

基于 MVC 架构的足球联赛数据可视化分析软件

张懿 沈成泽 林宇萧 万钧涵 李嘉俊

一、问题及背景

竞技体育作为历久不衰的娱乐项目，一直拥有广大的受众群体。而足球作为“世界第一运动”，其关注度在所有竞技体育项目中高居前列。随着近年来观众数量的增长，球迷们对于专业数据分析的需求也有所增加，本软件的设计目的即为帮助球迷获取最新的联赛数据统计以及可视化的图表分析，最终获取需要的联赛信息和大致走势。

二、设计思路及技术难点

本程序的几个主要模块分别为数据获取，数据可视化分析及结果展示（UI 界面）。首先从数据源网站通过爬虫方法将数据获取到本地，之后按照设计的数据存储结构进行分类封装和保存。用户在 UI 界面的操作会作为指令传给数据控制模块，之后由控制模块从本地数据中获取用户指定的数据，交由可视化模块进行绘制，最后返还给界面进行展示。

本程序的技术难点有二，其一为数据获取。大多数权威数据网站对于爬虫程序有所防范，我们的数据源网站也不例外。因此我们这里利用了 python 环境下的模拟浏览器功能，利用 chrome 浏览器对特定数据页面实现自动访问和数据爬取。

三、开发计划及分工

整体开发计划分为以下几个模块，分工一并给出：

- 1、数据源的获取以及数据结构设计：沈成泽 李嘉俊
- 2、可视化对象（数据图表）的控制功能：林宇萧 万钧涵
- 3、UI 界面设计及连带的控制器模块：张懿

几个模块的设计可以并行，不过由于可视化模块及部分界面元素依赖数据进行测试，由负责数据处理端的同学提供了简单的样本数据（数据量较小）。

在最后进行模块拼接处理，由于数据获取模块相对独立，因此不需要特别进行拼接。



在基础的 MVC 架构中，View 层的主要内容为 qml 设计的 UI 界面元素以及通过 Qchart 实现的数据可视化模块。control 模块接收由用户点击信号在 view 层自动生成的字符串信号，由于本程序用户端不会更改数据库，我们的 model 模块逻辑十分简单，只需根据 control 给出的字符串信号返回特定数据即可，因此这里 model 模块与 control 模块的连接几乎被省略。

四、效果讲解

























3.1 数据获取模块及数据结构

数据获取模块被设计为通过驱动自动批量访问数据源网页并获取数据。

获取的数据以文件的形式存储于根目录下。

DeJia	2017/7/14 8:54	文件夹	
DeJiaPlayer	2017/7/14 8:54	文件夹	
teamlst	2017/7/14 8:54	文件夹	
Xijia	2017/7/14 8:54	文件夹	
XijiaPlayer	2017/7/14 8:54	文件夹	
Yijia	2017/7/14 8:54	文件夹	
YijiaPlayer	2017/7/14 8:54	文件夹	
Yinchao	2017/7/14 8:54	文件夹	
YinChaoPlayer	2017/7/14 8:54	文件夹	
DeJia.txt	2017/7/14 8:54	TXT 文件	2 KB
XiJia.txt	2017/7/14 8:54	TXT 文件	2 KB
YiJia.txt	2017/7/14 8:54	TXT 文件	2 KB
YinChao.txt	2017/7/14 8:54	TXT 文件	2 KB

如图，以联赛名命名的文件夹下存储的是该联赛的近期比赛数据，以联赛名+“Player”命名的是该联赛内所有球员的赛季数据。根目录下的 txt 文件为联赛的积分榜数据。

