

中期报告

当前进度

本组实验任务为实现黄金矿工小游戏，目前按照MVVM框架实现基本功能。现在已实现开始界面到游戏界面的转换，并且已完成游戏界面的基本操作，如可实现3个玩家同时游玩，并且单独操控自己的钩子，对矿石抓取的判定，对不同矿石抓取后钩子返回速度的处理、每局游戏矿石的随机生成、倒计时结束后结束本局游戏等等基本的业务逻辑。商店页面的视图已经完成，但还没有和数据层挂钩。

技术难点与解决情况

- MVVM的搭建

MVVM框架指Model-View-ViewModel框架，具有低耦合、可重用、独立开发等优势。在刚开始进行项目开发，本组并未有目的地根据该框架进行开发，导致第一轮迭代的成果耦合性高且代码重用性低，对于后续的迭代造成了极大的困难。为了解决该问题，本组成员向课程班中其他同学寻求帮助，进一步了解了MVVM框架在项目开发中的具体应用和实现；同时，在网络上搜集资料，了解MVVM框架的原理及与其他框架对比的优缺点。在对于MVVM框架有足够的理解后，对于第一轮迭代的代码进行重构，在总体上分为View层、Model层和ViewModel层。其中，View层负责实现用户所接触到的前端UI界面的绘制，Model层负责实现用户交互所产生的数据变化的处理，ViewModel层负责实现View层与Model层的数据封装，实现了View层和Model层的分离。

- QT图形库的使用

Qt是C++图形用户界面应用程序开发框架，对于GUI和非GUI的开发都有独特的优势。在第一轮迭代时，本组使用的图形库为EasyX图形库，虽然该库一直在更新和迭代，但是在UI界面的开发中仍然显得相对落后，具体体现在界面设计的繁琐和功能的缺少。

为了解决该问题，本组成员在搜集C++适配的图形库后进行了比较和讨论，并向一些小组咨询了意见，最终决定使用Qt进行项目的开发。通过对Qt教程的学习和案例示范，将第一轮迭代的成果改成了Qt框架。

- 游戏细节上的实现

一个例子是，将单人扩展为多人游戏的时候看，为了避免多个玩家的钩子对同一个矿石进行抓取判定，在矿石被一个钩子判定时就将其视为不存在，但是这样会导致矿石在被抓住后就在视图层消失，实际数据仍然存在，影响体验。于是将矿石的状态分为三种，正常，被抓取和已获取，解决了问题。

团队协作情况

张柏岩：负责view层的编写，包括页面的视图结构以及元素位置，以及视图间的转换，编写键盘接受指令的处理。

董家伟：负责model层的编写，包括基本数据逻辑的定义和业务逻辑的实现以及对外的访问接口。

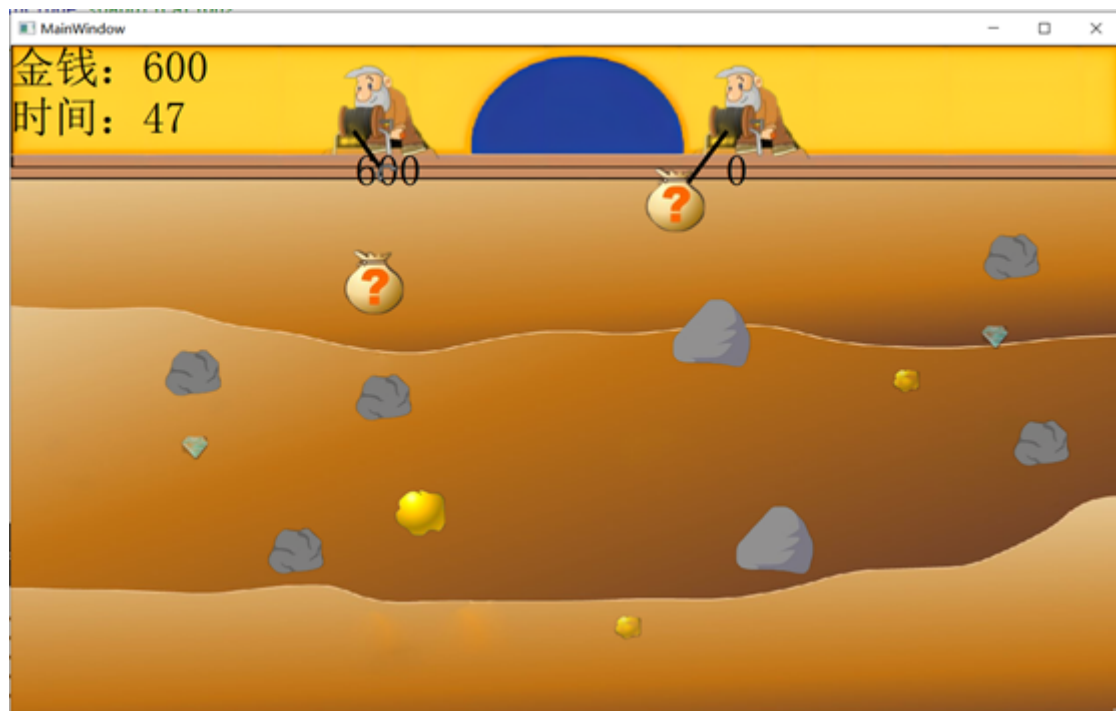
朱骏驰：负责viewmodel层的编写，存储和管理用户界面相关的数据，包括view和model之间的信息转换。

