

MINI-LAB: A TOOL TO VISUALIZE NORMAL MODES OF VIBRATION  
USING JAVA3D

by

QIN ZHANG

(Under the direction of Eileen T. Kraemer)

ABSTRACT

Vibrational analysis of polyatomic molecules requires tools with the capability to construct the system and display the normal modes. Mini-Lab, developed in this thesis work, is a 3D visualization tool and vibration analyzer that serves both educational and research needs. The goal is to help the user visualize the atoms and bonds, compute the normal modes for the system, and generate the animated vibrational pattern. Users can experiment with different masses and bonds to construct a vibration system and understand the normal modes when studying harmonic oscillation or small vibrations in classical mechanics. This tool has been used in research that analyzes the vibration of nanocrystals. Images of the molecule configuration and movies of the vibration can be generated by this tool.

INDEX WORDS: Java3D, Harmonic Oscillation, Normal Modes, Molecular  
Vibration

MINI-LAB: A TOOL TO VISUALIZE NORMAL MODES OF VIBRATION  
USING JAVA3D

by

QIN ZHANG

B.S., Shanghai Jiao Tong University, 1996

A Thesis Submitted to the Graduate Faculty  
of The University of Georgia in Partial Fulfillment  
of the  
Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2004

© 2004

Qin Zhang

All Rights Reserved

MINI-LAB: A TOOL TO VISUALIZE NORMAL MODES OF VIBRATION  
USING JAVA3D

by

QIN ZHANG

Approved:

Major Professor: Eileen T. Kraemer

Committee: Thiab R. Taha  
Steven P. Lewis

Electronic Version Approved:

Maureen Grasso  
Dean of the Graduate School  
The University of Georgia  
May 2004

## ACKNOWLEDGMENTS

First of all, I wish to acknowledge and express my appreciation to Dr. Eileen Kraemer, my major professor, for making this project possible. She took time from busy schedule to discuss with me and review the work for me. Without her inspirations, feedbacks and suggestions, I could not have finished the project so smoothly.

I would also like to thank Dr. Thiab Taha and Dr. Steven Lewis for serving on my advisory committee. I feel extremely lucky to have had the opportunity to learn from such an impressive and supportive committee. I owe a special thank you to Dr. Steven Lewis who provided me the needed and generous support. He dedicated a considerable amount of time discussing with me the vibration problem. These discussions greatly helped me in bringing the project into fruition.

Last but not least, I wish to thank my wife Yong, my parents and my entire family for their support over many years.

## TABLE OF CONTENTS

|  | Page |
|--|------|
| ACKNOWLEDGMENTS . . . . .                | iv   |
| LIST OF FIGURES . . . . .                | vii  |
| CHAPTER                                  |      |
| 1 INTRODUCTION . . . . .                 | 1    |
| 1.1 PURPOSE . . . . .                    | 1    |
| 1.2 USER GROUPS . . . . .                | 2    |
| 2 BACKGROUND . . . . .                   | 3    |
| 2.1 HARMONIC OSCILLATION . . . . .       | 3    |
| 2.2 3D VISUALIZATION WITH JAVA . . . . . | 6    |
| 3 TOOL OVERVIEW . . . . .                | 11   |
| 3.1 INSTALLATION . . . . .               | 11   |
| 3.2 MAIN FEATURES . . . . .              | 11   |
| 3.3 USER INTERFACE . . . . .             | 17   |
| 3.4 WALK-THROUGH SAMPLES . . . . .       | 21   |
| 4 DESIGN AND IMPLEMENTATION . . . . .    | 30   |
| 4.1 TOP LEVEL DESIGN . . . . .           | 30   |
| 4.2 DETAILED DESIGN AND CODING . . . . . | 31   |
| 5 CONCLUSION AND FUTURE WORK . . . . .   | 39   |
| 5.1 CONCLUSION . . . . .                 | 39   |

|                                  |    |
|----------------------------------|----|
|                                  | vi |
| 5.2 FUTURE WORK . . . . .        | 39 |
| BIBLIOGRAPHY . . . . .           | 41 |
| APPENDIX                         |    |
| A INSTALLATION . . . . .         | 43 |
| A.1 SYSTEM REQUIREMENT . . . . . | 43 |
| A.2 TEST RUN . . . . .           | 44 |
| B USER MENU . . . . .            | 45 |

## LIST OF FIGURES

|      |  |    |
|------|--|----|
| 2.1  | An Example of Tree Structure in a Scene Graph . . . . .        | 7  |
| 2.2  | An Appearance Node in Java3D . . . . .                         | 9  |
| 3.1  | Picking Atom and Bond in 3D Canvas . . . . .                   | 14 |
| 3.2  | User's Position and Orientation in 3D Scene . . . . .          | 15 |
| 3.3  | Mini-Lab Online Help . . . . .                                 | 17 |
| 3.4  | Main Layout of Mini-Lab . . . . .                              | 18 |
| 3.5  | Field of View . . . . .  | 19 |
| 3.6  | Initialization of Mini-Lab . . . . .                           | 21 |
| 3.7  | Dialog of Adding an Atom . . . . .                             | 22 |
| 3.8  | Appearance Settings in Configuration . . . . .                 | 23 |
| 3.9  | An Example of Atom Layout . . . . .                            | 23 |
| 3.10 | Dialog of Adding a Bond . . . . .                              | 24 |
| 3.11 | An Example of Bond Layout . . . . .                            | 25 |
| 3.12 | Modification of Symmetry Constraints . . . . .                 | 26 |
| 3.13 | An Example of System Layout after Adding Symmetries . . . . .  | 27 |
| 3.14 | Removal of Atom and Bond . . . . .                             | 28 |
| 4.1  | Top Level Module in Mini-Lab . . . . .                         | 31 |
| 4.2  | GUI for List . . . . .   | 32 |
| 4.3  | 3D Visualization for the Configuration of a Molecule . . . . . | 33 |
| 4.4  | Projection and Picking of an Object . . . . .                  | 34 |
| 4.5  | 3D Animation . . . . .   | 36 |

## CHAPTER 1

### INTRODUCTION

“In the matter of physics, the first lessons should contain nothing but what is experimental and interesting to see. A pretty experiment is in itself often more valuable than twenty formulae extracted from our minds.” *Albert Einstein*

The tool described in this thesis is designed to make the user feel like she is doing vibrational experiments for coupled oscillators. The experiments are not done in a real lab, but rather in a computer, a small virtual lab. Hence, the tool is named Mini-Lab.

#### 1.1 PURPOSE

Nanotechnology is an increasingly important research area. Much research in the field of nanoscience has been carried out to gain insight into interactions that occur at the atomic level. Among the various approaches employed, one useful method is to perform vibrational analysis, from which we can obtain information including the infrared spectra and the bonding properties. The harmonic approximation in this analysis leads us to calculate the normal modes of vibration. However, the quantity of data provided for the normal modes is very large. Looking at the results in their numeric form is cumbersome and impractical, even for simple molecules with only a few atoms. The alternative is to use a visualization tool to generate a 3D animation to permit better understanding of the physical system.

Another design goal of the project is to make a tool that can be used not only for research purposes, but also for education. As a basic physical phenomenon in nature, vibrations exist in the macroscopic world as well as the microscopic world. Thus, the harmonic vibrations are a crucial chapter in classical mechanics and quantum theory as well. Because large systems are hard to illustrate, the existing examples in the text book usually look only into small systems with very few oscillators. As a complement, the same visualization tool described for research applications will also be very helpful in an educational setting for the students to understand multiple coupled oscillators. The students can treat atoms just as balls with masses and treat the bonds as springs that connect those balls. They can adjust the masses or the spring constants to see different normal modes of vibration. Given a specification of the symmetry, the tool can generate the vibration modes with specific symmetry properties. This feature is also a good educational tool for graduate students for understanding the basic concepts in group theory, which is closely related to symmetry[14].

## 1.2 USER GROUPS

The potential users of the tool can be categorized into two groups. The first group consists of researchers who would like to perform vibrational analysis for their systems of interest and visually understand the results. The second group consists of undergraduate students who are studying the small-oscillation problem in classical mechanics and high-level graduate students who are studying group theory. The current demo version of the tool contains the basic functions to serve for both user groups. A dedicated version of the tool can be made later specifically for different user groups, in addition to the demo version.

## CHAPTER 2

### BACKGROUND

The development of Mini-Lab combines knowledge in both physics and computer science. In this chapter, a little background of Harmonic Oscillation and Java programming which this project involves is offered. A straightforward mathematical method in obtaining the normal modes of vibration is given. Java3D [1], an efficient 3D programming package for Java, is also briefly introduced.

#### 2.1 HARMONIC OSCILLATION

Mini-Lab visualizes the normal modes of vibration of coupled oscillators. The coupled oscillators are the mass points coupled by some potential energies. For example, we can consider some balls that we ignore the shape and some springs that interconnect these balls as a system of coupled oscillators. Molecules, nanocrystals and solids can also be modeled as coupled oscillators when we study their vibrations. In the molecular system, the oscillators are the atoms and the “springs” are the interactions among the atoms. In this paper and in the Mini-Lab tool, an oscillator is called an *atom*; an interaction between a pair of atoms is called a *bond*. The line passing through the atom pair is called the *axis* of the bond. The entire set of coupled oscillators is called a *system* or a *molecule*, although the real system could be the “balls and springs”.

