

# PaperPass旗舰版检测报告

## 简明打印版

### 比对结果(相似度):

总体 : 15% (总体相似度是指本地库、互联网的综合对比结果)  
本地库 : 9% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)  
期刊库 : 6% (期刊库相似度是指论文与学术期刊库的对比结果)  
学位库 : 7% (学位库相似度是指论文与学位论文库的对比结果)  
会议库 : 1% (会议库相似度是指论文与会议论文库的对比结果)  
图书库 : 3% (图书库相似度是指论文与图书库的对比结果)  
互联网 : 10% (互联网相似度是指论文与互联网资源的对比结果)

编号 : 5AE835937174970VW

版本 : 旗舰版

标题 : 基于Python的互动课堂应用研究与实现

作者 : 魏宏

长度 : 41231字符(不计空格)

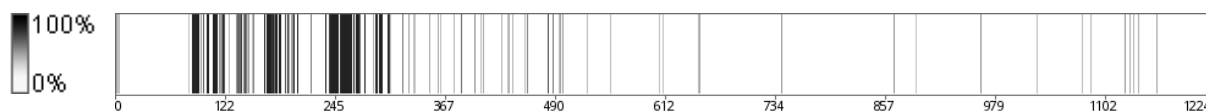
句子数 : 1224 句

时间 : 2018-5-1 17:38:27

比对库 : 学术期刊、学位论文、会议论文、书籍数据、互联网资源

查真伪 : <http://www.paperpass.com/check>

### 句子相似度分布图:



### 本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

- 1.相似度 : 2% 篇名 : 《国内外翻转课堂研究的现状与反思》  
来源 : 学术期刊 《重庆高教研究》 2017年5期
- 2.相似度 : 2% 篇名 : 《TopicQuester: 一种可交互的不同来源话题的可视化分析方法》  
来源 : 学位论文 天津大学 2015
- 3.相似度 : 2% 篇名 : 《基于Django和Python的Web开发》  
来源 : 学术期刊 《电脑编程技巧与维护》 2009年2期
- 4.相似度 : 2% 篇名 : 《Fuzzing工具的设计与实现》  
来源 : 学位论文 北京邮电大学 2010
- 5.相似度 : 2% 篇名 : 《基于J2EE的复杂网络多用户仿真平台实现》  
来源 : 学位论文 华中科技大学 2009
- 6.相似度 : 1% 篇名 : 《某部队保密综合信息系统的设计与实现》  
来源 : 学位论文 电子科技大学 2013

### 互联网相似资源列表 :

- 1.相似度 : 4% 标题 : 《PyQt5整体介绍 - 童小哥总是不开心。 - ...》  
<https://www.cnblogs.com/tongchengbin/p/pyqt5.html>
- 2.相似度 : 3% 标题 : 《转载 Python简介-云栖社区》

- <https://m.aliyun.com/yunqi/articles/41731>
- 3.相似度：3% 标题：《Python 简介 - CSDN博客》  
<https://blog.csdn.net/mrleeapple/article/details/78619986>
- 4.相似度：3% 标题：《python GUI之PyQt\_wx小鱼\_新浪博...》  
[http://blog.sina.com.cn/s/blog\\_aed19c1f0102wjok.html](http://blog.sina.com.cn/s/blog_aed19c1f0102wjok.html)
- 5.相似度：3% 标题：《pyqt5入门教程-土地公生活经验》  
<http://jingyan.todgo.com/shuma/2737056cbq.html>
- 6.相似度：3% 标题：《PyQt5 基本教程 - CSDN博客》  
[https://blog.csdn.net/NCTU\\_to\\_prove\\_safety/article/details/70194529](https://blog.csdn.net/NCTU_to_prove_safety/article/details/70194529)
- 7.相似度：3% 标题：《PyQt5 基本教程 - CSDN博客》  
<https://blog.csdn.net/blldxt/article/details/51691039>
- 8.相似度：3% 标题：《PyQt5模块 - CSDN博客》  
<https://blog.csdn.net/hehangjiang/article/details/78301267>
- 9.相似度：3% 标题：《PyQt5系列教程（1）：我与PyQt5有个约会...》  
<https://zhuanlan.zhihu.com/p/28012981>
- 10.相似度：2% 标题：《Eric6与pyqt5简单学习过程 - 软件开发...》  
<https://www.2cto.com/kf/201711/700441.html>
- 11.相似度：2% 标题：《翻转课堂之来龙去脉：国内外翻转课堂的现状与研究反...》  
[http://www.sohu.com/a/191155218\\_564470](http://www.sohu.com/a/191155218_564470)
- 12.相似度：2% 标题：《 翻转课堂之来龙去脉：国内外翻转课...》  
[http://www.360doc.com/content/18/0222/15/29487391\\_731463265.shtml](http://www.360doc.com/content/18/0222/15/29487391_731463265.shtml)
- 13.相似度：2% 标题：《常州代办银行流水翻转课堂之来龙去脉：国内外翻转课...》  
<http://www.qi-wen.com/html/7676751.html>
- 14.相似度：2% 标题：《Python - CSDN博客》  
<https://blog.csdn.net/zdwzzu2006/article/details/4203445>
- 15.相似度：2% 标题：《 作业一 统计软件简介与数据操作-布布扣-bub... 》  
<http://www.bubuko.com/infodetail-2381187.html>
- 16.相似度：2% 标题：《什么是Python? - 不一样。 - 博客园》  
<https://www.cnblogs.com/ChenNotepad/p/5184206.html>
- 17.相似度：2% 标题：《 Qt5.9 WebEngine 概述-布布扣-... 》  
<http://www.bubuko.com/infodetail-2309290.html>
- 18.相似度：2% 标题：《问题详情》  
[http://wenwen.m.sogou.com/qudetail.jsp?sid=AU0rOkfY9LhH4LshwSnN\\_B7A&g\\_f=15903&qid=266428384&g\\_ut=3&adapted=1](http://wenwen.m.sogou.com/qudetail.jsp?sid=AU0rOkfY9LhH4LshwSnN_B7A&g_f=15903&qid=266428384&g_ut=3&adapted=1)
- 19.相似度：2% 标题：《Python \_ 》  
<http://www.docin.com/p-1253488218.html>
- 20.相似度：1% 标题：《 浅谈“ 翻转课堂 ” 教学模式 》  
<https://www.douban.com/note/481218253/?type=like>
- 21.相似度：1% 标题：《博乐\_新浪博客》  
<http://blog.sina.com.cn/u/1748415530>
- 22.相似度：1% 标题：《浅谈“ 翻转课堂 ” 教学模式\_百度文库》  
<https://wenku.baidu.com/view/6cbaccfaled9ad51f11df22c.html?re=view>
- 23.相似度：1% 标题：《Python的前景怎么样？\_慕课猿问》  
[https://www.imooc.com/wenda/detail/362868?block\\_id=tuijian\\_yw](https://www.imooc.com/wenda/detail/362868?block_id=tuijian_yw)
- 24.相似度：1% 标题：《PyQt类库介绍 - codeAB - 博客园》  
<https://www.cnblogs.com/codeAB/p/5016405.html>
- 25.相似度：1% 标题：《 Python\_PyQt5\_库 》  
<http://www.mamicode.com/info-detail-2147793.html>

全文简明报告:

摘要

{46%：随着计算机技术与网络的飞速发展，起源于美国的“翻转课堂”成为促进国内教学课堂产生变革的新机遇，} {47%：信息技术在教育方面的应用也在极大程度上推动着“翻转课堂”的普及进程。} “翻转课堂”通过课前预习为课上学习打基础，作为主体与学生和教师通过相互沟通共同学习， {40%：在课后可以通过教学资源进行复习，充分调动学生的积极性，实现真正意义上的因材施教，} {44%：帮助学习能力强的学生快速学习所需要的知识，也可以为学习能力稍弱的学生提供学习交流的平台，} 通过教师 and 学生的智慧帮助自己快速成长起来，完成学业所需的各种应试。通过这种全新的教育模式，学生可以发挥最佳的学习效果，高效利用课堂学习时间，释放课后繁重的学习压力，利用合理的时间参与丰富的课后活动，构建一个色彩丰富的学生生涯。

本文详细的介绍了 “基于Python的互动课堂应用研究与实现” 在教师端的设计方案，各个模块的功能分配，并且通过相关的技术分析。本文针对教师端各个功能模块进行详细的论述，包括客户端内嵌网页，画板，截屏以及在截屏的图片上进行相应的操作，对学生进行分组等。本文所介绍的客户端的实现在编程语言的使用上选择Python，并且主要使用PyQt5[1]框架进行GUI编程。

关键字： 互动课堂，Python，PyQt5，录屏，截屏

#### Abstract

With the rapid development of computer technology and networks, the “flipping classroom” originated in the United States has become a new opportunity to promote the transformation of domestic teaching classrooms. The application of information technology in education has also greatly promoted the popularity of “flipping classrooms”. process. ” Flip the classroom” through the pre-learning class to lay the foundation for learning in class, as the subject and students and teachers through mutual communication and learning together, after class can be reviewed through teaching resources to fully mobilize the enthusiasm of students to achieve the true meaning of teaching students in accordance with their aptitude. Students who have strong learning ability can quickly learn the necessary knowledge, and they can also provide a platform for learning exchanges for students with weaker learning abilities. Through the wisdom of teachers and students, they can help themselves grow up quickly and complete the various exams required for their studies. Through this new education model, students can exert their best learning results, make efficient use of classroom study time, unleash heavy learning pressure after class, use reasonable time to participate in rich after-school activities, and build a colorful student career.

This article describes in detail the design of the “Python-based interactive classroom application research and implementation” on the teacher side, the distribution of functions of each module, and through the relevant technical analysis. This article discusses in detail the various functional modules of the teacher, including client-side embedded web

pages, drawing boards, screenshots, and corresponding operations on the screenshots of the pictures, grouping the students. The implementation of the client described in this article selects Python for the use of programming languages, and uses the PyQt5 framework for GUI programming.

Keywords: Interactive Classroom, Python, PyQt5, Screenshot, Screen record

## 目录

### 摘要I

### AbstractII

## 第一章 绪论1

### 1.1问题提出1

### 1.2国内外研究现状1

### 1.3 研究目标与意义2

### 1.4研究方法 with 内容3

#### 1.4.1研究内容3

#### 1.4.2研究方法3

## 第二章 开发平台介绍4

### 2.1开发语言介绍4

### 2.2开发工具5

### 2.3PyQt5框架6

### 2.4本章小结9

## 第三章 需求分析10

### 3.1客户端分析10

#### 3.1.1 网页浏览需求分析10

#### 3.1.2 截屏需求分析12

#### 3.1.3录屏需求分析14

#### 3.1.4画板需求分析16

### 3.2客户端性能需求18

### 3.3客户端运行环境19

## 第四章 系统设计20

### 4.1架构设计20

### 4.2概要设计20

#### 4.2.1 “ 网页浏览 ” 模块的概要设计20

#### 4.2.2 “ 屏幕截取 ” 模块的概要设计21

#### 4.2.3 “ 屏幕录制 ” 模块的概要设计22

#### 4.2.4 “ 简易画板 ” 模块的概要设计23

## 第五章 功能模块的实现25

### 5.1 主界面实现25

### 5.2 主要功能模块的实现26

#### 5.2.1 网页模块的实现26

#### 5.2.2 屏幕截取模块的实现29

#### 5.2.3 屏幕录制模块的实现41

#### 5.2.4 简易画板模块的实现45

## 第六章 系统测试47

### 6.1功能测试47

### 6.2性能测试48

## 第七章 总结与展望49

### 7.1 总结49

### 7.2 展望49

## 参考文献50

## 致谢52

## 第一章 绪论

### 1.1问题提出

随着互联网科技的发展以及“课堂”教育模式的深入人心，关于教育的本质到底是什么的问题越发引起社会的广泛关注， {41%：根据中国教育学会名誉会长、北师大资深教授顾明远先生剖析，我国的传统教育存在以下五大弊端，其一长期以来缺乏对教育本质的认识，}

