

北京化工大学

本科毕业设计（论文）

论文题目：计算化学数据处理软件开发

学院名称：化学学院

专业：应用化学

班级：应化 1902

学号：2019130145

学生姓名：刘玮卿

导师姓名：雷鸣

日期：2023 年 5 月 26 日

诚信声明

本人声明：

本人所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：_____ 日期：_____

指导教师签名：_____ 日期：_____

毕业设计（论文）任务书

设计（论文）题目： 计算化学数据处理软件开发

学院： 化学学院 专业： 应用化学 班级： 应化 1902

学生： 刘玮卿 指导教师： 雷鸣 专业负责人：

1. 设计（论文）的主要任务及目标

(1) 编程实现分子建模软件

2. 设计（论文）的主要内容

(1) 进一步开发分子三维可视化软件。

3. 设计（论文）的主要要求

(1) 软件要实现对分子的各种操作，使用户可以方便地绘制。

4. 主要参考文献

[1] SOZYKIN S A. GUI4dft-A SIESTA oriented GUI [J]. Comput Phys Commun, 2021, 262: 6.

[2] BAEK N, LEE H. OpenGL SC Implementation on the OpenGL Hardware [J]. IEICE Trans Inf Syst, 2012, E95D(10): 2589-2592.

5. 进度安排：

	设计（论文）各阶段名称	起 止 日 期
1	查阅文献，阅读文献，完成文献综述、开题报告	2022 年 11 月 1 日到 2022 年 12 月 26 日
2	开发分子可视化软件	2023 年 1 月 1 日到 2023 年 4 月 1 日
3	撰写毕业论文，论文答辩	2023 年 4 月 1 日到 2023 年 6 月 10 日

毕业设计（论文）具体标题

计算化学数据处理软件开发

摘要：本论文在原有化学建模软件的基础上进行了进一步的开发，增加了任意基团与现有分子相连接的功能，并对软件界面进行了优化，使其能够打开多个窗口。此外，本论文还增加了模板界面，使用户可以在基团中选择具体要键连的原子，增加了绘制多键功能和撤销重做功能，以及增加了用户可选择的基团等等。这些改进大大提升了软件的易用性和灵活性，使其更加适用于化学研究领域中的分子建模和分析。

为实现这些改进，本论文采用了 Python 编程语言、PyQt5 GUI 工具包和 OpenGL 图形库，通过编写各种函数和模块，实现了多种功能。同时，为了增加用户的交互体验，本论文还引入了模板界面，让用户可以直观地选择要键连的基团。

实验结果表明，本论文所开发的化学建模软件，具有较高的稳定性和良好的用户体验，能够有效地提高化学研究人员的工作效率和准确性。在未来的研究中，本软件还有进一步的优化和拓展空间，以更好地满足不同用户的需求。

关键词：化学建模软件；任意基团连接；用户界面优化；多窗口；模板界面；绘制多键；基团选择。

TITLE OF THE DISSERTATION

Development of Computational Chemistry Data Processing Software

Abstract: This thesis further develops the existing chemistry modeling software by adding the functionality to connect any functional group to existing molecules, optimizing the software interface to enable opening of multiple windows, and introducing a template interface that allows users to select specific atoms to bond in functional groups, as well as adding features such as drawing multiple bonds and undo/redone options, and allowing users to select from a variety of functional groups. These improvements greatly enhance the usability and flexibility of the software, making it more suitable for molecular modeling and analysis in the field of chemistry research. To achieve these improvements, this thesis uses the Python programming language, the PyQt5 GUI toolkit, and the OpenGL graphics library to implement various functions and modules. In addition, to enhance the user experience, a template interface is introduced that allows users to visually select the functional groups they wish to bond. Experimental results show that the chemistry modeling software developed in this thesis has high stability and good user experience, effectively improving the efficiency and accuracy of chemistry researchers. In future research, there is further optimization and expansion space for this software to better meet the needs of different users.

KEY WORDS: chemistry modeling software, connecting arbitrary functional groups, user interface optimization, multi-window, template interface, drawing multiple bonds, functional group selection.

目 录

第 1 章 绪论	2
第 1.1 节 研究背景和意义.....	2
第 1.2 节 国内外研究现状.....	3
第 1.3 节 研究内容.....	5
第 2 章 可视化软件设计.....	6
第 2.1 节 软件框架的改进.....	6
第 2.2 节 新增功能及其实现方法.....	8
2.2.1 分子编辑功能.....	8
2.2.1.1 旋转，平移分子功能.....	8
2.2.1.2 旋转基团功能.....	10
2.2.1.3 对任意基团都适配的添加基团算法.....	10
2.2.1.4 两个基团选择界面.....	15
2.2.1.5 双击直接输入元素名称替换原子.....	16
2.2.1.6 选中原子功能改进.....	17
2.2.2 键编辑功能.....	17
2.2.2.1 选中键功能.....	17
2.2.2.2 删除键功能.....	17
2.2.2.3 多键增加功能.....	18
2.2.2.3.1 对大型分子的额外处理.....	18
2.2.3 界面功能.....	19
2.2.3.1 撤销重做功能.....	19
2.2.3.2 平铺和瀑布布局功能.....	20
2.2.3.3 绘制框图功能.....	21
2.2.3.4 在界面右边添加控制键宽，颜色控制面板	22
2.2.3.5 界面下侧添加命令行功能.....	22
2.2.3.5.1 删除命令	22
2.2.3.5.2 选中命令	23

2.2.3.5.3 Smiles 命令	23
2.2.3.6 界面焦点切换功能	23
结论	25
参考文献	26
致谢	29

前言

本文主题是使用 **PyQt5** 和 **OpenGL** 开发一个化学建模软件,旨在为化学研究人员提供一款高效、易用的化学建模工具,帮助他们更好地理解和分析分子的行为和化学反应的机制。

在这个项目中,我利用了 **Python** 编程语言、**PyQt5** GUI 工具包和 **OpenGL** 图形库,通过编写各种函数和模块,实现了多种功能,包括分子结构的可视化等。同时,为了提升软件的可靠性和用户体验,我还对软件进行了充分的测试和优化。

在本文中,我将会详细介绍本项目的背景、目的、方法、实现和结果,并探讨本软件在化学研究领域中的应用前景和价值。同时,我也将反思本项目的不足和未来的发展方向,希望能够为相关领域的研究和应用提供一些有益的启示。

第1章 绪论

第1.1节 研究背景和意义

随着计算机技术以及量子化学理论的发展,理论计算对于研究化学反应机理和分子结构及性质有着非常重要的作用,越来越多的研究人员采用了量子化学计算的方式对反应体系进行研究。通过使用计算化学程序模拟研究分子结构和化学反应,可以更好地理解分子的性质和化学反应的机制。通常量子化学计算流程分为三步:分子模型的建立、量子化学计算、数据处理。其中在第一步分子模型的建立中,需要研究人员通过建模程序建立合适的分子结构,建模程序应提供合理便捷的操作工具对分子模型进行编辑修改,然而,现有的化学建模软件可能不够灵活或者不够便捷。因此,开发一款能够满足研究人员需求的化学建模软件变得非常重要。

PyQt5 是一种流行的 Python GUI 工具包,提供了很多实用的 UI 控件和功能,使得开发出高效、易用、美观的应用程序变得容易^[1]。OpenGL 是一种跨平台的图形库,可以用于高效地渲染 3D 图形^[2,3]。将 PyQt5 和 OpenGL 结合使用可以创建出高效的、交互性强的 3D 应用程序。

本研究旨在开发一款化学建模软件,使用 PyQt5 和 OpenGL 技术实现对分子结构和化学反应的模拟和可视化。该软件可以提供多种功能,如可视化分子结构、计算能量和力场参数、分析分子动力学等,这些功能可以帮助化学研究人员更好地理解和分析分子的行为和化学反应的机制。

