

## C490 Homework #4

Points:	: 40 points
Due Date:	: July 23 <sup>rd</sup> (8:30am)
Submissions:	: Canvas and hardcopy

### PART I (30 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 ++ )
{
    ...
}
```

**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.