

C490 Homework #4

Points:	: 50 points
Due Date:	: July 22 nd (8:00am)
Submissions:	: Canvas and hardcopy

PART I (20 POINTS)

Design and code a Swing GUI for a two-player tic-tac-toe (noughts and crosses) game on a 3 by 3 game board. The JFrame should use a BorderLayout with a JLabel in the NORTH region to display messages (e.g. who won the game), and a JPanel in the CENTER region to display the game board. For the game board in the JPanel, use a GridLayout manager with a 3 by 3 layout of JButton's in each cell to display the game board. The button labels should initially be blank. When a player clicks on an empty button an appropriate "X" or "O" should be placed in the label field of the button. If there is a winner (three in a row) then the program should display the winner in the JLabel located at the top of the window. If all nine cells have been filled without a winner the program should indicate that there was a tie. Your program should behave like the binary code (*TicTacToe.class*) posted in the Canvas.

Hints:

- 1) You may want to define some instant variables to trace the status of the game, for example, whosTurn, totalStepsFinished, whoWins, etc.
- 2) You may want to define a private method that has a heading like the following;

```
private String checkWhoWins()  
//if nobody wins the game yet, return "U",  
//if O wins the game, return "O";  
//if X wins the game, return "X";  
//if game gets tied, return "T".
```

PART II (10 POINTS)

The bouncing ball example (in Canvas) is a program that bounces a blue ball inside a window. The ball begins moving when you click your mouse inside the window. When the ball hits the edge of the window, it bounces off the edge and continue in the opposite direction.

- (1) Compile and run the program, read the code. Some of the java.awt and java.swing classes are not covered in our lectures. Don't worry about them, focus on the part of how the program creates and executes threads.
- (2) Modify the program to add a new ball with random color each time the user clicks the mouse. Provide for a maximal of 20 balls.

Hint:

(1) You may want to define an **array** of Ball type in BallPanel.java. You may also want to define a counter to track how many balls you have created.

(2) You may want to modify the method of createBall() to be like this:

```
if (count < 20)
{
    ...
    count ++;
}
```

(3) You may also want to modify the method of paintComponent() to be like this:

```
for (int i = 1; i <= count; i ++ )
{
    ...
}
```

PART III (20 POINTS)

Write a program to edit text file for extra blanks. The program will replace any string of two or more blanks with a single blank. Your program should work as follows: Create a temporary file. Copy from the file to the temporary file but do not copy extra blanks. Then delete the original file, and rename the temporary file to be the original file.

- (1) Your program will ask the user for the name of the file to be edited.
- (2) The temporary file should have a name different from all existing files so that the existing files are not affected (except for the file being edited). However, your program will not ask the user for the name of the temporary file but instead will generate the name within the program. You can generate the name any way that is clear and efficient. One possible way to generate the name for the temporary file is to start with an unlikely name, such as "TempX", then to append a character, such as 'X', until a name is found that does not name an existing file. To check if a name conflicts with existing files, you could create an **File** object (of class **File** type), and use *.exists()* method to check if such a name has been used or not.
- (3) You will want to use other methods of class **File** to delete a file, and rename a file.
- (4) Use appropriate streams to read and write files.
- (5) You may want to use java.util.StringTokenizer to remove extra space. Following is an example:

```
String line;
while (inputStream.hasNextLine()) {
    line = inputStream.nextLine();
```

```
StringTokenizer st = new StringTokenizer(line, " ");

while (st.hasMoreTokens()) {
    word = st.nextToken();
    if (st.hasMoreTokens())
        outputStream.print(word + " ");
    else
        outputStream.println(word);
}
}
```

WHAT TO SUBMIT:

- Submit your code for all three parts to Canvas (using the "Assignments" function).
- Submit a hard copy of your code and test-run output (or screenshot).
- Make sure that you follow the "Assignment_style-guideline_C490" or you might lose credits.