

一、程序功能介绍

燕园经纬是以北京大学校园地图为基础，融合导航、游览、娱乐的趣味校园助手程序。

项目主要分为四个功能模块，信息查询、校园导航、地点收藏和校园动态，并创新性的加入了大模型、老虎机等板块以拓展体验。

1. 整体介绍

程序实现了多主题、中英文切换。用户可自由选择浅色 深色 蓝色 北大 清华主题，并进行中英文切换。

地图采用svg格式展示燕园地貌，左上角展示了当前时间。

2. 信息查询

(1) 地图部分

点击右上的放大和缩小按钮或滚动鼠标轮盘可自由实现地图的缩放，拖动移动按钮或点击键盘上下左右键可实现地图的移动。

鼠标悬停在建筑物上方可显示建筑物名称。

点击地图建筑物会高亮并显示建筑物名称及详细信息。主要介绍建筑方位、历史及功能，帮助同学们快速了解燕园建筑。

(2) 信息查询（按键）

点击信息查询

- 点击我想吃，可查询当前时间校园内食堂营业情况，同时也可自由查询其他时段食堂营业情况和各食堂全天营业时间，并附带对食堂简要介绍。
- 点击我想学，可根据需求推荐对应教室，如对插座和教室大小的要求。
- 点击问ai，我们可以询问北京大学ai助手各类问题，ai助手会根据问题并结合北大特色进行回答，这将成为用户了解燕园最方便的工具。

