发中我主要负责 model 及 viewmodel 层的实现，对于程序后端的逻辑设计有了更清晰的理解与掌握。课程中收获最大的就是对于 mvvm 框架的理解，在使用框架进行开发时，各模块的实现都被分离，团队协作变得更加容易，只需每名成员完成好自己负责的部分，不用过多考虑其他层的实现，大大增加了开发效率。我认为对于框架的掌握将使我在未来的学习过程中持续受益。总之很幸运自己在短学期选择了这门课，也很感谢老师的悉心教导，使我度过了一段很充实且有意义的时光。

高雪：

短学期的课程忙碌而充实。在前几节课，通过老师的讲授我了解了更多 C++ 在实际运用中需要注意的地方，与我们在课内更加看重的知识点理解及编程题不同，它更加注重在项目实践中的开发、协同与效果呈现，在课程后期的项目实践中，搭建框架、进行设计是对课堂内容的回顾与应用，在练习掌握工业生产中协作开发所需的各种工具的同时，既加深了我对框架的理解，又使团队协作顺利且高效。在本次项目开发中，我主要负责 common 及 app 层的实现，完成各层级间数据绑定及资源整合，并参与优化实现程序的多

轮迭代。短学期的课程和实践都让我受益匪浅，提升了团队沟通、协作开发的能力，也加深了对框架的理解，锻炼了编程能力，两周时间收获颇丰。

陈诺：

通过本课程我掌握了工业化协作开发需要的版本控制、持续集成等工具，几种典型的大型软件的框架设计和建构，了解了 C++ 编程陷阱以及学会了优化 C++ 程序的方法。在理论和实践双重探索中，我对 C++ 编程及 MVVM 框架架构有了更加清晰和深入的了解。在本次开发中我主要负责 view 层的实现，学习了 Qt 等工具的使用，锻炼了自己的编程能力，使自己代码更加规范更有鲁棒性。同时界面的设计、与其他层的衔接等也让我对工业化软件开发有了更明确的认知。在与队友的协作过程中，一方面加强了 git 等版本管理工具的使用，另一方面和队友们分工讨论，汲取经验，增进了我与他人合作的能力。

## 课程建议

希望本课程未来能增加一些团队协作开发工具的介绍和引导，理论部分可以将 MVVM 架构与 MVC 等的区别再稍微详细和突出一点。最后是上课是电脑连接教师机有时会连接不上，一定程度上影响到听课体验和效率。