**“隔镜杯”数据爬虫开发大赛**

**作品介绍文档**

**作品题目： 英雄联盟游戏沉迷自我分析系统**

**成员姓名： 季甜甜、陈榉烁、张斌**

**电子邮箱： zb2017051700@stu2017.jnu.edu.cn**

**提交日期： 2020年7月29日**

**目 录**

**1 引言 1**

1.1 编写目的 1

1.2 背景 1

1.3 运行环境 2

**2 爬虫技术 3**

2.1 Selenium工具 3

2.2 Selenium基本用法 4

**3 实现方法 4**

3.1 爬虫实现 4

3.2 数据分析 6

**4 运行与测试 12**

4.1 运行步骤 12

4.2 时间性能 12

**5 数据应用介绍 14**

5.1 游戏次数分析 4

5.2 游戏时间段分析 4

5.3 游戏习惯分析 4

# 1 引言

## 1.1 编写目的

《英雄联盟》（简称LOL）是由美国拳头游戏（Riot Games）开发、中国大陆地区腾讯游戏代理运营的英雄对战MOBA竞技网游。游戏里拥有数百个个性英雄，并拥有排位系统、符文系统等特色养成系统。《英雄联盟》致力于推动全球电子竞技的发展，形成了自己独有的电子竞技文化，广受大量青少年的喜爱。

然后随着游戏的迅猛发展，“网络游戏上瘾”问题也越来越受社会的关注，过量的游戏时间将会对玩家带来极大的负面影响，因此需要对玩家的游戏数据进行爬取和分析，以达到防沉迷的效果。

本作品利用爬虫技术，对玩家在《英雄联盟》里的游戏信息进行爬取，进而进行数据分析，将分析结果反馈给玩家，并提出合理的建议，玩家以此来监督和管理自己的游戏时间，起到良好的“防沉迷”效果。

系统根据爬取信息进行的分析如下：

1、分析每月游戏局数（至少/平均花费时间）

2、分析游戏的时间段

3、分析不健康游戏习惯

4、根据玩家特点，为其动态给出建议

5、显示详细对局情况

## 1.2 背景

**1、开发软件系统名称：**英雄联盟防沉迷系统

**2、开发者：**暨南大学信息科学技术学院本科生

**3、系统使用者：**英雄联盟玩家

## 1.3 运行环境

|  |  |
| --- | --- |
| 操作系统 | Windows10 |
| 运行配置 | (1)Python; (2)selenium; (3)csv; (4)pyqt5; |

**2 爬虫技术**

**2.1 Selenium工具**

Selenium 是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括IE（7, 8, 9, 10, 11），Mozilla Firefox，Safari，Google Chrome，Opera等。Selenium 是一套完整的web应用程序测试系统，包含了测试的录制（Selenium IDE）,编写及运行（Selenium Remote Control）和测试的并行处理（Selenium Grid）。

Selenium的核心Selenium Core基于JsUnit，完全由JavaScript编写，因此可以用于任何支持JavaScript的浏览器上。Selenium可以模拟真实浏览器，自动化测试工具，支持多种浏览器，爬虫中主要用来解决JavaScript渲染问题。

**2.2 Selenium基本用法**

**1、声明浏览器对象**

Selenium支持多个浏览器，也支持手机端的浏览器，除此之外还有Phantomjs，本作品创建的是谷歌浏览器对象（如下），依次类推可以得到其他浏览器对象。



**2、Selenium元素定位**

寻找单个节点的多种方法，返回的结果是WebElement类型的

如果是找多个节点，在element后面加s，结果是列表类型

|  |  |  |
| --- | --- | --- |
| **定位单个元素** | **定位多个元素** | **含义** |
| find\_element\_by\_id | find\_elements\_by\_id | 通过元素id定位 |
| find\_element\_by\_name | find\_elements\_by\_name | 通过元素name定位 |
| find\_element\_by\_xpath | find\_elements\_by\_xpath | 通过xpath表达式定位 |
| find\_element\_by\_link\_text | find\_elements\_by\_link\_tex | 通过完整超链接定位 |
| find\_element\_by\_partial\_link\_text | find\_elements\_by\_partial\_link\_text | 通过部分链接定位 |
| find\_element\_by\_tag\_name | find\_elements\_by\_tag\_name | 通过标签定位 |
| find\_element\_by\_class\_name | find\_elements\_by\_class\_name | 通过类名进行定位 |
| find\_elements\_by\_css\_selector | find\_elements\_by\_css\_selector | 通过css选择器进行定位 |

