Data Structures and Algorithms

COMP 103 2019-20 Semester 2

Lecture 08a

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

VICTORIA UNIVERSITY OF WELLINGTON TE HERENGA WAKA

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

- XMUT103 home
- Course Outline
- Lecture Schedule
- Weekly Timetable
- Assignments
- Submission
- Your Marks
- People
- Java Resources
- Java documentation

School of Engineering and Computer Science ▸ Courses/XMUT103_2020T1 ▸ Assignment2PartB

Introduction to Data Structures and Algorithms
Assignment 2 Part B: Using Collections

- Due 6 May , 7pm

Zip file

Resources and links

- Download zip file containing the necessary code and data.
- Java Documentation
- Submit your answers
- Marks and feedback
- Part A of the assignment

What To Hand In

- Part A
  - `WellingtonTrains.java`
- Part B
  - `MoleculeRenderer.java`
  - `Atom.java`
  - `Element.java`

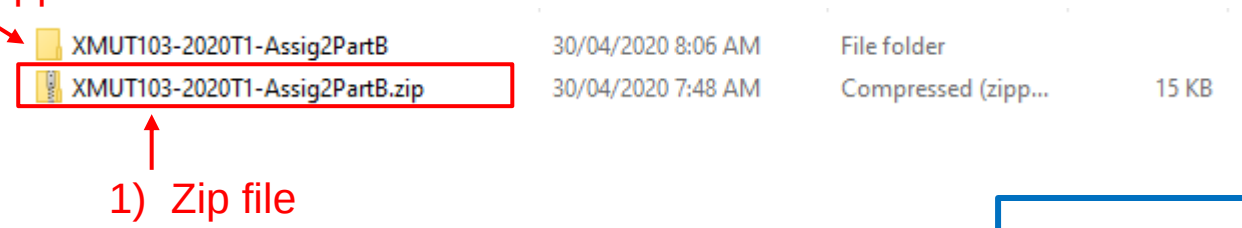
Do not rename these files.

Remember to submit all of these files. When you have submitted them, check that you can read the files listed on the submission page, and complete the submission process.