3. 校园导航

选择起点并选择终点，程序可给出两点之间的最短路径，并在地图界面绘制出对应路线、标注前进次序。

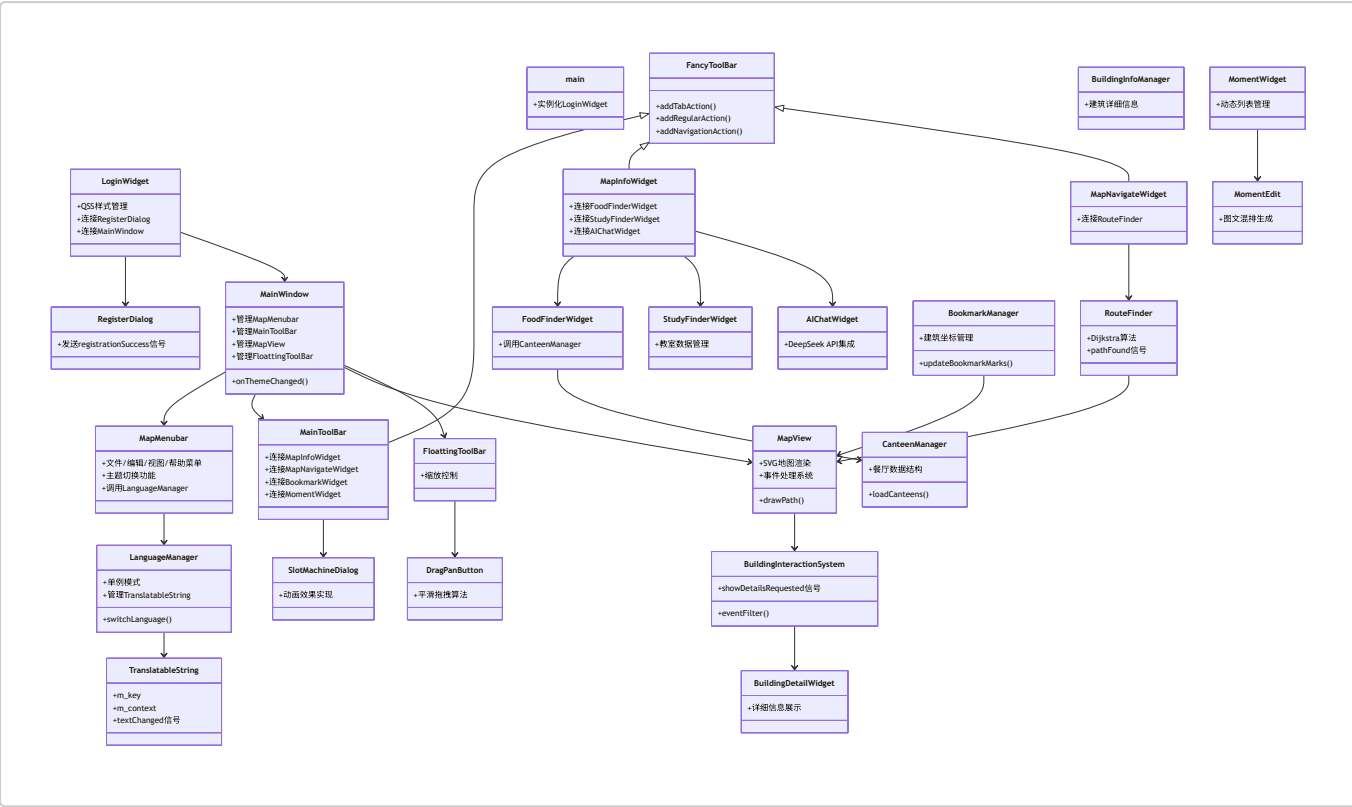
4. 校园动态

点击收藏地点，按住键盘shift键同时鼠标点击地图，即可收藏地点，程序会在地图界面为已收藏地点绘制星星标记。

按住键盘ctrl键同时点击地图可在对应位置发布动态，动态可发布文字与照片，分享在燕园的美好经历。后续可在燕园动态中查看已发布动态。

动态发布成功后可获得10金币奖励，使用金币可在燕园老虎机中游玩老虎机游戏。如有困难，还可调整难度为简单模式。

二、项目各模块与类设计细节



1. main类

作为程序入口，实例化了一个LoginWidget对象。

2. LoginWidget类

登录界面，由:/style/login_style.qss文件单独管理样式。布局了一个标题图片:/assets/title.png，两条用户名、密码输入框QLineEdit，以及三个注册、登录、取消按钮QPushButton，其中注册按钮连接了RegisterDialog类的一个实例，登录按钮连接了MainWindow类的一个实例。

3. RegisterDialog类

注册界面，由:/style/login_style.qss文件单独管理样式。布局了三个用户名、密码、确认密码输入框QLineEdit，两个注册、取消按钮QPushButton。其中还设置了registrationSuccess(username, password)信号传播给LoginWidget以实现注册成功自动填充账号密码。

4. MainWindow类

程序的主界面，统一管理此后所有控件的样式。依次布局了上方菜单栏MapMenubar，左侧工具栏MainToolBar，右侧地图区域MapView，右上角悬浮工具栏FloatingToolBar。槽函数onThemeChanged()负责接受主题更改信号并切换所有类的样式，普通函数showBuildingDetails()用以接受显示建筑详细信息信号并在左侧区域显示详细信息。

5. MapMenubar类

上方菜单栏，负责控制程序的基本功能，继承自QMenuBar。依次布局了四个菜单&文件、&编辑、&视图和&帮助。其中&文件菜单可以打开电脑文件或者保存文件至电脑，后续可以实现切换地图以及保存有关信息的功

能；&编辑菜单可以实现简单的复制粘贴功能；&视图中有菜单项&主题和&语言，&主题添加了6种QAction，分别释放不同的themeChanged(themeName)信号，由MainWindow的onThemeChanged()接受并进行切换，&语言添加了中文与English菜单项，统一调用单例模式LanguageManager中的函数switchLanguage(languageCode)对其管理的所有动态控件进行文本刷新，从而实现翻译功能。

6. TranslatableString类

用于管理可翻译的字符串对象，每个TranslatableString类的实例会绑定到一个动态控件，从而实现动态控件的文本翻译功能。其内储存了控件对应的原始文本m_key以及文本所处上下文m_context（在程序中上下文即为所处的类名），还会释放textChanged()信号通知绑定控件进行文本的更新。

