Data Structures and Algorithms

COMP 103 2019-20 Semester 2

Lecture 08b

Dr. Kerese Manueli

kerese.manueli@ecs.vuw.ac.nz

Victoria University of Wellington

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

Core:

Complete the `MoleculeRender` class so that it

- Reads the data from the element-info file into a Map of `Elements`
- Reads the data from a molecule file into a List of `Atom`s.
- Renders the molecule from the front

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
/**
 * Reads the file "element-info.txt" which contains information about each type of element:
 *   element name, atomic number, atomic weight, a radius, and red, green, blue, components of the color (integers)
 * Stores the info in a Map in the elements field.
 * The element name is the key.
 */
private void readElements() {
    /*# YOUR CODE HERE */

}

/**
 * Asks the user to choose a file that contains info about a molecule,
 * loads the information, resets the view angle to 0,  and
 * renders the molecule on the graphics pane.
 * Should call other methods to do the main steps
 */
public void loadAndRenderMolecule(){
    /*# YOUR CODE HERE */

}
```

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

Element name, atomic number, atomic weight, a radius, (red, green & blue values for the colour)

Example:

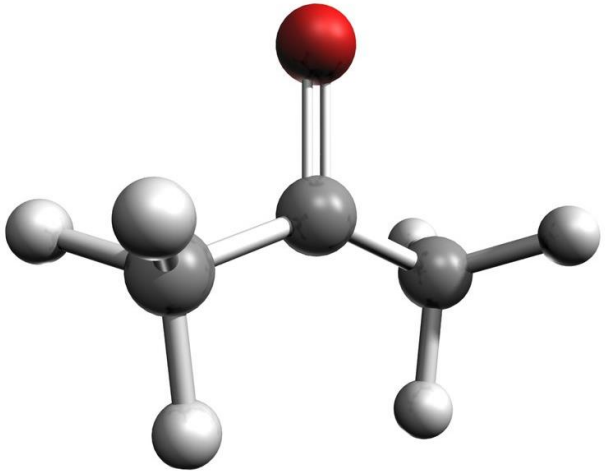
H	1	1.008	14	250	250	0
C	6	12.011	30	100	100	100
O	8	15.999	26	0	0	180

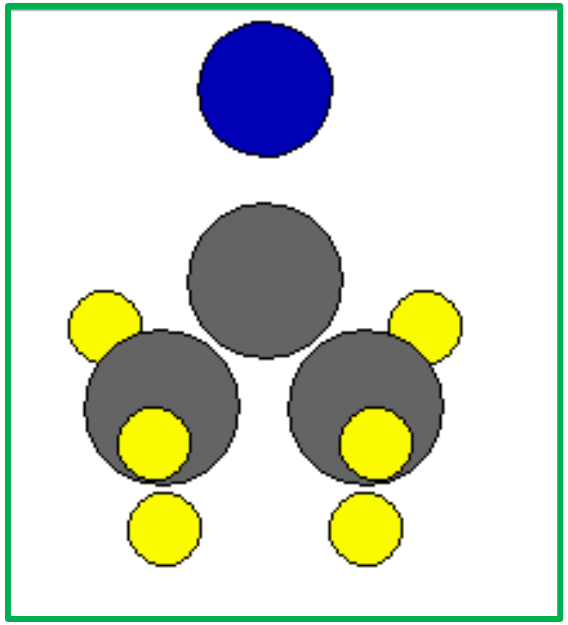
element-info.txt - Notepad

File Edit Format View Help

H	1	1.008	14	250	250	0
F	9	18.998	25	60	0	140
O	8	15.999	26	0	0	180
N	7	14.007	27	0	180	0
C	6	12.011	30	100	100	100
B	5	10.811	34	40	0	160
Be	4	9.012	35	20	0	180
Cl	17	35.453	39	200	0	0
I	53	126.904	80	80	0	80
S	16	32.065	41	180	0	20
P	15	30.974	43	160	0	40
Br	35	79.904	44	0	60	140
Co	27	58.933	45	20	180	0
Ni	28	58.693	45	0	200	0
Cr	24	51.996	46	80	120	0
Cu	29	63.546	46	0	180	20
Fe	26	55.845	46	40	160	0
Mn	25	54.938	46	60	140	0
Se	34	78.960	46	0	80	120
Si	14	28.086	46	140	0	60
As	33	74.922	47	0	100	100
Ge	32	72.640	48	0	120	80
V	23	50.942	48	100	100	0
Al	13	26.982	49	120	0	80
Ga	31	69.723	49	0	140	60
Ru	44	101.070	49	0	230	0
Zn	30	65.390	49	0	160	40
Ti	22	47.867	51	120	80	0
Mg	12	24.305	53	100	0	100
Sc	21	44.956	56	140	60	0
Na	11	22.990	61	80	0	120
Ca	20	40.078	68	160	40	0
Ba	56	137.327	77	0	20	180
K	19	39.098	39	180	20	0
Rb	37	85.468	84	0	40	160

Acetone molecule: C_3H_6O





Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
69  /**
70   * Reads the file "element-info.txt" which contains information about each type of element:
71   *   element name, atomic number, atomic weight, a radius, and red, green, blue, components of the color (integers)
72   * Stores the info in a Map in the elements field.
73   * The element name is the key.
74   */
75  private void readElements() {
76      /*# YOUR CODE HERE */
77
78  }
```