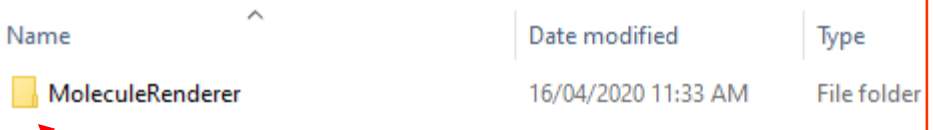
Assignment 2b - Core

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

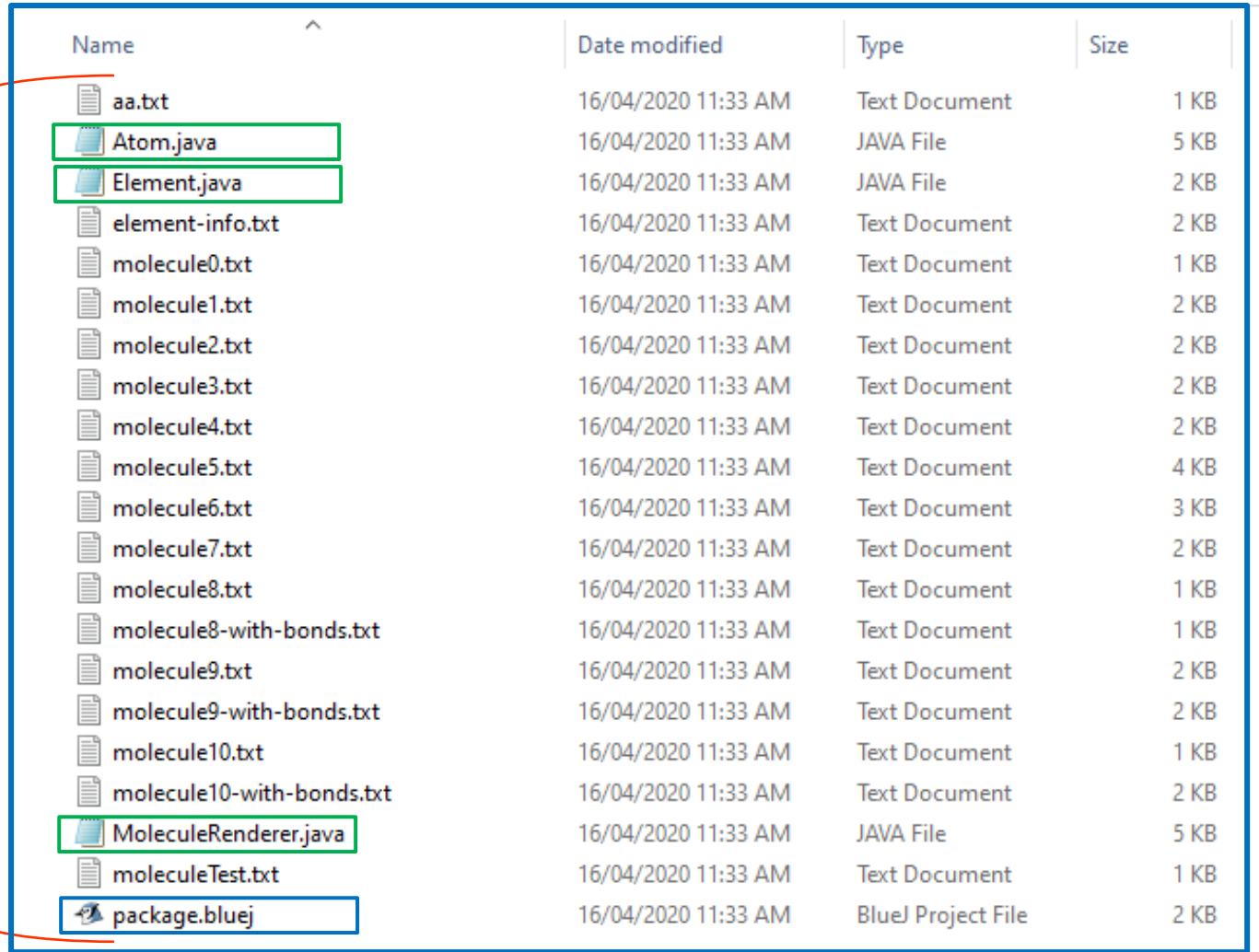
2) Unzipped folder



1) Zip file



3) Files inside the MoleculeRenderer folder



Name	Date modified	Type	Size
aa.txt	16/04/2020 11:33 AM	Text Document	1 KB
Atom.java	16/04/2020 11:33 AM	JAVA File	5 KB
Element.java	16/04/2020 11:33 AM	JAVA File	2 KB
element-info.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule0.txt	16/04/2020 11:33 AM	Text Document	1 KB
molecule1.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule2.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule3.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule4.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule5.txt	16/04/2020 11:33 AM	Text Document	4 KB
molecule6.txt	16/04/2020 11:33 AM	Text Document	3 KB
molecule7.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule8.txt	16/04/2020 11:33 AM	Text Document	1 KB
molecule8-with-bonds.txt	16/04/2020 11:33 AM	Text Document	1 KB
molecule9.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule9-with-bonds.txt	16/04/2020 11:33 AM	Text Document	2 KB
molecule10.txt	16/04/2020 11:33 AM	Text Document	1 KB
molecule10-with-bonds.txt	16/04/2020 11:33 AM	Text Document	2 KB
MoleculeRenderer.java	16/04/2020 11:33 AM	JAVA File	5 KB
moleculeTest.txt	16/04/2020 11:33 AM	Text Document	1 KB
package.bluej	16/04/2020 11:33 AM	BlueJ Project File	2 KB

Assignment 2b - Core

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

MoleculeRenderer (Weight: 1/3)

For this question, you are to complete a program that draws 3D representations of molecules, drawing the atoms in a molecule as coloured circles on the graphics pane. To help the user see the 3D shape of the molecule, the program can let the user rotate around the molecule to view it from different directions.

The program involves Maps, Lists, and Sorting.

Assignment 2b - Core

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

MoleculeRenderer (Weight: 1/3)

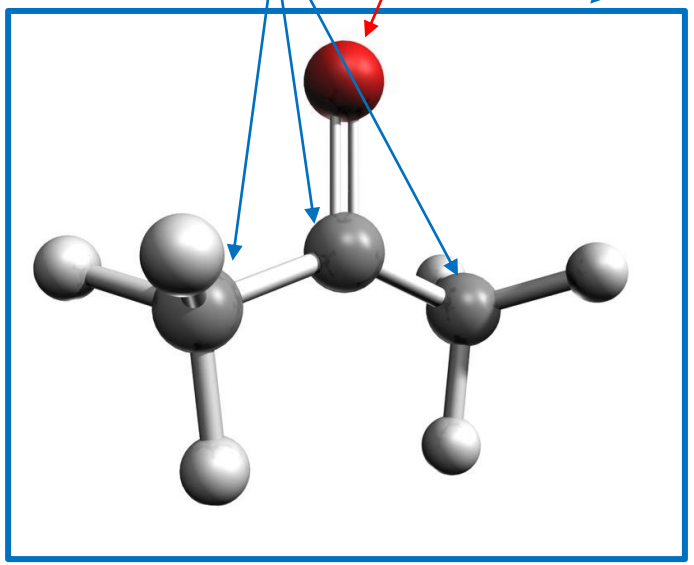
For this question, you are to complete a program that draws 3D representations of molecules, drawing the atoms in a molecule as coloured circles on the graphics pane. To help the user see the 3D shape of the molecule, the program can let the user rotate around the molecule to view it from different directions.

The program involves Maps, Lists, and Sorting.

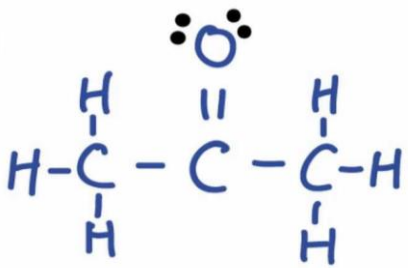
3D molecule

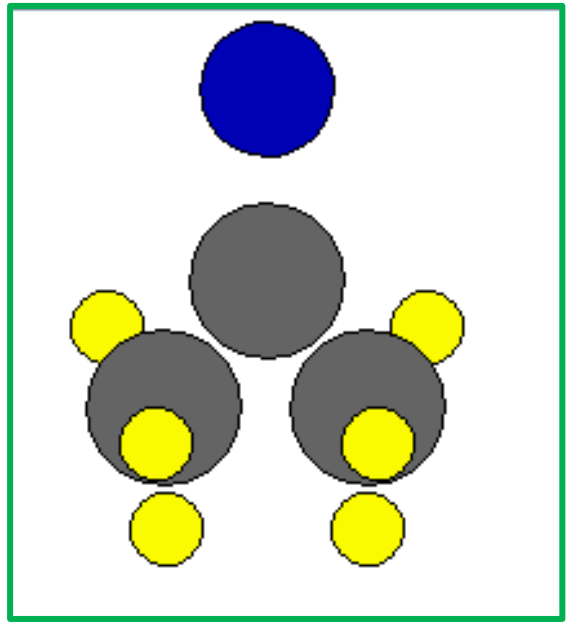
3D molecule to be created from your Java program

Acetone molecule: C_3H_6O



Hand written 3D molecule





Assignment 2b - Core

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

MoleculeRenderer (Weight: 1/3)

For this question, you are to complete a program that draws 3D representations of molecules, drawing the atoms in a molecule as coloured circles on the graphics pane. To help the user see the 3D shape of the molecule, the program can let the user rotate around the molecule to view it from different directions.

The program involves Maps, Lists, and Sorting.

The Problem

The shape of molecules can be very important for understanding their function, especially for large biological molecules. Using X-ray crystallography and other techniques, chemists or biochemists can identify the structure of a molecule by identifying the (x, y, z) coordinates of each atom in the molecule. However, tables of values are hard to make sense of, and it is better to be able to visualise the molecule by presenting a graphical rendering of the atoms. Because molecules are 3D shapes, is hard to make sense of them with just one viewpoint – a good visualisation will allow a user to view the molecule from different directions.

The `MoleculeRenderer` program produces a simple visualisation that draws each atom in a molecule as a coloured circle. To ensure a correct 3D visualisation, the atoms that are further from the viewer are drawn before atoms that are closer to the viewer, so atoms in front will cover any atoms behind. The `MoleculeRenderer` program also lets the user step left or right around the molecule in 5 degree steps to view the molecule from multiple (horizontal) directions.

