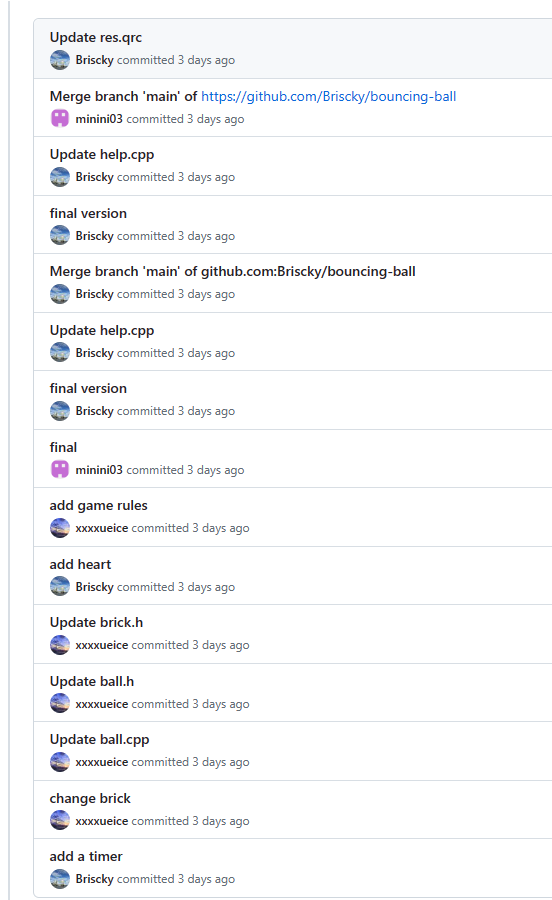
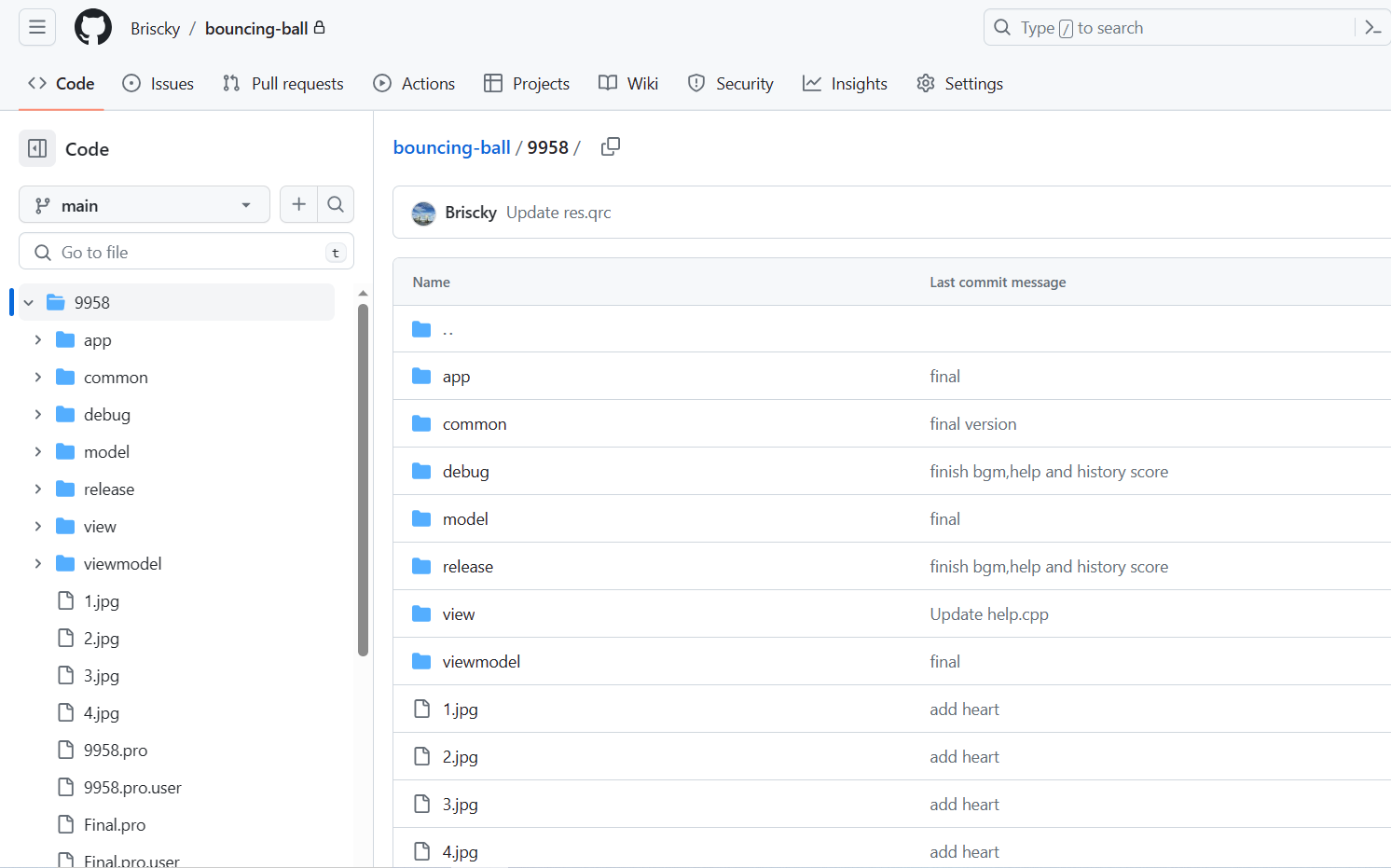
协作、开发和集成工具的使用：

