SchoolNova

IT101

Graphical User Interface
part III

MenuBar, Menu and MenuItem

- In this class we will continue working on the Tennis Game application. We will add a menu to Save the score to a file, and as a result learn two things:

    1. How to create menus in Java;
    2. How to work with Files.

- You can continue working with your Tennis Game from the previous classes, or download a new code base from http://www.schoolnova.org/student_area/tennis/TennisGame_v1.2.zip

- Menus in Java are part of the window (JFrame), not part of the canvas (JPanel).
- Menus consist of MenuBar, Menu and MenuItem. You add MenuItem to the Menu, Menu to the MenuBar, and MenuBar to the JFrame.

```
JMenuBar menubar = new JmenuBar();
JMenu fileMenu = new JMenu("File");
JMenuItem menuItem1 = new JMenuItem("Save");
fileMenu.add(menuItem1);
menubar.add(fileMenu);
frame.setJMenuBar(menubar);
```

- Compile and run the code to confirm that the menu was successfully added to the frame.

Capturing Menu Events

- Note that your game does not stop when you use the menu. In order to stop (suspend) the game, we need 'listen to events that occur to the menu' by adding a MenuListener:

```
fileMenu.addMenuListener(new MenuListener() {
                        public void menuSelected(MenuEvent e) {
                            game.suspend = true;
                        }
                        public void menuDeselected(MenuEvent e) {
                            game.suspend = false;
                        }
                        public void menuCanceled(MenuEvent e) {
                           // do nothing
                        }
                    });
```

- Note that when the menu is selected, we change game's "suspend" property to "true", and when the menu is deselected, we change "suspend" to "false". In order for that to work, add a "suspend" variable to the TennisGame object, and use it in the "move" method, as follows:

```
private boolean suspend = false;

if (!suspend) {
  ball.moveBall();
  racquet.move();
}
```

- Compile and run the code to confirm that the game stops when the menu is selected, and resumes when the menu is deselected.

Writing to a File

- Now make the menu write the score to a file. First create a method that writes the score to a file. We can use the java.io.FileWriter class to write content to a file.

```
/**
 * Saves the current score to a file.
 */
 private void saveScore() {
    try {
         FileWriter output = new FileWriter("/Users/?/Desktop/tennis_score.txt");
         output.write(String.valueOf(score));
         output.close();
    } catch ( IOException e ) {
         JOptionPane.showMessageDialog(this, "Score could not be saved.");
    }
 }
```

- Note that the file path structure will be different between Unix based operating systems, such as Macintosh, and Windows operating systems.
- Also, note that FileWriter may throw an IOException if it can not create and/or write to a file. In such a case we have to do one of two things: pass the exception to the calling method, or handle it using the try/catch block.

Capturing Menu Item Events

- Now add an ActionListener to the MenuItem, which invokes saveScore when the menu item is clicked.

```
menuItem1.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent event) {
                        game.saveScore();
                    }
                });
```

- Compile and run the code to confirm that the game writes the score to a file.

Homework

- Enhance your game by reading the score from the file when the game starts.
- Remember that the first method that gets executed when TennisGame starts is the TennisGame constructor.
- Use java.io.BufferedReader object, it has a “read” method that can read a file line by line.
- Use the Integer.valueOf method to convert the score from String to int.
- Here’s one possible way to read the file:

```
BufferedReader input = new BufferedReader(new FileReader(filePath));
                String line;
                while ((line = input.readLine()) != null) {
                    score = Integer.valueOf(line);
                }
```

- Remember to handle BufferedReader’s IOException.