The development of Mini-Lab requires basic knowledge of the elementary dynamical theorems for small oscillations. The physical system Mini-Lab visualizes is a set

of atoms that are making small vibrations about their equilibrium position. We can set up a classical expression for the potential energies and kinetic energies of the molecule in terms of the coordinates of the atoms [15]. Suppose there are  $N$  atoms in the system. Let  $x_s$ ,  $y_s$  and  $z_s$  be the coordinates of the  $s^{\text{th}}$  atom with mass  $m_s$  and  $x_{s0}$ ,  $y_{s0}$  and  $z_{s0}$  be the equilibrium position of the  $s^{\text{th}}$  atom. Then displacements from the equilibrium,  $\Delta x_s$ ,  $\Delta y_s$  and  $\Delta z_s$ , will be measured by  $x_s = x_{s0} + \Delta x_s$ ,  $y_s = y_{s0} + \Delta y_s$ ,  $z_s = z_{s0} + \Delta z_s$ . If we use generalized coordinates for the system, we can treat each x, y and z coordinate of each atom as one degree of freedom. Then we have in total  $3N$  degree of freedom,  $q_1, q_2, \dots, q_{3N}$ , where  $q_1 = x_1, q_2 = y_1, q_3 = z_1, q_4 = x_2, \dots, q_{3N} = z_N$ . If we shift the arbitrary zero of potential energy of the system to coincide with the equilibrium potential energy, then, as the atoms experience no force at their equilibrium position, the potential energy of the system can be written down as

$$V(q_1, \dots, q_{3N}) = \frac{1}{2} \sum_{i,j=1}^{3N} \left( \frac{\partial^2 V}{\partial q_i \partial q_j} \right)_0 \Delta q_i \Delta q_j, \quad (2.1)$$

where we have thrown away the higher order terms in the Taylor expansion assuming the oscillation is small. The above approximation is called the *harmonic approximation*, and a vibration within harmonic approximation is called a harmonic oscillation. The kinetic energy also can be written as a quadratic function of the velocities

$$T = \frac{1}{2} \sum_{i=1}^{3N} m_i \dot{q}_i^2, \quad (2.2)$$

where  $m_i$  is the mass of the atom that moves at the degree of freedom  $q_i$ . Apply Lagrange's equation [12] of motion

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0, \quad (2.3)$$

where  $L=T-V$ , we get

$$m_i \ddot{q}_i = - \sum_{j=1}^{3N} K_{ij} \Delta q_j, \quad (2.4)$$

where  $K_{ij} = \frac{\partial^2 V}{\partial q_i \partial q_j}$  forms a  $3N \times 3N$  matrix. The K matrix is also called the force-constant matrix as equation 2.4 is the generalization of Hooke's Law for a mass on a spring.  $K_{ij}$  is just the force on the degree of freedom at  $q_i$  due to a unit displacement from the equilibrium position in the degree of freedom at  $q_j$ . The K matrix is symmetric because  $K_{ij} = \frac{\partial^2 V}{\partial q_i \partial q_j} = \frac{\partial^2 V}{\partial q_j \partial q_i} = K_{ji}$ . If we try the  $\Delta q_i = \sum_{k=1}^{3N} C_k a_{ik} e^{-i\omega_k t}$ , where  $C_k$  gives the complex amplitude of the oscillation with the frequency  $\omega_k$ , and use  $\dot{q}_i = \Delta \dot{q}_i$ , the equation of motion can be transformed to :

$$\begin{pmatrix} K_{11} - \omega^2 m_1 & K_{12} & \cdots & K_{1,3N} \\ K_{21} & K_{22} - \omega^2 m_2 & \cdots & K_{2,3N} \\ \vdots & \vdots & \cdots & \vdots \\ K_{3N,1} & K_{3N,2} & \cdots & K_{3N,3N} - \omega^2 m_{3N} \end{pmatrix} \begin{pmatrix} \Delta q_1 \\ \Delta q_2 \\ \vdots \\ \Delta q_{3N} \end{pmatrix} = 0 \quad (2.5)$$

Equation 2.5 is a generalized eigenvalue problem with eigenvalue  $\omega^2$ , which implies a secular determinant with  $3N$  roots  $\omega_k^2$ ,  $k = 1 \dots 3N$ . The  $3N$   $\omega_k$ s are called *eigenfrequency*. The eigenvectors of the solution can transform the coordinates space  $\{q_i\}$  into a new set of generalized coordinates called *normal coordinates*. Each of the new coordinates corresponds to a vibration of the system with only one frequency, and the eigenvectors, or the component oscillations, are called *normal modes of vibration*. To focus more on the 3D visualization of normal modes, the demo uses a simple model to generate the force-constant matrix. In this model, each pair of atoms is coupled by a bond which has only two independent force-constants. The longitudinal force-constant  $K_l$  produces *longitudinal force* if two atoms have a relative displacement along the axis of bond. The transverse force  $K_t$  constant produces a *transverse force* if two atoms have a relative displacement perpendicular to the axis and opposite to each other. In this paper, the motion that causes longitudinal force is called *compressing or stretching* and the motion that causes transverse force is called *shearing*. For example, if atom A and atom B are both on the z axis, the force-constant matrix

can be made as

$$\begin{pmatrix} -K_t & 0 & 0 & K_t & 0 & 0 \\ 0 & -K_t & 0 & 0 & K_t & 0 \\ 0 & 0 & -K_l & 0 & 0 & K_l \\ K_t & 0 & 0 & -K_t & 0 & 0 \\ 0 & K_t & 0 & 0 & -K_t & 0 \\ 0 & 0 & K_l & 0 & 0 & -K_l \end{pmatrix}$$

After we get the force-constant matrix  $K$ , we can use equation 2.5 to obtain  $3N$  normal modes of vibration and  $3N$  corresponding eigenfrequencies. Any real physical vibration is just a linear combination of the normal modes. Degeneracy of the normal modes, in which two or more modes share the same frequency, reflects the symmetry of the system, which can be understood using Group theory. In this paper, rotation about an axis, reflection through a plane or inversion with respect to the origin is called a *symmetry operation* or *mapping operation*. If the system does not change under a symmetry operation, then the system contains this symmetry.

## 2.2 3D VISUALIZATION WITH JAVA

With the help of advanced Java technologies including Java 3D [1] and Java Advanced Imaging [2], we are able to program this tool to construct the molecular system and visualize the eigenmodes of vibration in three dimensions. The following sections do not go into details of Java3D programming [10][3], but rather give an overview of the Java elements used in this project.

### 2.2.1 INTRODUCTION TO JAVA3D

In Mini-Lab, 3-D reconstruction of the scene is produced by using the Java 3D API [1], a hierarchy of Java classes that serve as the interface to sophisticated 3D graphics programming. Java3D programming is at a high level and it gives an easy

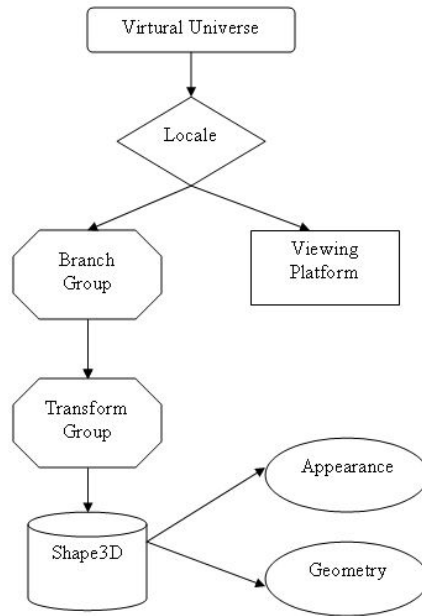


Figure 2.1: An Example of Tree Structure in a Scene Graph

way for the programmer to create and manipulate the 3D geometric objects without knowing the details of the 3D graphics rendering. For example, a scene graph can be constructed in a tree structure by instances of Java 3D objects, as shown in Figure 2.1. The scene graph completely specifies the contents of a virtual universe, which is to be rendered into 3D graphics. The term *virtual universe* commonly refers to the three dimensional virtual space populated by Java 3D objects. There are hundreds of fields and methods in the classes of the Java 3D API. The programming of Mini-Lab uses a few of them to realize the 3D visualization and animation for the vibration modes.

### 2.2.2 COORDINATE SYSTEM

The coordinate system of the Java 3D virtual universe is right-handed. Visualized in screen, by default the x-axis is positive to the right, y-axis is positive up, and z-axis is positive toward the viewer. In this tool, it uses a SimpleUniverse Class in Java 3D object. So the origin of the coordinate is chosen at the center of the 3D canvas panel.