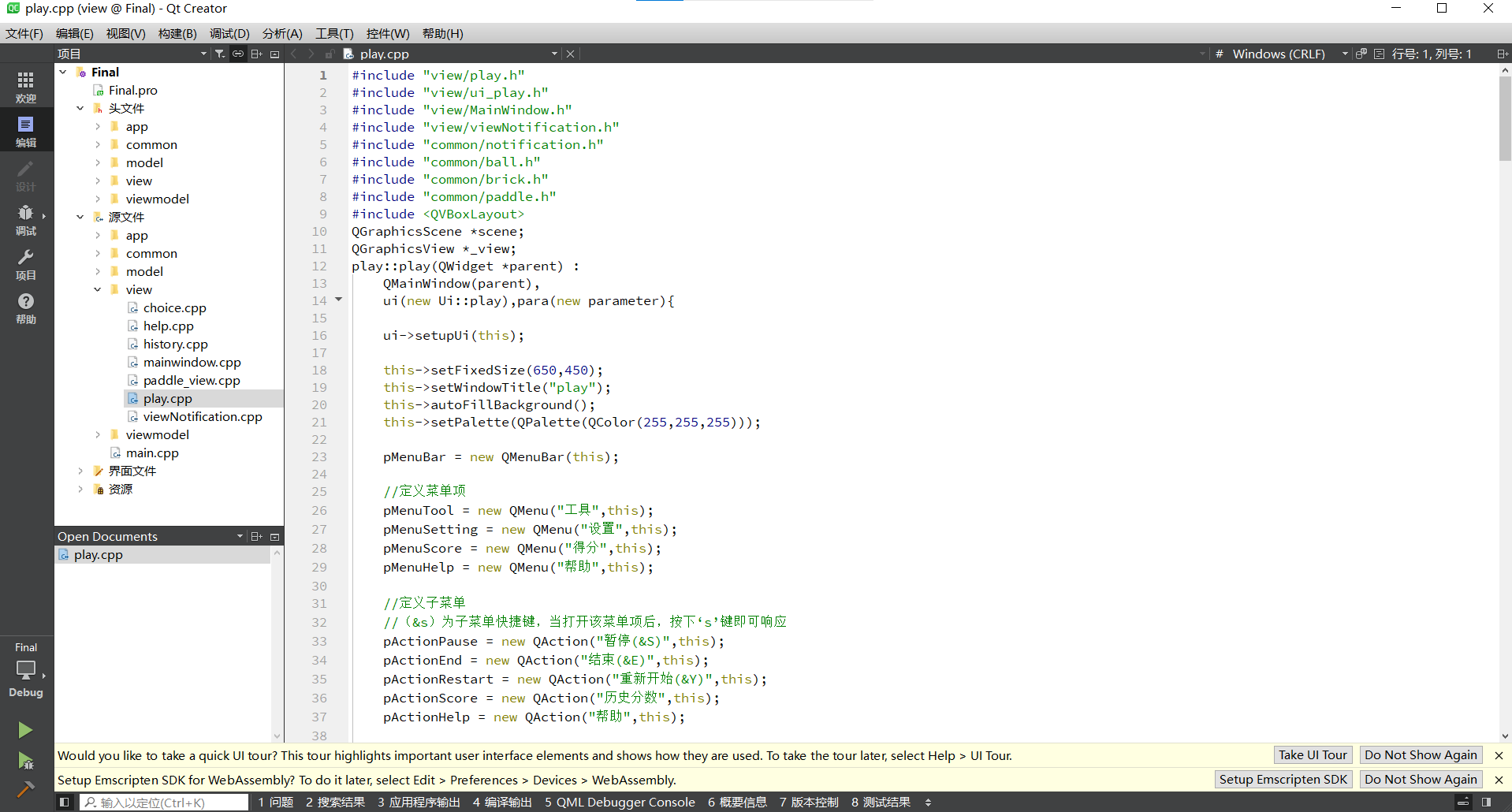
我们使用了 git 进行版本管理，GitHub 进行代码托管，部分协作过程截图如下：