 Augsburg.json	2017/7/14 8:54	JSON File	6 KB
 Bayer Leverkusen.json	2017/7/14 8:54	JSON File	4 KB
 Bayern Munich.json	2017/7/14 8:54	JSON File	4 KB
 Bayern.json	2017/7/14 8:54	JSON File	2 KB
 Borussia Dortmund.json	2017/7/14 8:54	JSON File	6 KB
 Borussia M.Gladbach.json	2017/7/14 8:54	JSON File	6 KB
 Darmstadt.json	2017/7/14 8:54	JSON File	6 KB
 Eintracht Frankfurt.json	2017/7/14 8:54	JSON File	6 KB
 FC Cologne.json	2017/7/14 8:54	JSON File	6 KB
 Freiburg.json	2017/7/14 8:54	JSON File	6 KB
 games.json	2017/7/14 8:54	JSON File	11 KB
 Hamburg.json	2017/7/14 8:54	JSON File	2 KB
 Hamburger SV.json	2017/7/14 8:54	JSON File	4 KB
 Hertha Berlin.json	2017/7/14 8:54	JSON File	6 KB
 Hoffenheim.json	2017/7/14 8:54	JSON File	6 KB
 Ingolstadt.json	2017/7/14 8:54	JSON File	6 KB
 Leverkusen.json	2017/7/14 8:54	JSON File	3 KB
 Mainz 05.json	2017/7/14 8:54	JSON File	4 KB
 Mainz.json	2017/7/14 8:54	JSON File	3 KB
 RasenBallSport Leipzig.json	2017/7/14 8:54	JSON File	7 KB
 Schalke 04.json	2017/7/14 8:54	JSON File	3 KB
 Schalke.json	2017/7/14 8:54	JSON File	3 KB
 Werder Bremen.json	2017/7/14 8:54	JSON File	6 KB
 Wolfsburg.json	2017/7/14 8:54	JSON File	6 KB

以德甲数据为例，文件夹内分球队保存了各队的比赛数据，这种保存模式会导致一定的数据

冗余（每场比赛被保存两次），但可以提高数据索引的效率。

球员数据的保存模式与之同理。

3.2 UI 界面



UI 界面的左上角为联赛选择区，用户点选时，被选择的联赛将以字符串形式传入控制区。

	球队	胜	平	负	积分
1	Real Madrid	29	6	3	93
2	Barcelona	28	6	4	90
3	Atletico Madrid	23	9	6	78
4	Sevilla	21	9	8	72
5	Villarreal	19	10	9	67
6	Real Sociedad	19	7	12	64
7	Athletic Club	19	6	13	63