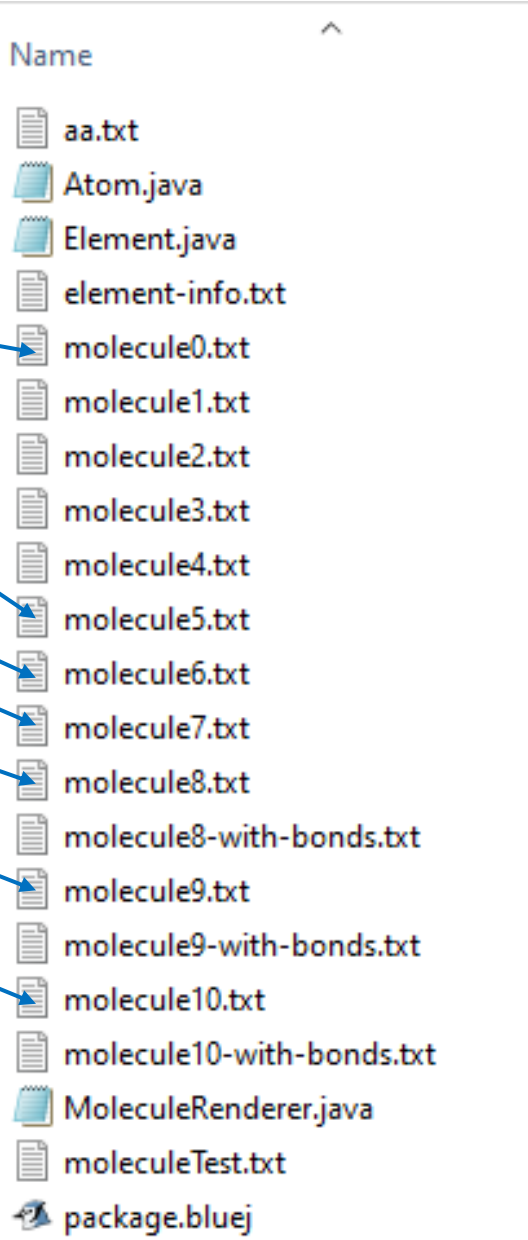
Complete the `MoleculeRenderer` program so that it reads a specification of the atoms in a molecule and then renders the molecule on the screen. It should be initially shown from the front, but should then allow the user to view the molecule from different directions.

Assignment 2b - Core

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

There are several molecule files including

- acetone (`molecule0.txt`),
- a small part of the hemoglobin protein that carries oxygen in our blood stream (`molecule5.txt`),
- a *buckyball* — an unusual form of Carbon (`molecule6.txt`), and
- curcumin — a critical component of Turmeric (`molecule7.txt`)
- trinitrotoluene (`molecule8`)
- pregestoerone (`molecule9`)
- rose bengal - a dye (`molecule10`)



The program has three parts:

1. Reading information about elements (including size and colour) from a file into a Map of `Element` objects, indexed by the element type ("C", "H", " etc).
2. Reading the description of a molecule from a file (atom types and positions) into a List of `Atom` objects.
3. Drawing the molecule by drawing all the atoms in the molecule in order after sorting the list of atoms in the molecule according to the desired viewi direction, so that atoms that should be further from the viewer come earlier in the ordering, and atoms that should be closer come later in the ordering.

Element Information data.

The `element-info.txt` file contains the size and colour for drawing each kind of element. Your program needs to read this data into a Map of `Eleme` objects, with the name of the element ("O", "C", "N", etc) as the key of the map. The `Element` class is already written for you.

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, (for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Assignment 2b - Core

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

There are several molecule files including

- acetone (`molecule0.txt`),
- a small part of the hemoglobin protein that carries oxygen in our blood stream (`molecule5.txt`),
- a *buckyball* — an unusual form of Carbon (`molecule6.txt`), and
- curcumin — a critical component of Turmeric (`molecule7.txt`)
- trinitrotoluene (`molecule8`)
- pregestoerone (`molecule9`)
- rose bengal - a dye (`molecule10`)

The program has three parts:

1. Reading information about elements (including size and colour) from a file into a Map of `Element` objects, indexed by the element type ("C", "H", "N", etc).
2. Reading the description of a molecule from a file (atom types and positions) into a List of `Atom` objects.
3. Drawing the molecule by drawing all the atoms in the molecule in order after sorting the list of atoms in the molecule according to the desired viewing direction, so that atoms that should be further from the viewer come earlier in the ordering, and atoms that should be closer come later in the ordering.

Element Information data.

The `element-info.txt` file contains the size and colour for drawing each kind of element. Your program needs to read this data into a Map of `Element` objects, with the name of the element ("O", "C", "N", etc) as the key of the map. The `Element` class is already written for you.

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, C for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Assignment 2b - Core

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

There are several molecule files including

- acetone (`molecule0.txt`),
- a small part of the hemoglobin protein that carries oxygen in our blood stream (`molecule5.txt`),
- a *buckyball* — an unusual form of Carbon (`molecule6.txt`), and
- curcumin — a critical component of Turmeric (`molecule7.txt`)
- trinitrotoluene (`molecule8`)
- pregestoerone (`molecule9`)
- rose bengal - a dye (`molecule10`)

The program has three parts:

1. Reading information about elements (including size and colour) from a file into a Map of `Element` objects, indexed by the element type ("C", "H", "N", etc).
2. Reading the description of a molecule from a file (atom types and positions) into a List of `Atom` objects.
3. Drawing the molecule by drawing all the atoms in the molecule in order after sorting the list of atoms in the molecule according to the desired viewing direction, so that atoms that should be further from the viewer come earlier in the ordering, and atoms that should be closer come later in the ordering.

Element Information data.

The `element-info.txt` file contains the size and colour for drawing each kind of element. Your program needs to read this data into a Map of `Element` objects, with the name of the element ("O", "C", "N", etc) as the key of the map. The `Element` class is already written for you.

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, C for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Assignment 2b - Core

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

There are several molecule files including

- acetone (`molecule0.txt`),
- a small part of the hemoglobin protein that carries oxygen in our blood stream (`molecule5.txt`),
- a *buckyball* — an unusual form of Carbon (`molecule6.txt`), and
- curcumin — a critical component of Turmeric (`molecule7.txt`)
- trinitrotoluene (`molecule8`)
- pregestoerone (`molecule9`)
- rose bengal - a dye (`molecule10`)

The program has three parts:

1. Reading information about elements (including size and colour) from a file into a Map of `Element` objects, indexed by the element type ("C", "H", "N", etc).
2. Reading the description of a molecule from a file (atom types and positions) into a List of `Atom` objects.
3. Drawing the molecule by drawing all the atoms in the molecule in order after sorting the list of atoms in the molecule according to the desired viewing direction, so that atoms that should be further from the viewer come earlier in the ordering, and atoms that should be closer come later in the ordering.

Element Information data.

The `element-info.txt` file contains the size and colour for drawing each kind of element. Your program needs to read this data into a Map of `Element` objects, with the name of the element ("O", "C", "N", etc) as the key of the map. The `Element` class is already written for you.