### 2.2.3 3D TRANSFORMATION

The transformation changes the position, orientation and size of the 3D shape objects, such as the sphere and cylinder. The transformation is realized by the Transformgroup node in the rendering tree. The Transformgroup node contains a transform3D object which internally represents a  $4 \times 4$  double-precision floating point matrix. The Java 3D model for  $4 \times 4$  transformations is:

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} T_{00} & T_{01} & T_{02} & T_{03} \\ T_{10} & T_{11} & T_{12} & T_{13} \\ T_{20} & T_{21} & T_{22} & T_{23} \\ T_{30} & T_{31} & T_{32} & T_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \quad (2.6)$$

This transformation is in the homogeneous coordinates [13]. Instead of being represented by  $(x, y, z)$ , each point in the virtual universe is represented by a 4-element quaternion  $(x, y, z, w)$ .  $(x, y, z, w)$  represents the same point as  $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}, 1)$  if  $w \neq 0$ . If  $w = 0$ , the point is at infinity. The transformation matrix  $T$  can perform translations, rotations, scaling and shearing [11]. For example,

$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

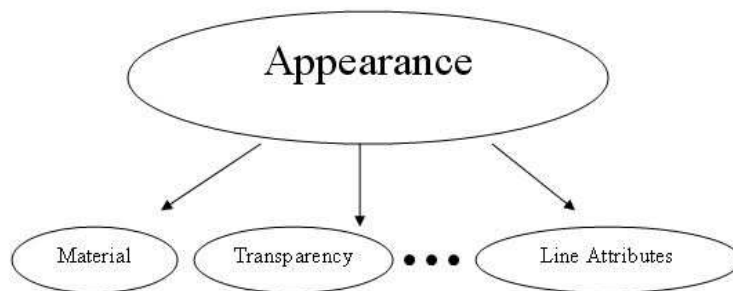


Figure 2.2: An Appearance Node in Java3D

makes  $(x, y, z, 1)$  a translation to  $(x+d_x, y+d_y, z+d_z, 1)$ , and

$$\begin{pmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

makes  $(x, y, z, 1)$  a shearing to  $(x+sh_x \cdot z, y+sh_y \cdot z, z, 1)$

#### 2.2.4 GEOMETRY AND APPEARANCE

Mini-Lab uses Shape3D objects in Java3D to generate 3D shapes such as atoms and bonds. Shape3D objects may reference both a Geometry and an Appearance object. The Geometry object specifies the per-vertex information of a visual object. The Appearance object refers to several different appearance attribute objects, as shown in Figure 2.2. For example, in Mini-Lab a Material object is defined for the appearance of an atom or bond.

### 2.2.5 INTERACTION AND ANIMATION

The changes of objects as a result of user actions are called Interaction and the changes that occur without direct user actions are called Animation [3]. Interaction and animation are specified with Behavior objects which are used to change the scene in response to some stimulus. For example, the Mouse Behavior class uses the mouse as input for interactions including translating (moving in a plane parallel to the image plate), zooming (moving perpendicular to the image plate), and rotating. The stimulus, the reason for change in the time-based animations, can be made by an Alpha object in Java3D. An alpha object produces a value, called the alpha value, between 0.0 and 1.0, changing with respect to time. Interpolators, customized behavior objects for animation, use the Alpha object to provide animations of visual objects. Technically, the visualization of the vibration modes in Mini-Lab is implemented by interaction and animation methods in the Java3D API.

### 2.2.6 JAVA ADVANCED IMAGING

Java Advanced Imaging offers a Java based open-specification, cross platform, extensible imaging API [2]. The Image File I/O is used in programming of this tool to export the images of the configuration and generate the movies for the vibration.

## CHAPTER 3

### TOOL OVERVIEW

#### 3.1 INSTALLATION

Programmed in Java, Mini-Lab is a cross-platform application. The prerequisite to run Mini-Lab is to have J2SE [4] Runtime Environment set up and Java3D installed. The latest version of J2SE and Java3D can be downloaded from Sun's web site [5]. The Java class files of Mini-Lab have been archived in a Jar file. For detail of the installation, availability, and portability, the user can refer to appendix A.

#### 3.2 MAIN FEATURES

Mini-Lab includes many useful features that are required to visualize the vibrational modes of coupled oscillators under harmonic approximation. A user friendly interface including menu and tool bar makes the configuration of the system easy. The atom and bond information can be viewed immediately in the information panel if they are selected in the virtual 3D world. The viewer's position and the symmetry information of the system are also easy to obtain from the main layout. The program can compute the eigenmodes after the system is configured. A list of eigenfrequencies will be shown on one panel in the main layout. If one eigenfrequency is selected, the animated normal modes of the system will be shown in the virtual 3D canvas. The configuration can be saved and reloaded later. Images of the system configuration and small movies of the vibrational modes can be exported to files in standard formats.

### 3.2.1 ATOM CONFIGURATION

Atoms or oscillators are considered as points with certain masses in our model for calculating the normal modes. Atoms' positions should be specified at the beginning of the system construction. If the user clicks the "Add Atom" menu item or the corresponding tool-bar button, a dialog will be shown to ask for the input of the atom configuration. Although atoms are considered mathematically as points without shape in our calculation for modes, they are visualized as spheres with a certain appearance in 3D space. The proper radius can be used as an indication of the atom types. As the radius is a manually specified parameter, the user has the flexibility to specify any radius for any type of atoms. In the program, the radius parameter is also used to judge if two atoms are at the same position. In other words, two atoms cannot overlap each other when they are viewed as spheres. The program judges the overlap of two atoms by their positions and radii. The program will give a warning if the user adds an atom at a space already occupied by another atom. In addition to radius, atoms can also be distinguished by different colors in the visualization. The appearance of atoms can be selected at will with different color, including ambient color, diffuse color, specular color and emissive color. The ambient color sets the color reflected off the surface of the material; the diffuse color sets the color of the material when illuminated; the specular color sets specular color or highlights of the material; the emissive color sets the light the material emits.

### 3.2.2 BOND CONFIGURATION

The bonds are specified after the atoms have been entered into the system. The user can select the atom pairs and click the "Add Bond" menu item or the corresponding tool-bar button. Then a dialog similar to the dialog for adding an atom will be shown. The program uses a thin cylinder to indicate a bond. One can specify the

bond between each pair of atoms. Just like the atoms, the bonds also contain a radius parameter. Note that this parameter has no effect on the calculation of the modes. When a bond is added, the user should first choose the atoms to be connected with bonds. Then the program will present a dialog through which the user may specify the force-constant and the radius of the bond. The user can manually use different appearances and radii to indicate different types of bonds. The program has no constraint for the user to specify the radius or the color of the bond. The bonds can overlap or crossover. The only constraint for the bonds is that the user cannot specify more than one bond between two atoms.

### 3.2.3 SELECTION OF ATOM AND BOND

The selection of atoms or bonds allows further operations on those atoms or bonds. After selection, the user can modify or delete objects. For example, the user can change the appearance of the selected objects. To add bonds, user also must select the atom pairs first. There are two ways for the user to select. One can use the mouse to *pick* the object in the 3D canvas by clicking the left mouse button. To select more than one object, the user can simply press the Ctrl key and continue clicking. The selected object will blink, and, at the same time, the items for the selected objects in the information panel will also be shown selected. Another way to select a 3D object is to choose the atom or bond from the list in the information panel on the main layout. The user can also select multiple objects in the same manner. For example, to select consecutive items, click the first item, press and hold down Shift key, and then click the last item. To select nonconsecutive items, press and hold down Ctrl key, and then click each item. In this case, the selected object will also blink in the 3D canvas. In both ways of selection, detailed information of the selected atoms and bonds will be shown in the information box, as shown in Figure 3.1.

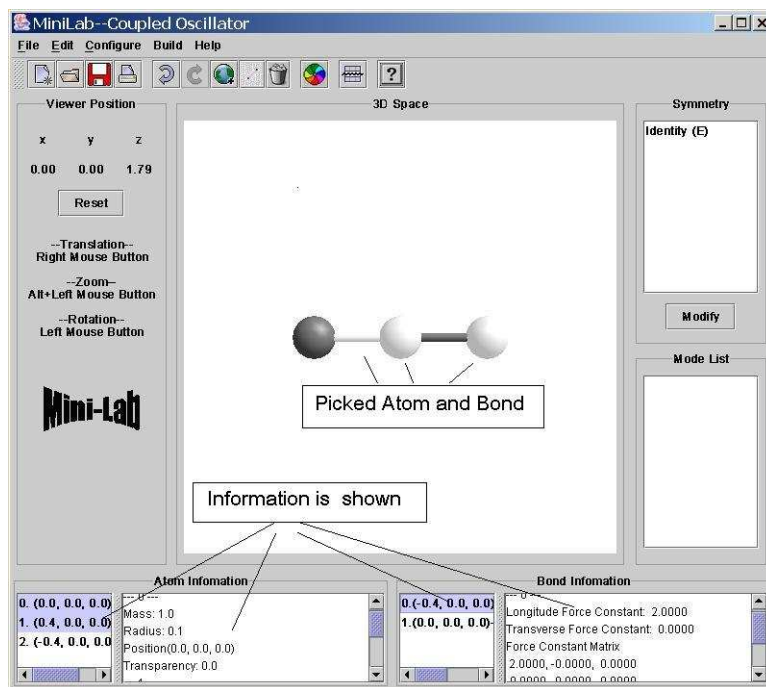


Figure 3.1: Picking Atom and Bond in 3D Canvas

### 3.2.4 NAVIGATION OF THE 3D SPACE

In general, navigation through the 3D space can be considered either as a rotation and translation of the molecular system, or as a change of the viewer's position. In this program, the atoms' coordinates remains unchanged. Only the viewer's position changes.