其二人才观、质量观、学生观的误区，其二轻视职业教育的制度设计，其三学校发展不均衡，其四正确的评价制度的缺失，其五传统陈旧的人才培养模式。那么面对以上这些传统教育的弊端，到底该以怎样的方式，怎样的态度去正视当前的教育问题呢？

{44%：随着翻转课堂的概念传入我国，其先进的教学理念引起众多教育学者以及教育工作者的关注。} {100%：传统的教学模式是以老师课堂讲授为主，布置作业让学生回家复习巩固。} {100%：所谓的“翻转课堂”打破了传统的教学模式，把传统教学模式的教学结构翻转过来的一种教学形态，} {87%：简而言之就是教师先把学生需要学习的内容录制成微课，并发布到互联网上；} {100%：学生在上课前一天可以通过网络在家中或课外观看教学微课视频，视频可以重复播放，方便做学习内容的记录；} {100%：在课堂上，师生可以进行互动交流，并完成课堂练习作业。} {100%：翻转课堂是教学视频和师生面对面的互动以及个性化的交流相结合的学习活动，如运用所学知识解答疑难问题等，} {100%：使所有的学生积极主动地学习，使所有的学生都得到个性化教育，从而达到更好的教育效果。} 由于我国基本国情所决定的教育资源不均衡的问题，在我国实属难以解决的问题，并非通过一朝一夕解决，如果可以通过一个平台使得省市，不同民族，不同习俗的学生都可以享受到国内顶级教师的教学方法，那么在一定程度上可以极大的补齐国内教育资源的短缺问题， {53%：使得不同地区的学生都可以享受到最佳的教学质量。}

由此，为推进翻转课堂在我国的普及进程，以及提高全国学生的学习效率，以“翻转课堂”为理论基础的互动课堂的概念由此而生。

## 1.2国内外研究现状

{72%：自从2007年美国科罗拉多州落基山林地公园高中的两位化学老师——乔纳森伯尔曼和亚伦萨姆斯发明“翻转课堂的教学模式以来”，} 在美国以“翻转课堂”为基础理论的教学模式迅速传播开来，相继影响美国各个大洲的教学模式，在应用层面收到一致好评。{98%：2011年，美国 Lake Elmo小学将 Moodle平台引入翻转课堂，让学生观看完课程视频后，} {100%：在 Moodle平台上做检测题，教师再根据学生的自学情况，组织课堂教学。} {100%：一年后，大部分教师表示不愿再使用传统方式教学，因为翻转课堂得到了学生和家的一致好评[2]。} {98%：除Lake Elmo小学以外，美国高地村小学、河畔联合学区、克林戴尔高中的一线教师也纷纷“翻转”课堂。} 2011年，翻转课堂被《环球邮报》评为“影响课堂教学的重大技术变革”。

{100%：我国已有重庆聚奎中学、深圳南山实验学校、南京九龙中学、广州市第五中学等多所学校实施翻转课堂，} {100%：但从整体上来说，我国翻转课堂仍处在实验探索阶段。} {97%：2013年8月，华东师范大学牵头20所国内著名高中成立了 C20慕课联盟，} {100%：C20慕课联盟官网能为注册用户免费提供2万多个微课视频，} {98%：此外，高校精品课程网、中国中小学教育教学网、超星视频教学网、网易公开课程网、微课网等也积累了丰富的教学视频资源。} {100%：优质教育资源的开放，加快了我国翻转课堂实践的进程。} 于此同时在国内，也有诸如科大讯飞，东师理想等软件开发商开发类似能实现以上基本需求的形形色色的软件， {54%：更是极大的推进翻转课堂在我国应用的进程。}

## 1.3 研究目标与意义

{49%：多媒体教室、学校资源管理利用、教育局网络信息化管理一直以来都是学校、教育局信息化建设的重点，} {58%：通过将文字、图形、图像、声音、资源管理平台集合在一起，采用生动活泼的声像显示以及网络新型资源平台、教育管理平台，} 不仅丰富了教学手段和管理手段，扩充了教学资源和管理资源，而且更提高了课堂教学效果、丰富了教师资源和领导管理信息化。 {100%：充实、形象、生动的授课内容，声像并茂的教学形式、管理形式，



激发学生的学习兴趣和老师的学习热情，} {75%：强调了“以学生为主”的教学新思路，极大得提高了教学质量，颠覆乏味的 PPT授课，} 与课堂互动无缝融合。

本文所讨论的基于 Python的互动课堂应用，仅仅作为互动课堂项目的一个分支，作为pc端，教师可以通过一系列的操作，对整个课堂进行宏观的控制，帮助课堂有条不紊的进行下去。旨在通过此项目，可以帮助教师实现，在本程序上可以通过本软件，直接浏览网页，市面上大多数的产品并不具备这样的功能，{45%：而在实际的调查中发现，不同的学校对此均有需求。} {42%：其二，教师可以通过本程序，对学生的作业进行更改，在教师拿到学生作业的图片后，} 可以加载到画板中，使用画笔对其进行更改，批注，方便拿出来与学生分享。其三，教师可以通过本程序，进行截图，并且可以对截图进行一定的操作，以便在课上与大家共同分享。

通过此项目的开发，可以推进国内互动课堂的普及，让更多的学生可以发挥更佳的学习潜能，使教师资源得到最大程度的利用。{59%：减轻教师与学生的负担，促进学生与教师之间的沟通。} 运用智慧课堂最大的好处就是在学习的过程中，随时跟老师进行沟通和分享，无论从知识和情感层面都需要倾诉，而课堂派不仅提供跟老师同学交流的机会，还能够获得别人的指导和鼓励，让整个学习过程更加轻松。

{55%：互动课堂旨在帮助教师和学生课堂上更加活跃，提高师生间的联系。} {46%：通过互动课堂旨在达成以下几点目标：}

{71%：师生互教互学，相互学习，形成真正的学习共同体系。} {68%：学生与教师的关系是平等的，民主的，整个教学过程是师生共同研讨，论述，丰富课堂。} {86%：在教学的互动中，学生发挥自己的个性和创造能力。}

{68%：师生之间的交流信息面十分广泛，知识，技能，态度，情感，价值观，人生观都得到充分的交流，} {80%：通过这些交流，师生间能够相互沟通，互相影响，相互补充学习，达成在教学过程中成为学生发现问题，} 提出问题，解决问题的过程。

{96%：师生共同参与，互相作用，能够创造性的实现教学目标。} {60%：形成师生之间的合力，促进学生的主动发展，提高教学效率，调动学生主观能动性，达成教学的最优化。}

{78%：通过优化互动教学的方式，即通过调节师生关系及其相互作用，形成师生互动，生生互动，} {62%：学生个体与教学纽带的互动，强化人与环境的交-经影响，通过教学形成在以上关系中的共振，} 达到提高教学效果的一种教学结构模式，真正实现推动我国教学教育水平的发展，减少教育资源不均衡，{41%：对我国教育发展所带来的负面影响，通过强少年强民族，}

{55%：强国家，使得中华民族屹立于世界大国之林，为人类发展做巨大贡献。}

## 1.4研究方法内容

### 1.4.1研究内容

本文主要内容是基于Python[2]的互动课堂的应用研究与实现，主要目标实现以下几个工作：

{42%：（1）调查并研究国内开发类似产品的需求，确定开发内容}

（2）研究相关的理论技术，使用工程的概念进行开发，对其进行详细的需求分析，概要

设计，功能模块设计，构建总体框架。

{61%：（3）确定开发平台， 开发语言以及开发工具。}

#### 1.4.2研究方法

##### 一、文献检索法

通过文献检索，了解国内外发展现状，全面客观地掌握实现客户端所需要的技术。以便在设计之初，对开发可能遇到的问题进行提前预知，并分析项目实操的可行性。

##### 二、统计分析法

通过浏览各大开发类似产品的公司网站，了解其项目开发基础，项目的具体功能，梳理脉络，为后续的创新研究打下基础。

##### 三、实践研究法

获取需求之后，设计项目基本框架，设计各个模块的详细功能，进行代码编写，以及测试工作。

## 第二章开发平台介绍

### 2.1开发语言介绍

{91%：Python[3]是一种简单易学，功能强大的编程语言，它有高效率的高层数据结构，能简单而有效地实现面向对象编程。} {100%： Python简洁的语法和对动态输入的支持，再加上解释性语言的本质，使得它在大多数平台上的很多领域都是一个理想的脚本语言，} 特别适用于快速的应用程序开发。 {100%：Python支持命令式编程、面向对象程序设计、函数式编程、面向切面编程、泛型编程多种编程范式。}

{100%：Python是一种很灵活的语言，能帮你轻松完成编程工作。} {100%：并可作为一种原型开发语言，加快大型程序的开发速度。}

{94%：虽然Python可能被粗略地分类为“脚本语言”（script language），但实际上一些大规模软件开发计划例如Zope、Mnet及BitTorrent，谷歌也广泛地使用它。} {100%：Python的支持者较喜欢称它为一种高级动态编程语言，原因是“脚本语言”泛指仅作简单编程任务的语言，} {100%：如 shell script、 JavaScript等只能处理简单任务的编程语言，并不能与 Python相提并论。} {81%：此外，由于Python对于C和其他语言的良好支持，很多人还把Python作为一种“胶水语言”（glue language）使用。} {100%：使用Python将其他语言编写的程序进行集成和封装。} {93%：在谷歌内部的很多项目使用C编写性能要求极高的部分，然后用Python调用相应的模块。}

{69%：Python具有广阔的应用范围，图形处理可以通过PIL、Tkinter等图形库支持，能方便进行图形处理。} {87%：在数学处理方面， NumPy扩展提供大量与许多标准数学库的接口，在文本处理方面， python提供的 re模块能支持正则表达式，} {100%：还提供 SGML， XML分析模块，许多程序员利用 python进行 XML程序的开发。} {93%：在数据库编程方面，可通过遵循 Python DB-API（数据库应用程序编程接口）规范的模块与 Microsoft SQL Server，} Oracle， Sybase， DB2， Mysql、 SQLite等数据库通信。 {100%：python自带有一个Gadfly模块，提供了一个完整的SQL环境。} {91%：在网络编程方面，提供丰富的模块支持sockets编程，能方便快速地开发分布式应用程



序。} Web编程应用的开发语言，支持最新的XML技术，有许多优秀框架比如Django, Tornado, Flask等。 在多媒体应用方面Python的PyOpenGL模块封装了“OpenGL应用程序编程接口” {100%：，能进行二维和三维图像处理。} 除此之外Python还可以编写游戏软件，PyGame模块提供编写游戏的各种API。

{74%： Python拥有众多的特性， Python是一种代表简单注意思想的语言，阅读一个良好的 Python程序就像在读英语一样简单，} 他是你能够专注于解决问题而不是语言本身。 {76%：易学，Python极其容易上手，源于其拥有极其简单的语法。} {87%：免费，开源，简而言之，可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。} {51%：高层语言，当使用Python编写程序时，无需关注诸如内存等一些列底层问题。} Python具有良好的可移植性，在Windows, Linux, mac, Unix均有良好的表现。 {41%：Python同样也是一种面向对象语言，与C，C#，Java一样对面向对象支持良好。} {51%：Python具有可嵌入性，可以将Python的代码嵌入到C/C++等代码中，也可以调用其他语言写的类库，从而向程序用户提供脚本功能。} {55%： Python最强大的功能还在于其拥有十分丰富的库， Python的标准库异常强大，它可以帮助你处理各种工作，} {100%：包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、} {100%：XML-RPC、HTML、WAV文件、密码系统、GUI（图形用户界面）、Tk和其他与系统有关的操作。} 记住，只要安装了Python，所有这些功能都是可用的。 这被称作Python的“功能齐全”理念。 {100%：除了标准库以外，还有许多其他高质量的库，如wxPython、Twisted和Python图像库等。} 但是任何一种语言，都有其劣势之处，比如 Python不能编写操作系统， Python的速度虽然也还可以， 但是和 C/C++这种编译型语言的运行速度是没法媲美的。

