强调一下版本控制的重要性：

1. 每一个小版本只更新了一个或两个功能，那后面如果出错了，排查错误、进行回滚就可以根据message方便的进行；

2. 团队协作的时候，必然会出现合作、共同使用代码的情况，比如我们两个人一同更新mainwindow这个文件，那就极可能出现冲突，像我们都在第1000行新写了一个函数，那版本就无法识别到底最后选择哪个，而当这些情况积累起来之后要处理就很困难而且容易出错了，甚至导致覆盖了别人的代码，而覆盖的情况下没有以往多个小版本的信息又很难恢复，所以合作的时候一定要多点merge，多写message，也就相当于交流沟通了；

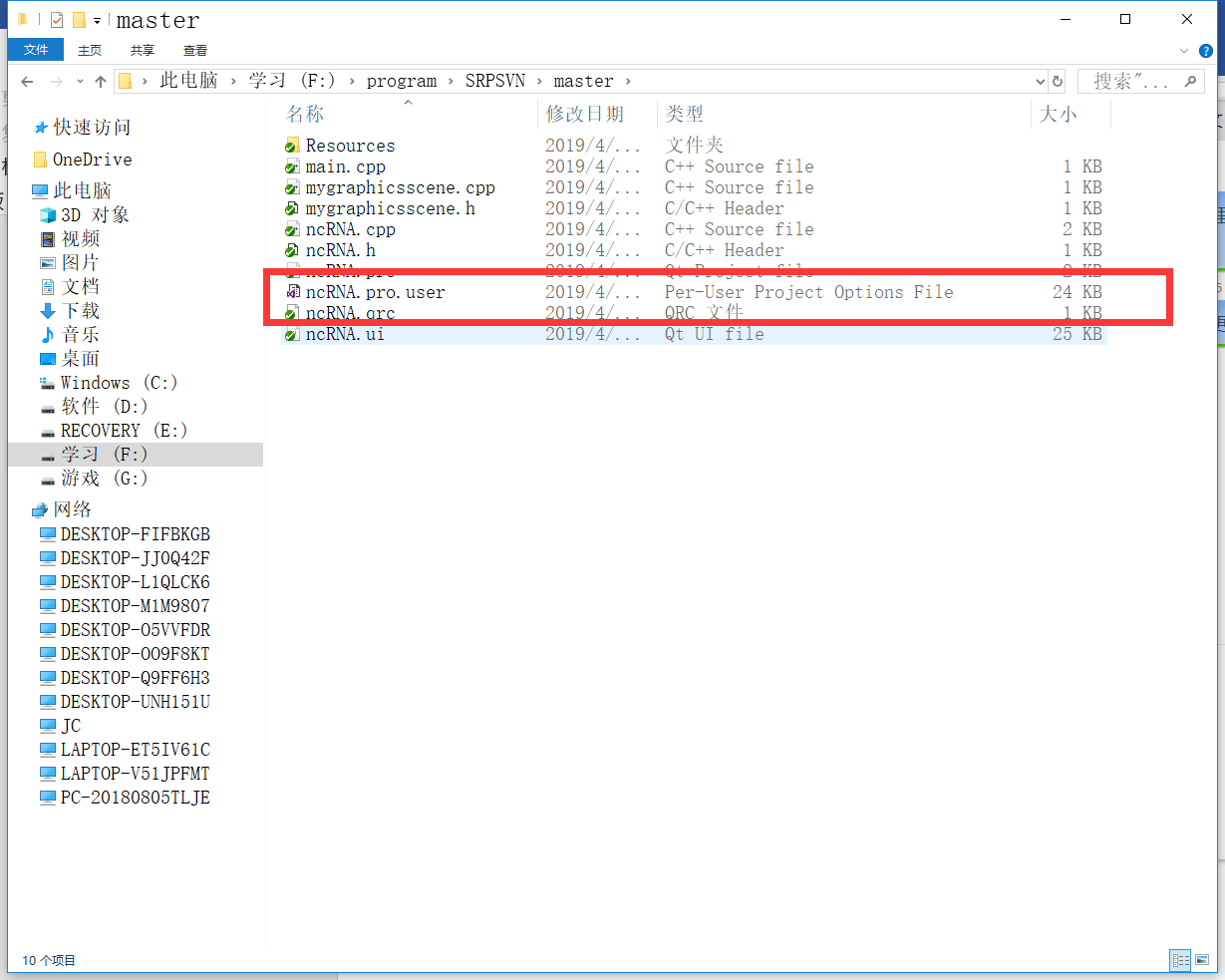
3. 所以正常的工作流程就应该是，工作前update同步到最新进度，每工作一会把阶段性成果先add、commit到本地保存，工作结束后再把所有本地版本merge到master中，然后就可以今晚吃鸡啦。