此外,本软件具有以下几个优点:

1.易于使用:软件提供了友好的图形用户界面,让用户可以方便地使用各种功能。

2.高效性:软件采用 OpenGL 技术,可以高效地渲染 3D 图形,让用户可以更直观地观察分子结构和化学反应的过程。

3.灵活性:软件采用 Python 编写,用户可以自由修改和扩展软件的功能^[4]。

综上,本研究开发的化学建模软件可以帮助化学研究人员更好地对分子模型进行编辑修改,并可视化地展示量化计算数据,帮助理解和分析分子的性质和化学反应的机制,并且具有易于使用、高效性和灵活性等优点,是一款非常有价值

的化学建模工具。

第1.2节 国内外研究现状

在化学建模领域，有许多知名的软件可供选择。这些软件的特点各有不同，其中 Gaussian、ChemDraw、MOPAC、GAMESS、PCModel、Chimera 和 Pymol 是最常用的软件之一。

Gaussian 是一款强大的分子建模软件，被广泛应用于计算化学、材料科学等领域^[5]。Gaussian 的主要特点是能够实现精确的计算，并且可以用于计算分子性质、优化分子结构和模拟反应过程。Gaussian 支持多种计算方法，包括从简单的哈特里-福克方法到复杂的密度泛函理论^[6]。这使得 Gaussian 在模拟复杂分子系统和反应机理方面具有很高的可靠性和准确性。此外，Gaussian 配套的 GaussView 软件提供了丰富的可视化工具，可以实现分子建模、Gaussian 输入文件的生成和输出文件的数据展示，使用户可以方便地分析计算结果。

ChemDraw 是一款广泛使用的化学绘图软件，该软件旨在帮助化学家和生物学家轻松地绘制化学结构、分子反应和化学方程式等图形^[7]。

ChemDraw 的特点在于其直观的用户界面和强大的功能。它提供了各种化学结构的绘制工具，包括单个原子、官能团、化学键和杂环等。此外，它还具有自动布局和格式化工具，使用户可以轻松创建专业且易于阅读的图形。

ChemDraw 还拥有强大的搜索和替换功能，可用于快速查找和替换分子中的特定原子或官能团。此外，它还支持多种输出格式，如 PDF、PNG 和 SVG 等，使用户可以轻松地将其绘图导出到其他应用程序或文档中。

在化学和生物科学研究中，ChemDraw 已成为必备工具之一。它的易用性、功能强大和可扩展性使其成为化学家和生物学家在实验室中进行研究和交流的理想选择。

MOPAC 是一款快速而可靠的分子建模软件，特别适用于大型分子系统的计算。MOPAC 的主要特点是其快速而准确的计算速度，以及广泛的化学应用范围。MOPAC 支持多种计算方法，包括从简单的能量最小化到复杂的反应动力学计算^[8,9]。此外，MOPAC 还具有良好的可扩展性和并行化能力，可以有效地处理大型分子系统的计算任务。

GAMESS 是一款可以用于计算分子性质和优化分子结构的强大的分子建模

软件。GAMESS 的主要特点是其高度优化的计算引擎和多种计算方法^[10]。

GAMESS 支持从量子化学到分子动力学的多种计算方法，以及多种电子相关性和基组方法。此外，GAMESS 还具有灵活的输入文件格式和丰富的可视化工具，使用户可以方便地进行计算和分析。

PCModel 是一款分子模拟软件，用于模拟化学反应、分子动力学以及其他分子关联计算。该软件的应用范围广泛，包括药物设计、材料科学、生物技术等。其主要特点包括以下方面：可视化工具：PCModel 提供强大的分子图形和可视化实用工具，以使用户直观地编辑和操作分子结构，显示分子的各种属性和结果；多种计算：该软件涵盖了多种计算模型和计算算法，包括分子力场、量子力学、蒙特卡罗模拟、分子对接等，能够模拟各种分子相关的物理和化学性质^[11]。

Chimera 是一款流行的分子可视化软件，它具有强大的分子图像渲染功能和许多交互式工具。Chimera 的开发旨在用于蛋白质、DNA、RNA、药物以及其他小分子等化学和生物学领域^[12]。它广泛应用于大型计算和生物医学研究中。

Chimera 支持多种文件格式，如 PDB、Mol2、MRC 等，可以轻松地处理和可视化分子结构。它具有强大的数据分析和计算功能，包括分子对接、分子动力学、蒙特卡罗模拟等^[13]。Chimera 也支持人与计算机的互动，使用户可以进行非常有用的技术教学以及相关研究。

Pymol 是一款流行的生物分子可视化软件，在生命科学和药物研发领域得到了广泛应用。Pymol 可用于表现静态或动态的三维结构，包括蛋白质、DNA、RNA 和其他小分子等生物大分子，以及晶体学和电子显微学中的相关结构。

Pymol 提供了多种可视化方式，包括线框图、球棍图、球面图、卡通图等，使用户可以方便地查看、分析和处理分子结构，并可在可视化窗口中进行旋转和缩放操作^[14]。此外，Pymol 还具有智能化的分子渲染和绘图功能，可以生成非常逼真的分子图像和动画。

Pymol 不仅有着卓越的分子可视化功能，还有着丰富的插件和扩展功能。通过编程语言 Python 与 Pymol 交互，用户可以实现更多的分子建模和可视化操作^[15-19]。此外，Pymol 支持乙烯环的石墨化，大大提高了其渲染效率，使得其在计算机硬件性能相应较低的设备上也能得到流畅的使用体验。

总之，这些软件在化学建模领域中都发挥着重要作用，并且在不同的应用场景中具有各自的特点和优势。选择合适的软件取决于用户的具体需求和研究方向。

第1.3节 研究内容

本论文致力于使用 PyQt5 和 OpenGL 开发一个化学建模软件，在该软件中，使用 Python 作为主要编程语言，以 OpenGL 图形库作为实现分子模型显示的主要工具，以 PyQt5 GUI 工具包开发图形界面，通过编写各种函数和模块，实现了多种分子建模和显示功能。

第2章 可视化软件设计

第2.1节 软件框架的改进

本研究提出了一种基于 Qt 的多文档界面(MDI)创建的多窗口界面方案，其初始设置包括模板窗口和绘制窗口两个主要部分。绘制窗口可以通过用户界面按钮进行扩展，而模板窗口则仅限于单一存在，并不能被关闭，除非整个界面被关闭。模板窗口的功能主要是接收并展示用户在分子选择界面所选中的基团，而绘制窗口则是提供分子绘制的操作场所。

为实现上述方案，接下来的任务主要是对总体界面进行调整和改进，以适应已完成的窗口界面。具体步骤包括将原有主窗口的工具栏、菜单栏和停靠窗口加入至当前主界面，并将这些组件在绘制窗口中隐藏。当多窗口界面的工具栏按钮被点击，将触发被隐藏在绘制窗口中的相应函数。此设计保证了工具栏信息同步的实现，即便工具栏并未在原来的绘制窗口显示。

同时创建了一个模板窗口。当焦点在不同绘制窗口中切换时，将改变模板窗口依附的绘制窗口^[20]。这意味着将此时焦点所在的绘制窗口的一些同步信号绑定到模板窗口上。当此绘制窗口改变数据时，例如在各种分子选择界面中选中了某个分子信息，模板窗口也能接收到信息并同步将分子显示在模板窗口界面上。

在焦点在不同绘制窗口中切换时，还需要重新绑定多窗口主界面的工具菜单栏所对应的函数，将它们全部绑定到当前焦点所在窗口对应的函数。这样，按下按钮时，对应窗口就能做出响应。但是，如果这样设计，当在当前绘制窗口按下按钮后，切换到另一个窗口时，还需要再按一次按钮，现在的绘制窗口才能收到信息。为了方便用户操作，我将当前按钮所代表的信息传递给模板窗口。由于模板窗口无法关闭，所以信息可以由模板窗口保存。当焦点切换时，模板窗口将按钮信息传递给当前的绘制窗口，从而使得用户不必重新按下按钮即可在新的绘制窗口使用之前选中的按钮功能。