## 2.2开发工具

编写 Python项目，可以选择的编译器还是非常多的，比如 Linux上的 Vim编辑器，Python内置的 IDLE，微软公司的 Visual studio系列， Mac上的 emacs编辑器，当然如果不喜欢以上的几款比较比较沉重， 也可以选择 sublime或者 visual studio code搭配 Anaconda进行项目开发。 本项目使用的开发工具主要是Pycharm 2017.3.4，也有一小部分通过visual studio code编写。

{41%：随着编程技术的发展，为了可以提高程序员的开发效率，各大企业均开发各种各样的编译器，} 来加快企业项目的开发的进度，以达到缩减项目开发成本，其中 Pycharm是 JetBrains开发的 Python专业开发工具之一， 其具有两个版本，一个是专业版需要收取一定的费用，另一个是社区版，是一个开源版本， 以供 Python爱好者可以免费使用Pycharm的一部分功能。

Pycharm支持多种Python框架，包括Django, Flask, Web2Py, AngularJS等。 通过virtualenv，可以在每个项目中单独使用Python包，避免因多个项目共同开发时，因为包管理混乱导致项目异常的问题。 通过安装 PEP8，可以让开发人员在代码格式上保持高度的统一，对项目整体的开发具有极高的价值， {63%：也对于以后项目代码的维护，打下了坚实的基础。}

Pycharm内置了很多实用的功能，可以通过点击两次 Shift使用搜索，其中文件名，symbol， action等 project相关的的都可以搜索，不过不区分模块名和文件大小写以及下划线。 Pycharm还具有代码补全功能，在各个编程语言中，每个库或者框架都存在各种各样的函数，一个人又很难技术其中所有内容， 同理在数不清的 Python包中，若想记住所有的函数名，显然是不现实的，此时通过编译器的提示功能， 就能实现在无形中增加开发人员的开发效率。

在数据库连接方面，Pycharm提供了很方便的操作，比如MySQL等RDB的连接，数据浏览操作都可以在GUI中完成，又如在Django中可以通过ORM间接操作数据库。对于开发人员来说，开发过程最大的问题在于调试bug。PyCharm自有debug功能，单步执行、断点设置、停止后改变变量、Watch等，一般debug的功能基本都可以做到。除此之外，还可以使用Remote Debug，通过ssh连接远程进行调试。当然作为一款相当优秀的编译器，其优点远远不止这些，以下便不再赘述。

## 2.3PyQt5框架

PyQt5是一款优秀的GUI框架，是一套Python绑定Digia QT5应用的框架。它可用于Python 2和3。PyQt5以一套Python模块的形式来实现功能。它包含了超过620个类，6000个方法和函数。{100%：它是一个多平台的工具套件，它可以运行在所有的主流操作系统中，包含Unix，Windows和Mac OS。} PyQt5采用双重许可模式。开发者可以在GPL和社区授权之间选择。

PyQt 是一个著名的Python应用程序框架。{80%：你并不能说它只是一个 GUI 库，因为 PyQt十分庞大，并不仅仅是 GUI 组件。} {91%：使用 PyQt，在一定程度上你获得的是一个“一站式”的解决方案：} {70%：不再需要研究 STL，不再需要Python的[string]，不再需要到处去找解析 XML、连接数据库、访问网络的各种第三方库，因为 PyQt自己内置了这些技术。}

{100%：PyQt5的类被划分在几个模块中，下面列出了这些模块：}

{98%：QtCore模块—涵盖了包的核心非 GUI功能，此模块被用于处理程序中涉及的} {98%：时间、文件、目录、数据类型、文本流、链接、QMimeData、线程或进程等对象。}

{93%：QtGui模块—涵盖了多种基本图形功能的类，包括但不限于：} {97%：窗口集、事件处理、2D图形、基本的图像和界面、字体和文本类。}

{96%：QtWidgets模块—包含了一整套UI元素控件，用于建立符合系统风格的Classic界面，非常方便，可以在安装时选择是否使用此功能。}

{94%：QtMultimedia模块—包含了一套类库，用于处理多媒体事件，通过调用API接口访问摄像头、语音设备、收发消息（Radio Functionality）等。}

{97%：QtBluetooth模块—包含了处理蓝牙活动的类库，其功能包括：} 扫描设备、连接、交互等行为。

{84%：QtNetwork模块—包含了用于进行网络编程的类库，通过提供便捷的TCP/IP及UDP的C/S代码集合，使得基于Qt的网络编程更容易。}

{96%：QtPositioning模块—用于获取位置信息，此模块允许使用多种方式实现定位，包括但不限于：} 卫星、无线网、文字信息。 {100%：此模块一般用在网络地图定位系统中。}

{92%：Enginio模块—用于构建客户端的应用程序库，在运行时访问Qt Cloud 服务器托管的应用程序。}

{98%：QtWebSockets模块—包含了一组类程序，用于实现WebSocket协议。}

{100%: QtWebKit模块——包含了用于实现基于WebKit2的网络浏览器的类库。}

{100%: QtWebKitWidgets模块——提供了一组类库，用于实现一种由Widgets包构建的、基于WebKit1的网络浏览器。}

{92%: QtXml模块——包含了用于处理XML的类库，此模块为SAX和DOM API 的实现提供了函数。}

{97%: QtSvg模块——通过一组类库，为显示矢量图形文件的内容提供了函数。}

{96%: QtSql模块——提供了数据库对象的接口以供使用。}

{88%: QtTest模块——包含了通过单元测试，调试PyQt5应用程序的功能。}

{97%: QtHelp模块——包含了用于创建和查看可查找的文档的类。}

QtOpenGL模块——使用OpenGL库来渲染3D和2D图形。 该模块使得Qt GUI库和OpenGL库无缝集成。

{98%: QtXmlPatterns模块——所包含的类实现了对XML和自定义数据模型的Xquery与XPath的支持。}

QtDesigner模块——所包含的类允许使用PyQt扩展Qt Designer。

{100%: Qt模块——将上面模块中的类综合到一个单一的模块中。} {96%: 这样做的好处是你不用担心哪个模块包含了哪个特定的类；} {100%: 坏处是加载到整个Qt框架中，从而增加了应用程序的内存占用。}

{100%: uic模块——所包含的类用来处理.ui文件，该文件由Qt Designer创建，用于描述整个或者部分用户界面。} 它可以将.ui文件编译为.py文件，以便其他Python程序调用。

{69%: 下图一至图三展示了，PyQt5中重要的继承关系。}

图2-1 QPaintDevice继承关系图

图2-2 QWidget继承关系图

图2-3 QDialog继承关系图

由于在后续模块将会重点使用到QWebEngine模块，在此重点讲述QWebEngine的一些特性以及重点介绍。 {92%: Qt WebEngine模块提供了一个web浏览器，在不使用本地浏览器的情况下，它可以很容易地把Web内容嵌入到Qt应用程序中。} Qt WebEngine为渲染HTML， XHTML和SVG文档，使用CSS和JavaScript，提供了C 类和QML类型。 Qt WebEngine为渲染HTML， XHTML和SVG文档，使用CSS和JavaScript，提供了C 类和QML类型。 Qt WebEngine的功能分成下列模块:

Qt WebEngineWidgets模块： 用于创建基于Widget的web应用。

Qt WebEngine模块： 用于创建基于Qt Quick的web应用。

Qt WebEngine Core模块： {85%: 与chromium交互页面渲染和JavaScript执行

从GUI进行分离到了Qt WebEngine进程，它是一个独立的应用库。}

{45%：模块之间相互的关系如下图2-4 Qt WebEngine Widgets模块关系图所示：}

图2-4 Qt WebEngine Widgets模块关系图

一个web engine view 是Qt WebEngine模块的主要Widget。 {86%：它可以在各种各样的应用中加载网页内容.在视图( view)里，一个 web engine page有一个主frame，} {97%：它响应网页内容，浏览链接历史以及操作( actions)。} view 和 page 非常相似， 它们提供一组公共函数。

所有pages都属于web engine profile(配置)， 它包含共享设置， 脚本和cookies。 {96%：Profiles可以使用页面分离， 一个典型应用是专用浏览模式的专用配置文件，其中没有永久保存的信息。}

{90%：应用程序可以使用URL和HTML字符串加载页面到WebEngineView， 并使用事务历史进行导航。} {99%： 在默认情况下， 不同页面的链接会加载到相同的WebEngineView对象， 但是web sites可能会请求打开新的tab， window和dialog。}

Qt WebEngine Core提供了一个由Qt WebEngine和Qt WebEngine Widgets共享的API，用于处理针对Chromium网络堆栈发布的URL请求并访问其HTTP Cookie。

实现QWebEngineUrlRequestInterceptor接口并在配置文件上安装拦截器可在拦截，阻止和修改URL请求（QWebEngineUrlRequestInfo）到达Chromium的网络堆栈之前启用它。

可以为配置文件注册QWebEngineUrlSchemeHandler以添加对自定义URL方案的支持。 然后将该方案的请求作为QWebEngineUrlRequestJob对象QWebEngineUrlSchemeHandler requestStarted()。

QWebEngineCookieStore类提供了访问Chromium的HTTP Cookie的功能。 这些功能可用于同步Cookie和QNetworkAccessManager，以及在导航过程中设置，删除和拦截cookie。

{93%：应用程序可以使用 URL和 HTML字符串加载页面到 WebEngineView，并使用事务历史进行导航.在默认情况下，} {80%：不同页面的链接会加载到相同的WebEngineView对象，但是 web sites可能会请求打开新的 tab， window和 dialog。}

## 2.4本章小结

{40%：本章介绍了设计和实现基于 Python的互动课堂的客户端所学要的开发工具和关键技术，} 包括 Python开发语言，十分优秀的 PyQt5框架，以及信号与槽机制[4]，主要的功能模块， {40%：类之间的关系，类图，以及即将用到的 Pycharm集成开发工具。}

## 第三章 需求分析

### 3.1客户端分析

本客户端是互动课堂中具有承上启下作用的一部分，面向云平台客户端可以从网络中主动请求教师所需要的各种资源， 面向学生端可以对学生平板进行一定的控制，包括对学生平板进行联网监控，控制学生屏幕， 对学生进行课上管理等功能。

本文所讨论的互动课堂主要功能在于以下几点：

(1) 客户端进行网络浏览，可以不通过第三方浏览器方便教师进行网络浏览，以及ppt，word等展示。

(2) 对教师机屏幕进行截屏并且进行一定的处理，方便教师处理学生作业图片，并对图片做一定的处理。

(3) 教师可以通过画板对学生上传的文件进行图片处理，方便教师与学生通过作业或者其他方式做更深刻的探讨

(4) 对教师屏幕进行录制，可以为缺课的学生提供课后自行补课的需求，也可以将录制的视频上传到云平台，可以分享给其他学校的学生。

{62%：总体结构如下图3-1 客户端总体架构图所示：}

图3-1 客户端总体架构图

### 3.1.1 网页浏览需求分析

客户端是重要的一个模块，作为教师课上联网的重要媒介之一，对提高教师课上提高效率有相当大的影响。在通过对一些常用浏览器基本功能分析之后，从教师使用浏览器的角度来设计，网页浏览模块应该包含以下几个功能：

(1) 教师可以通过输入框输入指定网址，浏览指定网站。

(2) 浏览器在进行页面浏览时，应该具有后退，前进，刷新，关闭当前页面的功能。

{51%：网页浏览流程图如下图3-2 网页浏览模块流程图所示:}

图3-2 网页浏览模块流程图

网页浏览用例规约:

用例名称教师

角色网页浏览模块

{41%：用例说明该用例描述了教师在在使用该模块进行操作的过程}

{43%：前置条件教师必须启动客户端，并且打开网页浏览模块}

{42%：基本事件流教师打开网页浏览模块，显示首页}

教师输入网址刷新网页

{51%：教师可以进行前进，后退，刷新，关闭网页的操作}

关闭当前模块

表3-1 网页浏览用例规约



教师端网页浏览模块用例图如下图3-3 教师端网页浏览用例图所示;