Assignment 2b - Core

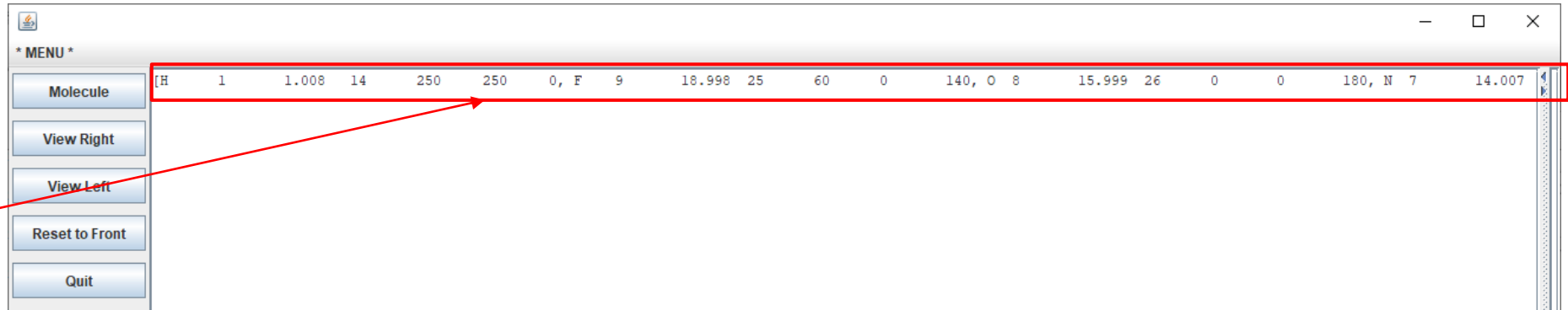
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
69  /**
70   * Reads the file "element-info.txt" which contains information about each type of element:
71   *   element name, atomic number, atomic weight, a radius, and red, green, blue, components of the color (integers)
72   * Stores the info in a Map in the elements field.
73   * The element name is the key.
74   */
75  private void readElements() {
76      /*# YOUR CODE HERE */
77
78  }
```

Code line 77 has sample Java code to read the contents of the element-info.txt file and store it into the allLines ArrayList