**3、浏览器操作方法**

|  |  |
| --- | --- |
| **方法** | **说明** |
| set\_window\_size() | 设置浏览器的大小 |
| back() | 控制浏览器后退 |
| forward() | 控制浏览器前进 |
| refresh() | 刷新当前页面 |
| clear() | 清除文本 |
| send\_keys (value) | 模拟按键输入 |
| click() | 单击元素 |
| submit() | 用于提交表单 |
| get\_attribute(name) | 获取元素属性值 |
| is\_displayed() | 设置该元素是否用户可见 |
| size | 返回元素的尺寸 |
| text | 获取元素的文本 |

**4、获取结点信息**

获取节点信息与两种方法，第一种是使用page\_source属性，获取到目标网页的源代码之后，使用正则表达式，css,xpath ,bs4等工具进行抓取信息。第二种就是直接使用selenium的一些方法和属性。

(1) 获取属性：首先选中要匹配解析的节点，然后调用get\_attribute方法来获取节点的属性；

(2) 获取文本：依然是先打开网址，然后定位到目标节点，再使用text属性来获取文本；

(3) 获取ID，位置，标签名和大小：方法和前两个类似，直接调用即可获得相关的值。

**5、切换frame**

网页有一种很常见的节点叫做iframe，相当于是子frame，也就是页面的子页面。

结构和外部网页的结构一致，当我们使用selenium打开页面，默认是在父页面里执行的，但是呢，这种网页我们往往是获取不到子页面的节点，所以要使用switch\_to.frame（）来跳转页面，然后再进行相对应的操作。

之后尝试获取父页面的某个节点，如果获取不到就报错，接下来就换回父页面去获取该节点，就能成功了。

**6、延时等待**

有些时候，网页有额外的Ajax请求，并没有那么快就能加载出来，所以我们需要等待。

等待分为两种，隐式等待和显式等待

(1) 隐式等待：查找某个节点的时候，隐式等待就会等待固定的时间，如果到了时间还没有来，就会出现报告操作。

(2) 显示等待：在等待期间会进行提示。

**7、拖动滑动条**

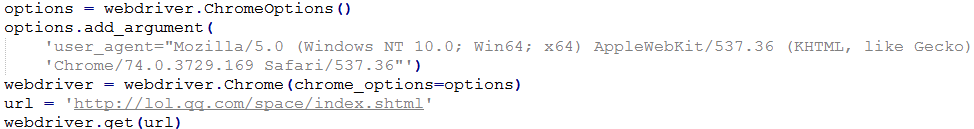
使用爬虫爬取网页时，经常会看到使用滑动条的页面，然后会显示加载中，没多久就会加载出新的页面出来。如果直接爬取，往往只能爬到一页甚至是前几页的信息。

