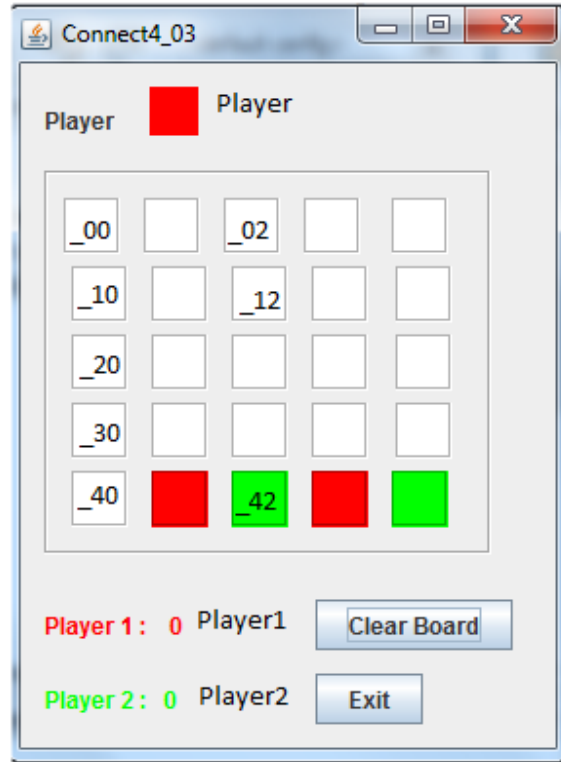


Fundamentals of Programming II Assignment 05

5x5 connect



```
public int[][] board = {{10, 10, 10, 10,10}, {10, 10, 10, 10,10}, {10, 10, 10, 10,10}, {10, 10, 10, 10,10}, {10, 10, 10, 10,10}};
```

```
private void _00MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event__00MouseClicked
    if(board[4][0]==10)
        play40();
    else if(board[3][0]==10)
        play30();
    else if(board[2][0]==10)
        play20();
    else if(board[1][0]==10)
        play10();
    else
        play00();
} //
```

```
public void play00() {
    if (_00.getBackground().getRGB() == Color.WHITE.getRGB()) {
        _00.setBackground(Player.getBackground());
        if (_00.getBackground().getRGB() == Color.RED.getRGB()) {
            board[0][0] = 0;
        } else if (_00.getBackground().getRGB() == Color.GREEN.getRGB()) {
            board[0][0] = 1;
        }
    }
    checkWinner();
    switchPlayer();
}
```



```
}  
}
```

```
private void _10MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event__10MouseClicked  
    if(board[4][0]==10)  
        play40();  
    else if(board[3][0]==10)  
        play30();  
    else if(board[2][0]==10)  
        play20();  
    else  
        play10();  
  
} //
```

```
public void play10() {  
    if (_10.getBackground().getRGB() == Color.WHITE.getRGB()) {  
        _10.setBackground(Player.getBackground());  
        if (_10.getBackground().getRGB() == Color.RED.getRGB()) {  
            board[1][0] = 0;  
        } else if (_10.getBackground().getRGB() == Color.GREEN.getRGB()) {  
            board[1][0] = 1;  
        }  
        checkWinner();  
        switchPlayer();  
    }  
}
```

```
private void _20MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event__20MouseClicked  
    if(board[4][0]==10)  
        play40();  
    else if(board[3][0]==10)  
        play30();  
    else  
        play20();  
  
} //
```

```
public void play20() {  
    if (_20.getBackground().getRGB() == Color.WHITE.getRGB()) {  
        _20.setBackground(Player.getBackground());  
        if (_20.getBackground().getRGB() == Color.RED.getRGB()) {  
            board[2][0] = 0;  
        } else if (_20.getBackground().getRGB() == Color.GREEN.getRGB()) {  
            board[2][0] = 1;  
        }  
        checkWinner();  
        switchPlayer();  
    }  
}
```

```
private void _30MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event__30MouseClicked  
    if(board[4][0]==10)  
        play40();  
    else if(board[3][0]==10)  
        play30();  
  
} //
```

```
public void play30() {  
    if (_30.getBackground().getRGB() == Color.WHITE.getRGB()) {
```



```
_30.setBackground(Player.getBackground());
if (_30.getBackground().getRGB() == Color.RED.getRGB()) {
    board[3][0] = 0;
} else if (_30.getBackground().getRGB() == Color.GREEN.getRGB()) {
    board[3][0] = 1;
}
checkWinner();
switchPlayer();
}
}
```

```
private void _40MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event__40MouseClicked
    play40();
} //
```

```
void play40()
{
    if (_40.getBackground().getRGB() == Color.WHITE.getRGB()) {
        _40.setBackground(Player.getBackground());
        if (_40.getBackground().getRGB() == Color.RED.getRGB()) {
            board[4][0] = 0;
        } else if (_40.getBackground().getRGB() == Color.GREEN.getRGB()) {
            board[4][0] = 1;
        }