```
75      private void readElements() {
76          /*# YOUR CODE HERE */
77          ArrayList<String> allLines  = readAllLines("element-info.txt");
78          UI.println(allLines);
```

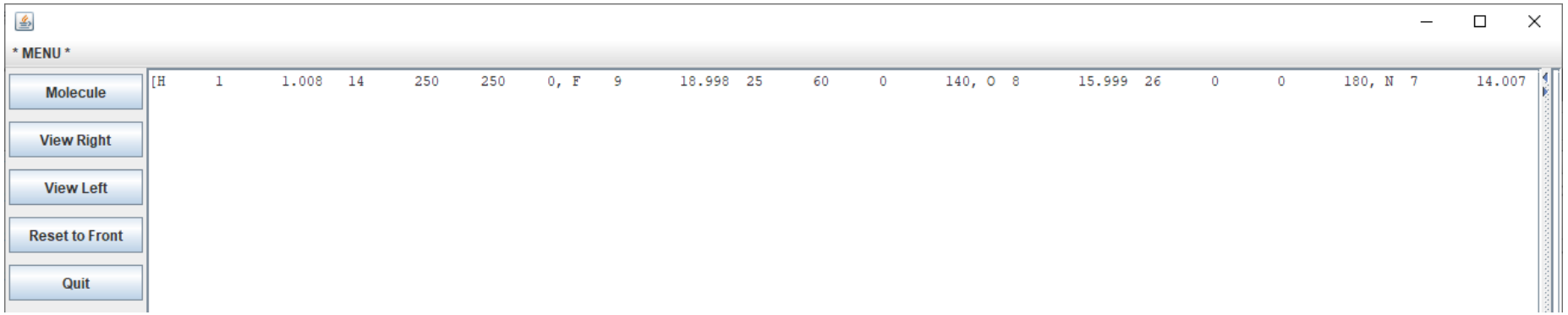
Compiling and running the program, produces the shown output.
Note that the contents of the element-info.txt file is now stored in the allLines ArrayList
All printed on one line.



Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
69  /**
70   * Reads the file "element-info.txt" which contains information about each type of element:
71   *   element name, atomic number, atomic weight, a radius, and red, green, blue, components of the color (integers)
72   * Stores the info in a Map in the elements field.
73   * The element name is the key.
74   */
75  private void readElements() {
76      /*# YOUR CODE HERE */
77
78  }
```



Element name, atomic number, atomic weight, a radius, (red, green & blue values for the colour)

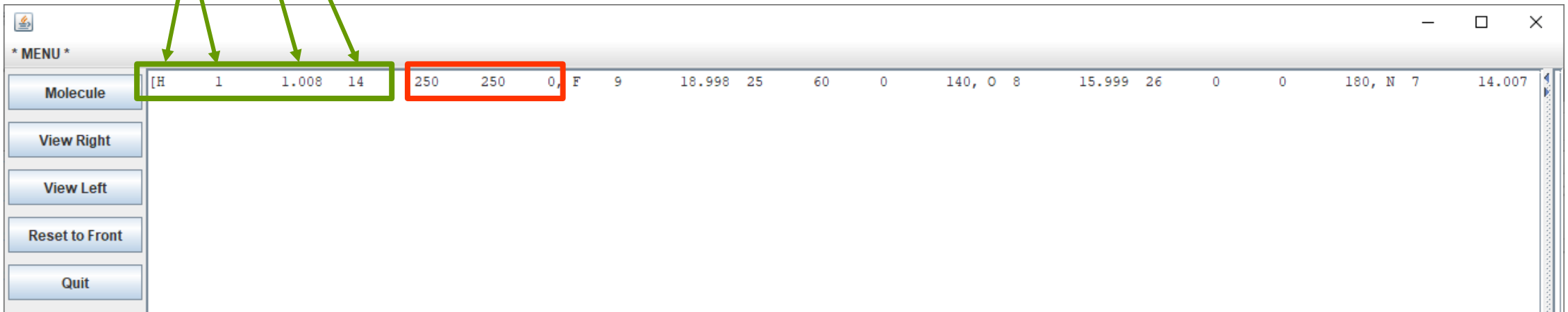
Example:

H	1	1.008	14	250	250	0
C	6	12.011	30	100	100	100
O	8	15.999	26	0	0	180

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

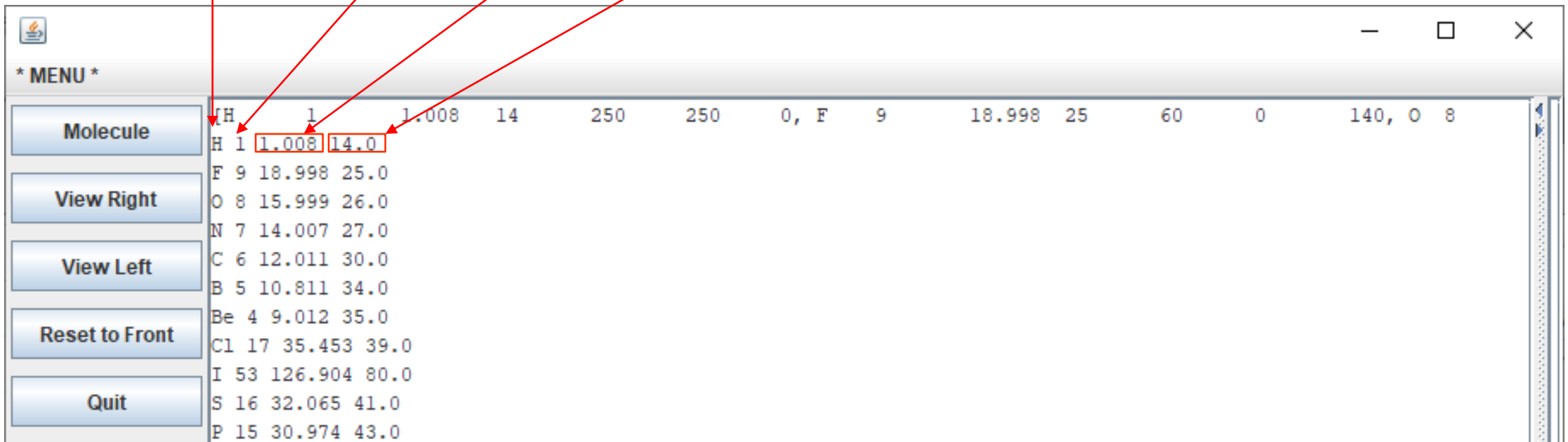
```
75  private void readElements() {
76      /*# YOUR CODE HERE */
77      ArrayList<String> allLines  = readAllLines("element-info.txt");
78      UI.println(allLines);
79
80      for (String line : allLines) {
81          Scanner scan = new Scanner(line);
82          String elementName = scan.next();
83          int number = scan.nextInt();
84          double weight = scan.nextDouble();
85          double radius = scan.nextDouble();
86          UI.println(elementName+" "+number+" "+weight+" "+radius);
```



Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
75  private void readElements() {
76      /*# YOUR CODE HERE */
77      ArrayList<String> allLines  = readAllLines("element-info.txt");
78      UI.println(allLines);
79
80      for (String line : allLines) {
81          Scanner scan = new Scanner(line);
82          String elementName = scan.next();
83          int number = scan.nextInt();
84          double weight = scan.nextDouble();
85          double radius = scan.nextDouble();
86          UI.println(elementName+" "+number+" "+weight+" "+radius);
```



Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
27      // Map containing info about each type of element.
28      private Map<String, Element> elements;
29
```

```
75  private void readElements() {
76      /*# YOUR CODE HERE */
77      ArrayList<String> allLines  = readAllLines("element-info.txt");
78      UI.println(allLines);
79
80      for (String line : allLines) {
81          Scanner scan = new Scanner(line);
82          String elementName = scan.next();
83          int number = scan.nextInt();
84          double weight = scan.nextDouble();
85          double radius = scan.nextDouble();
86          UI.println(elementName+" "+number+" "+weight+" "+radius);
```

```
elements = new HashMap<String, Element>();
```