7. LanguageManager类

用于统一管理所有的可翻译动态控件。为此采用了特殊的**单例模式**实现方法，确保该类在程序中始终只有一个实例，只提供了该实例的全局访问点。其私有成员包括QHash<QObject*, TranslatableString*> m_widgetBindings作为所有QObject对象的管理器，QHash<QAction*, TranslatableString*> m_actionBindings作为所有QAction对象的管理器。为了减少代码重复，在其中提供了所需动态控件的生成函数，包括createTranslatablePushButton()、createTranslatableLabel()、createTranslatableAction()、createTranslatableAction()、createTranslatableToolButton()、createTranslatableGroupBox()、createTranslatableMenu()、createTranslatableMenuAction()，分别给对应的控件绑定一个TranslatableString类的实例、连接textChanged()信号进行静态控件更新、注册至对应的可翻译控件的管理器中。在最后的switchLanguage()函数中，遍历管理器中的对象进行文本刷新从而实现翻译功能。所需的翻译文本储存在:/translations/lang_CN.qm和:/translations/lang_US.qm两个文件中，分别对应中文语言和English语言下的文本，由设计者自行翻译生成。

8. FancyToolBar类

程序中所有工具栏控件的基类，负责派生出所有的工具栏控件。其中addTabAction()用于添加可切换标签页的按钮；addRegularAction()用于添加常规工具按钮，不会隐藏工具栏本身；addNavigationAction()用于添加导航动作，不关联任何界面元素；addSeparator()用于添加导航动作

9. MainToolBar类

左侧工具栏，负责管理程序的主要功能，继承自FancyToolBar类。布局了四个TabAction，对应信息查询、校园导航、收藏地点、校园动态四种功能，分别连接到MapInfoWidget、MapNavigateWidget、BookmarkWidget、MomentWidget四类的一个实例，实现标签页的切换。还有一个常规动作RegularAction对应老虎机功能。

10. MapInfoWidget类

主工具栏中的信息查询功能，继承自FancyToolBar类。布局了三个TabAction，对应我想吃、我想学、问AI三种功能，分别连接到FoodFinderWidget、StudyFinderWidget、AIChatWidget三类的一个实例，实现标签页的切换。还有一个导航返回动作。

11. CanteenManager类

对餐厅信息进行统一管理。管理所需的源文件储存在:/info/canteens.txt中，包含餐厅的名称、详细信息、具体开放时间等内容。为餐厅建立了canteen结构体，保存了对应信息。类中还含有loadCanteens()对信息

进行导入，`findOpenCanteens()`负责查询相应时间开放的餐厅，`getOpeningHoursForPlace`负责查询相应餐厅开放的时间以及`getAllCanteenNames()`返回所有餐厅名称用于`QComboBox`的内容添加。

12. FoodFinderWidget类

“我想吃”功能的实现界面。主要包括按时间查询开放餐厅和按餐厅查询营业时间两个部分和一个返回按钮。按时间查询开放餐厅部分包含一个`QTimeEdit`控件用于输入时间，并匹配一个`QPushButton`连接到槽函数`onCheckOpeningClicked()`；按餐厅查询营业时间部分包含一个`QComboBox`控件用于进行餐厅的选择，并匹配一个`QPushButton`连接到槽函数`onCheckHoursClicked()`。

13. StudyFinderWidget类

“我想学”功能的实现界面。具体信息的源文件储存在`:/info/classrooms.txt`，包含教学楼、教室号、是否大教室、是否有插座等内容。界面主要包含一个返回按钮和两个`QGroupBox`用于选择是否需要大教室、是否需要插座，并匹配一个`QPushButton`连接到槽函数`onConfirmClicked()`查询对应的教室信息。

14. AIChatWidget类

“问AI”功能的实现界面。其中定义了常量`API_KEY`和选用的模型类别`MODEL_NAME`，接入了`deepseek`的API实现AI询问功能。通过日志文件`ai_chat_log.txt`储存了谈话历史，并借助Qt自带的`Json`文件有关方法处理获取的`Json`文件，实现了流式输出。

15. MapNavigateWidget类

主工具栏中的校园导航功能，继承自`FancyToolBar`类。布局了一个`TabAction`，对应规划路径一种功能，连接到`RouteFinder`类的一个实例，实现标签页的切换。还有一个导航返回动作。

