

Fundamentals of Programming II Assignment 03

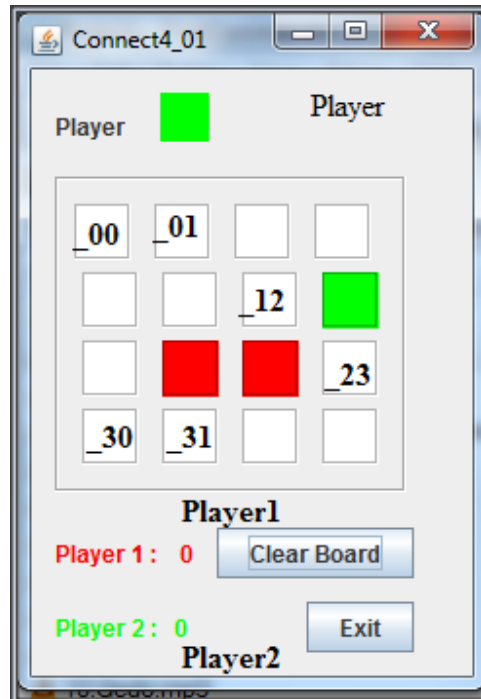
4X4 Connect4 – I

Connect4 - I

Build 4x4 connect 4 board.

Use green and red colors to present users.

Use matrix to presents the board status, to improve “checkWinner” method



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

```
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);
    }
}
```

```
public int findWinner() {
    int r, c, sum;
    // checking rows
    for (r = 0; r < 4; 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;
        }
    }
    // check columns
    for (c = 0; c < 4; 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;
        }
    }
    // 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;
    }
    // diagonal 2
    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;
    }

    return 0;
}

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 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 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);
    }
    if (_03.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _03.setBackground(Color.BLACK);
    }
    if (_10.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _10.setBackground(Color.BLACK);
    }
    if (_11.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _11.setBackground(Color.BLACK);
    }
    if (_12.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _12.setBackground(Color.BLACK);
    }
}
```



```
    }
    if (_13.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _13.setBackground(Color.BLACK);
    }
    if (_20.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _20.setBackground(Color.BLACK);
    }
    if (_21.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _21.setBackground(Color.BLACK);
    }
    if (_22.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _22.setBackground(Color.BLACK);
    }
    if (_23.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _23.setBackground(Color.BLACK);
    }
    if (_30.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _30.setBackground(Color.BLACK);
    }
    if (_31.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _31.setBackground(Color.BLACK);
    }
    if (_32.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _32.setBackground(Color.BLACK);
    }
    if (_33.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _33.setBackground(Color.BLACK);
    }
    for (int r = 0; r < 4; r++) {
        for (int c = 0; c < 4; c++) {
            board[r][c] = 10;
        }
    }
}
```

```
private void _00MouseClicked(java.awt.event.MouseEvent evt)
{
    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 _33MouseClicked(java.awt.event.MouseEvent evt) {
    if (_33.getBackground().getRGB() == Color.WHITE.getRGB())
    {
        _33.setBackground(Player.getBackground());
        if (_33.getBackground().getRGB() ==
Color.RED.getRGB()) {
            board[3][3] = 0;
        } else if (_33.getBackground().getRGB() ==
Color.GREEN.getRGB()) {
            board[3][3] = 1;
        }
        checkWinner();
        switchPlayer();
    }
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
    _00.setBackground(Color.WHITE);
    _01.setBackground(Color.WHITE);
    _02.setBackground(Color.WHITE);
    _03.setBackground(Color.WHITE);
    _10.setBackground(Color.WHITE);
    _11.setBackground(Color.WHITE);
    _12.setBackground(Color.WHITE);
    _13.setBackground(Color.WHITE);
    _20.setBackground(Color.WHITE);
    _21.setBackground(Color.WHITE);
    _22.setBackground(Color.WHITE);
    _23.setBackground(Color.WHITE);
    _30.setBackground(Color.WHITE);
    _31.setBackground(Color.WHITE);
    _32.setBackground(Color.WHITE);
    _33.setBackground(Color.WHITE);
    Player.setBackground(Color.RED);
}
}
```



جامعة الأهرام الكندية
AHRAM CANADIAN UNIVERSITY

```
FIRST:event_jButton2ActionPerformed  
System.exit(1); }
```