当然，如果遇到大规模重构、大范围更新，导致短期你的代码暂时无法运行，可暂时不merge到master，但也要及时保存到本地，以及update同步master。

Update、merge的过程其实也是实时地交流、了解团队进度的一个过程、

好啦，下面就是以往常规的操作内容：

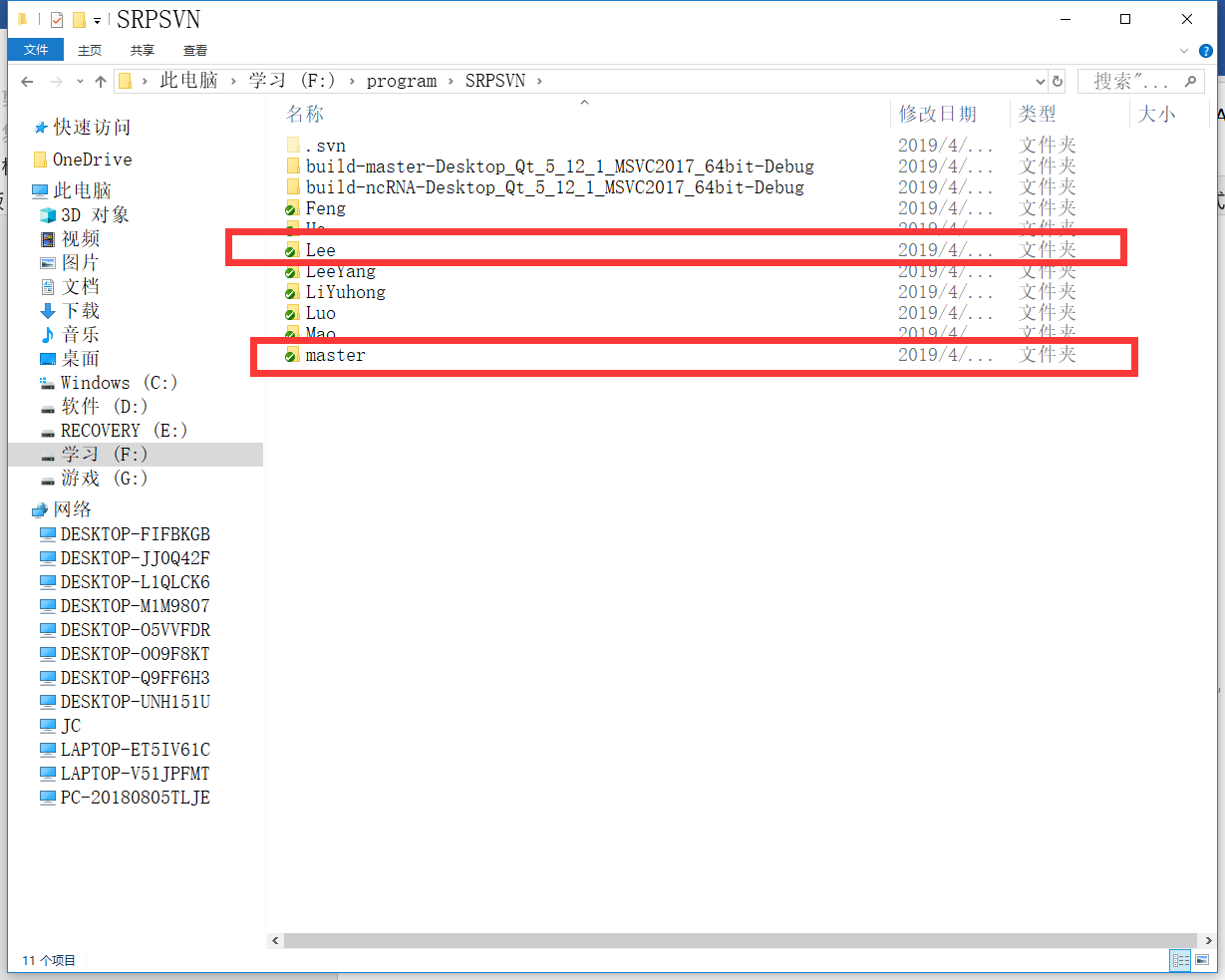
1.我们共享，进行版本控制仅为源代码，请理解这一点。所以编译，调试的部分不要上传到云端，更不要merge到master中。同理，一些不需要的文件，如xxx.user该类文件记录特定用户的属性，不需要进行共享。不需要共享，又出现在仓库目录下的东西添加到ignorelist即可，参考项目说明文档。