图3-3教师端网页浏览用例图

### 3.1.2 截屏需求分析

教师对于学生所上传的资源，通常需要进行一定的处理，之后将处理后的结果进行保存，之后在课上方便与同学们交流。教师通常会将学生的一些比较好的解题思路通过截取学生上传的作业资源，并在截取的图片上进行一定的绘制标记。也可以通过课上将一些很典型的题目截取下来，发送给学生们，方便在课下学生与学生，学生与教师沟通。通过对教师课上课下可能存在的一些功能需求，以及对已经存在的截图工具进行分析总结，截图模块应该包含以下几点功能：

（1）教师应该可以截取整个屏幕，或者截取屏幕的某一部分，并且可以将截取的图片保存到某个位置，方便以后使用。

（2）教师应该可以对截取屏幕图片进行一定的操作，比如在图片上插入文字，在图片上绘制图形。

（3）教师对于绘制的图形可以进行撤销和重绘功能。

（4）教师可以对于所截取的图片进行，保存或者放弃此次截图。

{49%：截取屏幕流程图如下图3-4 屏幕截取流程图所示：}

图3-4 截取屏幕流程图

截取屏幕用例规约：

用例名称教师

角色截取屏幕模块

用例说明该用例描述了教师在使用截取屏幕模块进行的相应的操作

前置条件教师必须启动客户端，并且打开屏幕截取模块

基本事件流教师打开截取屏幕模块，启动屏幕截取模块相应的功能

{46%：教师通过鼠标选取想要截取的目标区域，截取指定的屏幕}

教师在截取的屏幕上对截图进行文字输入，图形绘制等。

{49%：教师可以撤销之前的画图操作，或者重绘刚刚撤销的操作。}

5.教师可以保存此次的截图或者放弃本次截图

表3-2 屏幕截取用例规约

{43%：教师端屏幕截取模块用例图如下图3-5 教师端屏幕截取用例图所示;}

图3-5 教师端屏幕截取用例图

### 3.1.3录屏需求分析

{84%：“翻转课堂”起源于美国科罗拉多州落基山林地公园高中的两位化学老师——乔纳森伯尔曼和亚伦萨姆斯，} 最初这二位教师在上课时因班上有位学生因为身体原因无法正常出席。他们二人通过将讲课内容录制成视频，并且将视频发送给那位未能正常出席的学生。从此以后，二人通过一定的研究将这种教学方法形成一种独特的教学模式，并且很快在美国各个州部迅速发展开来，并且取得较为可观的成果。

根据此篇观点，互动课堂的客户端在屏幕录制方面便是必不可少的，当前教学模式中，教师通过 PPT，Word，Excel等办公软件，高效向学生传递知识。教师在一节课上的主要知识点，均记录在教师通过查找各种资料，以及教师通过多年的经验总结所累积而成的结果。

通过将教师屏幕录制下来，并分享给学生帮助学生在课前以及课后进行知识的预习以及知识的复习，极大地提高学生的学习效率。{43%：通过以上分析教师端屏幕录制模块应该包含以下几点功能:}

- (1) 最基本的功能需求是将教师端屏幕录制成一个完整阶段的视频。
- (2) 指定录制的视频的视频保存位置。
- (3) 保存或者放弃所录制的视频。

{45%：屏幕录制流程图如下图3-6 屏幕录制流程图所示：}

图3-6 屏幕录制流程图

屏幕录制用例规约：

用例名称教师

角色屏幕录制模块

用例说明该用例描述了教师在使用屏幕录制模块进行的屏幕录制操作

前置条件教师必须启动客户端，并且打开屏幕录制模块

基本事件流教师打开屏幕录制模块

{65%：教师点击开始录制按钮，进行屏幕录制}

教师点击结束按钮，结束屏幕录制

教师选择保存或者放弃保存按钮

表3-3 屏幕录制用例规约

教师端屏幕录制模块用例图如下图3-6 教师端屏幕录制用例图所示;

图3-6 教师端屏幕录制用例图

### 3.1.4画板需求分析

画板作为传统且原始的画图工具，在翻转课堂的教学中也拥有比较重要的地位，教师通过画板可以方便的对学生上传的作业图片直接进行，必要的修改，写一些批注，方便教师收集学生典型的错误题型。通过高效的题型收集，有利于教师通过课堂更加高效的专注学生平时把握不是很精准的知识点，通过专题专攻以达到人人都能理解难点，实现提高教学水平，整体提高国家竞争力的目标。根据以上分析，且本文所实现的画板仅仅是简单版本的画板，

实现的功能主要有如下几点：

可以在画板上进行一定的画图操作

可以打开文件，关闭文件，保存文件。

{58%：画板画图流程图如下图3-7 画板绘图流程图所示：}

图3-7 画板绘图流程图.

画板绘图用例规约：

用例名称教师

角色画板模块

用例说明该用例描述了教师在使用画板模块进行的画板绘图操作

{43%：前置条件教师必须启动客户端，并且打开画板模块}

基本事件流教师打开画板绘图模块

教师导入图片

{47%：教师教师对图片进行简单的文字或者手绘制图处理}

保存或者放弃保存文件

表3-4 屏幕录制用例规约

教师端屏幕绘图模块用例图如下图3-8 教师端画板绘图用例图所示;

图3-8 教师端画板绘图用例图

### 3.2客户端性能需求

在操作的实时性上考虑，教师机客户端在使用上若想获得比较好的流畅性，以及用户使用的好评，和易用性，需要在性能需求的分析上，花费一定的时间，才能获得良好的反馈。客户端的处理中，对于截取屏幕，需要操作上拥有非常高的流畅性，以及截图实时处理的直观感受，需要竭力避免因编码错误而导致客户端异常宕机的情况。在进行屏幕录制的时候，尽可能考虑到平台问题，在屏幕录制的同时，不能因为录制屏幕而导致，用户产生客户端变慢，变卡顿的知觉，保证录制屏幕的同时也不能牺牲用户的体验。在进行绘图的时候，在绘图操作进行的时候，竭力让用户产生瞬时画画的感觉，每一次在画板上的动作都不应该存在丝毫的卡顿。在用户使用内置的浏览网页时，需要用户使用到流畅，舒适的体验，杜

绝拖拉，反应迟缓的问题。

在时间特性方面考虑，一般的界面切换操作响应延时不能超过1.2s，使用户感到有发请求立即得到响应的状态，提升用户在使用上的体验。

在操作的简易性上，客户端的操作要做到让人容易理解，无需通过复杂的培训或者其他操作即可实现快速上手。

{43%：在计算机内存上考虑，客户端在运行时力求占用较少的计算机资源，} 当计算机空闲资源丰富的情况下也会更加能够保证客户端运行的稳定性。

教师端界面要求布局合理，颜色舒适，界面布局合理，拥有统一的元素标准，简单易用的主题。

### 3.3客户端运行环境

本客户端是基于Windows平台，如今计算机领域有Linux家族，Unix家族以及一些其他的操作系统，其中Windows的市场份额是以上平台中份额最大的。 {46%：在学校的教师机上运行的都是基于 Windows的不同版本的操作系统， Windows Xp，} Windows7，Windows8.1， Windows10，处理器为 Intel系列处理器，或者 Amd系列处理器， 不过二者均能兼容的运行 PyQt5所开发的 GUI应用程序。

## 第四章系统设计

### 4.1架构设计

设计基于 Python的互动课堂整体系统架构时，通过教师 PC端作为中间媒介， PC端通过向云端服务器发送 request请求， {42%：在服务器接收到来自客户端的 request请求后，通过在服务器中路由的寻找，找到服务器中响应的函数方法，} 之后在函数内部通过一定的数据库操作，将从数据库中取到的资源以 json或者 ajax等方式发送到客户端指定的端口的服务中， {49%：客户端通过解析 json或者 ajax返回的数据，将数据显示在教师机客户端上。} 在教师机面向学生端的通信过程中，学生端首先以教师机作为本地的服务器，学生通过教师机所发散的 wifi信号， 构建本地局域网，学生之间的通信，借助学校课堂的教师机，而不是通过学校的主机房，一方面可以减少机房服务器的压力，另一方面可以保证学生端在线的稳定性，不会线路连接不稳定而产生课堂断线的情况，保证课堂上课的连续性与系统架构的稳定性。 {46%：在学生端通过教师机连接互联网时，教师机可以对学生机进行网络白名单控制，} 教师机控制学生可以登录的网址，学生只能通过教师机所允许的网址访问互联网资源。

{42%：架构的设计需要让系统具有强大的适应性，良好的可靠性，较佳的安全性，以及一定的可维护性，} 除此之外还需具有良好的可拓展性，只有通过科学的架构设计，以及应用良好的设计模式的前提下， 系统在面向日后的变化时才能拥有更加良好的适应性。 只有增强系统的架构设计才能在日后，方便系统的拓展和维护工作，提高系统的复用性，以及可靠性。

本文所讨论的仅仅是教师端的 PC部分的一部分功能，通过教师机的主界面进行界面布局的控制， 在界面的布局上，通过界面一些按钮的使用来触发响应的功能。 主要通过按钮来执行教师机客户端，网页浏览功能的启动，进行屏幕录制，屏幕截取，画板功能的实现。通过 PyQt5的 Qt Designer将主界面的界面布局与代码逻辑相分离，将界面 UI与程序逻辑解耦合， 尽量将程序逻辑代码内部结构以很高的内聚的方式，增强程序的稳健性。

通过将设计模式的运用，增强程序的拓展性，以及可复用性，在程序面临主界面扩展时，只需继承相应的类来实现程序的拓展。

## 4.2 概要设计

{42%：通过对教师端的功能模块的需求分析，在此基础上设计教师端重要的功能模块的概要设计。} 通过第三章的详细讲述，本文将教师端划分4个主要的模块，分别是“网页浏览”模块，“截取屏幕”模块，“屏幕录制”模块，“简单画板”模块。{58%：以下将对这四个功能模块的概要设计进行详细的介绍。}

### 4.2.1 “网页浏览”模块的概要设计

这个模块的功能主要是在教师通过本程序，在想请求互联网上的资源时，可以启动网页浏览模块，在这个启动浏览网页的功能后教师可以直接进行网上冲浪，可以以此模块为基础，将教学所需要的教学资源在这个浏览器中直接展示给学生。以下是“网页浏览”模块的某些重要的功能的业务流程图：

图4-1 网页浏览模块业务流程图

### 4.2.2 “屏幕截取”模块的概要设计

这个模块的主要功能是教师通过按钮点击截取屏幕，教师通过鼠标对截屏区域进行选取，之后在屏幕显示一个已经截取的屏幕的原型，之后教师可以在此原型的基础上，在图片上进行图形的绘制，表示对学生作业的某些内容进行突出标识，方便教师在课上直接将学生对于某些问题的特殊解决办法，或者学生常见的易错题直接展示出来。还可以在所截取的图片上进行相应的文字描述，在文字描述的基础上，学生在拿到教师的发的资料时，可以第一时间根据图片上所显示的信息进行自我问题查找，自行巩固知识点。以下是“截取屏幕”模块的重要功能的流程图：

图4-2 屏幕截取模块业务流程图

### 4.2.3 “屏幕录制”模块的概要设计

这个模块的主要功能是教师通过点击屏幕录制按钮，之后可以将整个屏幕录制下来，在点击录制按钮结束录制后，教师可以获取所录制的视频。在教师获得录制的微课视频之后，教师可以将视频资料上传到个人空间，来为学生提供课前预习或者课后复习的资源，也可以将自己的上课视频传到已有的微课平台比如网易云课堂，慕课网，或者待本系统全部开发完成，{41%：教师可以将自己所有的教学资源，放到教师的网络空间中，共享给其他学校或者学生观看，} 提高自己学校以及全国的学校的教学资源的共享的成果。根据上述的概要分析，“屏幕录制”模块概要分析的业务流程图，如下图4-3 屏幕录制模块的业务流程图：