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, C for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Assignment 2b - Core

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

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, C for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Rendering the Molecule

To draw the atoms on the graphics window the way they would appear from some direction, the program must first order the atoms so that the atoms further from the viewer are ordered before the atoms closer to the viewer (so that the further ones appear to be behind the closer ones). This means that it must reorder the atoms in different ways, depending on their `(x, y, z)` position and the viewing direction.

The *x* axis goes from left to right, the *y* axis goes from top to bottom, and the ***z*** axis goes from near to far. This means that viewing the molecule from the front should order the atoms from large ***z*** to small ***z***. Viewing from a different direction will give a different order.

Each atom should be drawn as a circle with the right size and colour specified by its element type. The 3D positions of the atoms specified in the file are relative to the centre of the molecule. The position of the atom on the screen will depend on where the molecule should be centred on the screen and on the viewing direction.

Look carefully at the methods in the `Atom` class: there are methods that will help in the sorting and methods to help with drawing.

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 `Atoms`.
- Renders the molecule from the front

Completion

Assignment 2b - Core

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

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, C for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Rendering the Molecule

To draw the atoms on the graphics window the way they would appear from some direction, the program must first order the atoms so that the atoms further from the viewer are ordered before the atoms closer to the viewer (so that the further ones appear to be behind the closer ones). This means that it must reorder the atoms in different ways, depending on their `(x, y, z)` position and the viewing direction.

The *x* axis goes from left to right, the *y* axis goes from top to bottom, and the ***z*** axis goes from near to far. This means that viewing the molecule from the front should order the atoms from large ***z*** to small ***z***. Viewing from a different direction will give a different order.

Each atom should be drawn as a circle with the right size and colour specified by its element type. The 3D positions of the atoms specified in the file are relative to the centre of the molecule. The position of the atom on the screen will depend on where the molecule should be centred on the screen and on the viewing direction.

Look carefully at the methods in the `Atom` class: there are methods that will help in the sorting and methods to help with drawing.

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 `Atoms`.
- Renders the molecule from the front

Completion

Assignment 2b - Core

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

Completion

- Complete the `showFromViewDirection` method in the `MoleculeRender` class to render the molecule from different viewing directions.

- Viewing molecules from just horizontal directions doesn't give the user a good enough feel for the 3D shape. Extend the program to allow the user to move their view point up and down (from above the molecule to below it), as well as moving around.
  You will need to extend the GUI and modify your `showFromViewDirection` method.

Challenge

- Molecules are often rendered with balls and sticks, where the sticks show the bonds between the atoms. We have provided three files (`molecule8-with-bonds.txt` ... `molecule10-with-bonds.txt`) that contains the pairs of atoms in the molecule that have a bond between them. Following the atom types and positions, there are lines with two numbers - the indexes of two atoms that have a bond. For example, the line with "42 1" means that atom number 42 has a bond to atom number 1 (the atoms are numbered from 0 in the order they appear in the file).
  Modify the rendering to show the bonds as a fat line between the atoms. You may need to reduce the sizes of all the circles in order to show the bonds nicely.
- Make the atoms themselves look spherical rather than flat.

References

- Sizes of Atoms: http://www.rsc.org/periodic-table (Each element is a link)
- molecule7 is curcumin: https://www.ccdc.cam.ac.uk/structures/Search?Compound=curcumin&DatabaseToSearch=Published
- The data for molecule7 (curcumin) was provided by Zarinah Amin.
- The acetone data (molecule0): https://pubchem.ncbi.nlm.nih.gov/compound/acetone
- Part of the hemoglobin protein (molecule5) is from a protein database: http://www.rcsb.org/pdb/
- The large molecule containing only C atoms (molecule6): http://www.gcsescience.com/a38-buckminsterfullerene.htm

Assignment 2b - Core

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

Completion

- Complete the `showFromViewDirection` method in the `MoleculeRender` class to render the molecule from different viewing directions.

- Viewing molecules from just horizontal directions doesn't give the user a good enough feel for the 3D shape. Extend the program to allow the user to move their view point up and down (from above the molecule to below it), as well as moving around.
  You will need to extend the GUI and modify your `showFromViewDirection` method.

Challenge

- Molecules are often rendered with balls and sticks, where the sticks show the bonds between the atoms. We have provided three files (`molecule8-with-bonds.txt` ... `molecule10-with-bonds.txt`) that contains the pairs of atoms in the molecule that have a bond between them. Following the atom types and positions, there are lines with two numbers - the indexes of two atoms that have a bond. For example, the line with "42 1" means that atom number 42 has a bond to atom number 1 (the atoms are numbered from 0 in the order they appear in the file).
  Modify the rendering to show the bonds as a fat line between the atoms. You may need to reduce the sizes of all the circles in order to show the bonds nicely.
- Make the atoms themselves look spherical rather than flat.

References

- Sizes of Atoms: http://www.rsc.org/periodic-table (Each element is a link)
- molecule7 is curcumin: https://www.ccdc.cam.ac.uk/structures/Search?Compound=curcumin&DatabaseToSearch=Published
- The data for molecule7 (curcumin) was provided by Zarinah Amin.
- The acetone data (molecule0): https://pubchem.ncbi.nlm.nih.gov/compound/acetone
- Part of the hemoglobin protein (molecule5) is from a protein database: http://www.rcsb.org/pdb/
- The large molecule containing only C atoms (molecule6): http://www.gcsescience.com/a38-buckminsterfullerene.htm

Assignment 2b - Core

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

Completion

- Complete the `showFromViewDirection` method in the `MoleculeRender` class to render the molecule from different viewing directions.

- Viewing molecules from just horizontal directions doesn't give the user a good enough feel for the 3D shape. Extend the program to allow the user to move their view point up and down (from above the molecule to below it), as well as moving around.
  You will need to extend the GUI and modify your `showFromViewDirection` method.

Challenge

- Molecules are often rendered with balls and sticks, where the sticks show the bonds between the atoms. We have provided three files (`molecule8-with-bonds.txt` ... `molecule10-with-bonds.txt`) that contains the pairs of atoms in the molecule that have a bond between them. Following the atom types and positions, there are lines with two numbers - the indexes of two atoms that have a bond. For example, the line with "42 1" means that atom number 42 has a bond to atom number 1 (the atoms are numbered from 0 in the order they appear in the file).
  Modify the rendering to show the bonds as a fat line between the atoms. You may need to reduce the sizes of all the circles in order to show the bonds nicely.
- Make the atoms themselves look spherical rather than flat.