此外，对于某些特殊按钮对应的打开分子界面功能，用户可以从界面中选择要绘制的原子或分子。从实用性角度考虑，应当使当前绘制窗口选中的原子或分子信息同步到所有其他绘制窗口中。因此，还需要将当前绘制窗口的一些数据发送到模板窗口。当焦点切换时，模板窗口再将数据发送给绘制窗口，这样做的原

因与上面一致。接着，根据基团名称的长度（例如将氢原子命名为 H，苯环命名为 benzene ring），如果名称字符长度大于二，则认为是基团，否则认为是单个原子。根据它们的类型不同，进行不同的操作。若是基团，则调用下面即将提到的基团添加函数。若是单个原子，则在鼠标点击在某个已存在的原子范围内时，直接替换它的原子特征信息，而其它信息不变。若未在已存在的原子范围内，则直接进行绘制。

同时，我还将当前打开的每个窗口标题，以及控制每个窗口最大化、最小化和关闭的按钮，按垂直方向放置在左侧任务栏中。这样，用户可以方便地操作各窗口。当用户单击两次最大化按钮时，界面将恢复到原始大小。最小化操作也是如此。效果如下图所示：

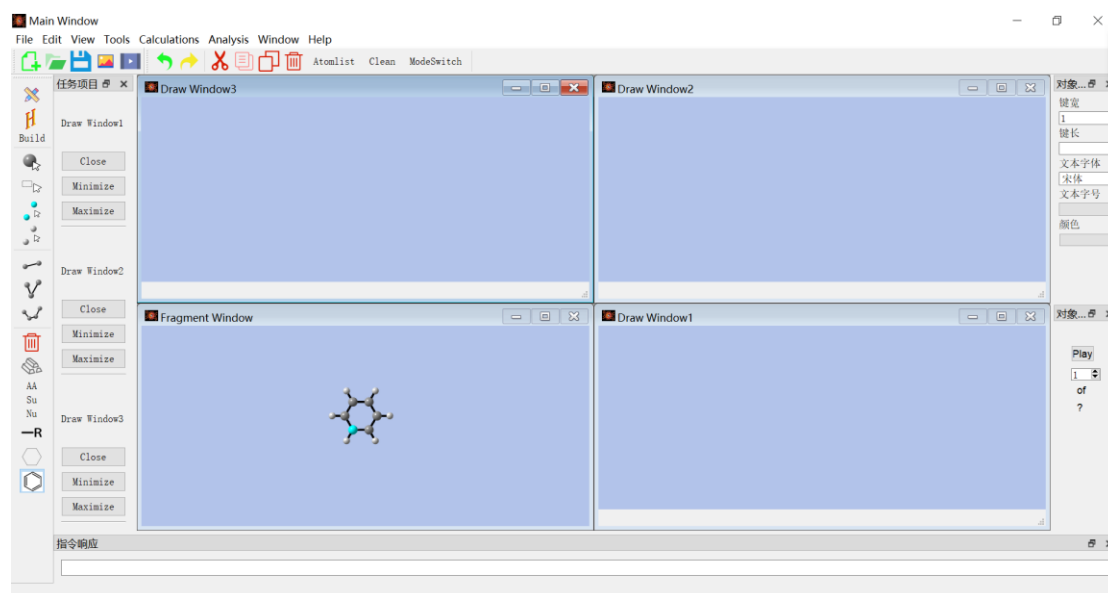


图 2-1 界面演示

Figure 2-1 Interface demonstration

第2.2节 新增功能及其实现方法

2.2.1 分子编辑功能

2.2.1.1 旋转，平移分子功能

在该软件中，我们可以使用键盘 r 键以及鼠标拖动来进行旋转操作，使用键盘 t 键以及鼠标拖动来进行平移操作。旋转和平移的实现方法基于同步关系。当用户按下键盘按键时，将记录下按下坐标，并设置一个标志允许按键事件进入主体函数。为了进入主体函数，我们还需要先触发鼠标移动事件。同时，也会记录鼠标移动后鼠标位置数据，但需要按键事件的触发此事件才会被触发。一旦进入主体函数，我们会记录当前时间。然后，只有当时间间隔满足条件时，才能再次进入主体函数。

旋转操作的主体函数实现方法如下：首先，我们禁用鼠标拖动旋转视点的功能。接着，我们会寻找当前所在的分子，并计算出其所有原子的坐标并求和算出形心。然后，我们会将鼠标点击的坐标转换为世界坐标，并根据点击坐标与形心的关系来判断鼠标相对位置的左右。接下来，我们会根据移动坐标与上一次移动坐标之间的差距来判断是上移还是下移，并相应地改变旋转角度。我们会计算旋转和平移矩阵，并先应用靠右的矩阵于坐标上，乘平移矩阵的意义在于因为二维旋转矩的旋转轴在 origin，只有将整体的旋转轴平移到 origin，才能使旋转矩阵正常工作。因此需要先对坐标乘以平移矩阵，再乘旋转矩阵，最后乘平移矩阵的逆矩阵。

平移矩阵公式如下：

$$\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2-1)$$

旋转矩阵形式如下：

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2-2)$$

平移矩阵的逆矩阵如下：

$$\begin{pmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2-3)$$

然后，我们会对该分子中的所有原子取前两个坐标进行变换，并加上第三个坐标来更新坐标。最后，我们会将移动后的坐标赋值给点击坐标，以便实时判断是在上移还是下移。旋转分子效果如下：

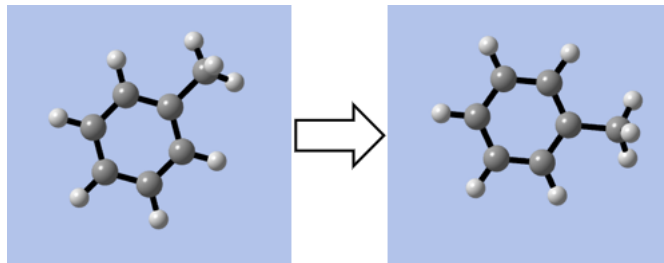


图 2-2 旋转分子效果展示

Figure 2-2 Rotating molecular effect demonstration

与旋转操作类似，平移操作也是分子模拟中常用的一种操作，其可以改变整个分子在二维或三维空间中的位置。不同之处在于，平移操作使用鼠标的世界坐标差来设置平移矩阵，而旋转操作则使用旋转角度来设置旋转矩阵。

平移矩阵如下：

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2-4)$$

其中 t_x 和 t_y 分别表示鼠标移动的世界坐标差，这样可以使得分子随着鼠标拖动的位置变化而变化。平移操作有两种形式，一种是先按下快捷键，长按鼠标（鼠标按下位置需要在某个原子坐标范围内）即可拖动整个分子，第二种是选择想要拖动的分子，之后鼠标长按选中的原子，直接拖动，效果是只拖动选中的原子，可以同时拖动多个原子。拖动选中的原子效果如图所示：