{77%：图4-3 屏幕录制模块的业务流程图}

### 4.2.4 “简易画板”模块的概要设计

这个模块的主要功能是教师通过点击画板功能按钮，调用画板模块，显示到主页面中，画板可以导入学生资源或者教师自己的资源，将资源进行处理。在互动课堂的教学过程中，教师将学生的作业可以导入到画板中进行处理，比如给学生的问题回答进行一定的批注，成绩评定，或者教师可以将自己的解决问题的思路写在学生的答案的附近，当将资源返回给学生看的时候，{42%：学生可以第一时间先通过自己的尝试重新解决某些问题，通过调高学



生自主的解决问题的能力，} {64%：以提高学生在学习过程中的学习能力。} 根据以上的概要分析，“简易画板”模块的概要分析的业务流程图，如下图4-4简易画板模块的业务流程图所示：

图4 - 4 简易画板模块的业务流程图

## 第五章功能模块的实现

### 5.1 主界面实现

本小节主要描述，互动课堂教师端主界面的界面布局以及主界面一些按钮的设置和布局。主界面的内容比较简单，主要是通过一个主界面，在界面上只有在左侧的一个工具栏。  
{43%：通过工具栏上的一些按钮，调用相关的功能模块，实现相应的功能。} {50%：主界面整体布局如图5-1 互动课堂主界面所示:}

图5-1 互动课堂主界面

下面本文将从代码的角度来分析此主界面的编写过程，首先创建一个继承 QMainWindow 主界面类， {52%：并且在代码中增添一个左侧的工具栏，代码如下所示：}

```
self.RightToolBar = QToolBar('ToolBar')

self.RightToolBar.setIconSize(QSize(30, 30))

self.addToolBar(QtCore.Qt.RightToolBarArea, self.RightToolBar)
```

在增加工具栏之后，向工具栏中增添一些功能，在 PyQt5中工具栏中，需要增加 QAction，通过点击或者其他事件触发，便可调用相关的模块功能，下面我将以添加浏览器模块的 Action为例，主要逻辑代码如下：

```
self.actionbrowser = QtWidgets.QAction('Browser')

icon = QtGui.QIcon()

icon.addPixmap(QtGui.QPixmap('../images/toolBarIcon/ie.png'),
QtGui.QIcon.Normal, QtGui.QIcon.Off)

self.actionbrowser.setIcon(icon)

self.actionbrowser.setObjectName("actionbrowser")

self.RightToolBar.addAction(self.actionbrowser)
```

同样通过如下的方法添加其他的QAction便可实现主界面左侧的工具栏的设置。

当然在此处为了可以调用其他模块的响应功能，在类的设计中，通过类包含，将画板模块，浏览器模块截屏模块，录屏模块，截屏模块添加到类的属性中。

通过以上的步骤设置好工具栏后，在进行相应的触发动作执行后需要通过 PyQt5中的信号与槽机制，来完成相应的信号传递以及对应的函数调用。下面以调用浏览器模块为例相应的信号与槽函数连接过程如下：

```
self.actionbrowser.triggered.connect(self.FormBrowser)
```

由于 Python 的易读性很容易看懂其中代码的含义，当触发浏览器模块时动作后，调用浏览器模块的槽函数，而在槽函数中编写具体浏览器模块被调用的具体实现。

主界面的实现比较容易而且易于理解，在此并不做过多的探讨，只是 PyQt5 基础知识的最简单的应用。下面的模块的实现中，也不再对信号与槽函数的具体的机制进行深入的解释，关于主界面的更多内容在此便不再更多描述。

## 5.2 主要功能模块的实现

### 5.2.1 网页模块的实现

网页模块的实现主要是基于 PyQt5 的 QWebEngine 模块实现，本篇论文所讨论的浏览器模块功能比较简单，所使用到的也仅仅是 QWebEngine 中一些简单的功能，通过将页面嵌入到应用程序中，为教师提供直接访问网络的接口。 {42%：网页浏览模块实现的结果下图5-2 网页模块的展示图所示：}

图5-2 网页模块的展示图

{43%：网页模块目录结构如下图5-3 网页模块的项目文件结构图所示：}

### 5-3 网页模块的项目文件结构图

下面将根据如图所示的结果对界面的布局以及功能的实现做简要的描述。

首先创建一个继承自 QMainWindow 的 Browser 类，用来展示网页浏览的主界面。 {53%：根据上图所示的结果，在主界面中主要包含以下内容：}

(1) 客户端主窗体展示所有的内容

(2) 主窗体中的主浏览器

(3) 主浏览器上方的工具栏

接下来将会依次分析这三个部分的代码，并且进行一定的解释。在这个窗体中主要实现了以下几个函数：

navigate\_to\_url(self) 用来将 url 指定到浏览器中

renew\_urlbar(self, q) 在输入页面 url 后通过刷新重新

closeEvent(self, event) 关闭当前所显示的内容

在界面布局上，由于界面继承自 QMainWindow，在类中存在这样的一个 CentralWidget 成员变量，表示继承自 QMainWindow 的核心控件类，在这里首先创建一个 QWebEngine 的实例，

```
self.browser = QWebEngineView()
```

```
url = 'https://www.baidu.com'
```

```
self.browser.setUrl(QUrl(url))
```

```
self.setCentralWidget(self.browser)
```

通过指定默认的 `url` 便可在打开本模块的时候自动打开浏览器所指定的 `url` 所对应的页面，之后将 `browser` 实例指定父对象，`browser` 的父对象为窗体的 `CentralWidget` 控件，之后在打开浏览器时，`browser` 实例就会显示在窗体的核心控件中。

在界面中除此之外，还有一个简易的工具栏，主要由4个按钮，分别是页面前进功能，页面后退功能，页面刷新功能和页面关闭功能，还有一个输入框用来输入和显示指定的 `URL`。  
{42%：工具栏部分通过 `QToolBar` 创建一个工具栏实例，之后将工具栏实例添加到窗口中，}

为了实现前进，后退，停止加载和刷新功能，还需要在工具栏中添加4个 `Action` 对象。

Toolbar代码：

```
navigation_bar = QToolBar(' Navigation' )
```

```
navigation_bar.setIconSize(QSize(16, 16))
```

```
self.addToolBar(navigation_bar)
```

4个 `QAction` 对象代码：

```
back_button = QAction(QIcon('../Web/icons/back.png' ), ' Back' ,  
self)
```

```
next_button = QAction(QIcon('../Web/icons/next.png' ), ' Forward' ,  
self)
```

```
stop_button = QAction(QIcon('../Web/icons/cross.png' ), ' stop' ,  
self)
```

```
reload_button = QAction(QIcon('../Web/icons/renew.png' ), ' reload' ,  
self)
```

将按钮放到导航栏上：

```
navigation_bar.addAction(back_button)
```

```
navigation_bar.addAction(next_button)
```

```
navigation_bar.addAction(stop_button)
```

```
navigation_bar.addAction(reload_button)
```

若要实现相应的点击功能需要将相应的对象与相应的槽函数连接，并在触发事件执行之后触发相应的动作，下面是对应的槽函数连接方法：

```
back_button.triggered.connect(self.browser.back)
```

```
next_button.triggered.connect(self.browser.forward)
```

```
stop_button.triggered.connect(self.browser.stop)
```

```
reload_button.triggered.connect(self.browser.reload)
```

设置url的槽函数：

```
def navigate_to_url(self):  
  
    q = QUrl(self.urlbar.text())  
  
    if q.scheme() == '':  
  
        q.setScheme('http')  
  
    self.browser.setUrl(q)
```

刷新页面的槽函数：

```
def renew_urlbar(self, q):  
  
    self.urlbar.setText(q.toString())  
  
    self.urlbar.setCursorPosition(0)
```

停止页面刷新的槽函数：

```
def closeEvent(self, event):  
  
    reply = QMessageBox.question(self, 'Message', "Are you sure  
to quit?", QMessageBox.Yes | QMessageBox.No, QMessageBox.No)  
  
    if reply == QMessageBox.Yes:  
  
        event.accept()  
  
    else:  
  
        event.ignore()
```

### 5.2.2 屏幕截取模块的实现

{41%：屏幕截取模块主要是通过PyQt5中的Qt模块，QtCore模块以及QtGui模块相关函数来实现。} 通过 QtGui模块来编写相应的界面 GUI，通过 QtCore中的 pyqtSignal来定义屏幕截取过程所用到的各种各样的信号，再通过信号与槽之间的通信实现各种简单功能，最终组合成完整的屏幕截取模块。截图模块的实现情况如下图5-4 屏幕截取模块的展示图，图5-5 屏幕截取模块功能展示图所示：

图5-4 屏幕截取模块的展示图

图5-5 屏幕截取模块功能展示图

下面将会根据以上的功能，以及具体的代码实现来详细分析屏幕截图模块的实现。

{41%：具体将本模块的实现分为以下几个子模块进行分析：}

(1) 不同事件的定义

(2) 文本输入框的实现

(3) 功能工具栏的实现

(4) 颜色工具栏的实现

(5) 屏幕截图的细节实现

{40%：首先整个模块的工程结构如下图5-6 屏幕截取模块工程目录结构图所示：}

图5-6 屏幕截取模块的工程目录结构图

首先定义不同操作的事件的定义，在这里将不同的操作，以不同的数字作为表示，下面是区域选择，移动选择区域，画矩形，画圆形，画箭头，画直线， {42%：自由输入文字，通过文本框输入文字， UnDo，保存，取消，确定的事件定义：}

```
ACTION_SELECT    =    0
```

```
ACTION_MOVE_SELECTED    =    1
```

```
ACTION_RECT    =    20
```

```
ACTION_ELLIPSE    =    21
```

```
ACTION_ARROW    =    22
```

```
ACTION_LINE    =    23
```

```
ACTION_FREEPEN    =    24
```

```
ACTION_TEXT    =    25
```

```
ACTION_UNDO    =    26
```

```
ACTION_SAVE    =    27
```

```
ACTION_CANCEL    =    28
```

```
ACTION_SURE    =    29
```

其中将所有的绘图操作添加到一个绘图操作的列表中，方便在之后的实现中使用，代码如下：

```
DRAW_ACTION    =    [ACTION_RECT,    ACTION_ELLIPSE,    ACTION_ARROW,  
ACTION_LINE,    ACTION_FREEPEN,    ACTION_TEXT]
```

在截图过程中鼠标位置的变化是比较复杂多样的，以下是鼠标位置的类定义：

```
class    MousePosition:
```



```
ON_THE_LEFT_SIDE    =    30

ON_THE_RIGHT_SIDE    =    31

ON_THE_UP_SIDE      =    32

ON_THE_DOWN_SIDE    =    33

ON_THE_TOP_LEFT_CORNER    =    34

ON_THE_TOP_RIGHT_CORNER    =    35

ON_THE_BOTTOM_LEFT_CORNER    =    36

ON_THE_BOTTOM_RIGHT_CORNER    =    37

INSIDE_AREA    =    38

OUTSIDE_AREA    =    39

WAIT_FOR_SELECT    =    40
```

此外还有对于鼠标是否在正确区域的常量定义，以及画笔颜色，画笔大小，字体大小以及字体颜色的常量的定义代码如下所示：

```
ERRORRANGE    =    6

PENCOLOR    =    ' #ff0000'    #    red

PENSIZE    =    1

FONTSIZE    =    10

FONTTYPE    =    ' ubuntu    mono'
```

文本输入框，是在截屏的基础上，通过在截取的屏幕上进行文字输入的一种形式。文本输入框主要是通过 PyQt5中 QtCore模块中的 pyqtSignal实现，通过定义输入文字相关信号， {40%：确定保存信号，取消信号来实现具体的功能，信号定义代码如下所示：}

```
inputChanged    =    pyqtSignal()

okPressed    =    pyqtSignal()

cancelPressed    =    pyqtSignal()
```