可以看到这里的.user是没有绿色下标的 因为我添加到了ignore list 。

2.为了解决跟富文同学分支合并的冲突我把每个人的分支处理了一遍，如果以后发现哪里出现问题可以随时问我进行解决。

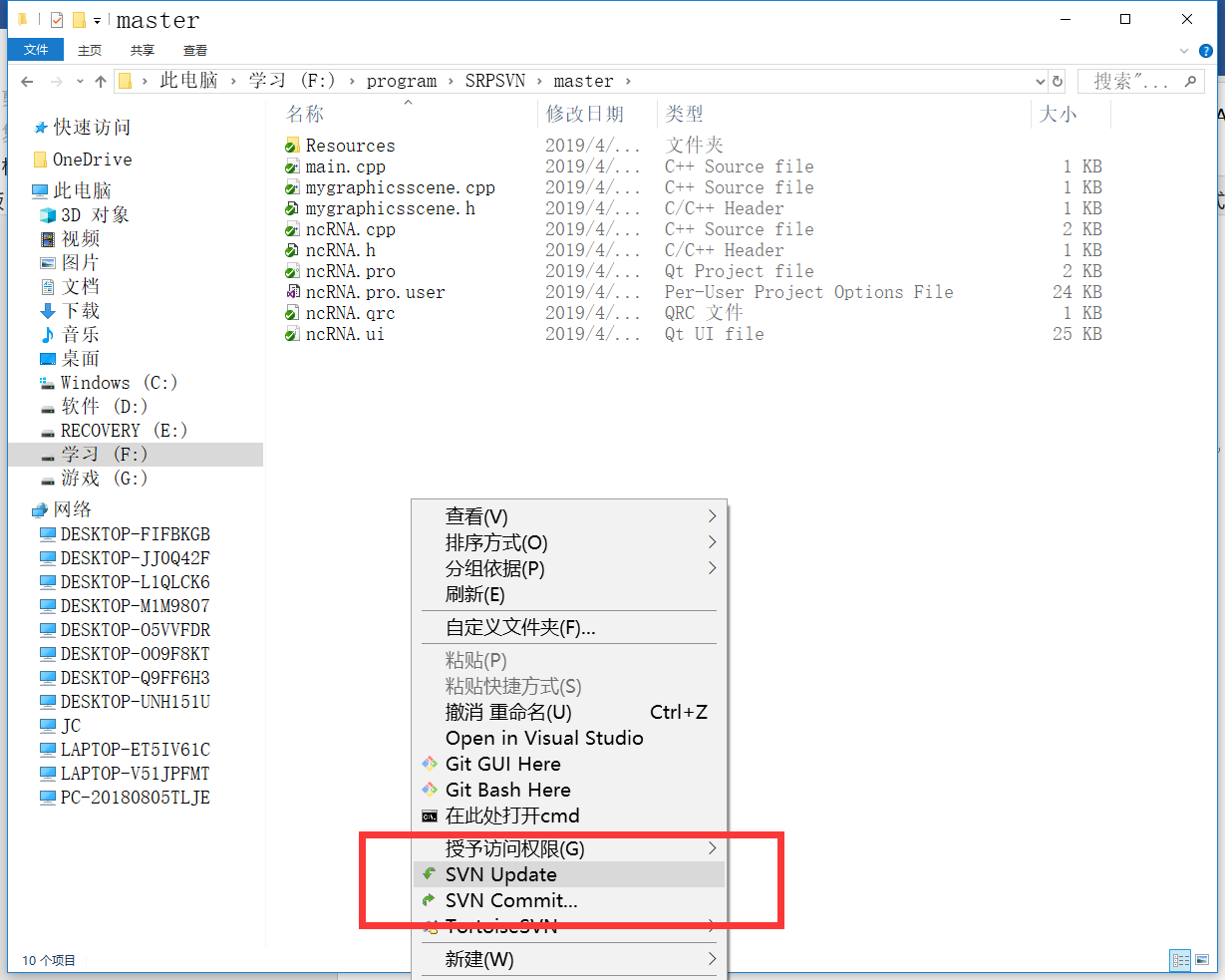
3.为了大家更快习惯SVN和Qt的版本控制流程，我给出一个详细的标准流程，按照这个流程应该就没有问题（项目安装参考上一份文档）：