在此可以直接模拟运行JavaScript，滑到网页的最底下。使用的是execute\_script()，并且可以设置一个网页提示框

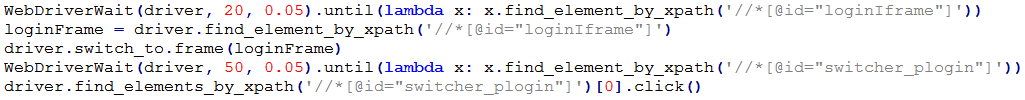
**3 实现方法**

**3.1 爬虫实现**

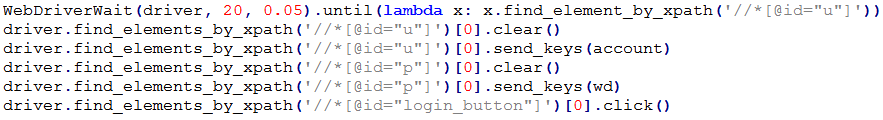
**1、访问页面**



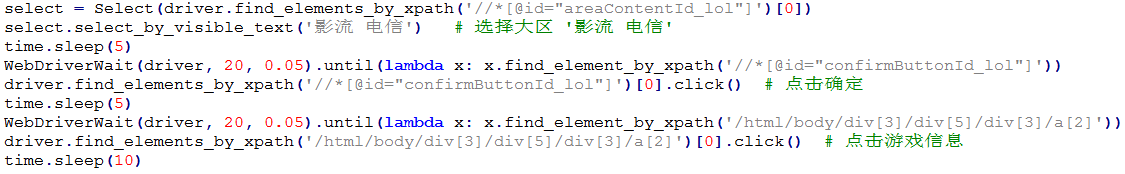
**2、登录frame**



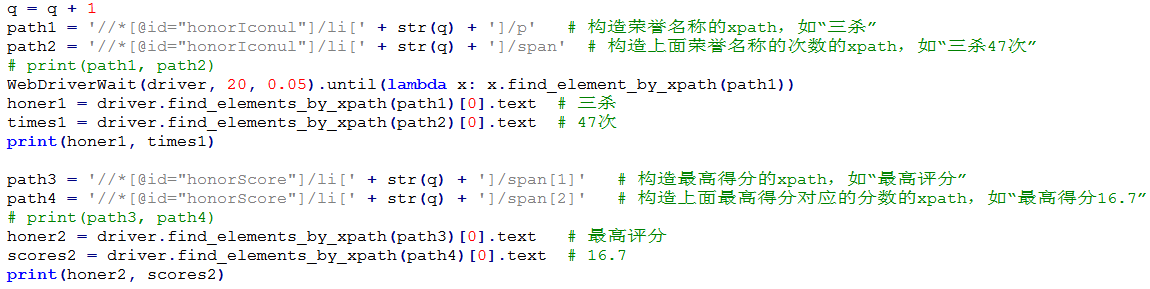
**3、账号登录**



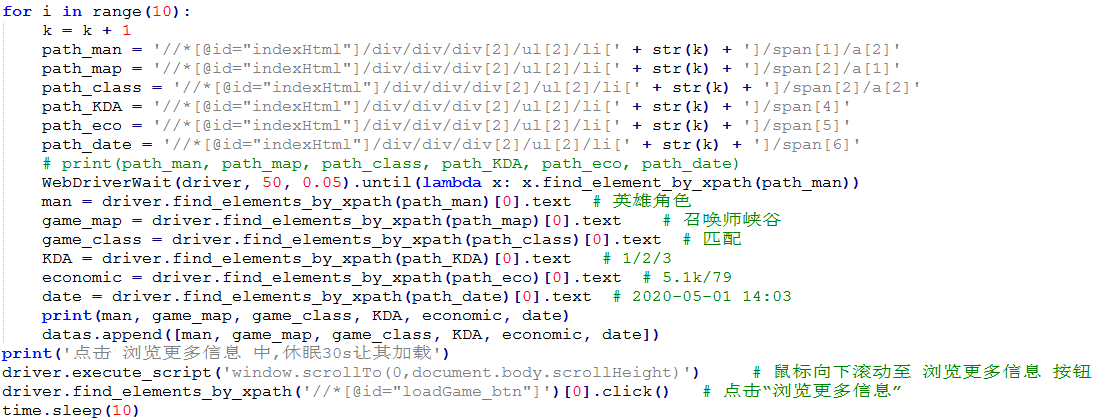