When the user starts the program or clicks the reset button on the left panel, the coordinate system of the Java 3D virtual universe is reset. After reset, the x-axis is positive to the right, y-axis is positive up, and z-axis is positive toward the viewer before the screen. Objects with x and y range of -1.0 to +1.0 at  $z=0$  plane can be fully viewed across. The user can change the viewer's position by dragging the mouse

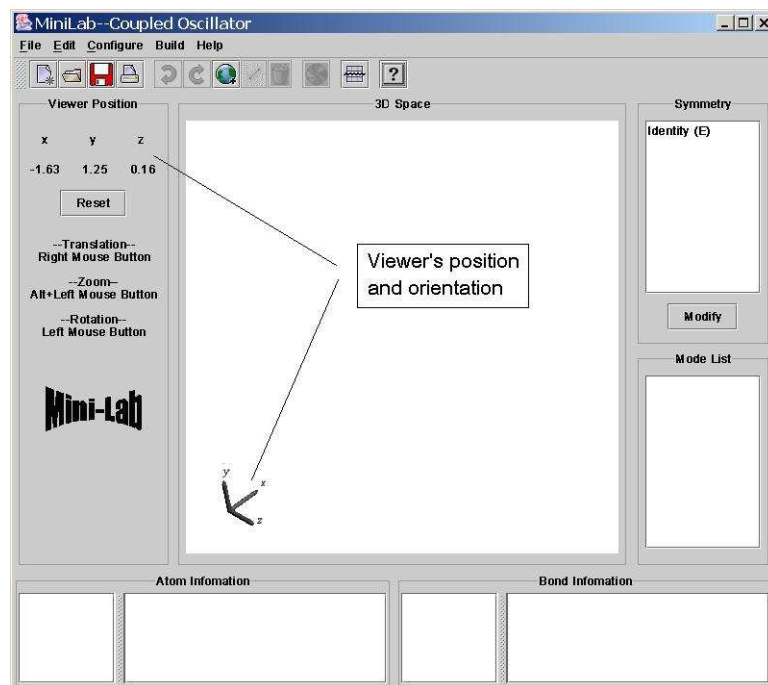


Figure 3.2: User's Position and Orientation in 3D Scene

and pressing some keys. Detail of how to navigate in 3D space is shown in section 3.3.1.

The viewer's position is shown in real time on the left panel of the main layout. The user can also set the option to show the coordinate system in the screen and thus know the viewer's orientation. Figure 3.2 shows the indication of the viewer's position and orientation.

### 3.2.5 CONFIGURATION SPEEDUP

The user can speed up the configuration of the system by specifying the symmetry operations of the system if any exist. This feature will be further developed to analyze mode symmetries within Group theory. In the current version, the user can specify

the symmetry operations in the system explicitly. Thus, the burden of configuring a large system will be greatly reduced if the system contains rotation, reflection and inversion symmetries. Details and examples of the symmetry configuration will be given in the next chapter.

### 3.2.6 COMPUTATION AND VISUALIZATION OF NORMAL MODES

After the bonds are specified, the user can let the tool compute the normal modes of the system by just clicking a button in the tool-bar or the corresponding menu item. The frequencies of the modes are shown in the list in the right panel of the main layout. If one mode is selected from the list, the animated picture of the selected mode will be shown in the 3D canvas. The user still can use the mouse to navigate to get a better view. The user can also set the vibrational period in the visualization. As the frequencies of the modes are different, the user can specify the vibrational period in visualization for the modes with the highest and the lowest eigenfrequencies. The vibrational periods of the other modes will be interpolated between the highest vibrational period and lowest vibrational period in the visualization. The user can also set a proper vibrational period to all modes by setting the highest and lowest period for visualization to be the same.

### 3.2.7 EXPORTING OF THE 3D SCENE

When one mode is selected in the mode list, the animated 3D picture will be shown in the 3D canvas. At the same time, the “Save Movie” item is enabled. The user can save the animated motion to a QuickTime movie file. To stop the animation, the user can deselect the mode by clicking the selected mode item when pressing the CTRL key. Then all atoms will stop vibration and return to their equilibrium position. At this time the “Save Movie” item is disabled, but the “Save Screen” item is enabled. Now the user can save the static configuration on the screen to a jpeg

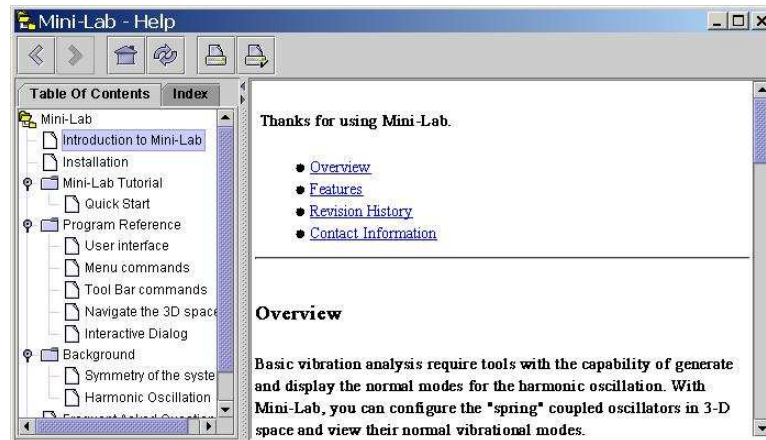


Figure 3.3: Mini-Lab Online Help

file. He can also print the 3D canvas to the printer. Moreover, the user can adjust the background and the light to get a better view or image.

### 3.2.8 MINI-LAB HELP

The tool Mini-Lab also offers online help as shown in Figure 3.3. The help window gives the table of contents in a tree structure on the left navigation panel and details for the selected topic on the right content panel.

## 3.3 USER INTERFACE

### 3.3.1 MAIN LAYOUT

The main layout of MiniLab, shown in figure 3.4, has the following main parts:

- Main Menu - The top-most part of the window is the Main Menu. It controls all functions of Mini-Lab. Functions of menu items are explained in detail in Appendix B.

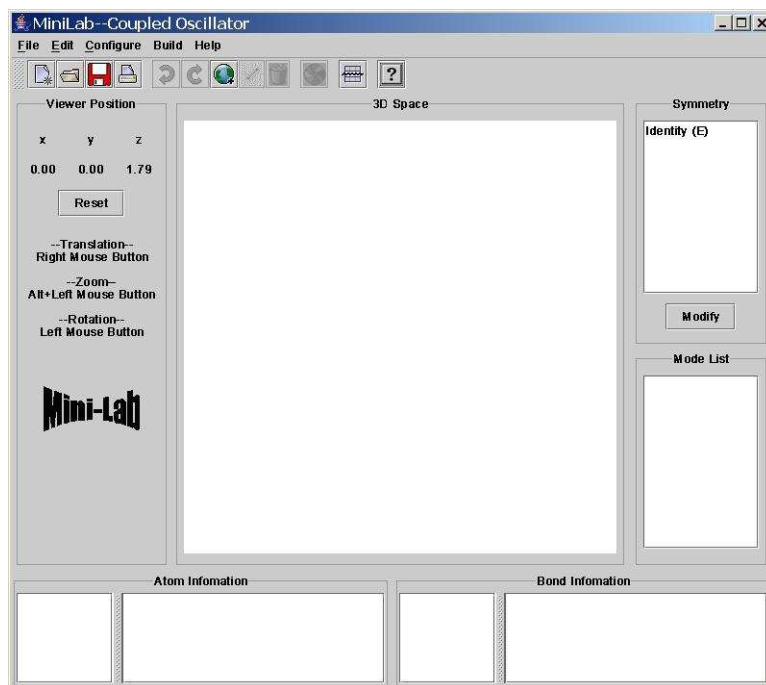


Figure 3.4: Main Layout of Mini-Lab

- **Toolbar** - The Toolbar is under the Main Menu. It is a collection of the most frequently used functions. Toolbar buttons can be separated into several groups including File input/output functions, System configuration functions, appearance modification functions, mode calculation functions and help functions. All these buttons in the toolbar can show the tip text corresponding to the menu item on mouse-over.
- **3D Canvas** - The 3D Canvas is the middle panel where the 3D image resides. When exporting images or movies, the content on this panel is saved. The user can navigate through the 3D space using the mouse. The rotation about the origin can be made via a mouse drag motion with the left mouse key pressed.

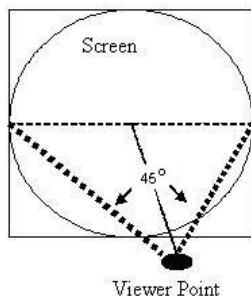


Figure 3.5: Field of View

Translation parallel to the screen can be made via a mouse drag motion with the right mouse button pressed. Zooming perpendicular to the viewer screen can be made via a mouse drag motion with the middle mouse button pressed or use Alt-key+left mouse button. Brief instructions on how to navigate through the 3D space are shown on the left panel of the main layout.

- Viewer information Panel - on the left panel, the viewer's position is shown in real time when the user navigates through the 3D space. Figure 3.5 shows the orientation with respect to the viewer. In this tool, the fixed field of view of  $\frac{\pi}{4.0}$  is used. If the user resets the viewer point by clicking the reset button on the left panel, the view point is moved to  $z=1.79$ . Thus, objects with  $x$  and  $y$  range of  $-1.0$  to  $+1.0$  at  $z=0$  plane can be fully viewed across.
- Atom and Bond List Panel - The atom and bond information is shown on the bottom of the main layout. When atoms or bonds are added, they will be shown both in the 3D canvas and their list panel. The selection of 3D objects in the canvas and in the list are concurrent. The selected object will blink in

the 3D canvas and will be highlighted in the list panel. When the objects are selected, the detailed information of them will be shown in the right panel of the lists.

- Symmetry List Panel - The symmetry list panel is on the right of the main layout. The user can read the symmetry information in the symmetry panel. The user can click the "Modify" button in this panel. A dialog will open for the user to modify the symmetry constraints.
- Mode List Panel - The mode list panel is below the Symmetry List Panel. After calculating the modes, the eigenfrequencies of the modes will be shown in the list in order. The user can pick one non-zero mode for animated visualization. To cancel the animation, she simply presses and holds the Ctrl key and clicks the selected mode again.