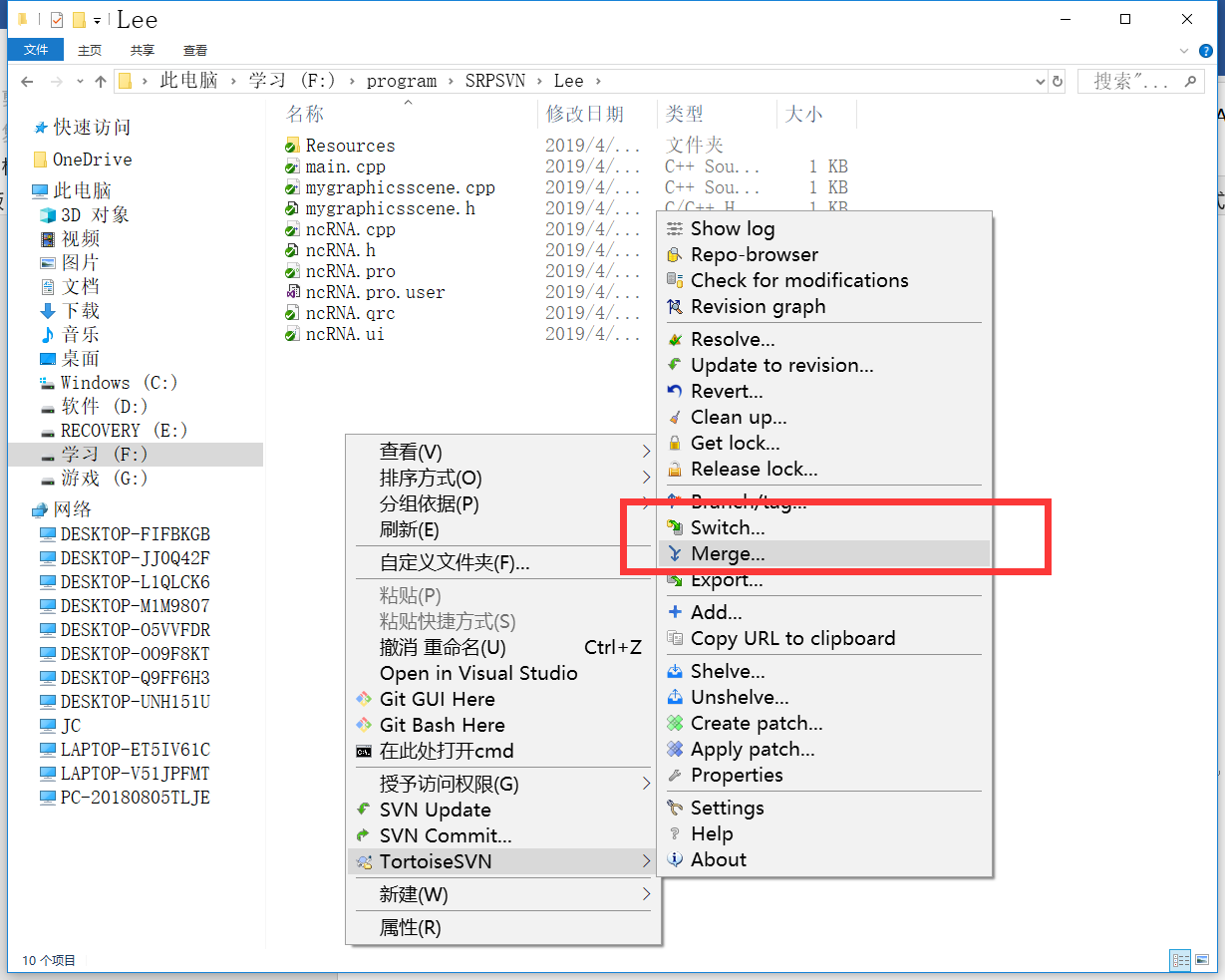
1.