16. RouteFinder类

规划路径功能的实现界面，可以通过输入起点和终点规划出最短路径，并在地图中可视化出具体路径。通过将地图抽象为图，结点信息储存在`:/info/building_info.txt`中，结点间的连接情况储存在`:/info/connected_vertices.txt`中。类中包含内部类`DijkstraFindPath`储存地图的抽象数据类型，包含邻接矩阵和结点信息，`dijkstra()`函数返回所需的单源最短路径。UI界面布局了两个`QComboBox`用于选择对应的起点和终点，以及一个`QPushButton`连接对应的槽函数`triggerFindPath`，槽函数会释放信号`pathFound(path)`传递给`MapView`进行具体路径的可视化显示。

17. BookmarkManager类

统一管理所有收藏的地点数据。建筑信息储存在`:/info/building_info.txt`中，包括对应的建筑名称，在`svg`图像中的具体坐标等信息。收藏数据作为应用数据储存在`marked_places`目录下。每次构造实例或者更改收藏地点时会导入或者保存收藏数据，从而实现记忆化。还有`updateBookmarkMarks()`函数，可以绘制五角星显示在`svg`地图上的对应收藏地点的位置；`bookmarksChanged()`信号，传递给`BookmarkWidget`实现及时更新。

18. BookmarkWidget类

主工具栏中的收藏地点功能。布局了一个返回按钮`QPushButton`连接`requestReturn`信号，一个已收藏列表`QListWidget`，一个可自动补全的添加控件`QComboBox`以及移除选中/清空全部两个操作按钮。为实现及时更新收藏地点，使用`updateBookmarkList`更新收藏列表并时刻监听`BookmarkManager`类的`bookmarksChanged()`信号。

19. MomentEdit类

创建动态界面实例。布局了位置信息QLabel、文本编辑框QTextEdit、图片显示区域QLabel和发布/取消两个操作按钮。通过函数createMomentImage()可以借助QPixmap和QPainter根据所输入文字和图片绘制对应的动态图像，并将其保存在数据文件moments中，用QDateTime生成唯一文件名，从而实现了记忆化。

20. MomentWidget类

主工具栏中的校园动态功能。布局了一个返回按钮QPushButton连接requestReturn信号，一个动态列表QListWidget储存了当前已发布的所有动态以及查看/删除/清空三个操作按钮。

21. MapView类

负责管理整个地图视图，实现与地图有关的输入输出设备交互功能。其中scene成员变量指向svg地图，地图储存在:/assets/ground_new.svg路径下。通过重写了mousePressEvent(), wheelEvent(), resizeEvent(), keyPressEvent()等事件实现了对输入流的响应，包括地图缩放移动、地点收藏、动态创建以及具体信息显示等功能；通过drawPath()函数实现了路径的可视化显示。另外还通过QLabel和动画效果QPropertyAnimation创建了对应风格的计时器。

22. FloattingToolBar类

右上角悬浮工具栏，负责提供按钮控制地图视图的缩放与平移功能。布局了放大、缩小两个QToolButton按钮分别连接MapView类的zoomIn和zoomOut函数，以及拖拽功能DragPanButton类的一个实例。

23. DragPanButton类

专门用于生成拖拽功能按钮，能够平滑地拖动地图视图。限制拖拽按钮在相应的圆形区域的圆心，通过重写了mousePressEvent()、mouseMoveEvent()、mouseReleaseEvent()，计算拖拽按钮相对圆心的偏移量计算相应的地图视图移动量，再利用MapView类的moveBy()函数实现平滑移动。在松开鼠标后，通过QPropertyAnimation创建返回动画，确保按钮回到圆心位置。

24. BuildingInfoManager类

用于管理所有地点的详细信息。源文件储存在:/info/building_details.txt资源文件中。详细信息储存在BuildingInfoManager命名空间下的BuildingInfo数据结构中，包含中文名、英文名、地点以及详细信息。通过一个QHash<QString, BuildingInfoManager::BuildingInfo>对象统一管理所有地点的详细信息。

25. BuildingInteractionSystem类

管理建筑交互系统的，主要包括鼠标停留显示缩略信息以及鼠标点击展示详细信息功能。通过重写了eventFilter()事件（其中不论如何都不对事件进行拦截是因为鼠标点击还有其他操作要处理），监控鼠标悬停的时长，达到阈值后即显示对应的缩略信息窗口。在鼠标点击事件后会调用showBuildingDetails()发出showDetailsRequested(name, details)信号给BuildingDetailWidget类接受并生成窗口。