### 3.3.2 OTHER INTERACTIVE DIALOGS

To assist the operation, many other dialogs will be shown. Some of them are just yes/no dialog. Some others require more inputs for customization. These Dialogs include:

- Dialogs to add Atom and Bond
- Dialog to change appearance
- Dialogs to change background and light
- Dialog to change the vibrational period in visualization
- Dialogs for broken symmetry
- Dialog for saving when exiting



Figure 3.6: Initialization of Mini-Lab

### 3.3.3 HELP WINDOW

The help window, as shown in Figure 3.3, will open when the help menu item is selected. The help information is displayed in a tree structure. The user can easily find the required information in the help contents.

### 3.4 WALK-THROUGH SAMPLES

In this section, we will go through simple examples to get familiar with the major functions of the tool. Before opening the Mini-Lab jar file, the user must make sure J2SE and Java3D are installed in the system. Details of the installation can be found in the Appendix A.

The opening of Mini-Lab may take a few seconds. A window with the progress bar, as in Figure 3.6, pops up to show the progress of initialization. After initialization, the main layout, as in Figure 3.4, is shown on the screen. When the user clicks the menu item Edit/Add Atom, a dialog will open nearby as in Figure 3.7. The user then sets the atom position to be  $(-0.4, 0, 0)$  and clicks the Add button. The user

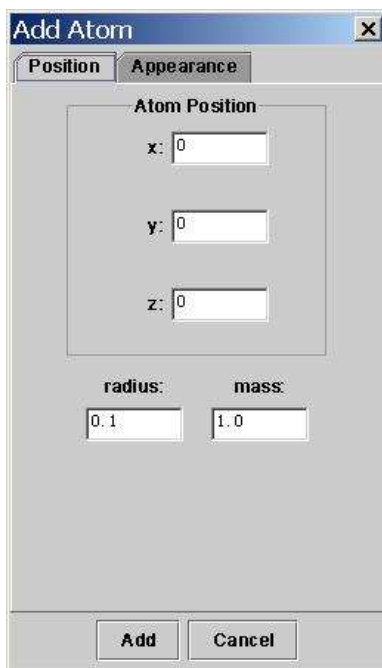


Figure 3.7: Dialog of Adding an Atom

then adds the second atom at  $(0.4, 0, 0)$ . Now two black spheres are shown in the corresponding position in the 3D canvas. In the above procedure, the user has chosen the default mass = 1 and radius = 0.1 and the default black color for each atom. These two atoms are properly shown because the viewing platform views x and y from -1 to 1 at  $z = 0$  by default. For a specific system, the viewer's point can be changed to make a better view, as discussed in section 3.2.4. The user now sets the third atom position to be  $(0, 0, 0)$ . Before he clicks the Add button, he presses the appearance tab in the dialog. The appearance settings as in Figure 3.8 are shown. The user chooses Red for the ambient color and then clicks "Add" button. Now three atoms indicated by spheres are shown on the screen. The list of the atoms is also shown in the Atom Information Panel as in Figure 3.9.

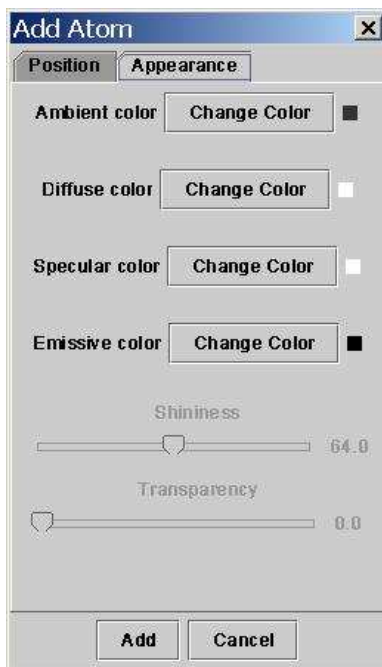


Figure 3.8: Appearance Settings in Configuration

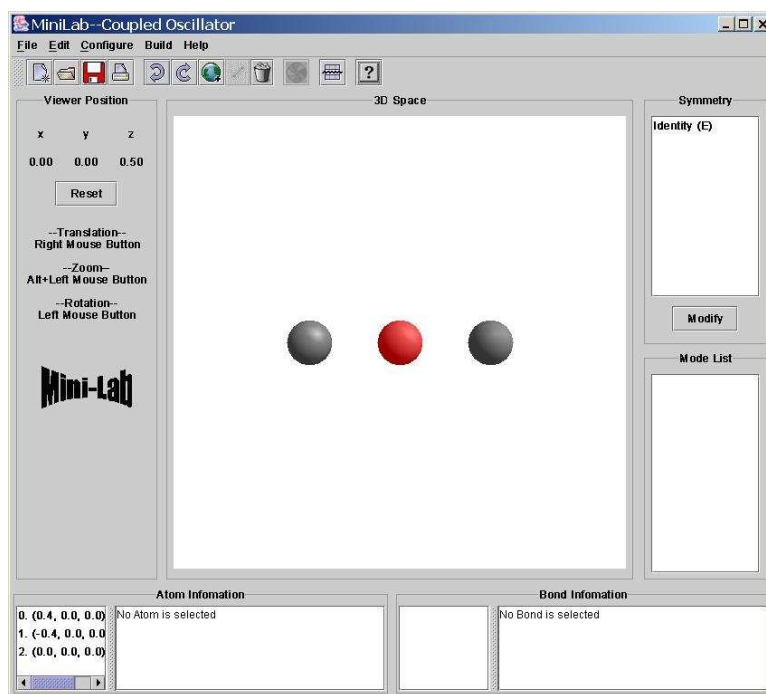


Figure 3.9: An Example of Atom Layout

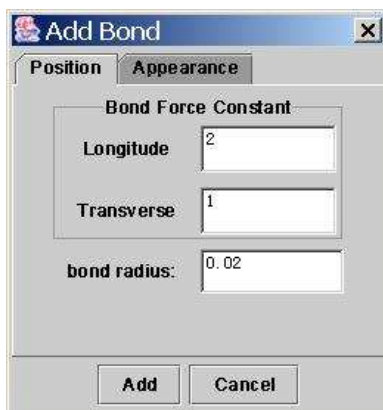


Figure 3.10: Dialog of Adding a Bond

To add bonds among atoms, the user first selects the atoms that are to be bonded. In this example, the user clicks the black atom at position  $(-0.4, 0, 0)$ . He either selects it from the atom list or selects it by picking it in the 3D canvas. He then presses the Ctrl key and clicks the Atom at  $(0, 0, 0)$ , either in the atom list or in the 3D canvas. Now two selected atoms are blinking. He then clicks the menu item Edit/Add Bond. A dialog for the bond parameter is shown as in Figure 3.10. In this example, the user sets the longitude force-constant to be 2 and transverse force-constant to be 1. This means that when the bond is stretched, compressed or sheared, it will give a restoring force to atoms which it interconnects. (Refer to Section 2.1 for details about the force-constant.) Now the user clicks the Add button. Two selected atoms are now attached by a grey cylinder that indicates the bond. The list of bonds is also shown in the Bond Information Panel. To deselect the blinking atoms, the user clicks any vacant space in the 3D canvas. Now the picture is as shown in Figure 3.11. As discussed in Section 3.2.5, the user can provide the symmetry information to make the configuration more quickly. By default, any system will contain the identity

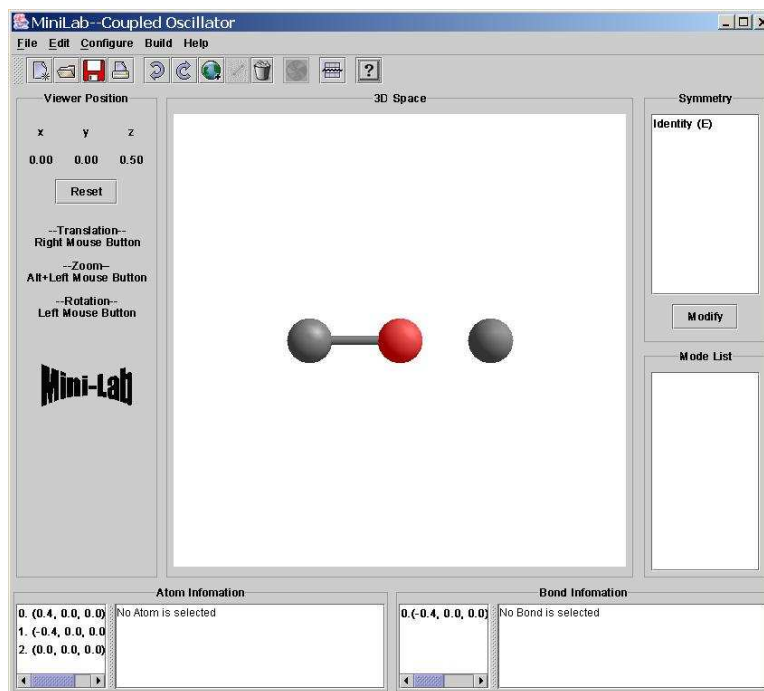


Figure 3.11: An Example of Bond Layout

symmetry, which is  $x \rightarrow x, y \rightarrow y, z \rightarrow z$ . In the following, the user continues the above example, and he is going to add two additional symmetries to the system. He clicks the "Modify" button for the symmetry in the right panel of the main layout. A window as in Figure 3.12 appears. First, he selects the symmetry " $x \rightarrow x$ " on the left panel and clicks the "Add" button to add the reflection  $x \rightarrow -x, y \rightarrow y, z \rightarrow z$ . Because there is no bond between the atom at  $(0,0,0)$  and the atom at  $(0.4,0,0)$ , the system originally doesn't have the symmetry " $x \rightarrow -x$ ". But the program will show a dialog to ask if a symmetry mapping bond between the atom at  $(0,0,0)$  and the atom at  $(0.4,0,0)$  will be added. If the "No" button in this dialog is clicked, the symmetry won't be added to the system. In this example, the user chooses "Yes". Then the mapping bond between the atom at  $(0,0,0)$  and the atom at  $(0.4,0,0)$  is