References

- Sizes of Atoms: http://www.rsc.org/periodic-table (Each element is a link)
- molecule7 is curcumin: https://www.ccdc.cam.ac.uk/structures/Search?Compound=curcumin&DatabaseToSearch=Published
- The data for molecule7 (curcumin) was provided by Zarinah Amin.
- The acetone data (molecule0): https://pubchem.ncbi.nlm.nih.gov/compound/acetone
- Part of the hemoglobin protein (molecule5) is from a protein database: http://www.rcsb.org/pdb/
- The large molecule containing only C atoms (molecule6): http://www.gcsescience.com/a38-buckminsterfullerene.htm

Assignment 2b - Core

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

Molecule data.

Each molecule is specified in a file that contains one line for each atom in the molecule. Each atom is specified by an element type (eg, O for oxygen, C for carbon, N for nitrogen, etc), and three numbers specifying the x, y, z coordinates of the atom, relative to the centre of the molecule.

Rendering the Molecule

To draw the atoms on the graphics window the way they would appear from some direction, the program must first order the atoms so that the atoms further from the viewer are ordered before the atoms closer to the viewer (so that the further ones appear to be behind the closer ones). This means that it must reorder the atoms in different ways, depending on their `(x, y, z)` position and the viewing direction.

The *x* axis goes from left to right, the *y* axis goes from top to bottom, and the ***z*** axis goes from near to far. This means that viewing the molecule from the front should order the atoms from large ***z*** to small ***z***. Viewing from a different direction will give a different order.

Each atom should be drawn as a circle with the right size and colour specified by its element type. The 3D positions of the atoms specified in the file are relative to the centre of the molecule. The position of the atom on the screen will depend on where the molecule should be centred on the screen and on the viewing direction.

Look carefully at the methods in the `Atom` class: there are methods that will help in the sorting and methods to help with drawing.

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 `Atoms`.
- Renders the molecule from the front

Completion

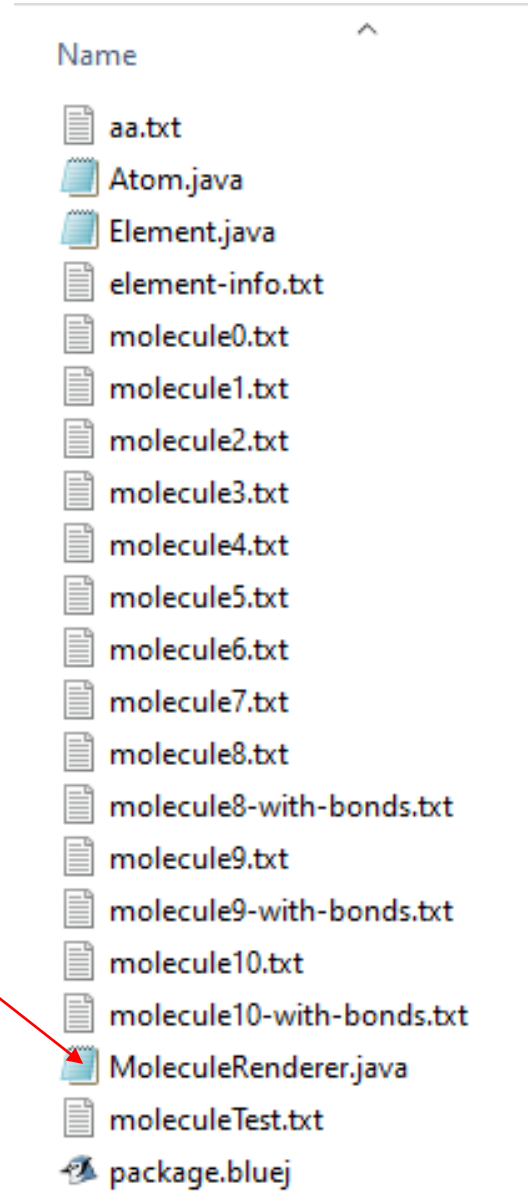
Assignment 2b - Core

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 `Atoms`.
- Renders the molecule from the front