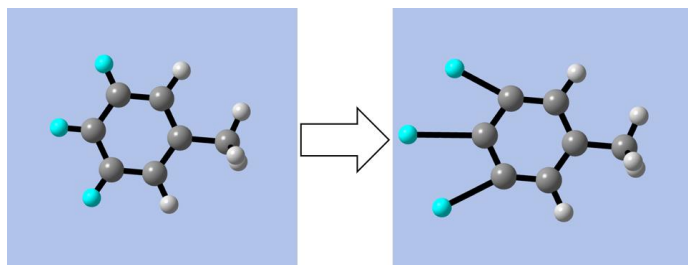


图 2-3 拖动选中的原子效果展示

Figure 2-3 Demonstration of dragging the selected atom

总之，平移操作和旋转操作一样，是分子模拟中不可或缺的一环，它们可以帮助研究者更加直观地理解分子的结构和性质。

2.2.1.2 旋转基团功能

此功能的实现需要找到选中原子所在的基团，我采取的方法是要求用户选择两个或以上的原子，这些原子需要和基团的连接点相连或者本身就是连接点，程序会把用户所有选择的原子通过键共同连接的原子作为基团和原本分子的连接点，接着根据用户选择的原子位于分子的哪一部分判断哪一部分原子是属于基团的。具体判断过程如下：当用户按下对应功能按钮后，程序遍历用户选择的原子，为每个原子创建一个空集合，然后将每个原子和所有键信息进行对比，判断是否在键中，若在键中，则将其此键中相关联的两个原子加入集合（集合具有互异性，因此不会重复添加相同原子）中，当遍历结束后，将每个集合进行交集运算，算出连接点原子，然后根据用户所选择的原子寻找在连接点一侧的所有原子，随后用户用鼠标拖动，可以对程序找到的基团进行旋转操作。旋转基团效果如下：

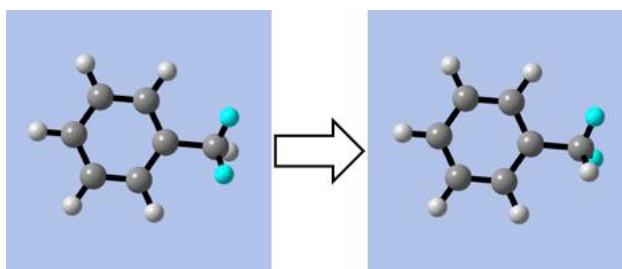


图 2-4 旋转甲基基团效果展示

Figure 2-4 Demonstration of the effect of rotating methyl groups

旋转基团功能的实现过程中需要考虑多个因素，包括原子的连接关系、位置关系等等，这些都需要经过精细的计算和判断才能准确实现旋转基团的功能。

2.2.1.3 对任意基团都适配的添加基团算法

该函数主要实现了以下功能，以满足化学信息处理中的基团操作需求。该函数会在用户选中旧基团的氢原子并选定新基团的键连原子后，添加新基团，并建立新旧基团间的化学键。同时，函数还将对新基团的角度进行调整，以确保其符合化学规律。

一种主要的功能是，在用户选择了新基团非氢原子并点击了旧基团的氢原子时，该函数能够在新旧基团的非氢原子间建立化学键。此过程还包括添加新基团，

并且调整新基团的角度以使其符合化学规律。

如果用户在选择基团但未点击旧基团的氢原子时，该函数则执行简单绘制操作，将新基团绘制在用户指定的位置。这种方式提供了简便快捷的操作选项，使用户可以直接在指定位置添加新基团。

如果用户在新旧基团中都选中了氢原子，函数将会使两个基团以氢原子为连接点进行链接。这种方式进一步扩大了用户的操作选择，允许用户以更细微的方式进行化学结构的操作。

此外，如果旧基团只包含一个原子，函数将首先删除该单个原子，然后完整地添加新基团。此设计考虑到了化学结构的完整性和准确性，尽量避免因基团更替而引发的可能错误。

函数实现方法：首先，要处理好逻辑。找到旧基团中，和鼠标点击选中的氢原子相连的原子，这个原子也就是等会要建立新键的，找到新基团中和选中原子（需要用户手动选择）相连的原子（一般是氢原子），这是新基团中等会要删除以便建立新键的原子，计算旧基团要替换的氢原子连接的原子指向相连旧基团要删除的氢原子的向量，计算新基团中要删除的相连氢原子指向选中原子的向量，在得出两个向量后将这两个氢原子删除，由于这两个新旧基团中向量代表了两个基团是否在同一方向，所以如果两个新旧基团中向量不同向，需要将新基团平移到旧基团要替换的氢原子连接的原子位置，根据之前所算得的新旧基团要替换的氢原子以及它们相关的原子构成的两个向量进行旋转新基团，由新基团的向量叉乘旧基团的向量，得到旋转轴，计算新基团的向量和旧基团的向量的夹角，接着调用旋转函数，旋转新向量所代表的整个基团至旧向量所在方向，接着需要逻辑上调整键长，也就是按照旧基团的方向向量方向进行整体平移一定距离，这时两个基团键并没有键存在。但若两向量同向或者反向，需要单独处理，若同向则不进行旋转操作，若反向，则以新基团中要成键的原子作为旋转点，任取一垂直于上述提到的新基团中向量的向量，以此为旋转轴，旋转 180 度，为什么可以任取？原因在于两向量本身就存在方向相反这个约束条件。

然后，需要找到新基团中的最长链以及新基团中的最长链，以它们头部的原子作为标记，为后续改变新基团与旧基团的二面角角度提供支持。为了找到最长链，首先需要寻找已经删除的氢原子连接的那个原子，和它相连接的所有原子，我们遍历数据，找到它连接的所有原子，放入一个列表，接着进行打擂台比链长大

小,使用每个原子这条链上所有原子长度进行对比,若长度相等,则长度相等的标志值加一,若长度更大,则覆盖之前数据,两者都要记录当前链头的原子下标,在唯一长度时,得到的原子下标唯一,长度有相等时,需要记录对应的原子下标,若相等的个数大于等于 3,则将对应相等的链上所有原子坐标相加,计算形心,如果形心坐标与两个已经删除的氢原子所连接的原子构成的向量在同一方向上,如两个甲烷相连变成乙烷的情况,这时两个甲基应该呈现反对称的状态,则我们删除最后一条对应链的所有坐标,再计算形心,这样形心就能代表另外两条最长长度链的位置了,随后在坐标数组中增加此数据,并把最大长度链头的原子下标设为坐标数组最后一位,使其指向所计算出来的形心,当相同长度为 2 时,随机选择链头原子当最大链的头部原子,设置对应标志位,在之后的旋转操作中使新基团旋转 90 度,当相同长度为 1 时,计算最大链的形心,此坐标加入坐标数组,最大长度链头的原子下标设为最后一位,经过上述计算,我们已得出 4 个点,在旧基团中代表最大长度链的链头原子,两个已经删除的氢原子所连接的原子,在新基团中代表最大长度链的链头原子,我们已经拥有了计算二面角的数据,于是计算二面角,但是如果直接用两平面的法向量进行点乘再除以两个法向量的模,最后使用反余弦函数计算二面角,若改变法向量的正负,也就是让它指向反方向,得出的二面角会变成原来计算的角度补角,也就是这个公式并不能完全反映二面角的真实值,于是我采用让由四个点中间两个点构成的轴以外的两个点投影到与它们所在的平面的法向量垂直以及轴的方向向量垂直的方向,实际上是在解向量,这个向量与所在的平面的法向量垂直以及与轴的方向向量垂直,我们解这个三元一次方程,对自由变量赋除零以外的值,即可解得向量,最后我们需要使用点乘判断计算所得的向量是否和原来的方向在同一朝向,若是相反则需要取反,这样我们得到了两个新向量。那么会不会出现四个点在一条直线的情况?在单条最长链且无其它链的情况下有可能出现,由于单链直接进行连接时符合化学规律,因此我们不用处理,直接跳过这一步即可。最后使用我们所计算的两个新向量进行点乘,判断结果正负,若是正的,则是锐角,若是负的,则是钝角,根据所得结论,对二面角进行修正。接下来,需要进行新基团的旋转。

由于我们的目标应是使两条最长链呈 180 度,按照理论来看只需旋转 180-二面角即可,但实则不然,若新基团中代表最大长度链的链头原子位于另外三个点所构成的平面之上,那么只需要正常旋转,但若这个点,位于平面之下,则需