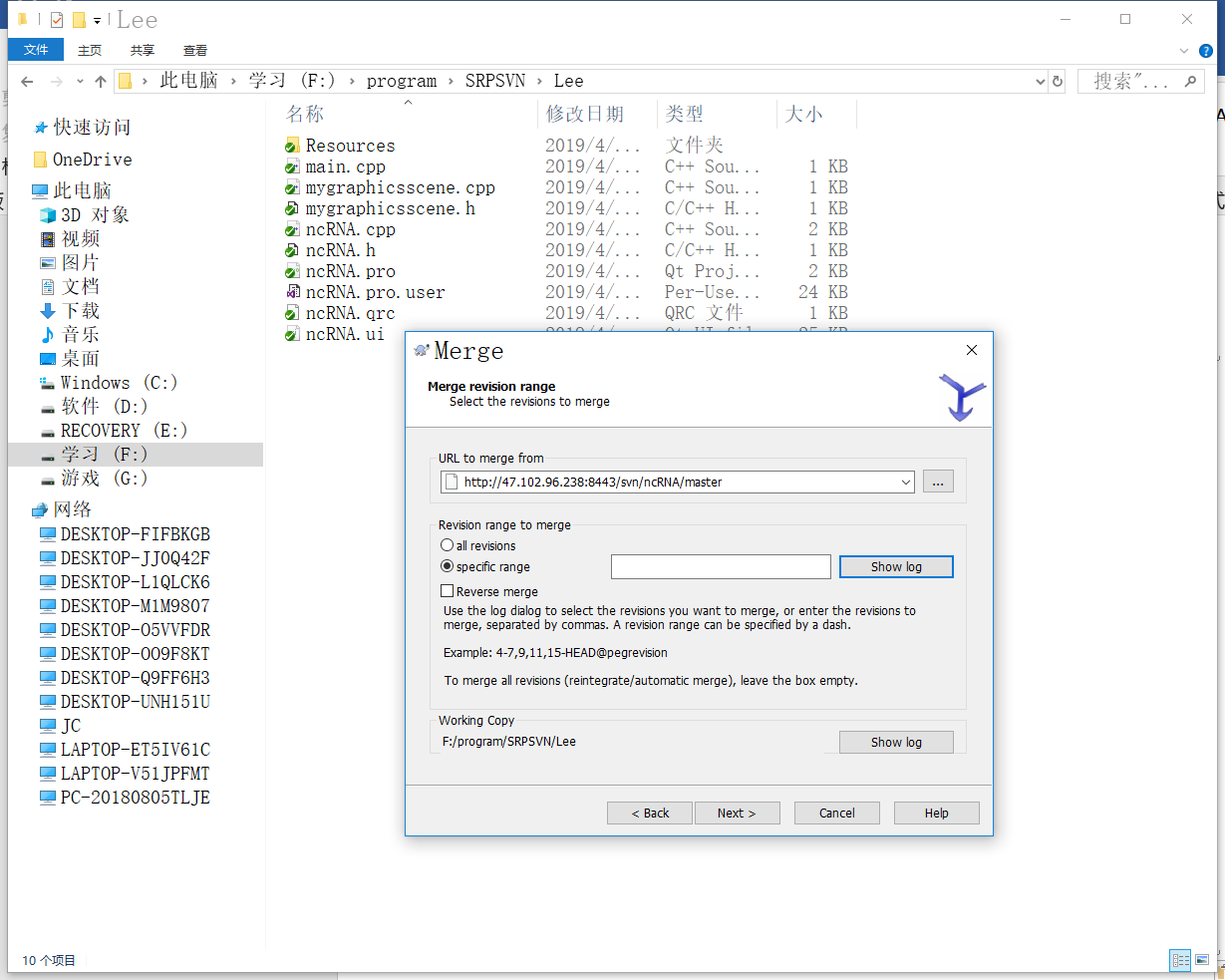
每个人仓库根目录应该至于有这两个文件夹，其他文件夹可以删除，也可以不管。

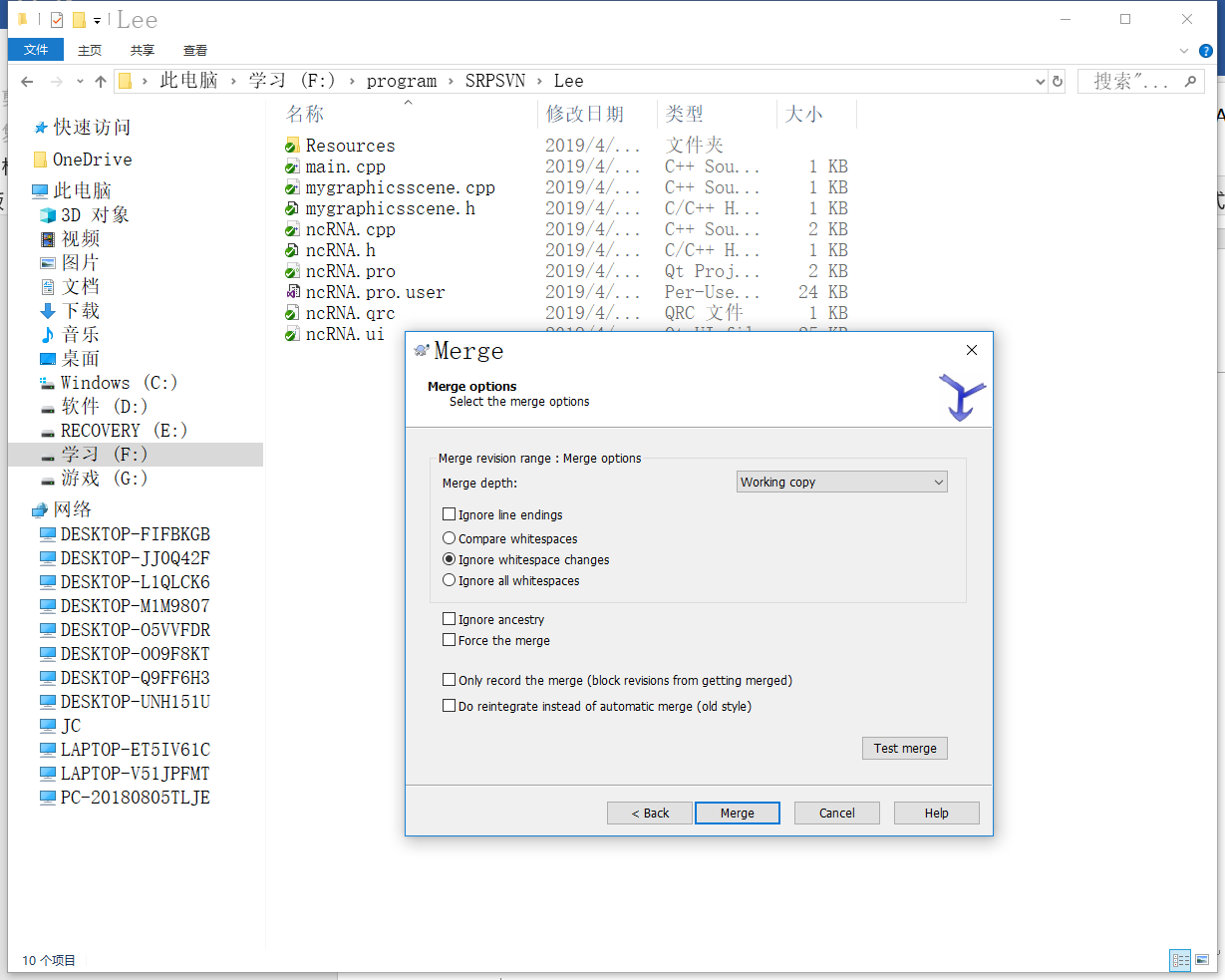
注意，在我们每个人眼里只有两个文件夹，自己的文件夹和master。我们从master同步所有人的更改，再在自己分支上进行修改，最后合并到master上。

