Design

The program follows the MVC principle. Therefore it is divided into three parts, which is model, view and controller. The model is the object model which is analysed. Moreover, the view is responsible for showing the information to users directly. The controller connects the model to view and control some actions happen when the user makes operations to interact with the program. The MVC principle makes the program more complex because every section of the program is separate. Moreover, the program followed MVC is easy to debug because it will not influence other two sections when one of the three parts was changed. Following is the design introduction of this assignment.

Model

There are three models in the model package which is Mandelbrot, MandelbrotCalculator and History. The first one is the object of the Mandelbrot set. It involves parameters that can constitute a Mandelbrot set image after calculating. Besides, it also has many functions that used to operate it with different parameters that transfer from the view. Another model is the History model that used to make user withdraw or redo the operation. The third model in the package is the MandelbrotCalculator that provided by the teacher, which used to calculate the Mandelbrot set.

View

The window is constituted by three parts, menu bar, panel and tool bar. Three components add into the window by the border layout. The menu bar is on the north of the window. There is an “About” button in the menu, which introduces how to operate. Moreover, there are two menu selections, which is File and Color. Users can execute the export or import function by clicking the menu item in the menu “File”. Moreover, the colour selections items can change the colour of the Mandelbrot set image, for instance, blue, red and green. Next, the panel is in the centre of the window. The Mandelbrot set image will be painted in the panel and users can make some operations in this area. Due to the panel implement the MouseMotionListener and the MouseWheelListener, users can drag the image by mouse or zoom the image by mouse wheel rotation. Moreover, the tool bar is in the south of the window, includes four buttons and two text fields. The user can reset, undo, redo or recalculate by the new max iteration. The text field is used to get the max iteration from the user, and another text field is used to display the times of zooming in.

Controller

Controllers are responsible for connecting view and model and making the reaction to users’ operation. It is like an intermediary; it will operate the model depends on the different action received from view, then it gives back the updated information of the model to view to display. There are three controllers corresponded to three sections, menu bar, panel and tool bar. The detail of the controllers’ function will appear in the implementation section.

Implementation

Show Instructions

There is an “About” button in the menu bar, which used to show the introduction of the program. To implement this function, the first thing is adding the JButton in the menu bar. Then, adding the action listener and compiling the action performance. The object JoptionPane is used in this case to show up the

message. The introduction message is written in the label, and the JoptionPane will show the label up when users click the button.

Display the Mandelbrot Set Image

The Mandelbrot involves parameters includes xResolution, yResolution, minReal, maxReal, minImaginary, maxImaginary, radiusSquared and maxIterations. These parameters will be used in calculating by the method in the model MandelbrotCalculator. In the default case, the model Mandelbrot will get the default value of the Mandelbrot set from the model MandelbrotCalculator. Then using the default parameters to calculate. After computing, it will obtain an array. Therefore, the panel class will traverse this array to draw the Mandelbrot set image. In the black and white case, If the number of the array equals the max iteration, it will draw the black line. Otherwise, it will draw the white line. The complete Mandelbrot set image will finish after traversing the whole array.

Zoom and Pan

In order to implement the zoom and pan, the panel class implements the MouseMotionListener and the MouseWhellListener to catch the action made by users. In the case of zooming, the rotation parameter of the mouse will be caught and makes pow operation with a default value. Then, using the pow result change the parameters of the Mandelbrot set, and recreate a new Mandelbrot image. At the same time, the tool bar will get the pow result as well. The pow result will be set in the text field to show users the zooming times. As for panning, the x coordination and the y coordination of the mouse will be used to recreate the parameter of the Mandelbrot set. Moreover, then, recreating a new Mandelbrot set by the new array of the set.

Recalculate with the Max Iteration

To catch the value printed in the text field by users, the recalculate button need to add the action listener. There will be some judgement for the value caught from the text field. If the value is greater than zero, it will execute the method in the model Mandelbrot, which recalculate the Mandelbrot with the new max iteration number. If the value less than zero, the window will show the label to alert users, which implements by the object JOptionPane. Similarly, if the user clicks the recalculate button without inputting the max iteration number. The wrong number format error will be caught and show the alert.

Undo and Redo

There are two stacks in the History model; the undo stack and the redo stack. The undo stack will store the current Mandelbrot set parameters. Moreover, if there any new operations caused by the user, new Mandelbrot parameters will add to the stack. When users withdraw the operation, the corresponding method in Mandelbrot model will be executed that the top element of the undo stack will pop out and add into redo stack and then the new top element of the undo stack will be used to create the Mandelbrot set image. Similarly, the top element of the redo stack will be used to calculate a Mandelbrot set when users want to redo the operation, but the element of the redo will not add into the undo stack. If there are no elements in the redo stack or the undo stack when users click the button, the alert will appear.

Export and Import

Furthermore, the program can export the Mandelbrot set parameters or load the outside parameters to create a Mandelbrot image. For this function, there is the export and the import method in the Mandelbrot model. The main point is to write (for save) or read (for load) the character string by I/O

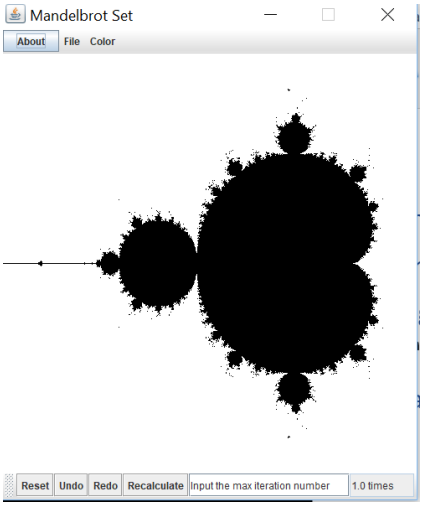
stream, then, give parameters to the method used to calculate. During the export and import, users can choose the file in the folder by the JFileChooser.