部分成果展示

- 开始界面



- 游戏界面



总体心得

在第一轮迭代中，本组因为贪图一时的方便，没有使用MVVM框架和Qt库，虽然轻松完成了第一轮迭代的成果，但是对于后续的开发造成了非常消极的影响。在添加新的功能的时候，我们发现由于代码的耦合性高，需要对代码多处进行修改，出现了多处bug。由于框架的落后，我们组在分工时也遇到了一定的困难。

在对第一轮迭代成果进行框架重构和图形库迭代后，我们组在进行第二轮迭代中充分体验到了MVVM框架相对于古老的框架的优势。虽然代码量在一定程度上增加了，但从开发的角度来看，三个层面彼此之前可以独立开发，debug的难度大大下降；同时，在进行功能迭代的时候，不需要再对原先的代码进行大面积的修改，极大程度上避免了逻辑上的错误。

在本次项目中，本组虽然走了一定的歪路，但我们对于MVVM框架的理解却更加深刻了，对于日后的项目开发积累了更多的经验和心得体会。

个人心得

- 朱骏驰

本次项目开发的目标为黄金矿工，实现的原理是相对简单的，所以我们在第一轮迭代中没有投入很大的精力在代码的框架上，而是专注于功能多样性的实现和界面的设计。但是在开发的过程中，我们逐渐感受到了代码耦合性高带来的影响，以及EasyX库在开发过程中功能的相对落后。

MVVM框架虽然带来了代码量的增加，但使得层级之间的关联下降，在小项目上可能没有很大的优势，但在大项目的开发中它的优势会充分地展现。耦合度低使得代码的修改不会导致大面积的改动，重用性高使得代码逻辑更加清晰且工作量降低。

本次项目开发为我提供了大量的开发思路和经验，相信对于日后的项目开发有着很大的帮助。

- 张柏岩

此次项目为开发一款游戏，在第一轮迭代及之前，我们对MVVM框架的理解不够到位，导致虽然完成了大部分开发目标，但是后期拓展与项目协作方面出现了较大的漏洞，同时一开始的图形库和项目框架过于古老，致使很多可以优化的地方并没有得到充分的重视，也因此导致了一些问题的出现。MVVM框架作为更为优化的项目框架，其低耦合性和在模块间信息的双向传递性上都有较大的优势，在项目开发中起着较为重要的作用，由于是第一次接触MVVM框架的编写，我们在理解和学习此框架中花费了较长的时间，最终成果也体现出来MVVM的巨大优势。

总体而言，这次的开发让我对现代程序框架有了更为明晰的认识，对于开发协作和程序架构也积累了宝贵的经验。

- 董家伟

其实游戏本身的实现并不复杂，主要的挑战在于如何理解并搭建MVVM框架，使得代码分工合作变得高效。一开始的实现还是古老且耦合性很高的方式，虽然达到了视觉和功能上的效果，但代码可读性低，并且不易扩展。在老师提醒后，开始将程序化为model、viewmodel、view层。我所负责model层实现的是代表真实状态内容的领域模型，并提供数据访问的接口。但在代码整合时我也是发现，MVVM确实带来了层级间关联的下降，总代码量也真实地增加了不少，在我们所进行开发的小游戏这样的小项目来说，体验感还是大于实用感。不过对框架的不断摸索也是为自己提供了开发的思路和经验，绝对会为未来的相关工作提供不小的帮助。