此时控制区经过筛选后返回该联赛积分榜数据给 view 层并显示。通过点选积分榜可以选择特定的球队并观看其数据分析。

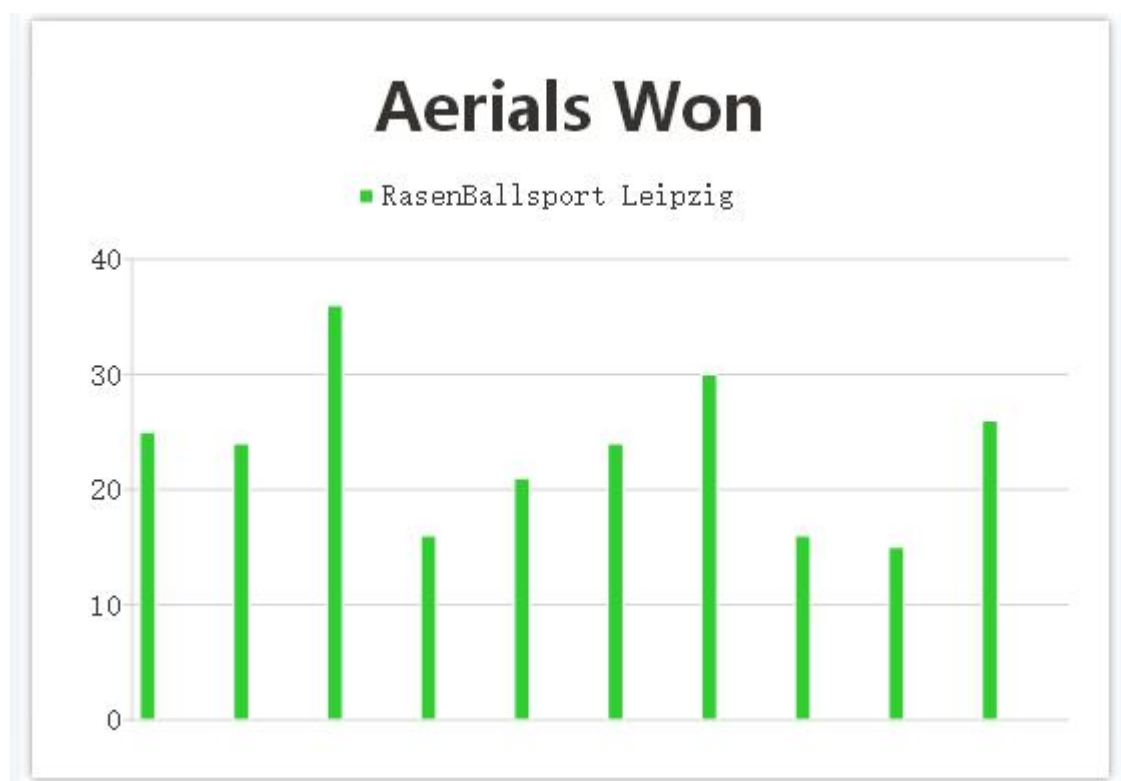


Real Madrid

Pass Accuracy	86.5%
Dribbles per game	11.7
Tournament	La Liga
Shots per game	17.4
Avg. Possession	54.8%
Goals per game	2.8
Seasons	2016/2017
Tackles	20.4

此处为球队信息区，根据积分榜的选择，view 层传给控制区另一个字符串，控制区据此查询该球队的宏观数据并展示给用户。此处的锁定功能用于对比分析，此处不展开。

3.3 数据可视化分析效果

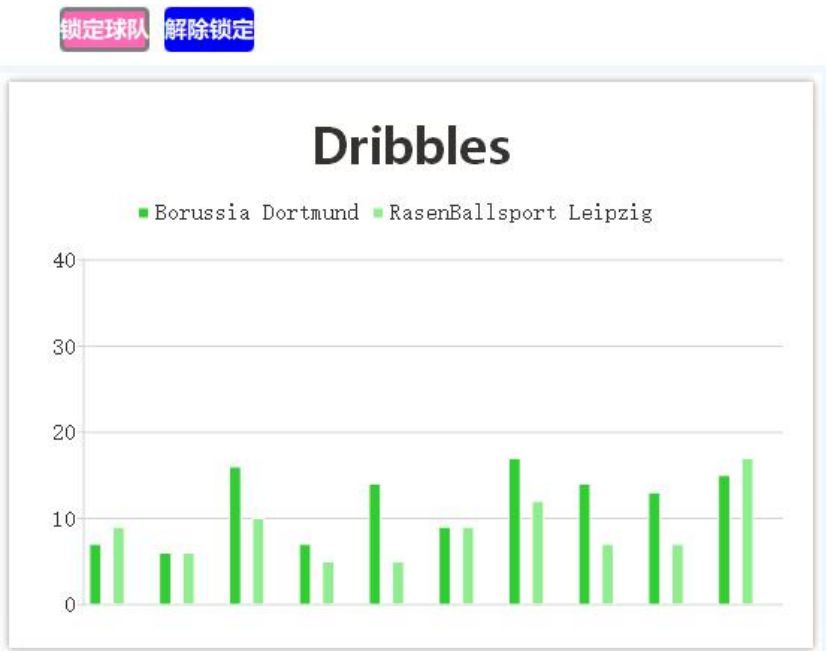


这里是可视化分析的直观展示。我们通过联赛和球队数据定位该球队的近期比赛数据，并按照比赛的时间轴数据以图表形式展现在分析区。此处为一支球队近期比赛的争顶数据图。



对于具有较为明显的趋势性的数据，我们这里采用折线图的方法进行描述，这里是该球队近

期传球成功率的图表描述。



之前提及的锁定功能在这里可以得到体现。点击锁定后，当前球队的数据会被作为对比样例保存，再点击其他球队即可看到两者的数据对比。点击解除锁定即可解除对比。



折线图也具有相同的功能。

五、最终运行效果图



整体运行效果如图，各功能均可正常执行。