2. 每天的工作就是一来先到master update 然后推到你自己的分支 然后在你分支干活 一天结束 然后把你的分支的修改merge 进入 master 大概就是这样



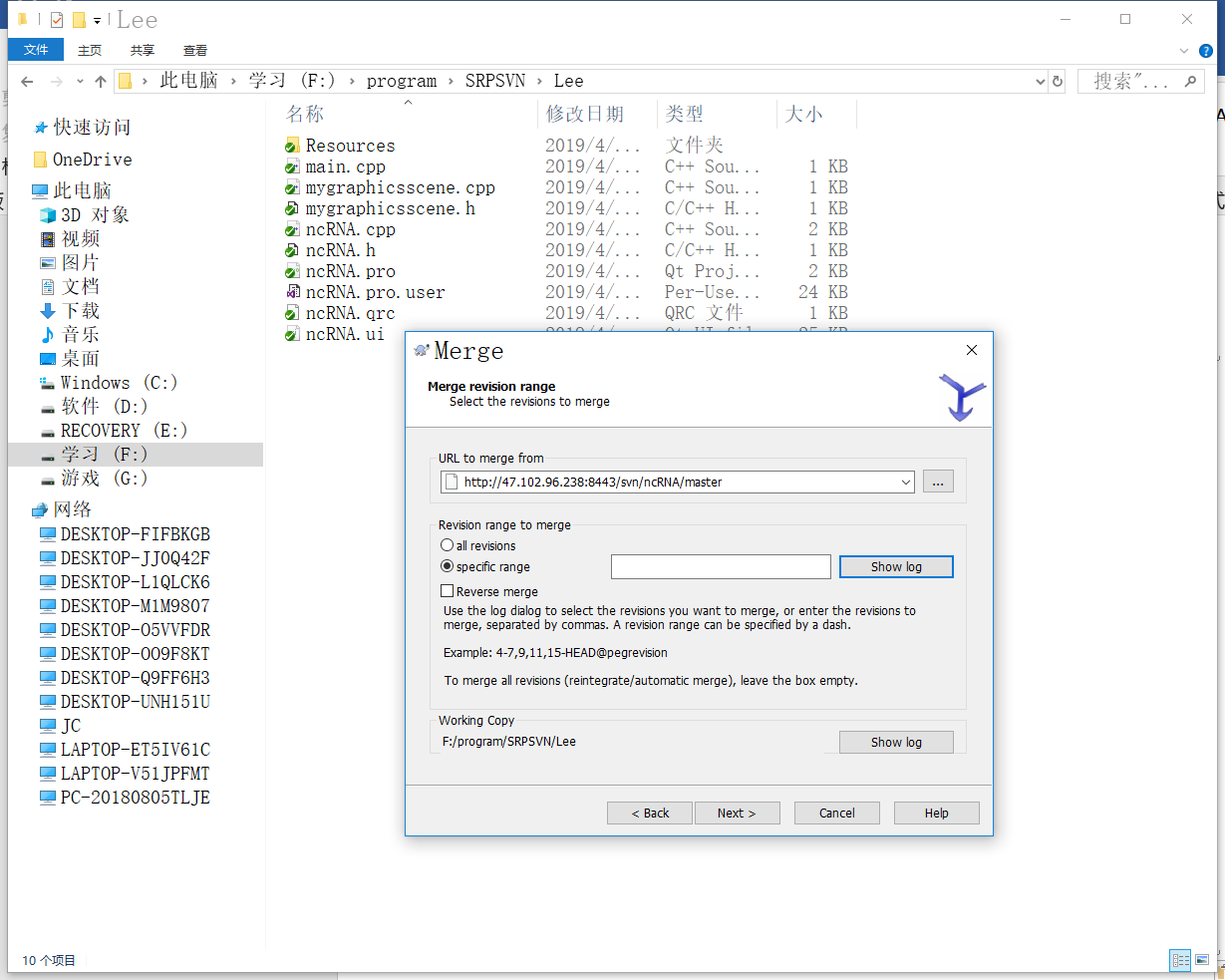






3.关于merge：

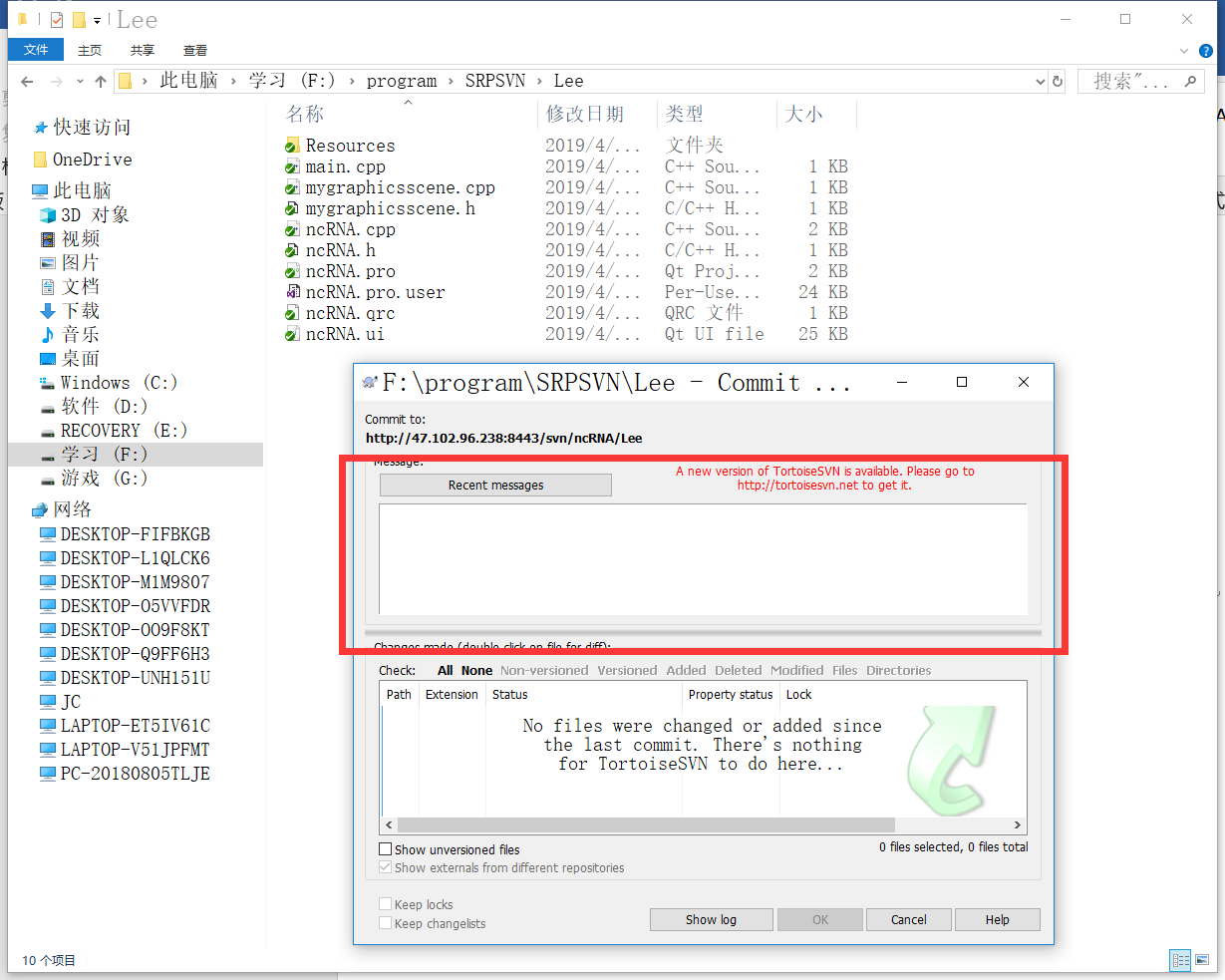
1，3按照图示即可，



选择all revisions会拉取对方的所有版本，specific range则挑选具体的版本。每次我们把一次更改推送到云端，就会产生一个版本。

由于解决与富文分支冲突的时候用了粗暴的复制粘贴，可能有部分人直接选择all revisions时会出现冲突，那每次挑选想要的版本更新即可。如果觉得有问题我可以重新开一次项目，刚开始文件不大也无所谓。