**4、进入个人中心**



**5、爬取信息**

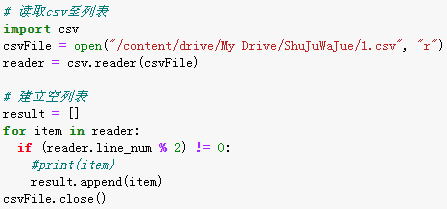


**6、爬取单局信息**

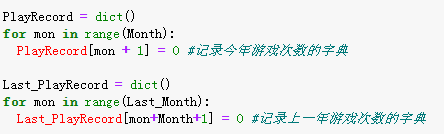


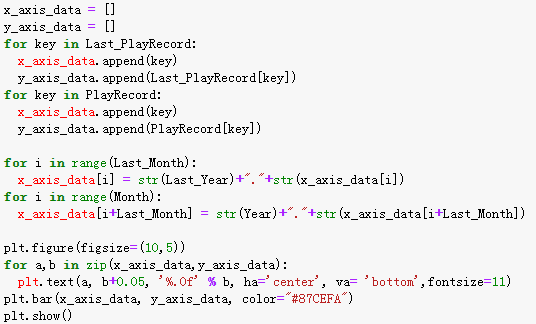
**3.2 数据分析**

**1、读取爬到的信息**

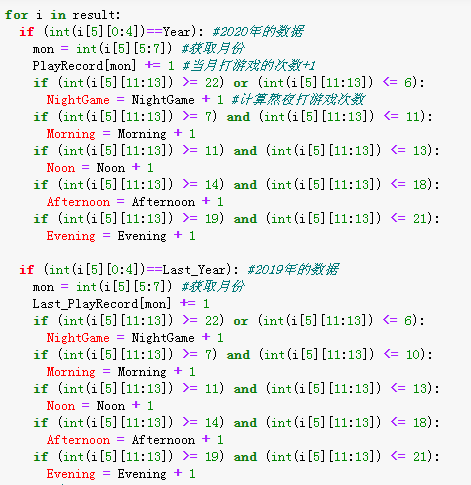


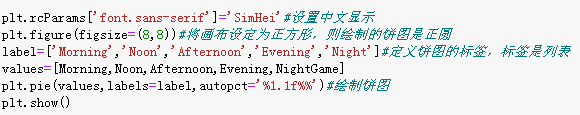