要旋转相反的角度。

于是我们根据前三个点构成平面的法向量与从平面指向第四个点的向量的点乘正负，判断点在平面的上侧还是下侧，若在上侧根据右手定则反方向旋转，角度为平面夹角-派，旋转后，两条最大支链呈 180° ，对于环或两条相同最大长度的链，将派改为派/2，当然，若一开始没有选中氢原子，就进行简单的复制黏贴，上述操作都无需执行。最后，在数据库中添加两个基团相连的键，再添加新基团的所有键信息，添加原子信息。通过以上步骤，我们就可以实现在化学分子中添加新基团并建立键，同时满足化学规律的要求。

上述旋转操作都使用了我写的旋转函数，该函数可以对三维空间中的原子或分子进行绕指定轴旋转。该函数使用了 Python 编程语言和 NumPy 库中的函数和方法。

该函数的输入参数包括三个参数，分别是旋转轴的起点，旋转轴的终点，旋转角度。其中旋转轴的起点和旋转轴的终点参数共同确定了旋转轴的方向，而旋转角度参数则决定了旋转的角度大小。

在函数内部，首先根据旋转轴的起点和终点，计算出旋转轴的方向向量。接着，对该向量进行归一化处理，得到单位向量。然后，通过对向量的坐标进行运算，计算出旋转轴的三个分量 a、b 和 c。

接下来，通过一系列矩阵变换，实现了对三维空间中的原子或分子进行旋转的操作。首先，通过构造平移矩阵：

$$\begin{pmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-5)$$

将旋转轴平移至原点。然后，通过构造两个旋转矩阵：

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-6)$$

$$\begin{pmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-7)$$

将旋转轴转化为与 z 轴重合的轴。接着，通过构造旋转矩阵：

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-8)$$

对旋转轴进行旋转。最后，通过构造旋转，平移矩阵的逆矩阵：

$$\begin{pmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-9)$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & \frac{b}{d} & 0 \\ 0 & -\frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-10)$$

$$\begin{pmatrix} 1 & 0 & 0 & -dx \\ 0 & 1 & 0 & -dy \\ 0 & 0 & 1 & -dz \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-11)$$

将旋转轴恢复到原来的位置，并将其平移回到原始位置。

最后，对于输入的几个原子或分子的坐标，通过矩阵相乘的方式，计算出旋转后的坐标值，并将其保存到坐标数据中。效果如图 2-5 所示。

综上所述，该函数通过一系列矩阵变换实现对三维空间中的原子或分子进行旋转的操作，具有较强的数学基础和实用性，可以广泛应用于化学、物理、生物等领域中的分子结构分析和模拟研究中。

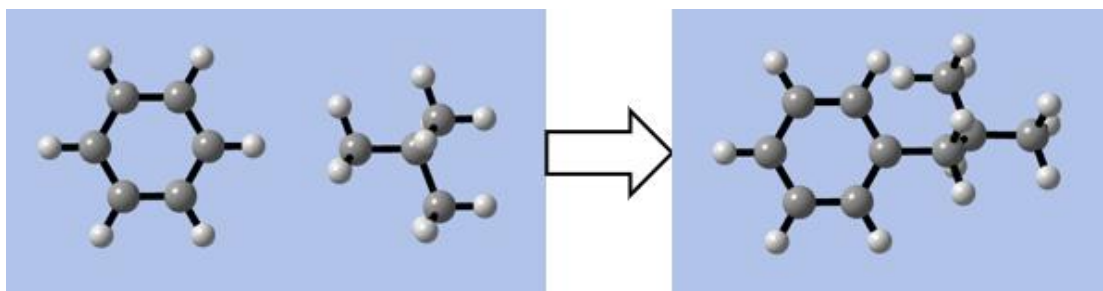


图 2-5 基团添加举例

Figure 2-5 Group addition example

2.2.1.4 两个基团选择界面

我在软件中新增了两个基团选择界面，一个是用于选择环状化合物，另一个用于选择常用的非环状结构，通过计数器，使这两个界面在关闭后可以再次打开，若多次点击打开按钮不会重复打开。

界面中的基团图标是使用 **rdkit** 提供的绘图方法进行绘图的，通过对高斯对应分子的输出文件读取，获取其中的原子坐标信息，键信息以及原子名称信息，随后根据 **rdkit** 可以接受的数据格式，将 **Gaussian** 中读取的数据进行修改，使其符合 **rdkit** 数据格式。

当鼠标悬浮在图标上，会相应地在鼠标旁边给出基团的名称。

若用户点击界面中的某个基团，则程序会取得基团名字，接着根据基团名字在数据库中寻找对应的项目，随后将此基团的原子坐标，原子信息，键连信息取出，传递给绘制程序。

环状化合物界面可选择的基团有：

benzene ,biphenyl ,naphthalene ,anthracene ,phenanthrene ,pyrene ,coronene ,pyridine ,pyrimidine ,pyrazine ,pyridazine ,cyclopropane ,oxirane ,cyclobutane ,cyclopentane ,cyclopentene ,cyclopentadiene ,furan ,tetrahydrofuran ,pyrrole ,thiophene ,cyclohexane-chair ,cyclohexane-boat ,cycloheptane ,,-cycloheptatriene ,cyclooctane ,cis-decalin ,trans-decalin ,pentalene ,iso-indene ,azulene ,fluorene ,bicycloheptane ,bicycloJoctane ,adamantane ,fullerene

非环状结构化合物界面可选择的基团有：

ethyl,n-propyl,iso-propyl,n-butyl,tert-butyl,vinyl,ethynyl,formyl,carboxyl,formamidyl

,nitro,nitroso,cyano,sulfonyl,thionyl,phosphonyl,mesylate,tosylate,ethyl,n-propyl,iso-propyl,n-butyl,tert-butyl,vinyl,ethynyl,formyl,carboxyl,formamidyl,nitro,nitroso,cyano,sulfonyl,thionyl,phosphonyl,mesylate,tosylate

目前环状化合物不提供分子图标, 非环状化合物提供分子图标, 如下图所示。

基团名	结构图	基团名	结构图	基团名	结构图
ethyl		ethynyl		cyano	
n-propyl		formyl		sulfonyl	
iso-propyl		carboxyl		thionyl	
n-butyl		formamidyl		phosphonyl	
tert-butyl		nitro		mesylate	
vinyl		nitroso		tosylate	

图 2-6 已绘制的非环状化合物图标

Figure 2-6 A drawn icon of a non-cyclic compound

2.2.1.5 双击直接输入元素名称替换原子

用户对某个原子进行双击后, 由于双击之前会触发鼠标释放动作, 因此要先将之前鼠标释放触发的事件进行撤销, 随后程序会在点击位置生成一个文本框,

用户可以输入原子的原子序号或者英文简写, 按下 **enter** 键后, 开始识别, 若识别失败, 在界面的状态栏中会进行错误提示, 如果识别成功, 则对选中原子进行修改, 将原子的颜色, 半径以及原子序数更新为用户输入的原子所具有的,

之后将文本框删除。

2.2.1.6 选中原子功能改进

在原始版本的选择函数中，代码实现了鼠标点击时在 OpenGL 渲染的场景中选择球体（原子）的功能。然而，它并未考虑球体在场景中的前后顺序，只要一个球体满足距离条件，函数就会选择该球体并停止寻找。

为了实现正确选择最靠前的球体，我对原始函数进行了以下修改：

使用一个变量 `closest_atom` 来记录当前最靠前的球体（原子）。

使用一个变量 `closest_distance` 来记录当前最靠前球体的 Z 轴距离。将其初始值设置为负无穷，以便在寻找过程中能更新为场景中球体的 Z 值。

在遍历场景中的球体时，检查每个球体是否满足距离条件。如果满足条件，将当前球体的 Z 轴距离与 `closest_distance` 进行比较。如果当前球体的 Z 轴距离更大（即更靠前），则更新 `closest_distance` 和 `closest_atom`。