4.注意：请妥善处理冲突！有冲突时搞清楚原因，与产生冲突文件的更改者协商，谨慎处理，切勿把别人的工作全部删除。当然删除了之后还可以恢复，不用过于担心。

请每次Merge到master时填写message，以便大家拉取时能够理解。

5.正式开始工作，（我建议用Qt creator进行编码工作而不是VS，只要下载了Qt都会自带Qt creator，使用VS在目录与组织形式会麻烦很多，涉及大量的ignore）

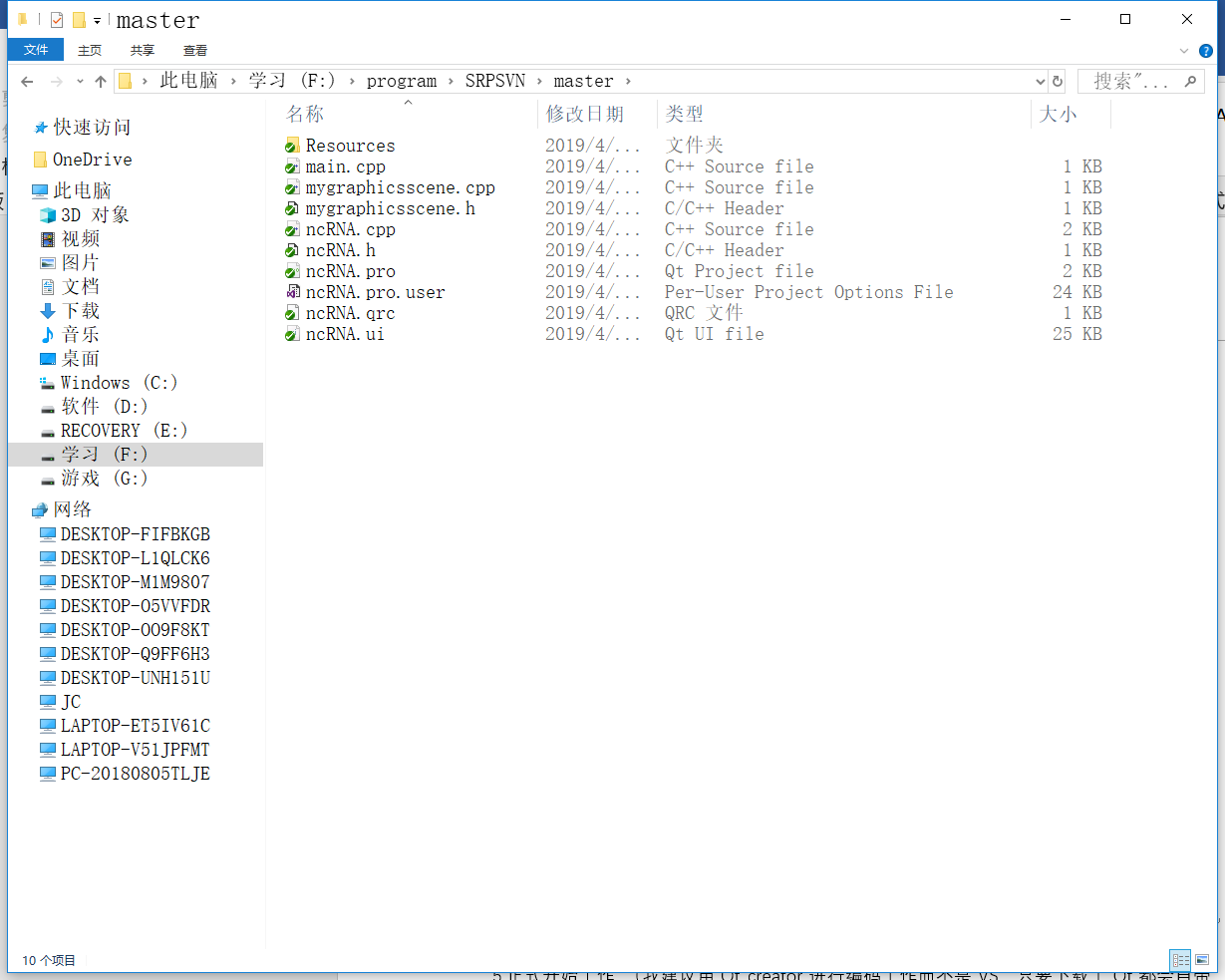
点击.pro文件，第一次打开要配置kit，直接点确定即可，然后qmake,run即可运行。

然后愉快地进行打码即可。运行有错误时，记住clean,qmake,run三连。

完成阶段性工作后即时commit，merge到master。

注意，当添加了新文件后，记得先add，再commit。

正确的样子：



希望大家习惯这种控制方式，多多指教^\_^