**2、游戏时间分析并绘制柱状图**





**2、游戏时间段分析并绘制饼状图**



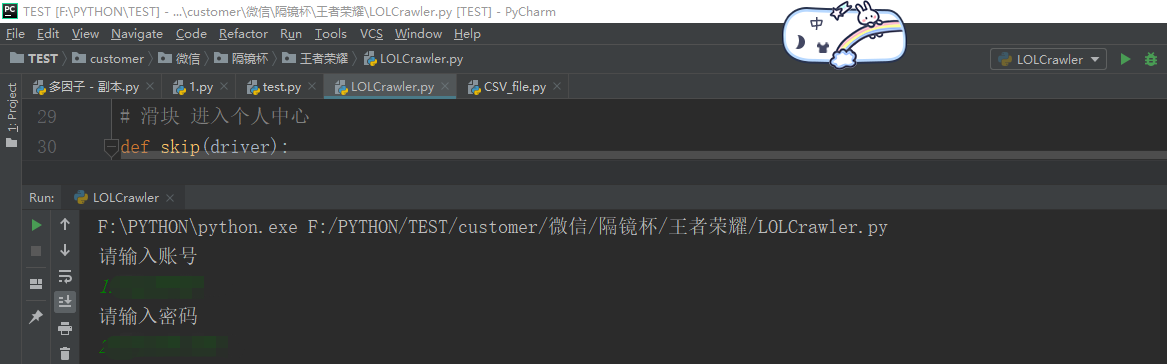


# 4 运行与测试

## 4.1 运行步骤

**1、根据系统指示输入账号密码，系统自动登录并进行安全验证**

用户在终端输入账号密码，之后程序会自动在登录界面输入账号密码

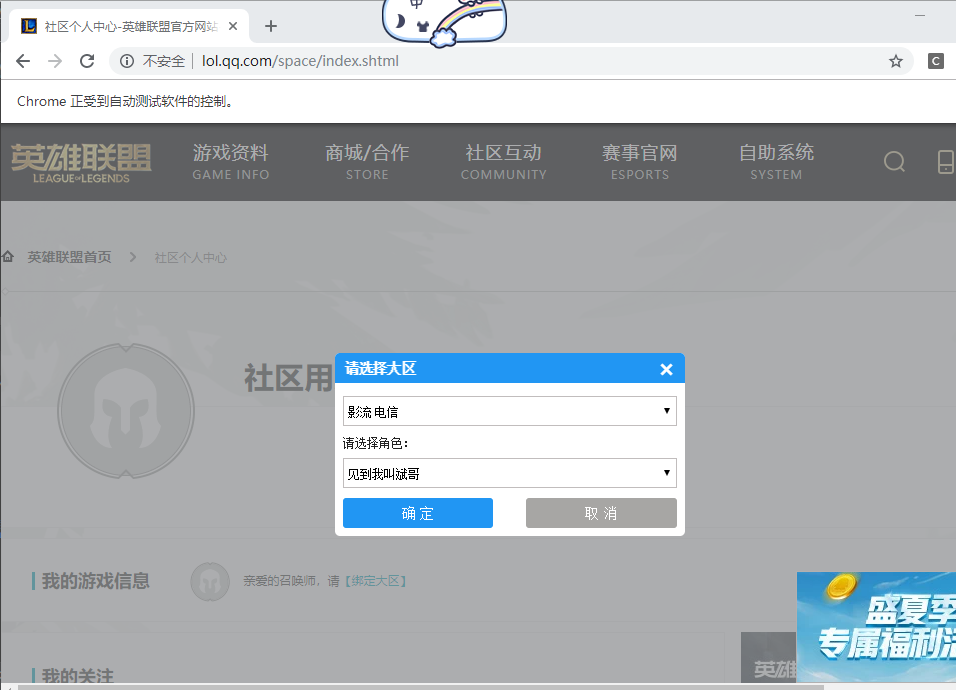


程序自动输入账号密码进行登录

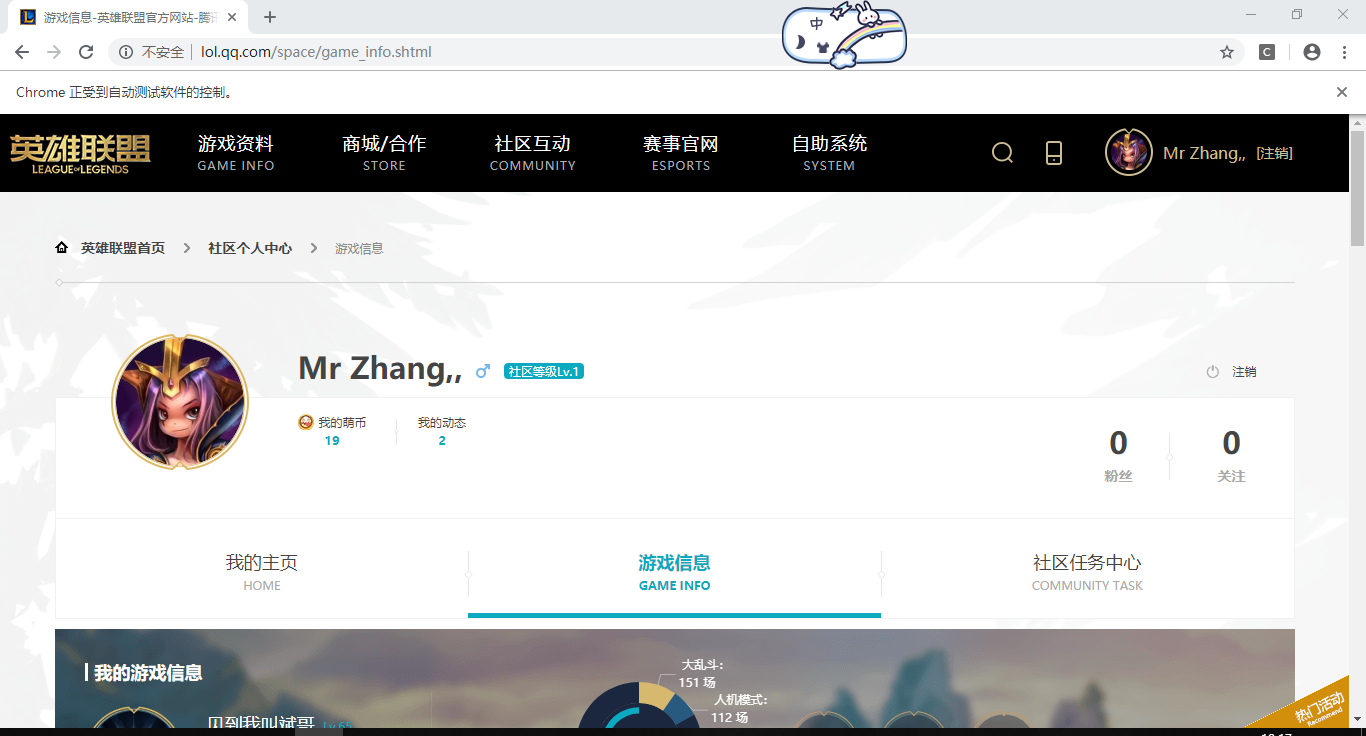


**2、系统自动选区并进入个人中心**

**程序自动选择大区并点击确定**

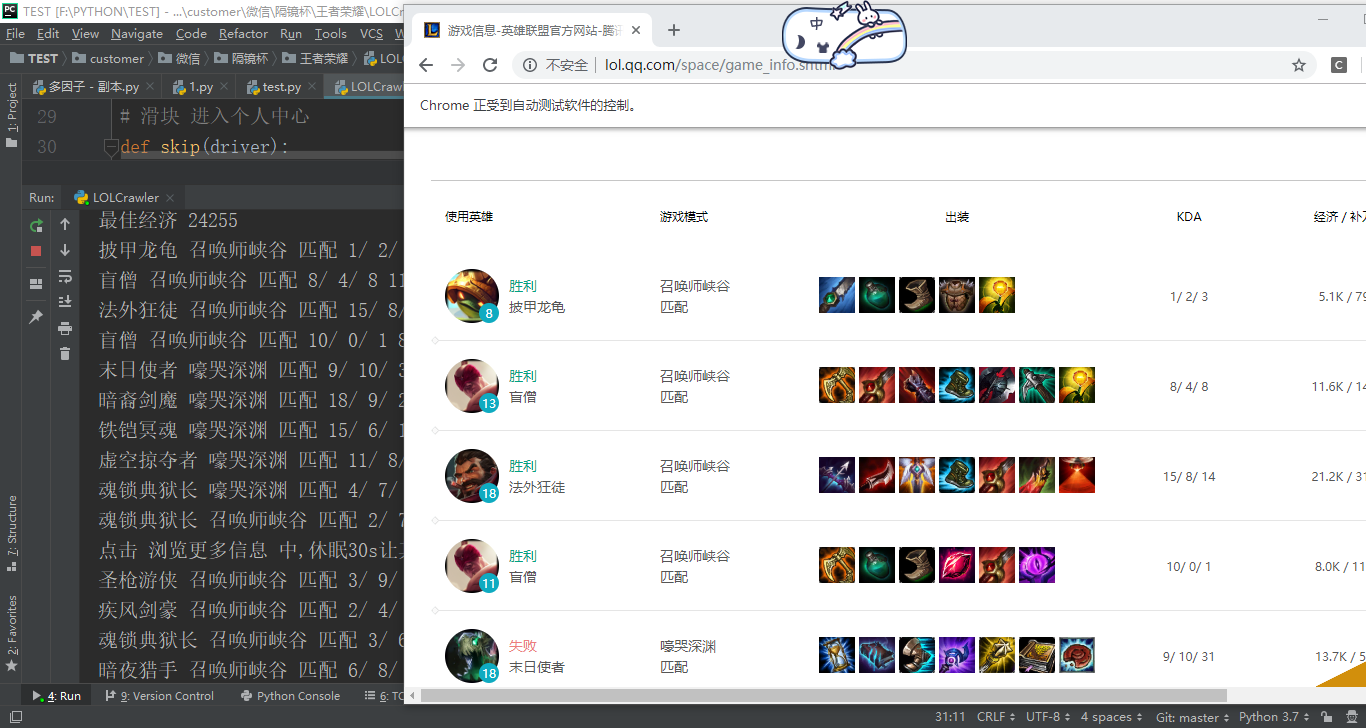