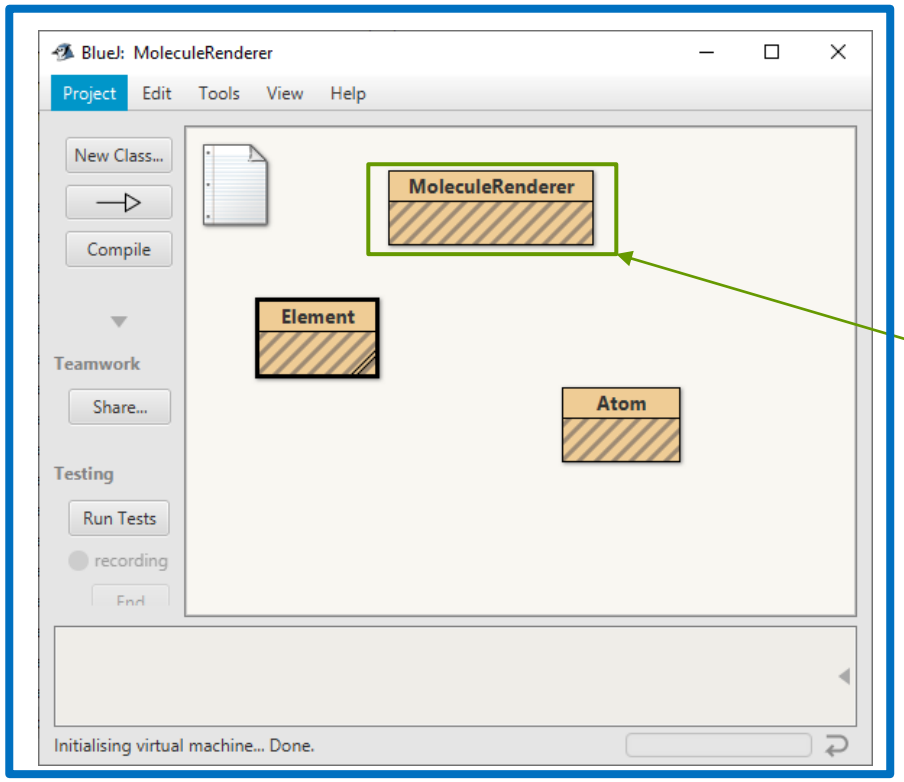
Assignment 2b - Core

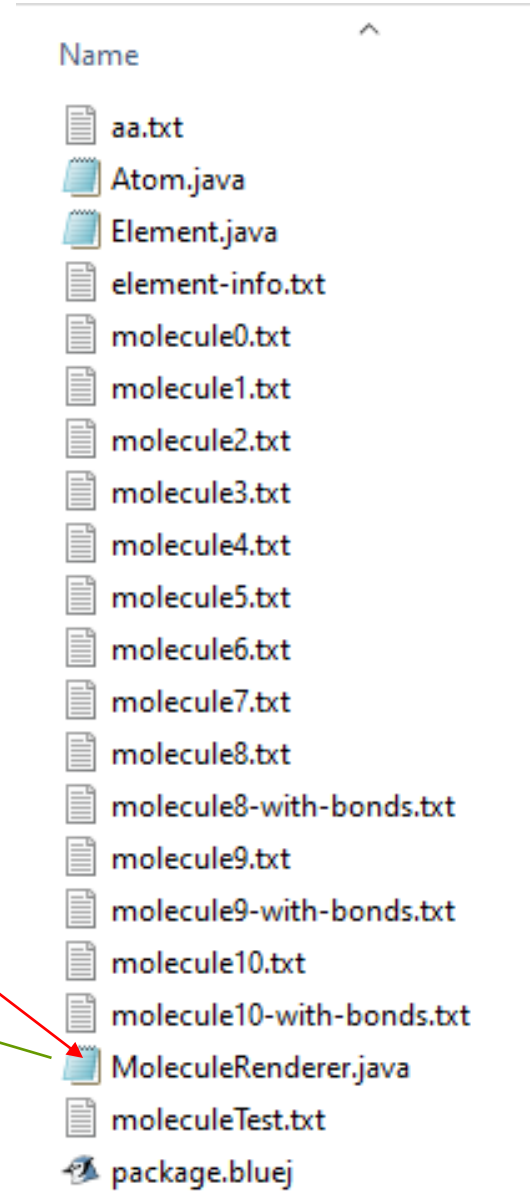
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

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 `Atoms`.
- Renders the molecule from the front





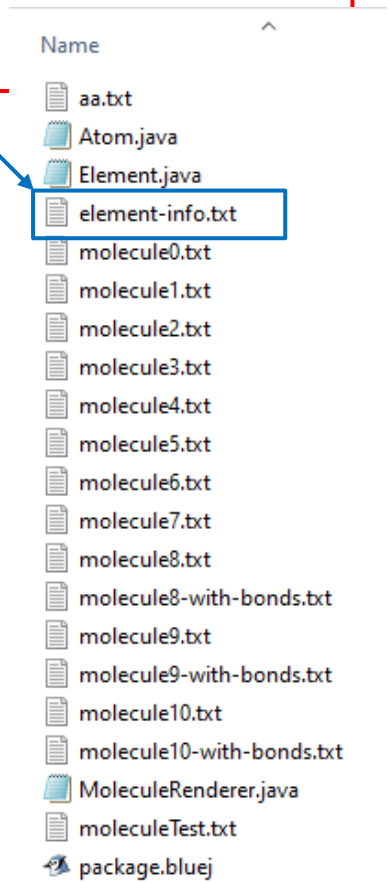
Assignment 2b - Core

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 `Atoms`.
- Renders the molecule from the front



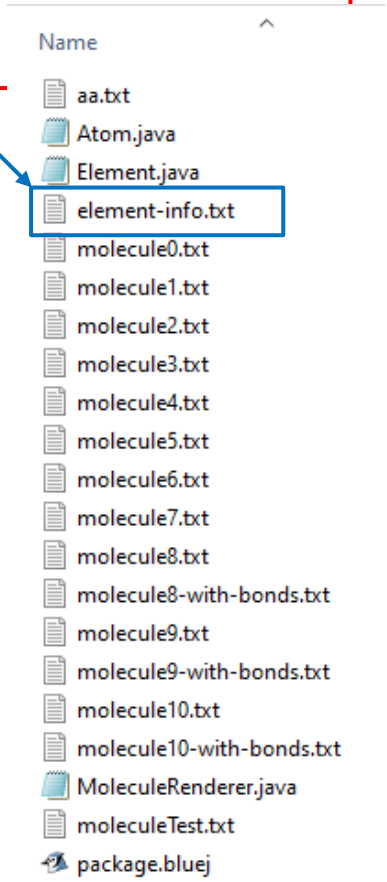
Assignment 2b - Core

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 `Atoms`.
- Renders the molecule from the front



Data inside element-info.txt

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

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 `Atoms`.
- Renders the molecule from the front

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

Data inside element-info.txt

Element Information data.

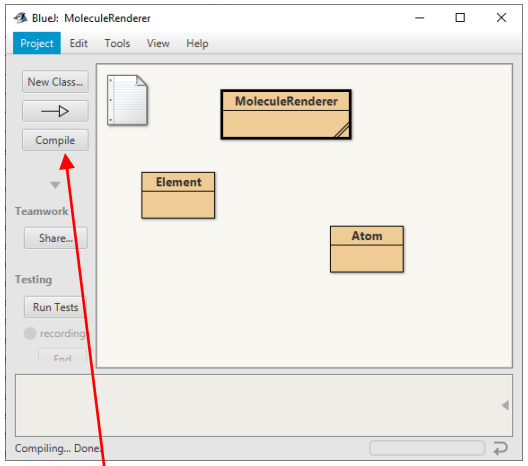
The `element-info.txt` file contains the size and colour for drawing each kind of element. Your program needs to read this data into a Map of `Element` objects, with the name of the element ("O", "C", "N", etc) as the key of the map. The `Element` class is already written for you.

Element name, atomic number, atomic weight, a radius, (red, green & blue values for the colour)
Example: H 1 1.008 14 250 250 0

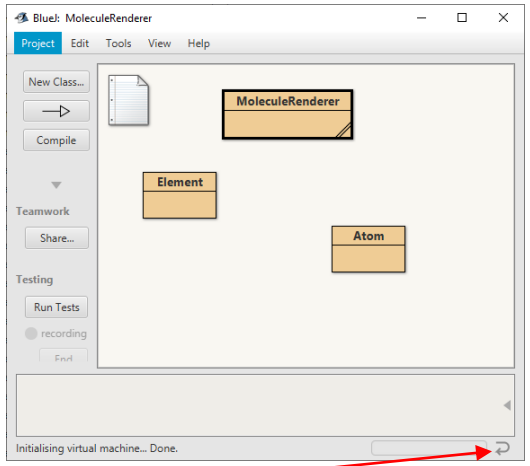
Mg	12	24.305	53	100	0	100
Sc	21	44.956	56	140	60	0
Na	11	22.990	61	80	0	120
Ca	20	40.078	68	160	40	0
Ba	56	137.327	77	0	20	180
K	19	39.098	39	180	20	0
Rb	37	85.468	84	0	40	160