{44%：文本输入框的具体实现如下图5-6 屏幕截取文本输入框展示图所示：}

{50%：图5-6 屏幕截取文本输入框展示图}

首先创建一个继承于 QWidget的 TextInput类，在类中通过添加 QVBoxLayout布局，在布局里添加一个 QHBoxLayout，在 QHBoxLayout中添加一个文本输入框和两个按钮具体代码实现如下所示：

```
self.mainLayout = QVBoxLayout()

self.textArea = QTextEdit(self)

self.buttonArea = QWidget(self)

self.buttonLayout = QHBoxLayout()

self.cancelButton = QPushButton(' 取消' , self)

self.okButton = QPushButton(' 确定' , self)

self.buttonLayout.addWidget(self.cancelButton)

self.buttonLayout.addWidget(self.okButton)

self.buttonArea.setLayout(self.buttonLayout)

self.mainLayout.addWidget(self.textArea)

self.mainLayout.addWidget(self.buttonArea)

self.setLayout(self.mainLayout)
```

下面将设置确定按钮，取消按钮，以及文本框内容改变的槽函数以便将内容显示到所截取的图片上，代码如下所示：

```
self.textArea.textChanged.connect(self.textChanged_)

self.okButton.clicked.connect(self.okButtonClicked)

self.cancelButton.clicked.connect(self.cancelPressed)
```

所对应的槽函数分别为：

```
def textChanged_(self):

    self.inputChanged.emit()

def cancelButtonClicked(self):

    self.cancelPressed.emit()

def okButtonClicked(self):

    self.okPressed.emit()
```

除此之外在TextInput类中，还定义了获取输入框文本函数，设置窗口焦点函数以及清空文本框的函数，代码分别如下所示：

```
def getText(self):

    return self.textArea.toPlainText()
```

```
def    getFocus(self):  
  
self.setFocus()  
  
self.textArea.setFocus()  
  
def    clearText(self):  
  
self.textArea.clear()
```

截图模块的功能相关的工具栏如下图5-7 屏幕截取功能选项工具栏展示图所示：

图5-7 屏幕截取功能选项工具栏展示图

工具栏中主要是通过 `QHBoxLayout` 实现，把所有的按钮添加到布局中，按钮主要分为三组，第一组画图形以及添加文字，第二组撤销和保存，第三组取消和确定按钮，具体的实现代码通过三个函数来添加。

画图组按钮代码如下所示，只以其中一个为例：

```
def    initDrawButtons(self):  
  
self.rectButton    =    QPushButton(self)  
  
self.rectButton.setIcon(QIcon("../icons/rect.png"))  
  
self.rectButton.setFixedSize(self.iconWidth,    self.iconHeight)  
  
self.rectButton.setCheckable(True)
```

其他组按钮实现代码如下所示，只以其中一个为例：

```
def    initOtherButtons(self):  
  
#    other    action    buttons  
  
self.separator1    =    QFrame(self)  
  
self.separator1.setFrameShape(QFrame.VLine)  
  
self.separator1.setFrameShadow(QFrame.Sunken)  
  
self.undoButton    =    QPushButton(self)  
  
self.undoButton.setIcon(QIcon("../icons/undo.png"))  
  
self.undoButton.setFixedSize(self.iconWidth,    self.iconWidth)  
  
self.undoButton.clicked.connect(self.otherButtonsClicked)
```

初始化整个界面的代码如下所示：

```
def    initWindow(self):
```

```
self.hlayout = QHBoxLayout()

self.hlayout.setSpacing(2)

self.hlayout.setContentsMargins(10, 2, 10, 2)

self.setLayout(self.hlayout)

self.initDrawButtons()

self.initOtherButtons()
```

在上述界面实现之后，通过点击某个按钮进行按钮切换的函数通过自定义buttonToggled函数来实现，具体代码如下所示：

```
def buttonToggled(self, button):

    if button == self.rectButton:

        self.trigger.emit(ACTION_RECT)

    elif button == self.ellipseButton:

        self.trigger.emit(ACTION_ELLIPSE)

    elif button == self.arrowButton:

        self.trigger.emit(ACTION_ARROW)

    elif button == self.lineButton:

        self.trigger.emit(ACTION_LINE)

    elif button == self.freePenButton:

        self.trigger.emit(ACTION_FREEPEN)

    elif button == self.textButton:

        self.trigger.emit(ACTION_TEXT)

    else:

        Pass
```

上述代码实现画图组按钮切换的信号发射，其他按钮的信号发射通过自定义的otherButtonClicked函数实现，具体代码如下所示：

```
def otherButtonsClicked(self):

    if self.sender() == self.undoButton:

        self.trigger.emit(ACTION_UNDO)
```

```
elif self.sender() == self.cancelButt

self.trigger.emit(ACTION_CANCEL)

elif self.sender() == self.okButton:

self.trigger.emit(ACTION_SURE)

elif self.sender() == self.saveButton

self.trigger.emit(ACTION_SAVE)
```

截图模块的颜色工具栏如下图5-8 屏幕截取模块的颜色工具栏所示：

图5-8 屏幕截取模块的颜色工具栏

根据以上图片的内容分析，截图的颜色工具栏主要分为三个部分，第一个部分是三种粗细的画笔， {54%：第二个部分是字体选择器，第三个部分是颜色选择按钮组。} 在此模块中画笔大小的改变，颜色的改变，字体的改变通过自定义的信号实现，代码如下所示：

```
penSizeTrigger = pyqtSignal(int)

penColorTrigger = pyqtSignal(str)

fontChangeTrigger = pyqtSignal(QFont)
```

在此处重点说明PyQt5中字体对话框的使用，在这个部分的代码中主要实现以下函数：

```
generateButtons(self, parent=None),

initWindows(self),

initPenSizeButtons(self),

initPenColorButtons(self),

initFontWidget(self),

showFontWidget(self),

showPenWidget(self),

colorButtonToggled(self, button),

sizeButtonToggled(self, button),

fontButtonClicked(self)
```

generateButtons函数通过colorlist的颜色列表，通过循环创建若干个按钮，主要的核心代码如下所示：

```
self.colorButtons = []
```



```
for color in self.colorList:

    button = QPushButton(parent)

    button.setObjectName(color[0])

    button.setStyleSheet(' QPushButton { background-color: s; }'
color[1])

    button.setFixedSize(self.iconWidth / 2, self.iconHeight / 2)

    button.setCheckable(True)

    self.colorButtons.append(button)
```

initWindows函数， initPenSizeButtons函数进行对窗口的初始化以及对于画笔相关的按钮的设置，在这里不作为重点描述， initPenColorButtons函数中通过调用 generateButtons函数，将产生的 button添加到布局的核心代码如下：

```
self.generateButtons()

self.colorButtonGroup = QButtonGroup(self)

for button in self.colorButtons:

    self.colorButtonGroup.addButton(button)

self.colorButtonGroup.buttonClicked.connect(self.colorButtonToggled)
```

产生如图中的2行buttons的核心代码如下所示：

```
tmp = 0

for x in range(0, 2):

    for y in range(0, int(len(self.colorList) / 2)):

        self.colorGrid.addWidget(self.colorButtons[tmp], x, y)

    tmp = 1
```

在initFontWidget函数中，创建文件对话框，在按钮点击是触发显示文件对话框的槽函数，具体代码如下所示：

```
self.fontDialog = QFontDialog()

self.changeFontButton = QPushButton(self)

self.fontDialog.setCurrentFont(QFont(' Sans serif' ))

self.changeFontButton.setText(' {0}
{1}' .format(self.fontDialog.currentFont().family(),
```

```
self.fontDialog.currentFont().pointSize()))
```

```
self.changeFontButton.clicked.connect(self.fontButtonClicked)
```

showFontWidget函数和showPenWidget函数通过调用相应控件的show()和hide()函数进行控件的显示和隐藏。colorButtonToggled函数实现代码如下所示：

```
def colorButtonToggled(self, button):  
  
    self.presentColor.setStyleSheet(' QPushButton { background-color: s;  
' button.setObjectName()  
  
    self.penColorTrigger.emit(button.setObjectName())
```

字体的转换函数fontButtonClicked函数的代码如下所示：

```
def fontButtonClicked(self):  
  
    ok = True  
  
    font = QFontDialog.getFont(self)  
  
    if font[1]:  
  
        self.changeFontButton.setText(' {0} {1}' .format(font[0].family(),  
font[0].pointSize()))  
  
        self.fontChangeTrigger.emit(font[0])
```

sizeButtonToggled函数的实现如下所示：

```
def sizeButtonToggled(self, button):  
  
    self.penSizeTrigger.emit(int(button.setObjectName()) * 2)
```

在上述简单功能实现之后，截图功能的主界面实现将在下面进行重点分析，首先将之前写好的子模块以包的形式导入到screenshot模块，代码如下：

```
from constant import *  
  
from toolbar import *  
  
from colorbar import *  
  
from textinput import *
```

首先创建一个 SnapShot类，继承于 QGraphicsView，通过初始化函数\_\_init\_\_()初始化类中需要用到的数据结构以及类的成员变量，所初始化的数据结构和变量主要有以下几个部分具体内容将在代码中展示：

```
#初始化画笔相关
```

```
self.penColorNow = QColor(PENCOLOR)

self.penSizeNow = PENSIZE

self.fontNow = QFont(' Sans' )

self.clipboard = QApplication.clipboard()

#初始化绘图相关

self.drawListResult = [] # DrawPicture list that sure to be
drew, [action, coord]

self.drawListProcess = None # the process to the result

self.selectedArea = QRect() # a QRect instance which stands
for the selected area

self.selectedAreaRaw = QRect()

self.mousePosition = MousePosition.OUTSIDE_AREA # mouse position

self.screenPixel = None

self.textRect = None

#初始化鼠标相关

self.mousePressed = False

self.action = ACTION_SELECT

self.mousePoint = self.cursor().pos()

self.startX, self.startY = 0, 0 # the point where you
start

self.endX, self.endY = 0, 0 # the point where you end

self.pointPath = QPainterPath() # the point mouse passes,
used by DrawPicture free line

self.itemsToRemove = [] # the items that should not
DrawPicture on screenshot picture

self.textPosition = None

#获取屏幕全部界面截图

self.getscreenshot()

#界面相关
```

```
self.showFullScreen()

self.setWindowFlags(Qt.WindowStaysOnTopHint | Qt.FramelessWindowHint)

#监听鼠标事件

self.setMouseTracking(True)

self.setHorizontalScrollBarPolicy(Qt.ScrollBarAlwaysOff)

self.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOff)

#设置布局以及样式

self.setContentsMargins(0, 0, 0, 0)

self.setStyleSheet(" QGraphicsView { border-style: none; } ")

self.toolBar = MyToolBar(self)

#界面按钮触发事件与槽函数的连接

self.toolBar.trigger.connect(self.changeAction)

self.penSetBar = PenSetWidget(self)

self.penSetBar.penSizeTrigger.connect(self.changePenSize)

self.penSetBar.penColorTrigger.connect(self.changePenColor)

self.penSetBar.fontChangeTrigger.connect(self.changeFont)

self.textInput = TextInput(self)

self.textInput.inputChanged.connect(self.textChange)

self.textInput.cancelPressed.connect(self.cancelInput)

self.textInput.okPressed.connect(self.okInput)

self.graphicsScene = QGraphicsScene(0, 0, self.screenPixel.width(),
self.screenPixel.height())

self.setScene(self.graphicsScene)

self.redraw()

#设置截图保存和取消的快捷键槽函数的连接

QShortcut(QKeySequence(' ctrl s' ),
self).activated.connect(self.saveScreenshot)

QShortcut(QKeySequence(' esc' ), self).activated.connect(self.close)
```

{55%：并且设置保存的快捷键是CTRL S，设置退出不保存的快捷键是ESC。} 在截图过程中，鼠标状态的控制是比较复杂的在不同的情境下鼠标的状态需要使用不同的状态记录。由于代码层次比较复杂，在此只展示部分代码，表现出鼠标控制的大体逻辑，具体代码如下所示：

```
def mouseMoveEvent(self, event):

    self.mousePoint = QPoint(event.x(), event.y())

    if self.action is None:

        self.action = ACTION_SELECT

    if not self.mousePressed:

        point = QPoint(event.x(), event.y())

        self.detectMousePosition(point)

        self.setCursorStyle()

        self.redraw()

    else:

        self.endX, self.endY = event.x(), event.y()

        # if self.mousePosition != OUTSIDE_AREA:

        # self.action = ACTION_MOVE_SELECTED

        if self.action == ACTION_SELECT:

            self.selectedArea.setBottomRight(QPoint(event.x(), event.y()))

            self.redraw()

        elif self.action == ACTION_MOVE_SELECTED:

            self.selectedArea = QRect(self.selectedAreaRaw)

        Pass
```