程序自动点击游戏信息进入需要爬取数据的页面



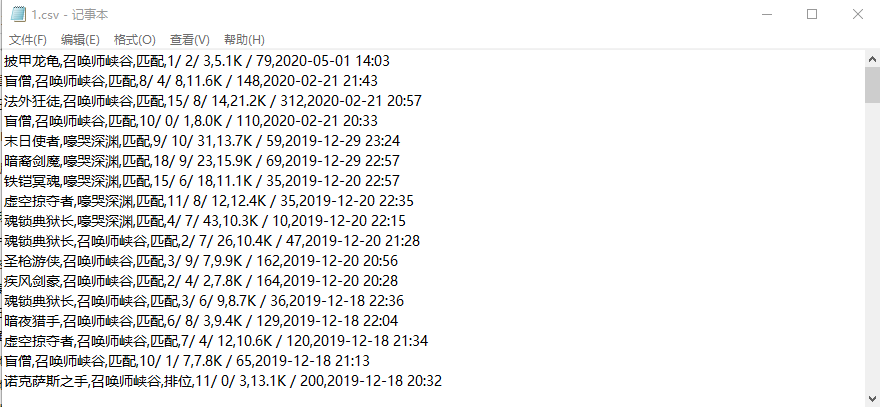
**3、爬取所需信息**





**4、将信息写入csv文件，进行数据分析**

**程序将数据写入csv文件中，为下一步的数据分析做准备**



# 5 数据应用介绍

## 5.1 游戏次数分析

通过对爬取的数据进行分析，统计近一年来该用户每个月进行的游戏次数，并以柱状图的形式绘制，以了解该用户近一年来的游戏习惯。

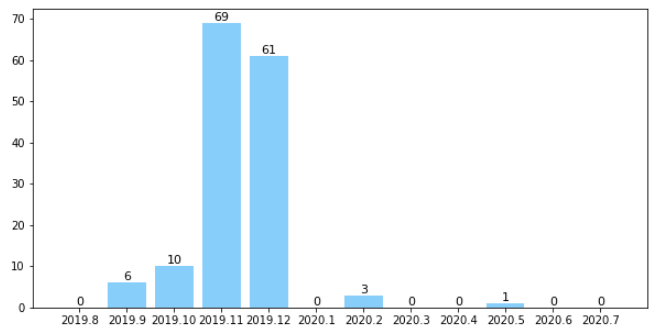


图5-1 游戏次数柱状图