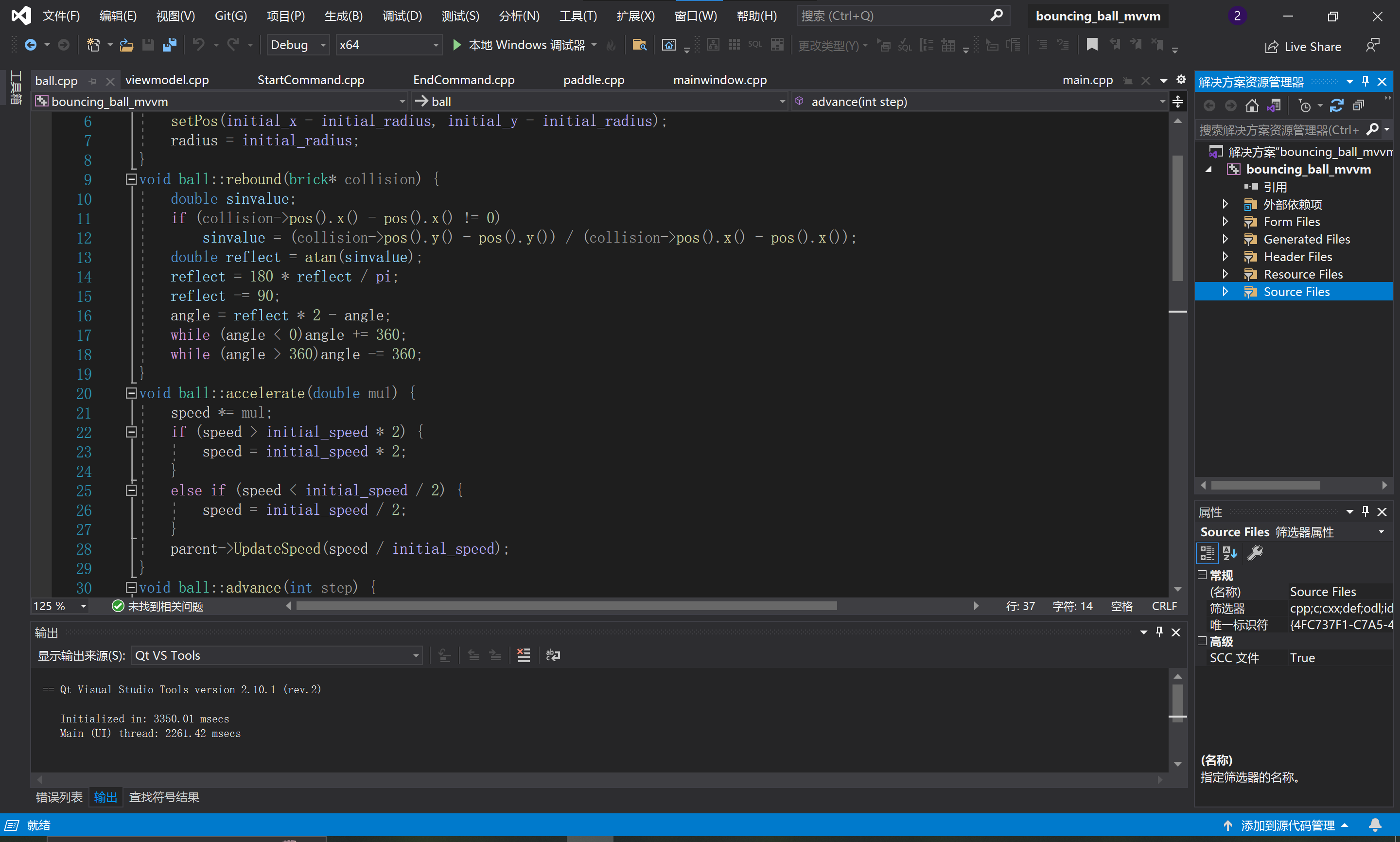


同时我们使用 Qt 和 Visual Studio 进行集成开发：

Qt：



VS：



给这门课的建议：

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。

个人心得：

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。