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
69  /**
70   * Reads the file "element-info.txt" which contains information about each type of element:
71   *   element name, atomic number, atomic weight, a radius, and red, green, blue, components of the color (integers)
72   * Stores the info in a Map in the elements field.
73   * The element name is the key.
74   */
75  private void readElements() {
76      /*# YOUR CODE HERE */
77
78  }
79
80  /**
81   * Asks the user to choose a file that contains info about a molecule,
82   * loads the information, resets the view angle to 0,  and
83   * renders the molecule on the graphics pane.
84   * Should call other methods to do the main steps
85   */
86  public void loadAndRenderMolecule(){
87      /*# YOUR CODE HERE */
88
89  }
```

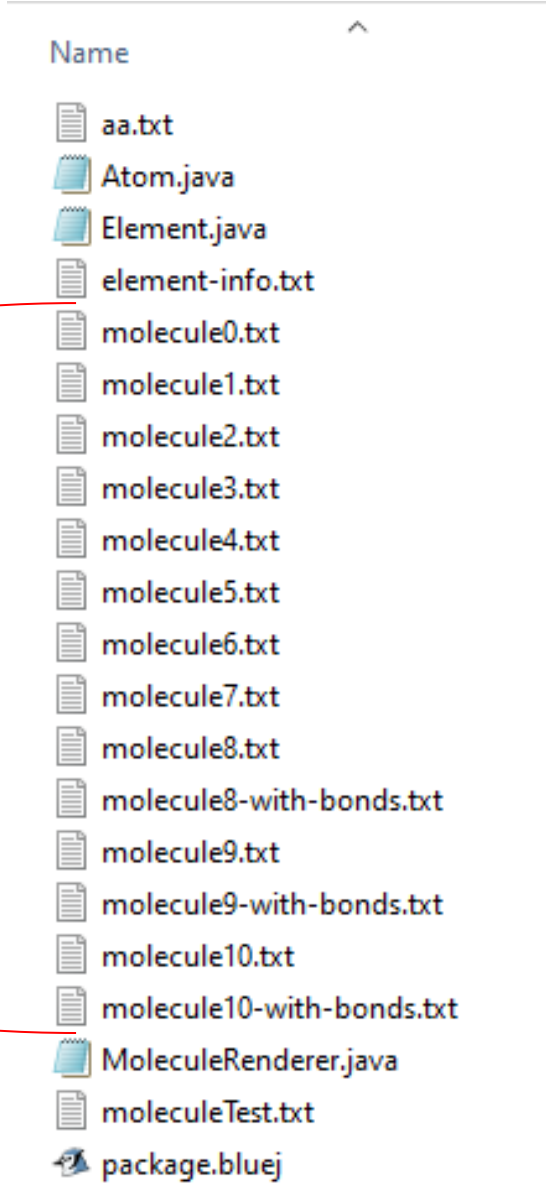
Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
/**
 * Asks the user to choose a file that contains info about a molecule,
 * loads the information, resets the view angle to 0,  and
 * renders the molecule on the graphics pane.
 * Should call other methods to do the main steps
 */
public void loadAndRenderMolecule(){
    /*# YOUR CODE HERE */
    String fn = (UIFileChooser.open());
    ArrayList<String> allLines  = readAllLines(fn);
    UI.println(allLines);

}
```

Ask user to choose a file containing info about a molecule from the list of files shown



Assignment 2b - Core

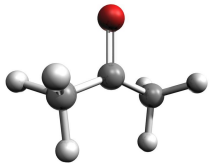
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

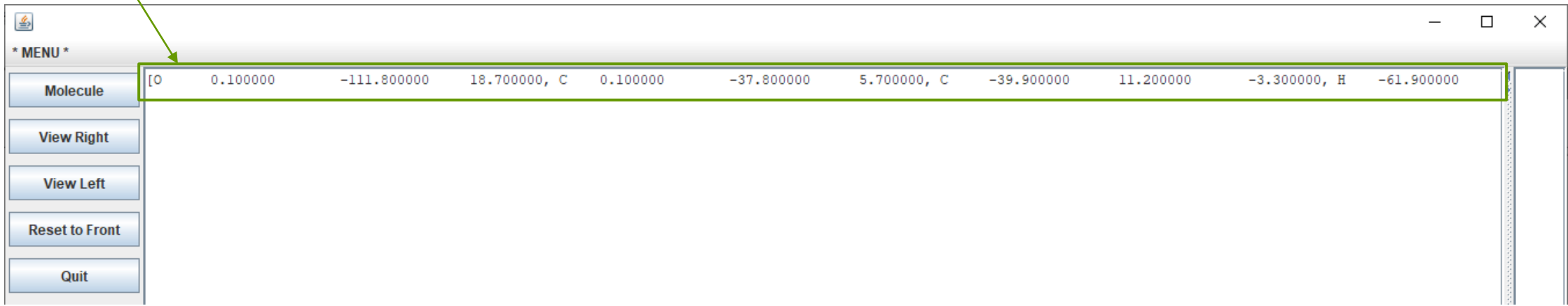
```
/**
 * Asks the user to choose a file that contains info about a molecule,
 * loads the information, resets the view angle to 0,  and
 * renders the molecule on the graphics pane.
 * Should call other methods to do the main steps
 */
public void loadAndRenderMolecule(){
    /*# YOUR CODE HERE */
    String fn = (UIFileChooser.open());
    ArrayList<String> allLines  = readAllLines(fn);
    UI.println(allLines);

}
```