遍历完所有球体后，检查 `closest_atom` 是否已更新。如果更新了，表示找到了最靠前的球体，然后根据球体的颜色执行相应的操作（选中或取消选中）。

这些修改使得选择函数能够在摄像机坐标系下正确地选择最靠前的球体，从而提高了程序的用户体验。

2.2.2 键编辑功能

2.2.2.1 选中键功能

若用户选中了键，程序会将键相连的两个原子添加到选中原子数据列表中，取消选中键，会将键相连的两个原子移出选中原子数据列表，这样可以更方便地满足用户修改键长，键类型以及二面角的需求，同时使修改键属性变得更加直观。

鼠标选中键的功能实现方法是：将键的两个原子在三维坐标系中使用旋转矩阵从初始坐标系旋转到观察坐标系中，然后将它们投影到平面上，用鼠标的二维坐标去匹配。首先算出两个原子坐标的方向向量，再求出垂直向量。使用点到直线（两个原子坐标的方向向量）的距离公式判断是否点击范围在键半径以内，同时使用方向向量的垂直向量以及两个端点分别构成的两条直线的距离来判断是否鼠标点击在键长内。同时，键长还减去了连接的两个原子的半径，使判断更加精确。

2.2.2.2 删除键功能

按照上述方法判断鼠标点击位置是否在某个键范围内，如果在，则判断键相

关的原子是否为选中颜色，如果是，则将原子的颜色更改为原来的颜色，并从选中原子列表中，删除这两个原子，此外，如果此键在选中键列表中，也需删除此键的选中信息，最后将此键的颜色信息，键信息从相应列表中删除。

2.2.2.3 多键增加功能

在过去，本软件并没有多键，原因在于多键增加方向难以判断。本文提出了一种多键增加方向判断函数的实现方法。（该方法首先计算每个原子所在分子的形心，并将其加入原子所在键的键信息中，以方便后续读取。接着，遍历每个键，计算垂直于该键的方向向量以及与形心构成的平面的法向量的向量，其中形心还起到了判断增加方向朝向的作用，最后将其传递给渲染程序。）

具体实现过程如下：首先建立一个集合，遍历所有原子，将它们所在分子的每个原子下标作为一个整体加入集合，若是重复的分子则不会添加入集合。然后，遍历集合中的每个分子，计算其形心，并再次遍历当前分子的每个原子，判断其在哪些键中。如果当前原子位于当前键中，则说明该键所在的分子的形心就是刚刚计算出来的形心，因此将形心坐标加入键信息中。接着遍历每个键，利用两个方程求解：第一个方程：求增加方向的向量与键的向量的点乘，第二个方程：求增加方向的向量与整个分子平面的法向量的点乘（使用形心构造）。在求解过程中，我们采用了 `sympy` 来求解方程。

考虑到乙烯这种分子，其并不存在平面，法向量可以旋转，因此可以直接将法向量看作是增加方向的向量（此时形心在键上，因为采用叉乘构造法向量，所以此时平面法向量计算得出的值为 0）。因此，我们只有一个方程，且有两个自由变量。为了使得增加方向在 xy 平面上增加，我们将 z 轴坐标设为 0，另一个自由变量为非零值，这样就能在屏幕上下方向增加键了。而对于正常情况下的苯环等分子，其法向量只能平移，因此最后的方程只有一个自由变量。所以我们需要进行分类讨论。此外，如果增加方向的向量与键中任意一原子指向形心坐标的方向相反（可以通过点乘的正负判断），则需要将增加方向的向量取反。最后，将所有计算得到的向量添加到一个新列表中，将添加在键信息中的形心坐标去除，并将数据传递给 `OpenGL` 作为坐标偏移方向。

2.2.2.3.1 对大型分子的额外处理

然而，这种方法不适用于大型分子，因为它们可以被看作是由较小的小分子组成。对于确定多键方向性，必须独立评估每个小分子。将整个分子视为一个整

体会导致偏差，偏离分子绘图约定。因此，在计算分子形心的过程中，我们建议遍历键信息以识别所有具有两个以上连接的原子，并排除连接氢原子的原子（因为氢原子那边无法构成分子，只有它一个原子，因此无需考虑氢原子那边的键增加方向）。这些原子连接多个小分子，我们从所有键信息中删除这些原子的键信息，将这些原子当成形式上已删除的原子（形式上已删除的原子指原子仍然存在，但是我创建了数据的备份，在备份中删除了原子的部分键信息，之后会在备份数据中进行搜索新分子操作），但是这里有一个问题，若某个原子被多个形式上已删除的原子包围，那么此时与这个原子相连的键都无法通过遍历每个原子然后在键信息的遍历中找到，从而无法为这些键提供键增加方向信息，因此需要将每个与形式上已删除的原子相连的原子找到，并计算所有形式上已删除的原子相连原子的交集，若下个判定需要删除键的原子所连接的原子出现在此交集中，我们不能删除此原子，此外如果此原子又与之前已经在形式上已删除的原子相连，我们也不能删除它，否则将找不到位于这两个原子之间的键（两种情况都无法下面所说的遍历方式找到所有键）。然后，我们通过遍历原子再遍历键信息找到删除后形成的小分子，而对于单个原子我们不判定为分子，接着计算每个小分子的形心。如果删除后小分子中的原子数小于三，则形心计算无效。在这种情况下，我们需要在原始数据中搜索与该原子相连的其他原子，直到找到三个原子，然后再按照之前的步骤进行计算。之后我们对每一个新形成的小分子中的原子进行遍历，再对每一个原子进行原始的键数据遍历，如果这个原子位于当前遍历的键中，则将原子所在分子的形心坐标加入键信息中，这样即使形式上删除了很多键，通过这种方式，仍然能把对应的形心坐标加入那些在备份数据中被删除的键的键信息中，因为可以通过原子找到原始键信息中对应备份数据中被删除的键信息，对于已加入过形心信息的键信息，不会二次加入数据。

2.2.3 界面功能

2.2.3.1 撤销重做功能

在程序中使用 QUndoStack 做撤销重做功能是一项关键的设计决策。随着用户在分子编辑器中的操作越来越复杂，他们可能会发现需要撤销或重做之前的步骤，以便更好地控制他们的工作流程^[21]。这正是 QUndoStack 的价值所在。

用户每进行一个修改分子的动作，程序将当前数据库里的数据压入栈中。这一过程实际上是程序将先前的状态保存在栈中，以便用户可以在任何时候返回到

该状态^[22]。这个过程并不会给用户带来太多的麻烦，但它确实确保了他们的数据得到了可靠的保护。

当用户想要撤销一个操作时，程序会将栈顶内容弹出并覆盖当前数据，这个过程可以有效地保护用户的数据免受不必要的更改，因为用户可以在任何时候回到之前的状态。

总之，设计撤销重做功能是一种明智的决策。它可以有效地保护用户的数据免受不必要的更改，同时也可以让用户更好地控制他们的工作流程。

2.2.3.2 平铺和瀑布布局功能

针对平铺和瀑布布局功能，Qt 中提供了 `QMdiArea` 类，该类是多文档界面的核心部件。`QMdiArea` 类允许用户在多个文档间进行导航，并支持平铺和瀑布布局，提高用户的操作效率。

在实现平铺和瀑布布局功能时，我们可以直接调用 `QMdiArea` 类提供的两个方法：`tileSubWindows` 和 `cascadeSubWindows`。其中，`tileSubWindows` 方法会将所有子窗口平均分布在可用的空间中，并将它们呈网格状排列；而 `cascadeSubWindows` 方法则会将每个子窗口以重叠的方式在主窗口中排列。效果如下图所示：