鼠标移动产生何种变化主要根据当前的事件类型决定，当鼠标处于不同的事件时，鼠标会根据事件类型表现出不同的鼠标属性。鼠标的释放事件也是具有一定的逻辑性，但是鼠标释放的动作与截图事件相关性较低，{40%：所以鼠标的释放事件相对而言比较简单，代码如下所示：}

```
if event.button() != Qt.LeftButton:

    return
```

```
        if self.mousePressed:

            self.mousePressed = False

            self.endX, self.endY = event.x(), event.y()

            if self.action == ACTION_SELECT:

                self.selectedArea.setBottomRight(QPoint(event.x(), event.y()))

                self.selectedAreaRaw = QRect(self.selectedArea)

                self.action = ACTION_MOVE_SELECTED

                self.redraw()

            elif self.action == ACTION_MOVE_SELECTED:

                self.selectedAreaRaw = QRect(self.selectedArea)

                self.redraw()

            # self.action = None

            elif self.action == ACTION_RECT:

                self.drawRect(self.startX, self.startY, event.x(), event.y(),
True)

                self.redraw()

            elif self.action == ACTION_ELLIPSE:

                self.drawEllipse(self.startX, self.startY, event.x(), event.y(),
True)

                self.redraw()

            elif self.action == ACTION_ARROW:

                self.drawArrow(self.startX, self.startY, event.x(), event.y(),
True)

                self.redraw()

            elif self.action == ACTION_LINE:

                self.drawLine(self.startX, self.startY, event.x(), event.y(),
True)

                self.redraw()
```



```
elif self.action == ACTION_FREEPEN:

    self.drawFreeLine(self.pointPath, True)

    self.redraw()
```

在截图过程中需要在鼠标下方某个位置产生一个类似于放大镜效果的小视窗，通过小视窗将截图的边界表现出来，为截图提供方便。 下面的代码描述了画矩形区域：

```
def drawRect(self, x1, x2, y1, y2, result):

    rect = self.selectedArea.normalized()

    tmpRect = QRect(QPoint(x1, x2), QPoint(y1, y2)).normalized()

    resultRect = rect tmpRect

    tmp = [ACTION_RECT, resultRect.topLeft().x(),
resultRect.topLeft().y(),

    resultRect.bottomRight().x(), resultRect.bottomRight().y(),

    QPen(QColor(self.penColorNow), int(self.penSizeNow))]

    if result:

        self.drawListResult.append(tmp)

    else:

        self.drawListProcess = tmp
```

{40%：其中画圆形，画箭头，画直线的做法均与其类似，在此便不再赘述。} 在截图完成之后，需要将截图保存到指定位置，下面是将截图保存的代码：

```
def saveOperation(self):

    filename = QFileDialog.getSaveFileName(self, ' Save file',
' ../Snapcut/screenshot.png', '*.png; *.jpg')

    if len(filename[0]) == 0:

        return

    else:

        self.saveScreenshot(False, filename[0], filename[1][2:])

        self.close()
```

至此关于截屏模块的主体功能，基本上可以实现，不过仍有大量的代码为能在本文章中表现出来。

### 5.2.3 屏幕录制模块的实现

屏幕录制模块的实现设计到Python中许多模块，Win32GUI，Pillow，PyQt5中的QtCore模块，moviepy模块。屏幕录制的主要过程通过点击开始录制按钮，之后会不断地截取屏幕当前的截图，并且将其保存到硬盘的某个位置上，在点击结束录制后，如果选择保存视频，则会在后台通过调用FFmpeg将截取的图片转换成MP4形式的视频保存起来。如下图5-9 屏幕录制模块展示图，图5-10屏幕录制过程中展示图所示：

图5-9 屏幕录制模块展示图

图5-10屏幕录制过程中展示图

屏幕录制子模块的项目工程目录如下图5-11 屏幕录制模块项目工程目录所示：

图5-11 屏幕录制模块项目工程目录

config.py中设置项目的绝对路径代码如下所示：

```
ABSOLUTE_PATH = os.path.dirname(os.path.realpath(__file__)).replace(' \\',  
' /')
```

首先根据截图内容分析截图界面GUI的实现，在界面中主要包含一个标题，一个复选框，还有五个按钮，功能分别是全屏选取，截取屏幕某个区域，开始录制，结束录制，保存录制的MP4文件。首先创建一个继承自base.Py中的Base类ScreenBar类，Base类继承自QWidget，在Base类中定义了窗口的样式，{56%：以及鼠标按下事件和鼠标移动事件，鼠标按下的事件代码如下：}

```
def mousePressEvent(self, QMouseEvent):  
  
    self.mMoveing = True  
  
    self.mMovePosition = QMouseEvent.globalPos() - self.pos()  
  
    return super().mousePressEvent(QMouseEvent)
```

鼠标移动事件的代码如下所示：

```
def mouseMoveEvent(self, QMouseEvent):  
  
    if self.mMoveing and (QMouseEvent.buttons() and Qt.LeftButton)  
    and (  
  
        QMouseEvent.globalPos() - self.mMovePosition).manhattanLength() ]  
    QApplication.startDragDistance():  
  
        self.move(QMouseEvent.globalPos() - self.mMovePosition)  
  
        self.mMovePosition = QMouseEvent.globalPos() - self.pos()  
  
    return super().mouseMoveEvent(QMouseEvent)
```

除此之外在类中还定义了mMoving类成员变量，以及鼠标位置。ScreenBar类中在初始

化函数中首先将界面的布局进行初始化，主要代码如下：

```
self.box = QVBoxLayout()

self.btn_box = QHBoxLayout()

self.tip_label = QLabel(' 屏幕录制 ')

self.is_record_full = QCheckBox()

self.open_btn = QPushButton()

self.start_btn = QPushButton()

self.end_btn = QPushButton()

self.close_btn = QPushButton()

self.recording_thread = ScreenRecordThread()

self.bind()

self.set_style()
```

set\_style()函数对界面进行简单的样式处理，在bind()函数中定义了如下几个函数：

```
open_signal

start_signal

end_signal

record_full_signal
```

即在执行 bind函数后，在 ScreenBar类中定义了4个信号，并且在定义4个信号之后将按钮的事件触发动作与对应的信号相连相连，代码如下所示：

```
self.open_btn.clicked.connect(open_signal)

self.start_btn.clicked.connect(start_signal)

self.end_btn.clicked.connect(end_signal)

self.close_btn.clicked.connect(QCoreApplication.instance().quit)

self.is_record_full.stateChanged.connect(record_full_signal)

self.recording_thread.trigger.connect(self.thread_trigger_signal)
```

在这个类中还有一个 thread\_trigger\_signal函数，在函数内部定义了一个 show\_success\_message函数，当保存录制的视频时将会调用此函数，详细的代码如下所示：

```
QMessageBox.information(self, "提示", str, QMessageBox.Yes)

self.end_btn.setEnabled(False)

self.open_btn.setEnabled(True)

self.is_record_full.setEnabled(True)
```

除此之外录制的视频的保存过程在函数的后半部分实现，代码如下：

```
path, tmp = QFileDialog.getSaveFileName(self,

"文件保存",

's/video.mp4' ABSOLUTE_PATH,

"All Files (*)")

if not ' ' == path:

self.save_mp4_thread = SaveMp4Thread(list, fps, path, gif_list)

self.save_mp4_thread.trigger.connect(lambda: show_success_message("保存成功"))

self.save_mp4_thread.start()

else:

show_success_message("取消保存")

shutil.rmtree('temp')
```

当保存或者不保存当前生成的MP4后，在temp目录下的临时生成的截屏文件以及temp文件夹均会通过shutil模块删除掉。

在 app.py中还定义了另一个继承自 QWidget类的 ScreenLabel类，在这个类的初始化函数中，使用 Label标签将要截取的屏幕区域放置在其中，并且设置鼠标检测事件为 True实时检测鼠标的移动动作以及鼠标的点击动作。并且初始化鼠标开始的点的状态，以及记录鼠标终点的坐标，并且通过pixmap变量保存当前的屏幕所截取的区域图像。

{43%：再 ScreenLabel类中重写了键盘点击事件，鼠标按下事件，鼠标移动事件，} 鼠标松开事件，重绘事件，另外定义了一个获取指定区域的函数 getRect()用来获取当前需要记录的区域，并且返回此区域，详细的代码如下所示：

```
def getRect(self):

width = qAbs(self.m_beginPoint.x() - self.m_endPoint.x())

height = qAbs(self.m_beginPoint.y() - self.m_endPoint.y())

x= self.m_beginPoint.x() if self.m_beginPoint.x() [
self.m_endPoint.x() else self.m_endPoint.x() )
```

```
        y= self. m_ beginPoint. y() if self. m_ beginPoint. y( )[
self. m_ endPoint. y() else self. m_ endPoint. y( )

        selectedRect = QRect(x, y, width, height)

        if selectedRect.width() == 0:

            selectedRect.setWidth(1)

        if selectedRect.height() == 0:

            selectedRect.setHeight(1)

        return selectedRect
```

上述的几个事件再次便不再过多描述，接下来将重点分析 `screen_record_thread.py` 模块，在这个模块中主要通过 `PIL` 模块获取指定的屏幕截图，通过线程和 `moviepy` 模块将截取的大量的图片通过 `ImageSequenceClip` 模块生成 `MP4` 文件。在这个模块下有两个类，一个是继承自 `Qthread` 的 `ScreenRecordThread` 类，另一个是同样继承自 `Qthread` 的 `SaveMp4Thread` 类。

在第一个类中通过 `run` 函数来获取屏幕的截图并且保存到系统生成的默认的路径中，详细的代码如下文所示：

```
self.image_gif_list = []

while self.recording:

    curX, curY = win32gui.GetCursorPos()

    if self.area is None:

        image = ImageGrab.grab()

        image.paste(imCursor, box=(curX, curY), mask=imCursor)

    else:

        image = ImageGrab.grab(self.area)

        image.paste(imCursor, box=(curX - self.area[0], curY -
self.area[1]), mask=imCursor)

        imageName = os.path.join(' temp', ' s.jpg' int(time.time() *
1e3))

        image.save(imageName)

        self.image_list.append(imageName)

        self.recordTime = time.time() - t
```

```
fps = len(self.image_list) / self.recordTime

self.trigger.emit(self.image_list, fps, self.image_gif_list)
```

最后将截图完成的信息通过信号的方式发送出去，等待其他函数对其进行处理。

此模块中另外一个重要的类是 `SaveMp4 Thread`，在这个类中同样通过 `run()` 函数，同时使用 `ImageSequenceClip` 函数将生成的截图文件序列，生成指定帧数的 MP4 文件，并且写入文件中，方便以后的使用，以下是 `run` 函数的具体实现过程：

```
clip = ImageSequenceClip(self.list, fps=self.fps)

clip.write_videofile(self.path) # to video

self.trigger.emit()

shutil.rmtree('temp')
```

在处理完毕，将向系统中发射已经保存 MP4 文件的信号，{47%：并在之后使用 `shutil` 模块的 `rmtree` 函数删除 `temp` 文件夹以及文件夹中生成的临时的截图文件，} 整个屏幕录制过程结束。

#### 5.2.4 简易画板模块的实现

画板模块主要通过 `PyQt5` 模块的 `QtWidgets` 中的 `QApplication`，`QWidget` 模块，`PyQt5` 中的 `QtGui` 模块中的 `QPainter` 和 `QPen` 模块除此之外还有 `QtCore` 模块中的 `Qt` 模块，此版本的画板所实现的功能很有限，仅仅可以在窗口中绘制简单的线条，尚处于未完成的状态，{42%：具体实现的样子如图5-12简易画板模块的功能展示图所示：}