26. BuildingDetailWidget类

负责生成地点详细信息窗口，同样继承FancyToolBar。布局了标题文本QLabel，详细信息界面QTextEdit和一个返回导航动作。

27. SlotMachineDialog类

实现了老虎机功能。有关数据保存在对应的数据文件中，实现了记忆化功能。布局了一个开始按钮 `QPushButton`，一个结果标签 `QLabel`，金币显示 `QLabel` 以及模式勾选框 `QCheckBox`。为了实现老虎机的摇动残影，利用 `QLabel` 创建了多个残影，并用 `QGraphicsOpacityEffect` 设置为半透明，再用 `QPropertyAnimation` 创建渐隐动画，实现丝滑摇动。

三、分工合作

各组员初步分工定位为组员 [姚远](#) 负责前端，[王宝骥](#) 负责后端，[张化东](#) 负责数据收集和处理。

组员 [王宝骥](#) 负责完成寻路算法，路径绘制，地图处理和显示，建筑物点击高亮等内容。

组员 [姚远](#) 负责框架的搭建，窗口的设计与美化（包括但不限于登陆界面，主界面，ai对话界面等），以及不同版本的整合。

组员 [张化东](#) 负责信息的收集和录入，包括地图的获取和处理，建筑坐标获取和录入，建筑信息录入等以及信息查询界面的相关工作。

组员 [姚远](#) 和 [张化东](#) 合作完成老虎机，收藏地点发布动态，食堂教室信息查询等功能。

组员 [王宝骥](#) 和 [姚远](#) 实现并完善建筑物点击高亮，悬停显示建筑信息等功能，合作优化地图界面。

四、项目总结与反思

1. 核心目标达成度：

已完成：

1. 地图交互功能的实现：地图的展示、放缩，常用地点的导航，鼠标悬停位置显示建筑信息，点击高亮交互。
2. 校内信息查询：教室信息，食堂开放情况，食物推荐。
3. 针对来访人员的功能：收藏地点，照片记录，娱乐功能。

未能实现：

1. 失物招领为代表的一系列方便学生日常使用的功能：原计划搭建一个新生适应了解北大的平台，包含更细颗粒度的导航，食堂、校内超市信息获取和推荐、失物招领与二手物品交易等功能。
2. 室内地图未能完成：一方面获取不了需要的信息（室内图纸，获取教室信息的权限等），另一方面工作量更大，需要掌握的技巧知识过多。

2. 关键问题反思：

1. 项目管理反思：组员都没有项目开发经验，对于在 `GitHub` 上构建 `repository` 同步各方面开发进度不熟悉，版本更新同步出现许多差错。未能很好地预估各部分工作量，分工不均衡，导致开发进程也由于局部的一些问题不能攻克而延缓。
2. 项目结构反思：缺乏早期的规划，导致类的设计以及类之间的关系不够明晰，进一步增大了组员之间合作的难度，同时也让程序的可维护性不佳。
3. 对 `QT` 已有构件不熟练导致的冗余低效：例如，在一开始采用自己设计的坐标放缩映射系统，在多次放缩操作后，误差累积达到肉眼可见的地步，既低效又不精准。

4. 不同人负责的部分不同，学习的内容不同导致知识共享的缺失：在图形界面，数据处理，算法设计方面各自学习完成，个人难以处理一些涉及到他人领域的问题。

3. 可改进的方向：

1. 室内导航扩展：录入教学楼，食堂等主要活动区域的信息，构建室内的导航，并能清晰的查看相关信息（例如理科一号楼内教室导航，查询教室占用情况）。
2. 三维立体导航构建：利用unity引擎构建教室建模并采用内置的路径规划算法完成导航功能。
3. 模板化方案：进一步完善文件标准化结构，提高程序的复用性。比如对鼠标事件的响应分散在MapView和BuildingInteractionSystem两个类中，在实现上就曾产生事件冲突，集中管理可以避免相应冲突。
4. 建立内部wiki以及增加每周技术分享的频率。