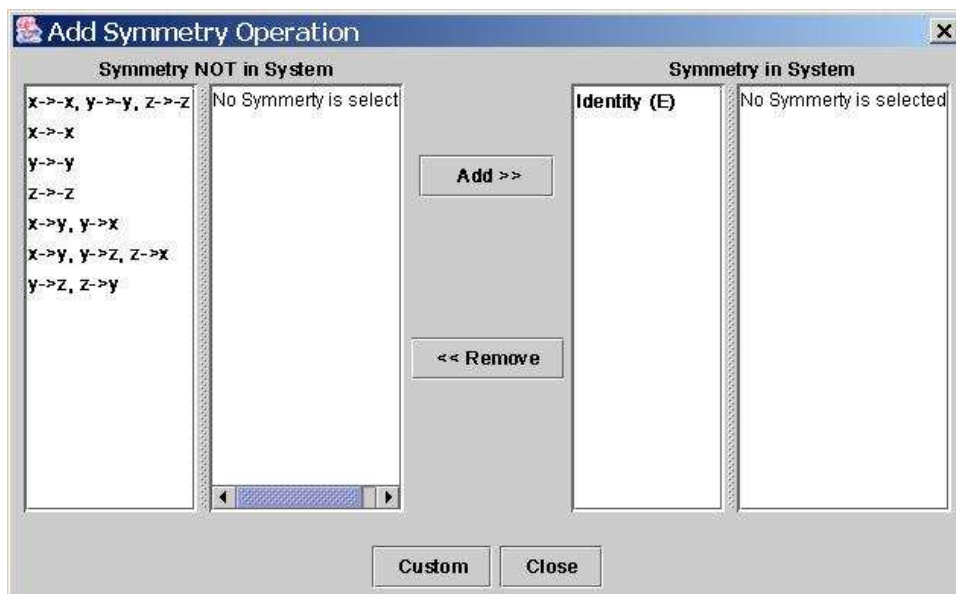


Figure 3.12: Modification of Symmetry Constraints

shown. Next, the user selects another symmetry “ $x \rightarrow y, y \rightarrow x$ ” and adds it to the system. Again, new atoms and new bonds will be added automatically to make this symmetry work. After the above two symmetries are added, the system will appear as Figure 3.13. After the user finishes the configuration, he can save it for later use. To visualize the eigenmodes of this system, the user can click the menu item Build/Calculate to compute the normal modes. After the computation, the normal frequencies will appear on the right panel of the main layout in the modes List. Then any modes with non-zero frequency will be shown in animation if they are selected.

The above example gives a straightforward way to configure the system and visualize the eigenmodes. The user can also modify the system at any time. For example, the user can change the appearance of any selected object.

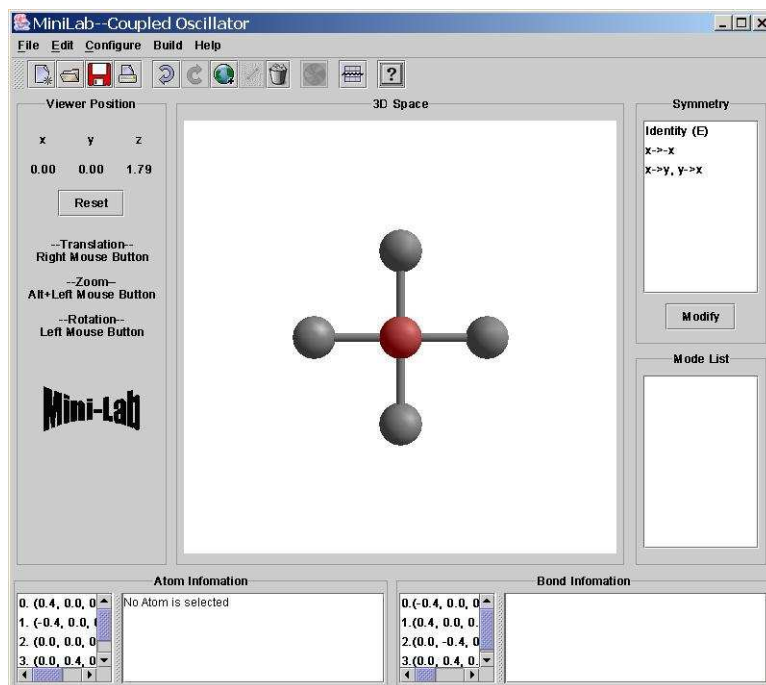


Figure 3.13: An Example of System Layout after Adding Symmetries

The delete function is also offered to the user to remove any unwanted atoms or bonds. For example, suppose the user wants to remove the black atom at  $(0,0,4,0)$ . He either selects it from the atom list or picks it directly from the 3D canvas, and then he clicks the menu Edit/Delete item.

Because removing this atom will break the symmetry, the tool will pop up a dialog as Figure 3.14. At the same time, all the atoms mapped with the atom to be deleted are shown highlighted in the 3D canvas. If the user only wants to delete the atom at  $(0,0,4,0)$ , he should just click the "YES" button in the dialog. Then the symmetry " $x \rightarrow y, y \rightarrow x$ " will be removed from the symmetry list. If the user chooses "NO" in the dialog, the tool will show another dialog to let the user confirm to removal of all mapped atoms. After the user confirms again, all of the symmetries

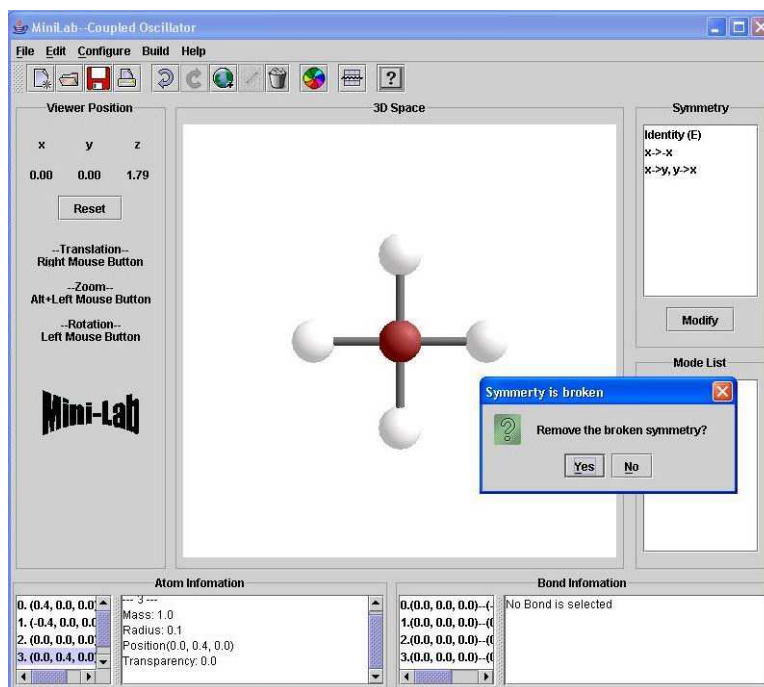


Figure 3.14: Removal of Atom and Bond

are reserved, but all black atoms are removed. A similar situation also happens when the user wants to add atoms or bonds to a system that contains some symmetries other than the identity symmetry. The tool will check whether a symmetry will be broken. If a breaking of the symmetry will be caused by adding an object, the tool will ask the user either to remove the broken symmetry or to keep the symmetry and let the tool add the mapped objects automatically. The automatic adding of the atoms and bonds under the specified symmetry will make the system configuration go much faster. In addition, the tool keeps all records of adding and removing the atoms or bonds. It offers unlimited undo and redo function to the user in case he performs some wrong operations.

The above examples only give an illustration of the operation in the demo tool. A special version for research can read the configuration directly from an input file instead of adding the atoms and bonds one by one. The force-constant matrix in the research version can be made more general than in the demo version.

## CHAPTER 4

### DESIGN AND IMPLEMENTATION

#### 4.1 TOP LEVEL DESIGN

As shown in Figure 4.1, the tool is considered to be divided into the following modules at the top level:

- Graphical User Interface - It offers a friendly user interface such as menu, toolbar, dialogs, etc.
- 3D Visualization - It visualizes the objects in 3D. It responds to 3D operation such as translation, rotation and mouse picking. It gives the animated picture if an eigenmode of the system is selected.
- Image Processing - It renders the screen into images and generates short movies for the animation of the modes.
- Modes Computation - It computes the eigenmodes for the system if the physical configuration is given.
- Symmetry - It records the symmetry constraints in the system. Any modification of the physical system will be sent to this module to check the symmetry status.
- 3D Objects - In this tool, they mainly are the atoms and bonds. They contain both physic informational and visualization information.

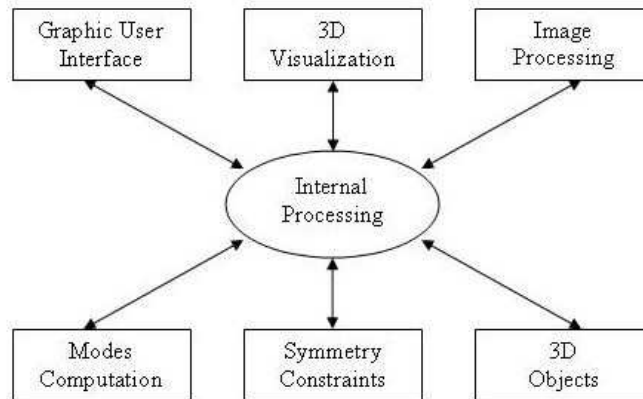


Figure 4.1: Top Level Module in Mini-Lab

- Internal Processing - The internal processing module coordinates all the modules above. The undo and redo functions are implemented within it.