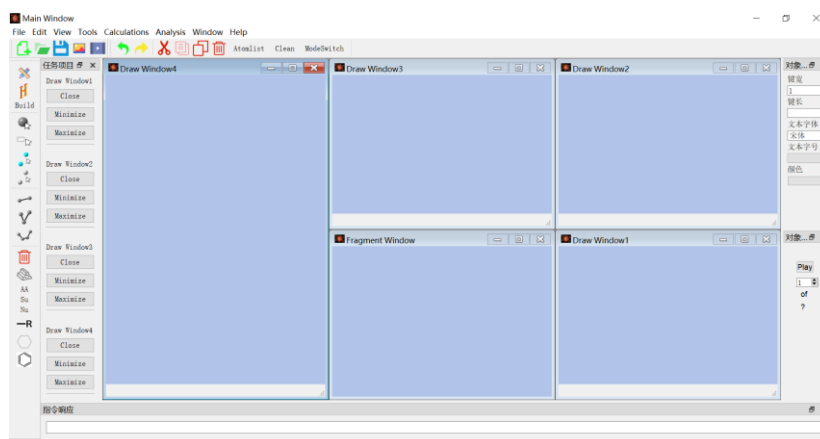


图 2-7 平铺效果展示

Figure 2-7 Tiling effect display

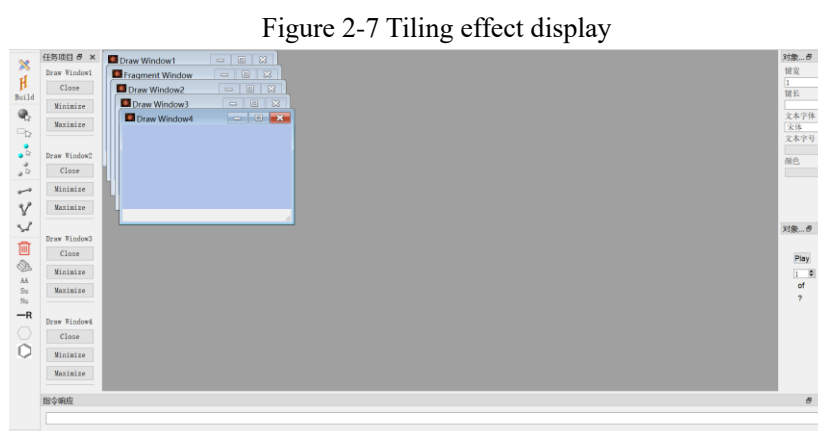


图 2-8 瀑布效果展示

Figure 2-8 Cascade effect display

2.2.3.3 绘制框图功能

我在框选功能中加入了框图的绘制。在计算机图形学中，绘制矩形是一个重要的应用^[23]。为了绘制矩形，需要对屏幕上的鼠标点击和移动事件做出响应，并将鼠标的坐标转换为 OpenGL 坐标系的坐标值。

首先，我们定义了一个类，其中包含了开始位置的 x 坐标和 y 坐标，结束位置的 x 坐标和 y 坐标这四个变量，它们分别代表矩形的左下角和右上角的坐标。类的构造函数会初始化这些变量并将清除标志设置为假，清除标志为真代表现在不应绘制。

接下来，我们定义了预备函数来准备着色器程序、VAO 和顶点缓冲区。着色器程序包括一个顶点着色器和一个片元着色器^[24]。顶点着色器只是将顶点的位置传递给 OpenGL，而片元着色器则将所有片元的颜色设置为橙色。（VAO 是

一个 OpenGL 对象，用于存储绘制矩形所需的状态。）顶点缓冲区存储了矩形的四个顶点坐标。

然后，我们定义了一个绘图函数来绘制矩形。该函数首先清除缓冲区，然后使用 VAO、着色器程序和顶点缓冲区来绘制矩形。我们使用 `glDrawArrays` 函数来绘制一个线框四边形。最后，我们使用交换缓冲函数来将缓冲区中的图像显示在屏幕上。

接下来是鼠标点击事件函数。如果鼠标左键被按下，我们将开始位置的 `x` 坐标和 `y` 坐标设置为鼠标坐标，并将清除标志设置为假，允许绘制图形。然后，我们使用更新函数来通知 OpenGL 重新绘制窗口。当鼠标释放时，清除标志为真，不允许绘制图形。

然后是鼠标移动事件函数，实时更新结束位置的 `x` 坐标和 `y` 坐标，并使用更新函数来通知 OpenGL 重新绘制窗口。

2.2.3.4 在界面右边添加控制键宽，颜色控制面板

用户先通过选中键功能，选择需要改变的键，随后修改控制面板的内容，如果选择了键宽的输入框，并输入数值，按下回车键，即可相应改变屏幕上选中的键的宽度，可同时改变多个。如果点击选择颜色按钮，会跳出颜色选择界面，这是 Qt 内置的颜色选择界面，可以自由选择颜色，当关闭颜色选择界面后，用户事先选中的键和原子的颜色会相应地改变，此功能可以帮助用户将需要突出的显示的内容标出。

2.2.3.5 界面下侧添加命令行功能

用户可以在这里的文本框中输入指令，通过正则表达式进行匹配，如果输入格式符合内置在程序中已写好的表达式，则进入相应函数进行处理^[25]。

目前提供的命令有删除命令，选中命令，`smiles` 命令。

2.2.3.5.1 删除命令

删除命令中，用户既可以选择删除某一个元素名称对应的所有原子，也可以按照原子增加的序号，选择单独要删除的原子，对于第一种情况，首先遍历已存在在画面上的原子的名称列表，逐一匹配，若是匹配成功，则将相应下标加入一个新列表，随后，遍历此新列表，并设置一个计数器，将下标减去计数器的值传入删除原子函数，然后计数器加一，这样做是因为每次删除一个原子，后序的原子在列表中都会往前移一位。

2.2.3.5.2 选中命令

选中命令的大致步骤和删除命令一样，区别在于不是删除原子，而是改变要选中的原子的颜色并将其加入选中原子的列表。

2.2.3.5.3 Smiles 命令

"Smiles 命令"作为一种技术方案，其运行机制涉及与 `rdkit` 模块的交互以及正则表达式的运用，旨在提升化学信息编码的便捷性与准确性。该方案的核心步骤主要包括接收用户输入，数据转换，以及分子的界面呈现。

首先，用户的输入会直接传递给 `rdkit` 模块进行后续的处理。这一过程中，`rdkit` 作为一个强大的化学信息处理库，能够准确解析并处理用户输入的 smiles 公式。

其次，方案对 `rdkit` 返回的数据进行了二次处理。这一步骤的实现主要依赖于正则表达式的使用，通过其对数据进行复杂且精准的模式匹配，将数据转化为本软件的数据结构。

最后，用户的输入方式及分子的呈现方式也被赋予了多样性。除了可以在命令行中直接输入 smiles 公式外，用户还可选择复制 smiles 公式后在绘制界面通过右键选择粘贴为 smiles，从而将 smiles 代表的分子呈现在界面中。此外，右键菜单中的 smiles 输入界面提供了手动输入 smiles 公式的选择，进一步增强了操作的灵活性和便捷性。

"Smiles 命令"方案通过独特的设计与实现，为用户提供了一种高效、准确并且用户友好的化学信息处理方式。

2.2.3.6 界面焦点切换功能

实现了在主窗口及其子窗口之间轮换焦点的功能。用户通过按下 '`Ctrl+Tab`' 快捷键在窗口间切换焦点。为实现此功能，我定义了一个名为 `switch_focus` 的方法，该方法根据当前子窗口焦点索引切换到下一个子窗口。

当焦点切换到最小化的窗口时，我实现了自动恢复窗口正常大小的功能。在 `switch_focus` 方法中，我检查了窗口是否处于最小化状态。如果是，我使用 `showNormal()` 方法将其恢复到正常大小。这样，用户可以在窗口之间切换焦点并继续操作。

为避免焦点在同一窗口间切换，我修改了 `switch_focus` 方法，使其在查找下一个活动窗口时跳过当前活动窗口。通过在 `switch_focus` 方法中添加循环来