Use the readAllLines method (utility method given at the end of the original moleculeRenderer.java file) to open the chosen file fn and store contents of the file into an ArrayList named allLines

Acetone molecule
C_3H_6O





Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
/**
 * Asks the user to choose a file that contains info about a molecule,
 * loads the information, resets the view angle to 0,  and
 * renders the molecule on the graphics pane.
 * Should call other methods to do the main steps
 */
public void loadAndRenderMolecule(){
    /*# YOUR CODE HERE */
    String fn = (UIFileChooser.open());
    ArrayList<String> allLines  = readAllLines(fn);
    UI.println(allLines);

}
```

Reset the view angle to 0

```
33      private double horizViewAngle = 0; // The current horizontal viewing angle ("Yaw")
34      // reset it to 0 when loading a new molecule!
```

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
/**
 * Asks the user to choose a file that contains info about a molecule,
 * loads the information, resets the view angle to 0,  and
 * renders the molecule on the graphics pane.
 * Should call other methods to do the main steps
 */
public void loadAndRenderMolecule(){
    /*# YOUR CODE HERE */
    String fn = (UIFileChooser.open());
    ArrayList<String> allLines  = readAllLines(fn);
    UI.println(allLines);
    this.horizViewAngle = 0; // reset to 0 before drawing
    this.vertViewAngle = 0;
```

Reset the view angle to 0

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
for (String line : allLines) {
    Scanner scan = new Scanner(line);
    String atomType = scan.next();
    UI.println(atomType);
    Element element = elements.get(atomType);

    double x = scan.nextDouble();
    double y = scan.nextDouble();
    double z = scan.nextDouble();

    Atom atom = new Atom(x, y, z, element);
    molecule.add(atom);
}
showFromFront();
```

Type the given Java code after the reset code lines for the horizViewAngle and the vertViewAngle as shown in the previous slide.

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
for (String line : allLines) {
    Scanner scan = new Scanner(line);
    String atomType = scan.next();
    UI.println(atomType);
    Element element = elements.get(atomType);

    double x = scan.nextDouble();
    double y = scan.nextDouble();
    double z = scan.nextDouble();

    Atom atom = new Atom(x, y, z, element);
    molecule.add(atom);
}
showFromFront();
```

```
/**
 * Renders the molecule from the front.
 * Sorts the Atoms in the List by their z value, back to front
 * Uses the default ordering of the Atoms
 * Then renders each atom
 */
public void showFromFront() {
    /*# YOUR CODE HERE */

    UI.clearGraphics();
    for(Atom atom : molecule) {
        atom.render(MIDX, MIDY, 0, 0);
    }
}
```

You need to add the Java code to ensure that the showFromFront method renders the molecule from the front

Assignment 2b - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

Only 1 line of Java code is required in the showFromFront method.

Hint

```
/**
 * Renders the molecule from the front.
 * Sorts the Atoms in the List by their z value, back to front
 * Uses the default ordering of the Atoms
 * Then renders each atom
 */
public void showFromFront() {
    /*# YOUR CODE HERE */

    UI.clearGraphics();
    for(Atom atom : molecule) {
        atom.render(MIDX, MIDY, 0, 0);
    }
}
```

Outcome for the molecule0.txt file