## 4.2 DETAILED DESIGN AND CODING

The detailed design of Mini-Lab is constructed using Java [5], Java3D [1], and Java Advanced Imaging [2]. The code is under version control of CVS [6].

### 4.2.1 GRAPHICAL USER INTERFACE

The graphical user interface is constructed using Java GUI Swing components[9]. The menus and buttons used in Mini-Lab use Java's event-handling mechanism. They implement the appropriate listener interface (e.g., ActionListener). If the listener is triggered by an event, the method which has been overridden in the listener interface (e.g., actionPerformed) will be executed. There are four lists used in the graphical

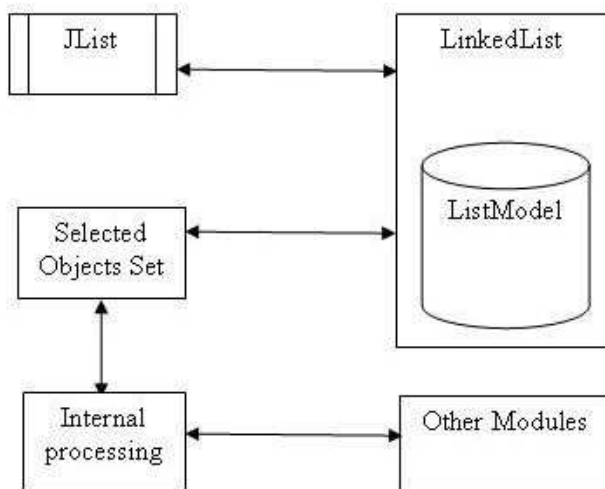


Figure 4.2: GUI for List

user interface. They are the atom list, the bond list, the symmetry list, and the normal modes list. To visualize these lists, a class that overrides `LinkedList` by adding the `ListModel` is created. The architecture for visualizing the list is shown in Figure 4.2. The Swing component `JList` can be added to the panel of the layout. When an object is required to be added to the list, the `ListModel` object will notify the `JList` object. Then the name of the object will be shown in the list panel. If the user selects the object from the panel for further operation, the `JList` object will notify the `ListModel` with the list index for the selected objects. Then the corresponding object in the `LinkedList` object will be added to the set of selected objects and further processing can be done for the selected objects.

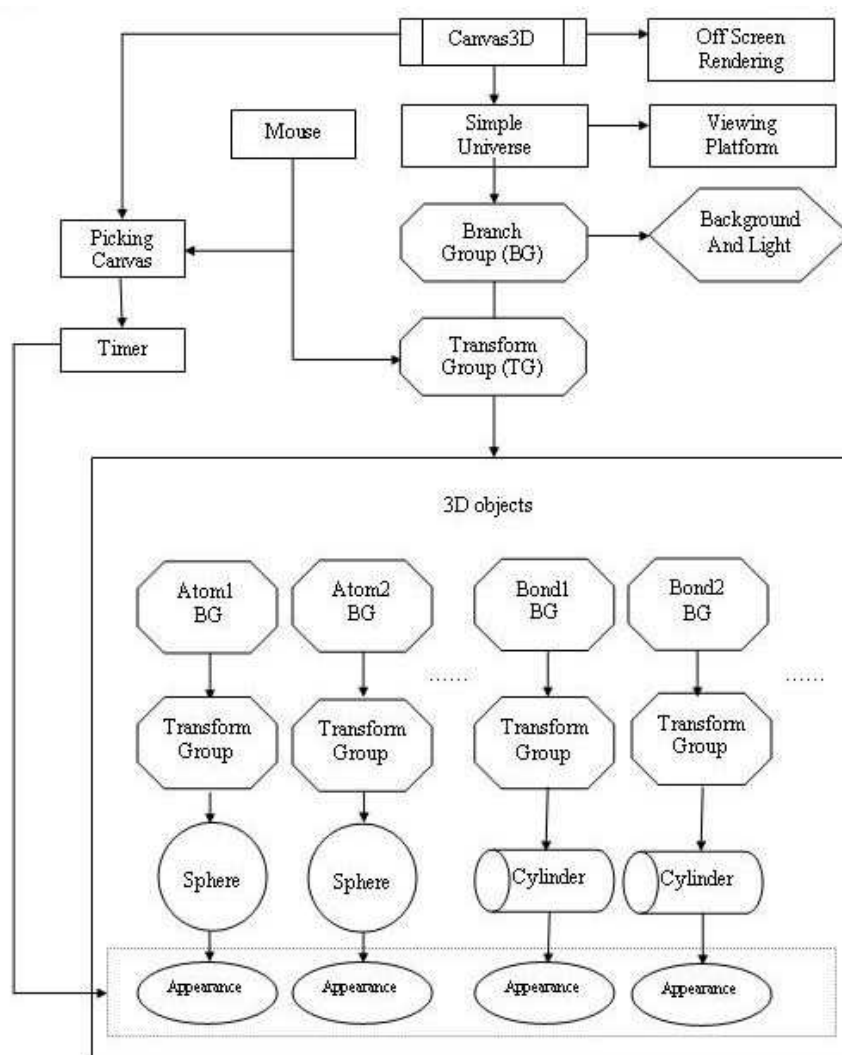


Figure 4.3: 3D Visualization for the Configuration of a Molecule

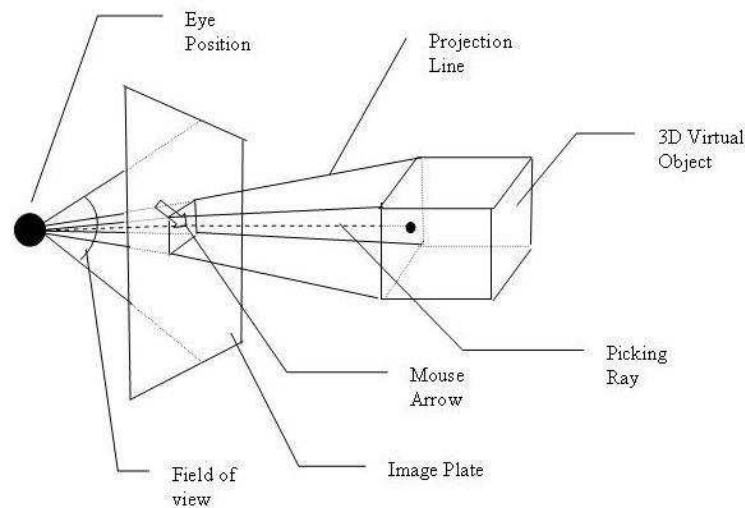


Figure 4.4: Projection and Picking of an Object

#### 4.2.2 3D VISUALIZATION

The 3D visualization is realized by using the Java3D API. The main blocks in this module are shown in Figure 4.3. The Canvas3D object provides the image plate where the visual 3D objects are rendered. It is embedded in the middle panel of the main layout. The scene graph is represented internally as a tree structure, as discussed in Section 2.2. The off-screen rendering function can generate a 2D image which is the same as the one on the screen. The image is put in an image buffer for further processing. Mini-Lab uses SimpleUniverse class in Java3D to create all the necessary objects to create a complete view branch graph for an empty scene. The projection from the virtual 3D objects to the image plate is made as in Figure 4.4. In the current version of Mini-Lab, the field of view is fixed to be  $\frac{\pi}{4}$ . By default or after reset, the objects will be shown on the image plate if its projection is between

-1.0 and +1.0 for x and y. The distance between the image plate and the view point is also fixed. In Mini-Lab, changing of the view is realized by using the mouse event to modify a TransformGroup that links to all virtual 3D objects. The picking is realized with the aid of the PickCanvas class in Java3D. A ray is projected into the virtual world from the position of the mouse pointer parallel with the projection, as shown in Figure 4.4. Intersection of this ray with the objects of the virtual world is computed. The intersected visual object closest to the image plate is set to be selected. The picked objects are put in a set upon which further operations can be performed. If the set for selection is not empty, a timer is started to change the appearance of the selected objects periodically. Thus, the selected objects will blink.

The data for animation is built inside the atom and bond objects. The atoms and bonds share a common Alpha object. When given a normal mode, the period of the vibration is divided into eight segments. At each time node, the atom positions are calculated with the max vibration amplitude no more than  $\frac{1}{10}$  of the length of the shortest bond. The bond's position, length and orientation at different times are decided by the interconnected atoms' positions at that time. The new values at each time node are given to the Interpolators which are a PositionPathInterpolator object and a RotPosScalePathInterpolator object used in Mini-Lab. The interpolator adjusts the Transformgroup object, which realizes the motion of the atom and bond, as shown in Figure 4.5.