查找下一个非当前活动子窗口，我确保焦点始终切换到不同的子窗口。

这些功能的实现不仅提高了用户在处理计算化学数据时的工作效率，而且改善了软件的可用性。在我们的应用场景中，用户可以快速地在绘制分子结构的主窗口和分子模板选择子窗口之间切换，从而提高了他们在计算化学领域的工作效率。

结论

本论文在原有化学建模软件的基础上进行了进一步的开发，增加了任意基团与现有分子相连接的功能，优化了软件界面，增加了多窗口、模板界面、绘制多键和撤销重做等功能。通过对实验结果的分析，本软件在易用性和灵活性方面都有了显著的提升，能够更好地满足化学研究领域中的分子建模和分析需求。

本论文的研究工作不仅提升了现有化学建模软件的功能和性能，也为进一步的研究提供了思路和方法。未来可以继续优化和拓展软件，比如增加更多的基团选择，支持更多的文件格式等等，以更好地服务于化学研究的需要。

最后，本论文所开发的化学建模软件，不仅可以为化学研究领域提供支持和帮助，也为软件工程和图形学领域的研究提供了一定的参考和借鉴价值。

参考文献

- [1]Sozykin S A. GUI4dft-A SIESTA oriented GUI [J]. Comput. Phys. Commun., 2021, 262: 1-6.
- [2]Baek N, Lee H. OpenGL sc implementation on the OpenGL hardware [J]. IEICE. Trans. Inf. Syst., 2012, J(10): 2589-2592.
- [3]Sun H, Li C, Zhang Z, et al. Computer-based method for providing advanced stereoscopic three-dimensional (3D) rendering system, involves transmitting that outputs to OpenGL application programming interface (API) to create 3D rendered image in original application, WO2021081568-A2; WO2021081568-A3[P/OL]. 2021-2-23.
- [4]Schwarz S, Uerlich S A, Monti A. Pycity_scheduling-A Python framework for the development and assessment of optimisation-based power scheduling algorithms for multi-energy systems in city districts [J]. SoftwareX, 2021, 16: 1-8.
- [5]Deringer V L, Bartok A P, Bernstein N, et al. Gaussian process gegression for materials and molecules [J]. Chem. Rev., 2021, 121(16): 10073-10141.
- [6]Cha T H. Floating shifts of s-type Gaussian type orbitals optimized in molecules local environments [J]. Chem. Phys. Lett., 2021, 773: 1-5.
- [7]Mah W H, Nazuan N H A, Yeap W S, et al. Deciding which is the best H-1 NMR predictor for organic compounds using statistical tools [J]. C. R. Chim., 2022, 25: 83-95.
- [8]Giesecking R L M. A new release of MOPAC incorporating the INDO/S semiempirical model with CI excited states [J]. J. Comput. Chem., 2021, 42(5): 365-378.
- [9]Faure Q, Jevermovic T T. Molecular dynamics and reaction kinetics analyses of gamma radiation impact on concrete hydration [J]. Nucl. Technol. Radiat Prot., 2020, 35(1): 1-15.
- [10]Hua W J, Tian G J, Luo Y. Theoretical assessment of vibrationally resolved C1s X-ray photoelectron spectra of simple cyclic molecules [J]. Phys. Chem. Chem. Phys., 2020, 22(35): 20014-20026.
- [11]Abraham R J, Cooper M A. The use of MM/QM calculations of C-13 and N-15 chemical shifts in the conformational analysis of alkyl substituted anilines [J]. Magn. Reson. Chem., 2020, 58(6): 520-531.
- [12]Zhao Y, Zhang L, Zhang Z, et al. Designing protein fusion and maintaining activity of

subunit by performing alpha helix fusion with protein A D domain, screening using Chimera software, predicting secondary structure, calculating energy and comparing structural stability, CN111462813-A; CN111462813-B[P/OL]. 2020-10-20.

[13]Nishimura K, Nishimori H, Katzgraber H G. Griffiths-McCoy singularity on the diluted Chimera graph: Monte Carlo simulations and experiments on quantum hardware [J]. Phys. Rev. A., 2020, 102(4): 18-35.

[14]Lu X J. DSSR-enabled innovative schematics of 3D nucleic acid structures with PyMOL [J]. Nucleic. Acids. Res., 2020, 48(13): 8-15.

[15]Mooers B H M. Shortcuts for faster image creation in PyMOL [J]. Protein Sci., 2020, 29(1): 268-276.

[16]Mooers B H M. A PyMOL snippet library for Jupyter to boost researcher productivity [J]. Comput Sci Eng, 2021, 23(2): 47-53.

[17]Tao Y W, Zou W L, Nanayakarra S, et al. PyVibMS: a PyMOL plugin for visualizing vibrations in molecules and solids [J]. J. Mol. Model., 2020, 26(10): 1-12.

[18]Liu H, Jin Y, Ding H J. MDBuilder: a PyMOL plugin for the preparation of molecular dynamics simulations [J]. Brief. Bioinform., 2023, 24: 1-7.

[19]Kagami L P, Das G M, Timmers L, et al. Geo-Measures: a PyMOL plugin for protein structure ensembles analysis [J]. Comput. Biol. Chem., 2020, 87: 1-8.

[20]Wang J. Method for switching window, involves using window switching instruction for indicating relationship, and determining focus window after switching window according to focus window and window switching instruction, CN113093961-A; CN113093961-B; WO2022218152-A1[P/OL]. 2017-10-20.

[21]Kim M G, Park J M, Lee J, et al. Interaction undo support device for supporting undo of interaction performed between virtual objects, has interaction undo interface which is activated in response to gesture not corresponding to user interaction gesture, US2021055802-A1; KR2021021720-A; KR2306392-B1[P/OL]. 2021-9-30.

[22]Li C, Liao D, Wang L, et al. Method for automatically superimposing and archiving new and old file data, involves replacing historical archive data with archive data when archive data is repeated with historical archive data, and storing replaced per-item data into target database, CN114461880-A[P/OL]. 2022-5-10.

[23]Liu L, Wen W, Xue S, et al. Spherical rectangular drawing method for computer graphics, involves using vertex P1 and vertex P3 to calculate spatial coordinates of vertex P2 and vertex P4, and mapping small triangle from plane onto sphere, CN115482357-A[P/OL]. 2022-12-16.

[24]Luo Y, Xiao B. Method for rasterizing computer model, involves transferring linear expressions of code from fragment shader to vertex shader, and executing code comprised in vertex shader on computer model, WO2023280291-A1; US2023011882-A1[P/OL]. 2023-1-12.

[25]Han W, Han Y, Li M, et al. Regular expression extracting method, involves selecting regular expression with highest matching degree as optimal regular expression of target text according to numerical value of matching degree corresponding to regular expression, CN111400449-A[P/OL]. 2020-7-10.

致谢

在完成本论文的过程中，我得到了许多人的支持和帮助，在此向他们表达我最真挚的感谢。

首先，我要感谢我的导师雷鸣教授。他不仅在我论文的选题和研究方向上给了我很多指导和启示，还在整个研究过程中给了我很大的帮助和支持。他的认真负责的态度和严谨科学的精神，一直激励着我努力学习和钻研。

其次，我还要感谢我的实验室同事们。他们给予我很多技术上的指导和帮助，还与我分享了很多有趣的科研成果和经验，让我受益匪浅。特别是刘阳秋同学，在我的研究过程中给了我很多有用的建议和帮助，在此向他表示衷心的感谢。

此外，我还要感谢我的家人和朋友们。他们一直给予我鼓励和支持，让我在研究和生活中都能够保持积极向上的精神状态。

最后，我要感谢所有为本论文提供帮助和支持的人。他们的关心和帮助，对我的论文研究和学习生活都有很大的促进作用，我将永远铭记在心。