Assignment 2b - Core

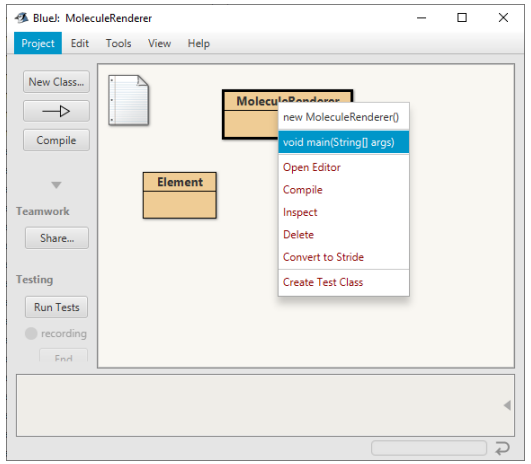
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB



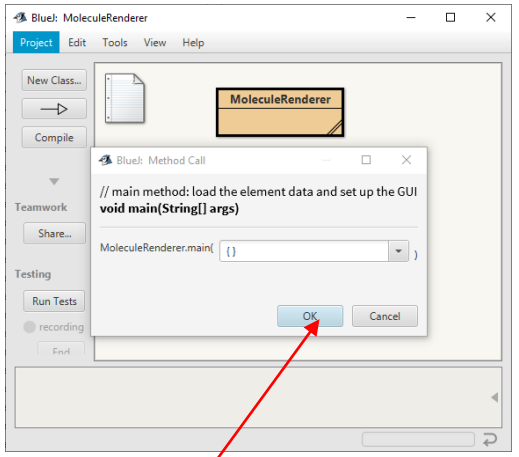
1) Compile



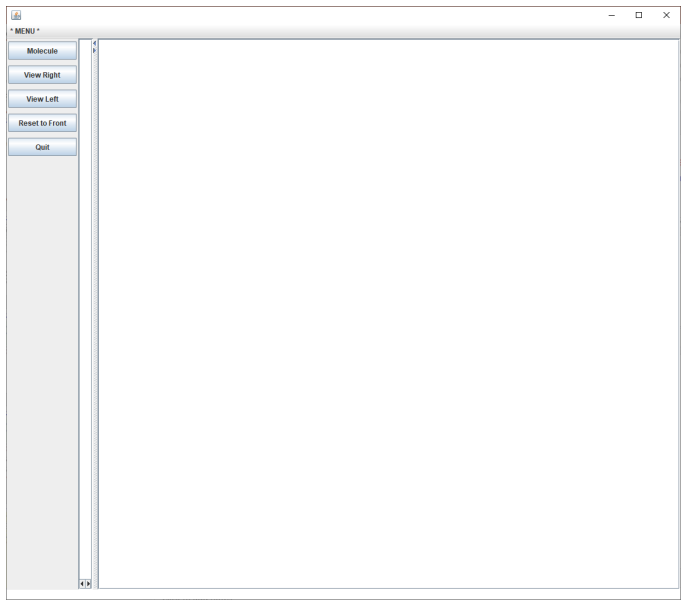
2) Reset Java Virtual Machine



3) Run the program



4) Click OK button



5) End result

Assignment 2b - Core

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

MoleculeRenderer - MoleculeRenderer

Class Edit Tools Options

MoleculeRenderer ×

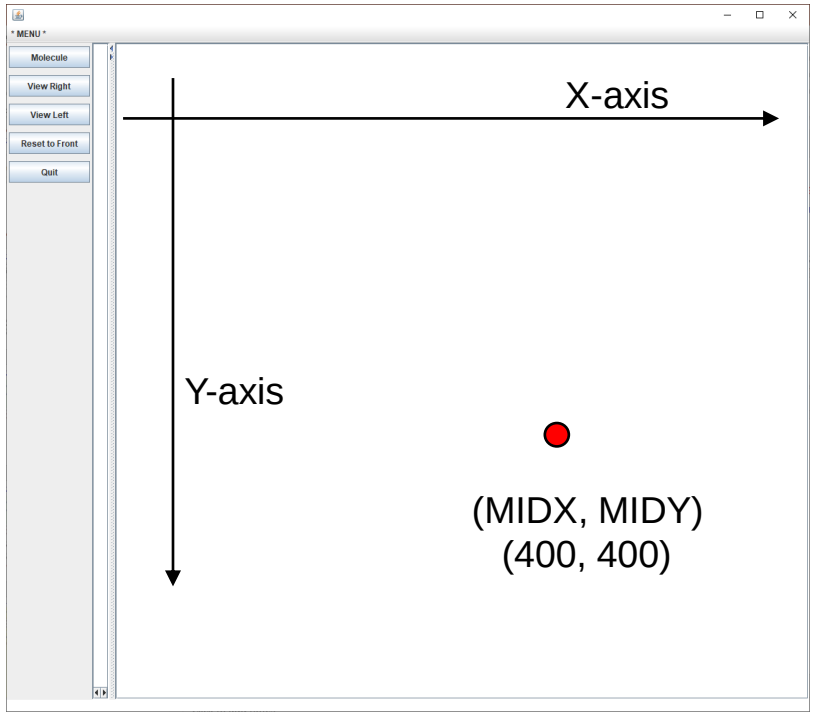
Compile Undo Cut Copy Paste Find... Close Source Code

```
1  // This program is copyright VUW.
2  // You are granted permission to use it to construct your answer to a XMUT103 assignment.
3  // You may not distribute it in any other way without permission.
4
5  /* Code for XMUT103 - 2020T1, Assignment 2
6   * Name:
7   * Username:
8   * ID:
9   */
10
11 import ecs100.*;
12 import java.awt.Color;
13 import java.util.*;
14 import java.io.*;
15
```

Assignment 2b - Core

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

```
16 /**
17  *  Renders a molecule on the graphics pane from different directions.
18  *
19  *  See the assignment page for a description of the program and what you have to do.
20  */
21
22 public class MoleculeRenderer {
23     // The molecule should be centered at (MIDX, MIDY) on the graphics pane:
24     public static final int MIDX = 400;
25     public static final int MIDY = 400;
```



Assignment 2b - Core

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

```
16 /**
17  *  Renders a molecule on the graphics pane from different directions.
18  *
19  *  See the assignment page for a description of the program and what you have to do.
20  */
21
22 public class MoleculeRenderer {
23     // The molecule should be centered at (MIDX, MIDY) on the graphics pane:
24     public static final int MIDX = 400;
25     public static final int MIDY = 400;
26
27     // Map containing info about each type of element.
28     private Map<String, Element> elements;
29
```

Assignment 2b - Core

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

```
16 /**
17  *  Renders a molecule on the graphics pane from different directions.
18  *
19  *  See the assignment page for a description of the program and what you have to do.
20  */
21
22 public class MoleculeRenderer {
23     // The molecule should be centered at (MIDX, MIDY) on the graphics pane:
24     public static final int MIDX = 400;
25     public static final int MIDY = 400;
26
27     // Map containing info about each type of element.
28     private Map<String, Element> elements;
29
```