### 4.2.3 ATOM AND BOND

The atom is visualized by a Sphere shape. The Sphere object is given the mass and the position to specify an atom. The visualization of an atom also requires information including the radius and appearance. Dialogs are built to ask the user to provide these inputs. A Transformgroup object is initialized by the position information. The information of the set of bonds that connect to this atom is maintained. When an

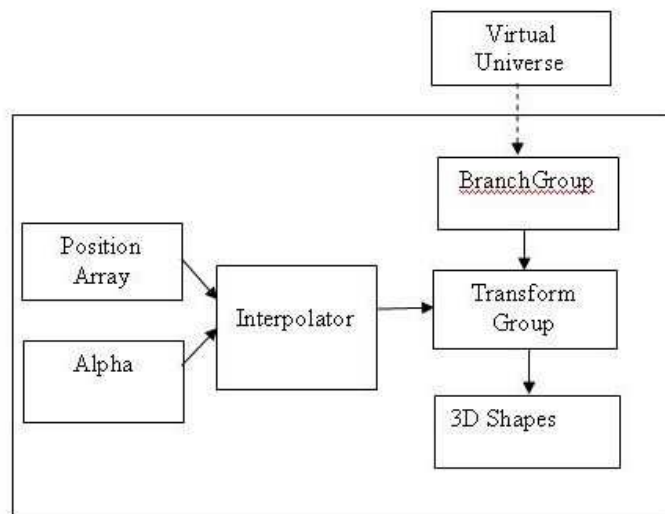


Figure 4.5: 3D Animation

atom is removed, the bonds that connect to this atom is also removed. Each atom also maintains a set of positions that is used for animations, as discussed in Section 4.2.2.

The bond is visualized by a cylinder shape. It is required to give the index of the atom pairs that this bond interconnects. The position and axis of the bond is calculated by the atoms pairs. Mini-Lab, allows the user to add bonds only when more than two atoms are selected. Among the selected atoms, the nearest neighbors are connected with bonds.

#### 4.2.4 IMAGE PROCESSING

The image is prepared in a buffer by rendering a copy of the Canvas3D object that is on screen. The generation of a single image file is done by using the static method

in ImageIO. To generate a movie for the animation, a series of temporary images are produced for the snapshot of vibration at different time point. Then, the sample codes offered on Sun's web site [2] using Java Advanced Imaging are embedded in the tool to batch the images and generate a QuickTime movie.

#### 4.2.5 SYMMETRY OPERATION

A symmetry list is maintained in the program to keep all the symmetries the user specifies for the system. The symmetry operation is realized by a  $3 \times 3$  transformation matrix that transforms all the atom positions. The program has defined some basic symmetries such as inversion, reflection and rotation. The user can also define his own. By default, the program automatically assigns the identity symmetry (i.e.  $3 \times 3$  identity matrix) to the symmetry list. The visualization of the symmetry list is discussed in Section 4.2.1.

When the user adds atoms or bonds, the symmetry operations in the symmetry list are checked one by one to see if any more atoms or bonds should be added to maintain the symmetry. If so, the program will ask the user either to add the new objects or to remove the broken symmetry. Note that adding new atoms and bonds could cause more unmapped atoms or bonds. Therefore, this procedure is recursive, and the newly added atoms or bonds are temporarily put in a set. If the newly added atoms or bonds in the set produce closure under all symmetries in the system, all of them will be added to the the system. If the user cancels in the middle of the procedure, no atoms or bonds will be added.

Similar to adding atoms or bonds, when the user removes atoms or bonds, the symmetry operations in the symmetry list are also checked one by one to see if any symmetries are broken. If so, the program will find the unmapped atoms or bonds and ask whether they should also be deleted. Note that deleting the mapped object could cause a third object to be unmapped by some symmetry. In this case, a set of

all deleted objects are also maintained. If all the atoms and bonds in the set that are to be deleted form closure under all symmetries in the system, all of them will be deleted finally. If the user cancels in the middle of the procedure, the atoms or bonds in the set will be put back to the canvas.

When the user adds a new symmetry to system, the program also checks if new atoms and bonds should be add to make this symmetry work. This function makes the configuration of the system much faster. There are some constraints for the  $3 \times 3$  transformation matrix. The symmetry operation should be a unitary matrix. Also, some symmetry operations cannot be put together. For example, a three fold rotation and a four fold rotation about the z axis conflict. The current version does not check for these possibilities. Future work is discussed in Section 5.2. In the current version, if an improper symmetry is added to the system, the program will continue asking to add new mapped atoms and bonds. If the user finds that additions do not stop, he can cancel additions at any time in the middle of the procedure and then the improper symmetry will not be added to the system.

#### 4.2.6 EQUATION SOLVING

The generalized eigenvalue problem, i.e. Equation 2.5, can be solved by functions in Lapack [7], a well-known linear algebra package written in Fortran language. Mini-Lab uses jlapack, a java version of Lapack package that is generated by a Fortran-to-Java Compiler [8].

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

#### 5.1 CONCLUSION

This project employs the knowledge of coupled-oscillator vibration and Java 3D API to develop Mini-Lab, a research and education tool, for interactive and collaborative visualization of numerical data in analyzing the normal modes of vibration. The present demo version combines many features for easy configuration of the physical system in 3D. The tool can generate the animated vibration pattern for the normal modes of vibration. The user can adjust the color, light and viewer position to produce a suitable visualization in the 3D space. The configuration can be stored; the images can be exported; and the vibration pattern can be saved to movies. Moreover, the physical system can be constructed automatically by the symmetry operation. Although the demo version still requires further development to meet the requirements from the specific user, it has been used in the author's research on Titanium Carbide nanocrystals.

#### 5.2 FUTURE WORK

In the demo version, the symmetry operation is mainly used to speed up the configuration of the system. Although the program offers a custom input for the symmetry by giving a  $3 \times 3$  matrix, the program doesn't check if the new symmetry the user specifies is compatible with the existing symmetry. Also, not all  $3 \times 3$  matrices are symmetry operators. In the next version of Mini-Lab, a fully group-theory-based

symmetry module will be introduced in the program. The program will assign the molecule's symmetry group instead of giving its individual symmetries. The program will also link the normal modes of the system to the corresponding irreducible representation of the group. A more challenging work is to let the program determine the symmetry of the molecule and assign a corresponding symmetry group to it.

## BIBLIOGRAPHY

- [1] <http://java.sun.com/products/java-media/3D/>.
- [2] <http://java.sun.com/products/java-media/jai/>.
- [3] <http://java.sun.com/developer/onlineTraining/java3d/>.
- [4] <http://java.sun.com/j2se/>.
- [5] <http://java.sun.com/>.
- [6] <http://www.cvshome.org/>.
- [7] <http://www.netlib.org/lapack/>.
- [8] <http://www.cs.utk.edu/f2j/>.
- [9] H. M. Deitel and P. J. Deitel. *Java How to Program*. 1999.
- [10] H. M. Deitel, P. J. Deitel, and S. E. Santry. *Advanced Java 2 Platform; how to Program*. 2001.
- [11] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics; principles and practice*. 1990.
- [12] Herbert Goldstein. *Classical Mechanics*. 1980.
- [13] E. A Maxwell. *General homogeneous coordinates in space of three dimensions*. 1959.

- [14] Michael Tinkham. *Group Theory and Quantum Mechanics*. 1964.
- [15] E. Bright Wilson Jr., J. C. Decius, and Paul C. Cross. *Molecular Vibrations; the Theory of Infrared and Raman Vibrational Spectra*. 1955.

## APPENDIX A

### INSTALLATION

#### A.1 SYSTEM REQUIREMENT

In order to use Mini-Lab, the computer must meet the following minimum requirements:

- 256 MB of memory
- 100 MB of available disk space
- A Pentium 3 or equivalent processor (G3 on Mac)
- A 3D graphics accelerator (strongly recommended)
- One of the following supporting operating systems:
  - Windows XP/2000/98/Me
  - Mac OS X version 10.3(panther)
  - Linux(recent versions)
  - Solaris(recent versions)
- Install Java 2 Platform, Standard Edition (J2SE) Runtime Environment(JRE) and Java3D. They can be downloaded from Sun's web site: <http://java.sun.com>

## A.2 TEST RUN

The package can now be downloaded at

<http://www.physast.uga.edu/~qinzhang/project/MiniLab/index.htm>.

In the installation directory, the user can run command `java -jar MiniLab.jar` or she can just double click the jar file in Windows. The user may go through the samples in Section 3.4. If it goes well, the installation is successful. If the user has any problems that the online help cannot solve, please contact the author: [qinzhang@physast.uga.edu](mailto:qinzhang@physast.uga.edu).

## APPENDIX B

### USER MENU

- File menu
  - New

Create a new, empty 3D canvas on the 3D space window. If the old one is not saved, the user will be prompted to do so.
  - Open

Open an existing configuration.
  - Save

If the system already has a name, save the configuration to a file with this name. Otherwise, just operate as the menu item Save As.
  - Save As

Save the configuration to a file with specified name.
  - Save Screen

Save the 3D space on the screen to a jpeg file.
  - Save Movie

If the 3D is in the animated mode, it will save a serials of jpeg files and make a \*.mov QuickTime file.
  - Print

Print the 3D space on the screen to a printer.

- Exit

Exit the Mini-Lab.

- Edit menu

- Undo

Undo the last operation.

- Redo

Redo the last operation.

- Add Atom

Add the atom to the 3D space. The Add Atom Dialog will open.

- Add Bond

Add the bond between selected atoms. The Add Bond Dialog will open.

- Add Bond to All Atoms

Add the bond between the nearest atoms.

- Remove

Remove selected bonds or atoms.

- Change Appearance

Change the color of the selected bond or atoms.

- Configure menu

- Background

Set the background color of the 3D canvas.

- light

Set the light color of the 3D canvas.

- Show Coordinate Direction

Show the direction of the coordinates.

- Vibration Period

Show the dialog to modify the vibration period. The Vibration Period Dialog will open.

- Build menu

- Calculate Modes

Calculate the normal modes of the vibration.

- Help menu

- Contents

Display the help document.

- About

Display information about Mini-Lab.