

程序功能介绍

环境配置

本次项目需要用到以下第三方库：`pypinyin`、`PyQt5`。

建议使用 python 3.10/3.11 版本测试代码。

下载 `pypinyin` 指令如下：

```
pip install pypinyin
```

下载 `PyQt5` 指令如下：

```
pip install PyQt5
```

功能与使用方式

在这个项目中，我们实现了一个支持根据 `tag` / 名称搜索社团，添加收藏，支持增加 `tag`，并且能根据历史搜索记录进行用户画像针对性推荐社团的社团推荐系统。

用以下指令可以测试代码功能。

```
python test_qt.py
```

弹出界面后，可以在文本框内输入名称/标签及其拼音缩写和谐音词来搜索对应社团。

对于搜索到的社团，后台将记录搜索频率进行用户画像从而分析用户偏好。

同时可以单击社团对应的空心五角星标志收藏，被收藏的社团可以通过“我的收藏”进行查看，收藏后，五角星变为实心，单击可以单个取消收藏，“清空收藏夹”可以撤销所有收藏。

在每个社团的文本框中输入文本后，点击“tag维护”可以增加tag，单击“还原tag”可将所有 tag 重置。

“清空历史记录”可以清零所有历史搜索数据。

单击每个社团对应的“详情”可以查看对应的详细资料。

文件说明

核心功能文件：

[algorithm.py](#) 后端算法

[test_qt.py](#) 前端界面

[tag.json](#) 当前 tag 存储

[prefer.json](#) 当前收藏夹

[his.json](#) 当前历史记录存储

辅助功能文件

[bg.jpg](#) 背景图片

[bgm1.mp3](#) 等背景音乐

项目各模块和类的设计细节

前端界面部分

前端界面主要由 `test_qt.py` 文件提供，使用 `PyQt5` 框架构建了一个交互式的桌面应用程序。

其中，自定义控件 `FlipCard`、`NeonLabel`、`ImageCarouselWidget` 和 `ParticleWidget` 等通过使用 `PyQt5` 的图形和动画功能，实现了3D卡片翻转、霓虹灯效果标题、图片轮播和粒子背景等视觉效果。

类 `ClubRecommendationApp` 继承自 `QMainWindow`，作为应用程序的主窗口，负责管理界面布局、主题切换、用户交互和与后端算法的连接。

函数 `initUI` 用于初始化界面元素，包括创建轮播图、搜索框、按钮、滚动区域等，并设置它们的样式和布局。

函数 `apply_theme` 用于应用当前选定的主题（明亮、暗色、蓝色），更新所有界面元素的样式。

函数 `recommend_clubs` 和 `prefer_clubs` 分别用于根据用户输入的关键词进行社团推荐和显示收藏的社团，通过调用后端算法模块（`algorithm.pinyin_search` 和 `algorithm.prefers`）获取数据并展示。

动画系统包括 `FlipAnimation`（卡片翻转动画）和悬停放大动画，通过 `QPropertyAnimation` 和事件过滤器实现。

此外，`display_recommendations` 函数负责将获取到的社团数据以翻转卡片的形式展示，每个卡片附带操作按钮（详情、收藏、标签维护）。

后端算法部分

后端算法支持主要由 `algorithm.py` 部分提供。

其中，函数 `load_prefs`、`save_prefs0` 和 `clear_prefs` 等通过使用 `json` 库，对对应的 `.json` 文件进行读写从而实现对收藏夹、历史记录、社团标签的读档，存档和初始化。

`list name`、`link`、`tag`、`historyscore` 分别存储社团名称、详情链接、标签和被搜索次数，`tag0` 用于存储初始标签，`Historyscore` 用于存储初始被搜索次数。

类 `Club` 用于存储每个社团的信息，`Tag`、`Name`、`Link`、`Score` 分别表示每个社团的标签、名称、详情链接、与当前搜索关键词/名称的吻合度和历史搜索次数的加权优先级评分。通过对 `clubs` : `[Club]` 以 `score` 为关键字排序就可以找出较为符合用户需求的几个社团。

类 `memoryClub` 效果与 `Club` 类似，但 `score` 计算时的加权权重不同。

函数 `search` 返回基于 `Club` 的与用户搜索关键词/名称逐字匹配程度最高的部分社团。

函数 `pinyin_search` 通过调用 `pypinyin` 库，返回基于 `Club` 的与用户搜索关键词/名称逐字匹配或同音匹配或按缩写匹配相似程度最高的部分社团。

函数 `memory` 返回基于 `memoryClub` 的与用户搜索关键词/名称逐字匹配程度最高的部分社团。

小组成员分工情况

杨子涵：界面部分及其功能解释 + 协同贡献 idea + demo 录制

林喆熹：算法部分及其功能解释 + 界面部分功能调试 + 前后端接口维护 + 协同贡献 idea

麻堃：资料收集 + 协同贡献 idea

项目总结与反思

总体来说，在小组成员的共同努力以及老师助教的支持和指导下，本项目得以完成。从程序实现到测试功能与 debug 的过程中，组员的代码实现能力和对 python 各项功能的掌握得到了一定程度的提升，对面向对象编程有了更为深刻的理解，很好地检验和提升了本门课程对于面向对象编程的学习效果。

存在的一些问题：

1. 由于前期未能约定好前后端 API 接口导致后期需要对前后端接口进行手动修改。
2. 完成过程中交流不够充分使某些环节出现了一些困难，但是得到有效解决。

反思：

1. 在小组合作中应充分发挥交流的作用，使合作更加高效可靠。
2. 在实现程序之前应当有一个恰当的完整程序构思。

感谢组长、各位组员的共同努力，也感谢老师和助教提供的支持和指导！