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 `Atoms`.
- Renders the molecule from the front

Assignment 2b - Core

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

```
/**
 *  Renders a molecule on the graphics pane from
 *
 *  See the assignment page for a description of
 */

public class MoleculeRenderer {
    // The molecule should be centered at (MIDX, MIDY) on the graphics pane:
    public static final int MIDX = 400;
    public static final int MIDY = 400;

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

    // The collection of the atoms in the current molecule being rendered
    private List<Atom> molecule = new ArrayList<Atom>();

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

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 `Atoms`.
- Renders the molecule from the front

Assignment 2b - Core

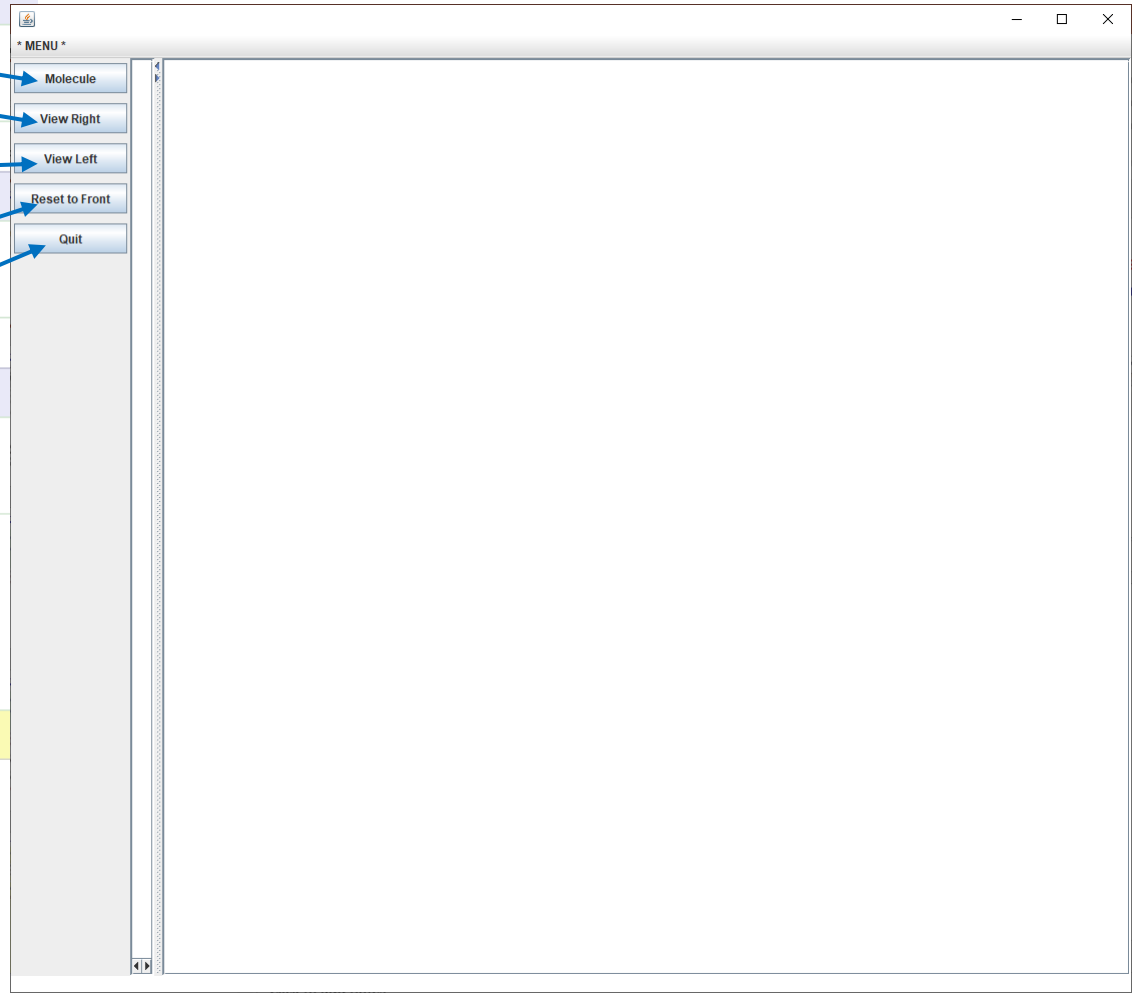
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartB

```
38  /**
39   * main method: load the element data and set up the GUI
40   */
41  public static void main(String args[]) {
42      MoleculeRenderer molRen = new MoleculeRenderer();
43      molRen.readElements();
44      molRen.setupGUI();
45  }
```

Assignment 2b - Core

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

```
47      /**
48       * Set up the user interface
49       */
50      public void setupGUI(){
51          UI.addButton("Molecule", this::loadAndRenderMolecule);
52          UI.addButton("View Right", ()->{
53                  horizViewAngle += 5;
54                  showFromViewDirection();
55              });
56          UI.addButton("View Left", ()->{
57                  horizViewAngle -= 5;
58                  showFromViewDirection();
59              });
60          UI.addButton("Reset to Front", ()->{
61                  horizViewAngle = 0;
62                  showFromFront();
63              });
64          UI.addButton("Quit", UI::quit);
65          UI.setWindowSize(MIDX*2, MIDY*2);
66          UI.setDivider(0.0);
67      }
```



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

```
94      /**
95       * Renders the molecule from the front.
96       * Sorts the Atoms in the List by their z value, back to front
97       * Uses the default ordering of the Atoms
98       * Then renders each atom
99       */
100     public void showFromFront() {
101         /*# YOUR CODE HERE */
102
103         UI.clearGraphics();
104         for(Atom atom : molecule) {
105             atom.render(MIDX, MIDY, 0, 0);
106         }
107     }
108
109     /**
110      * Renders the molecule from the current viewing angle.
111      * Sorts the Atoms in the List according to their distance at
112      * the current viewing angle
113      * Then renders each atom.
114      */
115     public void showFromViewDirection() {
116         /*# YOUR CODE HERE */
117
118     }
```

```
120     // Utility method to help with reading data files
121
122     /**
123      * Read all lines in a file into a list of Strings
124      */
125     public ArrayList<String> readAllLines(String filename){
126         try{
127             ArrayList<String> ans = new ArrayList<String>();
128             Scanner sc = new Scanner(new File(filename));
129             while (sc.hasNext()){
130                 ans.add(sc.nextLine());
131             }
132             sc.close();
133             return ans;
134         }
135         catch(IOException e){UI.println("Fail: " + e); return null;}
136     }
137
138 }
```