Change Color

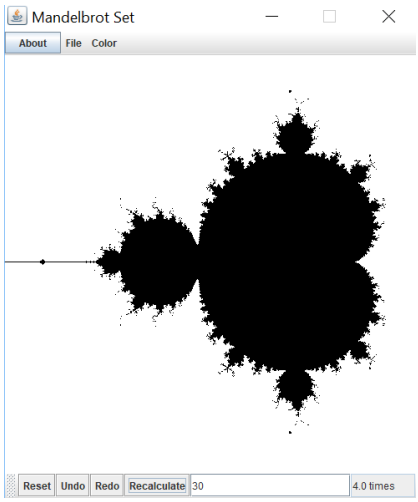
There is a method used to set the colour, and the colour value depends on the menu item that the user selects. Different colour menu item corresponds to the different event. For example, If the user selects the “Green” item, the parameter colour will be set green. Then, the draw method will judge the parameter colour and draw the Mandelbrot set. If the parameter colour is null, the image will paint with black and white.

Testing

The running program looks like the following picture 1-1. Moreover, the 1-2 shows the Mandelbrot set image that created by the new max iteration number.

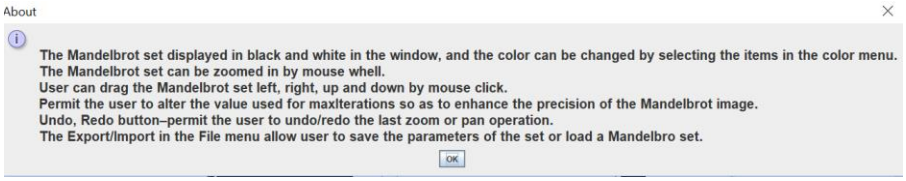


Picture 1-1



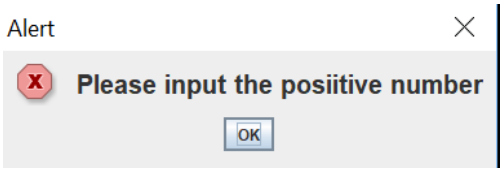
Picture 1-2

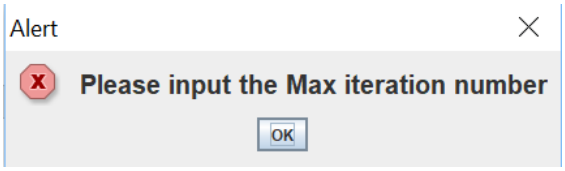
The picture 1-3 shows the instruction of the program, and it will show up when users click the “About” button in the menu bar.



Picture 1-3

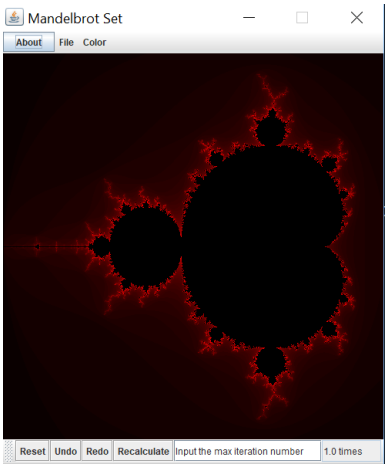
The following pictures show the alert when the user click the recalculate button without inputting the max iteration number or inputting the minus.

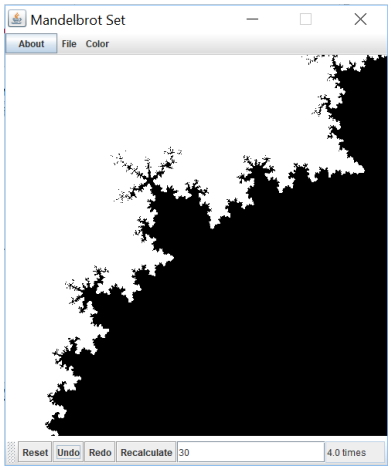




Picture 1-4 Picture 1-5

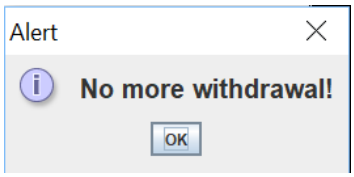
The following pictures show the colour changing function and the zooming in function. The times of zooming in will display in the right-bottom of the window.

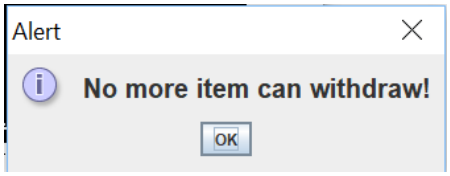




Picture 1-6 Picture1-7

These are the alert that the user clicks redo or undo button when there is no element in the stack.





Picture 1-9 Picture 1-10

The following data is the parameters of the Mandelbrot set, which exported from the program. And the export format is like this: xResolution: 500.0 yResolution: 500.0 MinReal: -2.0 MaxReal: 0.7 MinImaginary: -1.25 MaxImaginary: 1.25 MaxIterations: 50.0 RadiusSquared: 4.0

Evaluation

For some objects, they may not conform to the MVC principle completely. For example, the codes for the event performance that should be compiled in the controller object, but appears in the view object. Moreover, not familiar enough to the GUI components like how to change the menu size. Besides, it is hard to implement that drawing a rectangle when the user tries to select a district that used to zoom in or zoom out, therefore, the way for zooming changed to zoom by the mouse wheel.