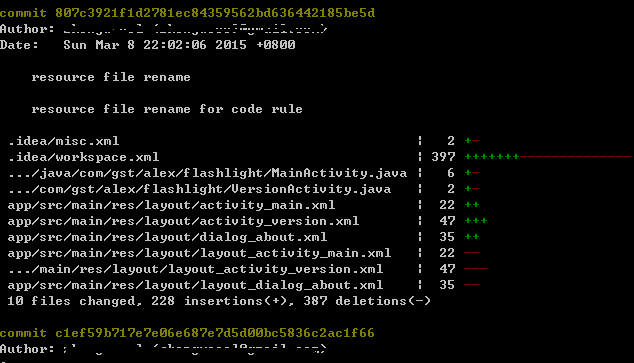
我们以一个具体的实例来熟悉git中log的命令的使用。   
情景：   
1、当前版本的代码编译运行出现了一个棘手的问题，我们打算**查阅最近的提交修改**看是否有头绪：

git log --stat

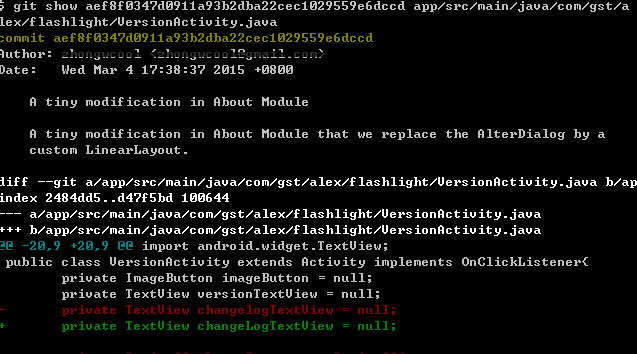
* 1

   
在提交记录我们可以看到每一次commit，都有哪些文件发生了改变，这里简洁的列出了相关文件基本信息。

2、我们打算**查看某个文件在某次提交中的修改**：

git show <hashcode> <filename>

* 1

   
这里的hashcode就是你想要查看的节点的哈希值，也就是图中黄色字体commit后的一串代码。

3、我们分析后发现不是这几次的修改造成的，所以我们希望查看这个文件所有的提交记录。   
**查看仅这个文件的所有历史记录**：

git log --pretty=oneline <filename>

* 1

4、在某些情况下我们可能希望查看目标文件两个版本之间的差异。 查看这个文件\*\*任意两个版本的差异\*\*：

git diff <hashcode-before-right> <hashcode> <filename>

* 1

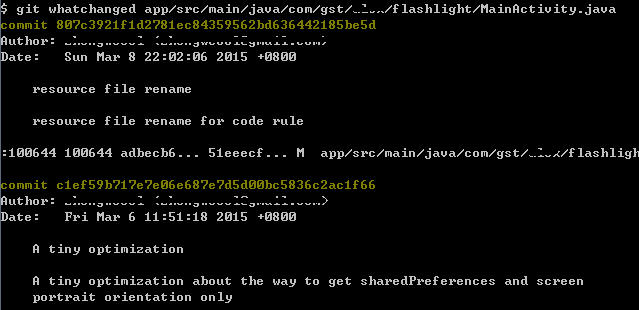
注意：filename在提交记录中的文件路径可能已经被缩略，我们在写filename一定要写上完整路径，但不该是绝对路径。

#### 补充两点

团队开发或者做Code Review的时你可能需要用到：   
查看**某个文件的包含提交人员，日期、版本号等记录信息，不包括修改详情**：

git whatchanged <filename>

* 1

   
在review时查看**某次提交修改详情**：

git show <hashcode>

* 1

也可以用

git log -p <hashcode>

* 1

熟悉如上命令，我们基本可以轻松按照subversion的习惯查阅版本提交和代码修改记录了