Chun-Wei Chen

Assignment #3 Written Report

2012/02/05

At the beginning, I tried to remind what we’ve learned in class and how to use them by looking and running sample programs. After that, I realized why we need an interface and how to let model notify all interested viewers when the state changes. Before working on the controller part, graphic display and text display were working fine. Every shape was on the right place and the text also showed correctly. Then I started the controller part; however, I didn’t do it in the right way initially. I added all the controller code in my graphic display class. Even though I could move the shapes in the window, the instructor told me I should separate the controller and view. I spent a lot of time to figure out how to separate them. I even tried to add the controller on the JPanel when I tried to make my code work. After discussing with the instructor about the problem and trying several times, I finally understood that I just needed to add a DrawingBoard as an instance variable, pass in a DrawingBoard as a parameter when constructing the controller, and change the state of the model (DrawingBoard) when the events happen. At that time, I could move the shapes perfectly even if controller and view were separated. Nevertheless, I encountered errors. When there wasn’t any selected shape, Dr. Java showed errors while I dragged the mouse. I found that I should add an if statement to take care that situation, then the program ran fine. When the user tries to move, s/he first clicks on a shape. At the same time, a MouseEvent is created and it is passed in as an argument to call mousePressed method to find the selected shape, which means to find the shape contains the point which the user clicks. When the selected shape is found, the shape becomes the topmost shape in the window and is drawn with black outline. Then the user drags the shape, another MouseEvent is created and it is passed in as an argument to call mouseDragged. And then mouseDragged method calls moveSelectedShape to move the selected shape to where the user wants. The program repeats to call mouseDragged method while the user keeps dragging. This is how my program works when the user moves the shape.

Afterwards, I added buttons to my program. I used RadioButton instead JButton in my program. I didn’t know how to use it at first, so I made use of java tutorial to make me understand how to use it. I coded the performance that the program should do when one button was clicked, created RadioButtons, used ButtonGroup to group them, and added them on the JPanel. After those, the RadioButtons showed on the window, but they didn’t work after I clicked them. Then I found that I forgot to use addActionListener to let the controller listen to RadioButtons. After I fixed them, the buttons worked the way I wanted. I could click “Circle,” “Tee,” or “Delta” buttons and then click on the window to create the shape I clicked; besides, I could also click “Not Adding Shape” button when I want to move the shapes.

After I finished the basic requirement of the assignment, I decided to add more feature to my program. I first changed the text display part. I replaced System.out by JTextArea. Another thing I added to my program was JSlider, which could let the user control the size of the shape they want to create. I added that in graphic display class, and also let controller class implement ChangeListener in order to make the slider to set the size of the shape the user wants to create. The controller rounds up the size value for some shape in order to make the size value valid to construct the shape. When the user wants to create a Delta, the controller rounds up the size value if it’s an even number; moreover, the controller rounds up the size value to the closest multiple of 10 if the user wants to create a Tee.

This assignment reinforces my knowledge of MVC. I thought I already understood the how to design a MVC project before I started the assignment; nonetheless, I found that I didn’t really fully comprehend MVC. By looking sample programs and the java tutorial again and again, and trying a lot of ways to make my program works fine, I finally grasped how to design a MVC project. This assignment also makes me learn how to use JComponents and how to implement the listeners to handle the event. Even though I was stuck with making my program work, I really learn a lot from this assignment. I hope this assignment can help me avoid the same problem I had in this assignment or figure out the problems more quickly in the future.