图5-12 简易画板模块的功能展示图

此版本的画板为之前的屏幕截图模块打下了一定的基础。下面将简述此版本的画板的具体实现过程，也间接还原之前截图中的简单绘画的过程。首先在模块中定义了一个继承自 `QWidget` 的 `Draw` 类。通过设置窗口的大小，和窗口出现的位置进行一定的调整，并且对画板窗口进行重新命名。

```
self.resize(600, 600)

self.move(100, 100)

self.setWindowTitle("画板")

{44%：设置鼠标轨迹检测设置为True，可以实时获取鼠标的位置信息，}

self.setMouseTracking(False)
```

并且使用 `pos_xy[]` 列表来保存鼠标移动后的点。在这个类中主要实现了以下几个函数：

1. 重写 `paintEvent` 函数

2. 重写鼠标移动事件



### 3.重写鼠标松开事件

paintEvent函数的具体实现代码如下所示：

```
painter = QPainter()

painter.begin(self)

pen = QPen(Qt.black, 2, Qt.SolidLine)

painter.setPen(pen)

if len(self.pos_xy) > 1:

    point_start = self.pos_xy[0]

    for pos_tmp in self.pos_xy:

        point_end = pos_tmp

        if point_end == (-1, -1):

            point_start = (-1, -1)

            continue

        if point_start == (-1, -1):

            point_start = point_end

            continue

        painter.drawLine(point_start[0], point_start[1], point_end[0],
point_end[1])

        point_start = point_end

    painter.end()
```

按住鼠标移动事件，将当前点添加到pos\_xy列表中，调用update函数也就是间接调用paintEvent函数，具体的代码如下文所示：

```
pos_tmp = (event.pos().x(), event.pos().y())

self.pos_xy.append(pos_tmp)

self.update()
```

通过重写鼠标按住后松开的事件，在每次松开后向列表中添加一个断点，然后在每次绘制时判断这个点是不是断点，如果是断点的话就跳过去，不和之前的点继续绘制，详细的代码如下文所示：

```
pos_test = (-1, -1)

self.pos_xy.append(pos_test)

self.update()
```

## 第六章 系统测试

### 6.1功能测试

{40%：功能测试的主要目的是检查所完成的模块是否能够正常工作，是否符合预期的工程项目的实际需求，} 基于 Python的互动课堂教师端主要的用户是各个学校的教师，所以本系统需要稳定状态下的平稳执行。

{55%：第一步通过单元测试，通过最微小的规模测试来检测某个功能或者代码块是否健全，} 通过测试样例检测内部程序的设计和编码的细节是否完善，逻辑是否合理。详细测试屏幕截取模块，区域选择，文字书写，图形绘制，画笔颜色改变以及画笔粗细的细小功能，通过点击屏幕录制的每个按钮来检查，界面 UI是否可用，调用网页模块，通过输入 url，刷新网页，前进，后退，终止刷新检查网页模块的功能能否正常使用。最后检查画板模块，画笔是否正常，绘制的图像是否有所关联，图像显示是否存在问题。

{48%：第二步通过集成测试将已经测试过的单元模块组合成一个组件之后测试二者之间的接口，检测接口是否完备。} 通过将单元测试的结果两个两个互相组合成新的集成测试集，检测功能是否齐全，界面 UI是否正常显示，每个模块的单独子功能能否正常使用，尽可能通过集成测试来识别出二者之间相互组合时所发现的问题， {59%：通过使用要求在组合单元前测试每个模块单元，并确保每个模块单元都可以承受得住在不同情况下的考验。}

{45%：第三步通过系统测试来检测所需要测试的系统，} 测试的环境是 Windows10.0.16299.371，系统的处理器为 Intel( R) Core( TM) I5-3470 CPU@3.20 GHZ，内存为 DDR38 G大小的常规内存，速度是1600 MHZ，硬盘是希捷120 G固态硬盘500 G东芝机械硬盘组合，网络是20 M有线宽带。 {47%：系统测试的测试是基于需求分析的需求说明书进行测试的黑盒类测试，通过将整个系统组装起来进行集中测试，} {51%：来验证系统需求是否满足需求分析阶段的关于需求规格的定义，找出与需求分析不相符或者与需求分析结果相互矛盾的地方。} 通过检测网页模块，画板模块，屏幕截取模块，屏幕录制模块，在通过将系统资源的有关依赖注入进来，在系统的实际环境下进行测试，截图模块基本上符合需求分析所分析的结果，屏幕录制模块在进行全屏录制的情况下可以正常进行工作，在进行局部区域选取录制时，存在一定的 bug，并且在不当的操作下可能会引起程序的不正常关闭。屏幕录制模块通过简单的测试，基本上可以满足教师在浏览网页上的需求。画板在进行系统测试时，在文本输入的功能上，可以满足需求，但是画板的整体是一个仍需补充的状态，需要对画板增加一定的功能才可以保证画板功能的完善性。

### 6.2性能测试

{42%：性能测试的主要目的是在不同的测试环境下验证软件在不同的环境条件以及系统条件下能否满足性能需求的指标，} 或者检查在执行新版本的应用程序时，会不会比旧版本的应用程序的效率低下，通过交替进行的负荷测试以及强迫测试，测试软件在艰苦条件下表现，通过数据来变现，软件在不同条件下所表现出来的性能情况。

对教师端系统进行测试时，主要测试网页模块的不断重复刷新网页的情况下，程序能否依

旧保持稳定的状态以及优秀的表现力，由于这一部分在 PyQt5发布之前已经经过一定的测试，在此并不需要过多的陈述。在屏幕截图的压力测试中，不断的对屏幕进行截图，以及在图片上书写以及绘画，程序均能平稳的正常运行，未见程序的宕机状态。在屏幕录制模块的压力测试中，屏幕录制的过程基本上不会影响到教师机客户端的运行效果，通过录制长视频，会对硬盘大小的需求有一定的限制，并且在视频录制结束后，将截屏图片转换成视频的过程需要一定的时间的等待。

## 第七章 总结与展望

### 7.1 总结

基于Python的互动课堂的应用实现了屏幕截取模块，屏幕录制模块，将网页内容内嵌到客户端中，简单的画板模块。{50%：在论文的第一章，阐述了互动课堂的研究背景与研究意义，国内外发展现状，} 互动课堂的目标与意义，介绍了开发过程所需要的研究方法与技术路线。在论文的第二章中，介绍了互动课堂开发所用到的开发工具，开发语言以及开发框架。第三章，介绍了互动课堂的需求分析，并且绘制了分析得到的用例图，活动图，并对其进行详细的阐述。第四章通过对第三章需求分析所得到的内容，进行详细的概要设计，详尽描述了网页浏览模块，屏幕截取模块，屏幕录制模块，简单画板模块的整体功能。论文的第五章，通过对上述四个模块进行实现细节的分析，以及项目相关的代码，简要描述系统开发过程中的环境配置以及各个功能模块的具体实现。{45%：第六章，介绍在客户端测试过程中所使用的测试方法包括，单元测试，集成测试，系统测试以及性能测试，} 来综合评估系统在压力测试方面的表现力，以及一些测试过程和主要功能点。

### 7.2 展望

教育决定着国家的未来，本系统尚处于开发阶段，拥有很多的不完善，期待在后续的继续开发过程中，可以互动课堂系统完整的开发出来，用来推进“翻转课堂”在我国的普及进程，减少各个省市之间教育资源的差异化，整体提高全国的教育质量的平均水平，以此增强国家的后备资源的储备，间接增强我国的平均国民文化水平，提高国民文化素质。

## 参考文献

- [1]肖文鹏.用PyQt进行Python下的GUI开发[J]. 中文信息, 2002(07): 73-75.
- [2]Python语言学习之我见[J]. 赵彦琦, 谢小西, 荀宇畅. 电子制作. 2013(13)
- [3]肖旻, 陈行.基于Python语言编程特点及应用之探讨[J]. 电脑知识与技术, 2014, 10(34): 8177-8178.
- [4]金繁, 崔培雷.嵌入式Qt中信号与槽机制的研究[J]. 电子设计工程, 2014, 22(24): 168-170.
- [5]刘丽霞, 边金松, 张琍, 穆森.基于FFMPEG解码的音视频同步实现[J]. 计算机工程与设计, 2013, 34(06): 2087-2092.
- [6]王学斌. 软件工程中基于模型驱动架构的模型转换技术研究[D]. 国防科学技术大学, 2006.
- [7]田爱丽, 刘冬梅.论教学微视频编制的原则与方法[J]. 上海教育科研, 2014(08): 7-10.

- [8]于青青,李晓明.MOOC视频制作平民化方法的探索与评价[J].现代远程教育研究,2015(01): 106-112.
- [9]马洪堂.基于FFMPEG的视频转换系统的模块研究[J].电脑知识与技术,2009,5(13): 3545-3546.
- [10]丁峰. 嵌入式Linux平台下基于FFmpeg库视频格式转换设计[D]. 哈尔滨工程大学,2013.
- [11]张茗芳.动态语言Python探讨与比较[J].企业科技与发展,2012(13): 57-60.
- [12]黄宇东,胡跃明,陈安.基于Qt的多线程技术应用与研究[J].软件导刊,2009,8(10): 40-42.
- [13]万长胜,金革,王坚,李昔华,郝黎凯,黄鲲,李峰,任间.用 Qt进行面向对象软件的设计与开发[J].核电子学与探测技术, 2004(03): 311-313.
- [14]软件工程导论[M]. 清华大学出版社 , 张海藩, 2007
- [15]朱吉佳,蔡家麟.基于Qt的业务监控系统界面设计与实现[J].计算机技术与发展,2008(03): 236-239 242.
- [16]席文强. 基于FFmpeg的高清实时直播系统设计与实现[D]. 长安大学,2017.
- [17]王学斌. 软件工程中基于模型驱动架构的模型转换技术研究[D]. 国防科学技术大学,2006 .
- [18]陈周国,王胜银,付国晴,邓赟.基于Linux QT技术的远程监控GUI设计[J].通信技术,2009,42(12): 234-236.
- [19]万长胜,金革,王坚,李昔华,郝黎凯,黄鲲,李峰,任间.用 Qt进行面向对象软件的设计与开发[J].核电子学与探测技术, 2004(03): 311-313.
- [20]用PYQT进行Python下的GUI开发[J]. 中文信息 . 2003(02)

## 致谢

转眼之间大学四年的学习生涯马上就要接近尾声,昨日种种如今回想起来,有喜悦,有辛酸,内心还真有些不舍。 回顾四年的美好时光,与老师,同学之间的点滴仍然历历在目,但这却已然成为过去。 我很幸运在这四年中,承蒙学院每一位老师和学长学姐的悉心教导, 在此,向由衷的向这些默默为其他人奉献的人说一声谢谢!

特别是我的毕业设计指导教师张慧杰教授,感谢您在百忙之中仍能在选题,开题报告,以及论文撰写过程中所给我带来的帮助与鼓励,我会牢记您所传授给我的人生经验。 在这次毕业设计的完成过程中,有许多的辛酸与难点,其中面临某些问题的时候, 也曾想过放弃,但是在老师的耐心指导下,从一开始的手足无措到慢慢产生想法, 最后将模块可以通过我的努力一点一点的实现出来。 在这个过程中,我收获到的成功的果实以及坚持做好一件事态度是十分宝贵的,在此由衷的向我的指导老师张慧杰老师表示感谢! 我将在以后的人生中,勤奋刻苦,脚踏实地来报答您对我的帮助。

还要感谢在我论文撰写以及项目开发过程中帮助我,鼓励我,给我坚实后盾的同学们,你

们的真心帮助对我来讲是无价的，谢谢你们!

{61%：后向在百忙中仍然抽出时间评审本文的评审教师表示感谢!}

检测报告由PaperPass文献相似度检测系统生成

Copyright 2007-2018 PaperPass