## 5.2 游戏时间段分析

通过对爬取的数据进行分析，统计该用户主要进行的游戏时间段，并以饼状图的形式反馈给用户，让用户能清晰地了解自己的游戏规律，以便后续的不良习惯分析。

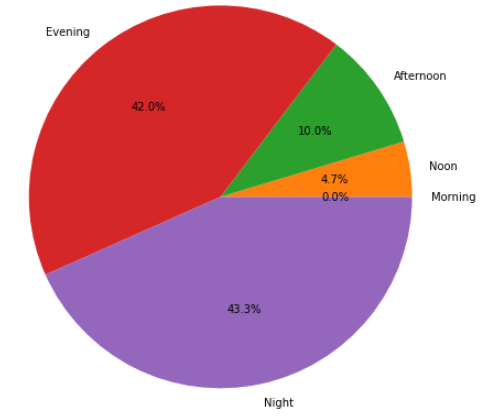


图5-2 游戏时间段饼状图

## 5.3 游戏习惯分析

对全年的游戏局数进行统计，分析至少/平均花费的游戏时间，并且分析该用户不健康的游戏习惯，根据玩家特点动态给出建议。

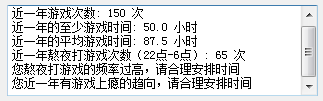


图5-3 游戏习惯分析

最终以用户界面的方式反馈给用户。

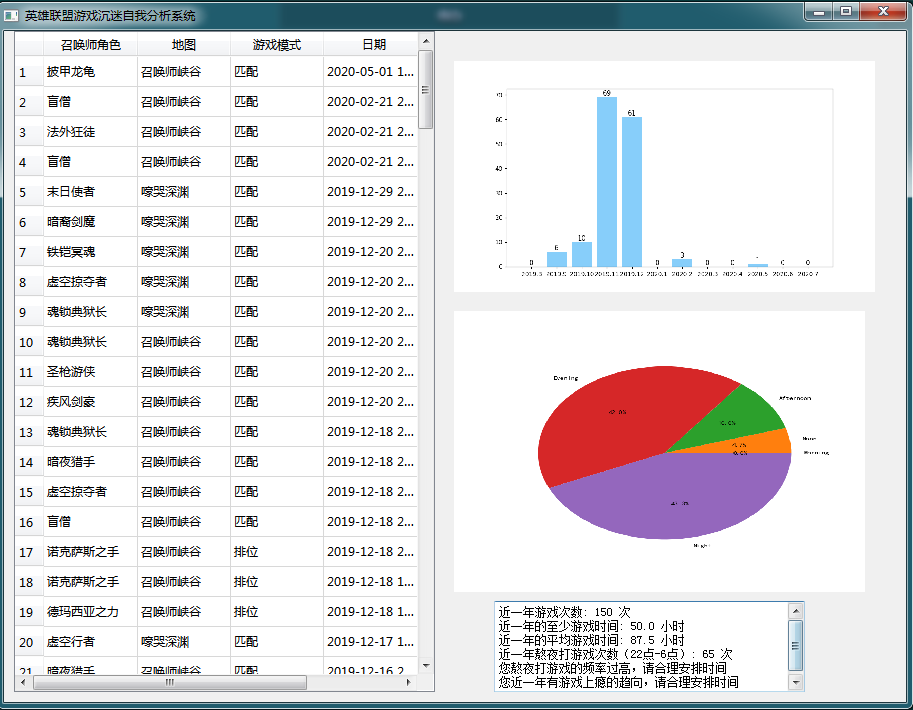


图5-4 用户界面反馈