        checkWinner();
        switchPlayer();
    }
}
```

```
public void checkWinner() {
    int w = findWinner();
    if (w == 1) {
        JOptionPane.showMessageDialog(this, "Plater 1 (Red) wins.");
        finishGame();
        updateScore(w);
    } else if (w == 2) {
        JOptionPane.showMessageDialog(this, "Plater 2 (Green) wins.");
        finishGame();
        updateScore(w);
    }
}
```

```
public int findWinner() {
    int r, c, sum;
    // checking columns
    for(c=0;c<5;c++)
    {
        sum=0;
        for(r=0;r<4;r++)
        {
            sum=sum+board[r][c];
        }
        if(sum==0) return 1;
        else if(sum==4) return 2;
    }
    sum=0;
    for(r=1;r<5;r++)
    {
        sum=sum+board[r][c];
    }
}
```

```
        if(sum==0) return 1;
        else if(sum==4) return 2;
    }
    // check rows
    for(r=0;r<5;r++)
    {
        sum=0;
        for(c=0;c<4;c++)
        {
            sum=sum+board[r][c];
        }
        if(sum==0) return 1;
        else if(sum==4) return 2;
    }
    sum=0;
    for(c=1;c<5;c++)
    {
        sum=sum+board[r][c];
    }
    if(sum==0) return 1;
    else if(sum==4) return 2;
}

// diagonal 1
sum = 0;
for (r = 0; r < 4; r++) {
    sum = sum + board[r][r];
}
if (sum == 0) {
    return 1;
} else if (sum == 4) {
    return 2;
}

sum = 0;
for (r = 1; r < 5; r++) {
    sum = sum + board[r][r];
}
if (sum == 0) {
    return 1;
} else if (sum == 4) {
    return 2;
}

// diagonal 2
sum = 0;
c = 3;
for (r = 1; r < 5; r++) {
    sum = sum + board[r][c];
    c--;
}
if (sum == 0) {
    return 1;
} else if (sum == 4) {
    return 2;
}

sum = 0;
c = 4;
for (r = 0; r < 4 ; r++) {
    sum = sum + board[r][c];
    c--;
}
if (sum == 0) {
```



```
        return 1;
    } else if (sum == 4) {
        return 2;
    }
    // diagonal 3
    sum = 0;
    c = 3;
    for (r = 0; r < 4; r++) {
        sum = sum + board[r][c];
        c--;
    }
    if (sum == 0) {
        return 1;
    } else if (sum == 4) {
        return 2;
    }

    // diagonal 4
    sum = 0;
    c = 4;
    for (r = 1; r < 5; r++) {
        sum = sum + board[r][c];
        c--;
    }
    if (sum == 0) {
        return 1;
    } else if (sum == 4) {
        return 2;
    }
    return 0;
}
```

```
public void finishGame() {
    if (_00.getBackground().getRGB() == Color.WHITE.getRGB()) {
        _00.setBackground(Color.BLACK);
    }
    if (_01.getBackground().getRGB() == Color.WHITE.getRGB()) {
        _01.setBackground(Color.BLACK);
    }
    if (_02.getBackground().getRGB() == Color.WHITE.getRGB()) {
        _02.setBackground(Color.BLACK);
    }
}
```

```
public void updateScore(int w) {
    if (w == 1) {
        int score = Integer.parseInt(Player1.getText());
        score++;
        Player1.setText(Integer.toString(score));
    } else if (w == 2) {
        int score = Integer.parseInt(Player2.getText());
        score++;
        Player2.setText(Integer.toString(score));
    }
}
```

```
public void switchPlayer() {
    if (Player.getBackground().getRGB() == Color.GREEN.getRGB()) {
        Player.setBackground(Color.red);
    } else if (Player.getBackground().getRGB() == Color.RED.getRGB()) {
        Player.setBackground(Color.GREEN);
    }
}
```



```
    }  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    for (int r = 0; r < 5; r++) {  
        for (int c = 0; c < 5; c++) {  
            board[r][c] = 10;  
        }  
    }  
}
```