

TicTacToe Project

ผู้จัดทำ

นายณัฐนันท์ วงษ์ศิริ รหัสนิสิต 6530300791

เสนอ

อาจารย์ผู้สอน ผศ.ดร.กุลวดี สมบูรณ์วิวัฒน์

วิชา 03603112 **Programming Fundamentals II**

ภาคเรียนที่ 2 ปีการศึกษา 2565

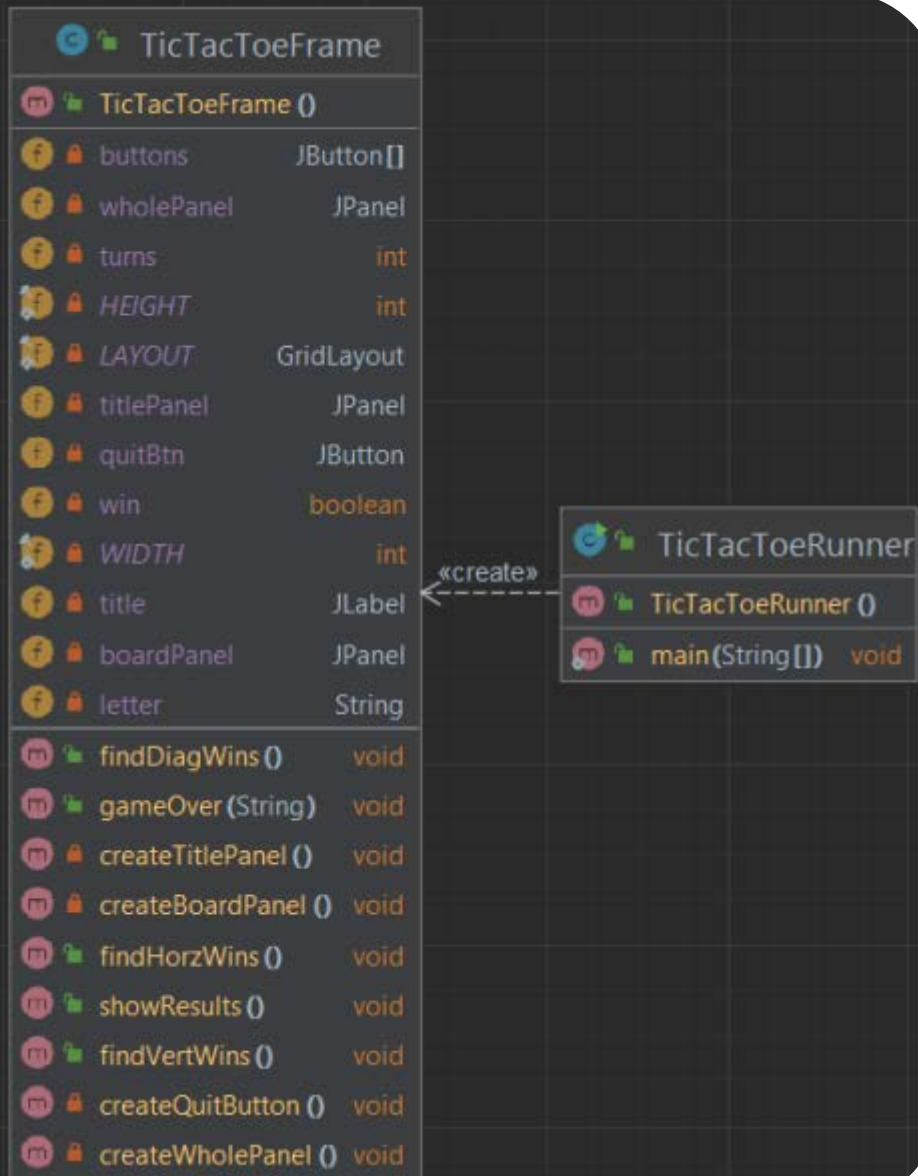
เหตุผลที่ทำ Project

Tic-tac-toe เป็นเกมที่เรียกว่าเป็นเกมเล่นกันทั่วไปที่ไม่ยุ่งยากและไม่ต้องใช้เวลานานในการเรียนรู้กฎเกม โดยเกมนี้เหมาะสำหรับทุกวัยทั้งเด็กและผู้ใหญ่ ซึ่งสามารถเล่นได้ทั้งเดี่ยวและเล่นกับเพื่อนๆ โดยไม่จำเป็นต้องมีอุปกรณ์หรือเครื่องมือเสริมเพิ่มเติมเลย

การทำโปรเจค **Tic-tac-toe** ยังช่วยให้ผู้เรียนได้ฝึกทักษะการเขียนโปรแกรมแบบพื้นฐาน และทักษะการสร้างโปรแกรมเกมง่ายๆ โดยโปรเจคนี้จะช่วยให้ผู้เรียนได้เรียนรู้หลักการพื้นฐานของการเขียนโปรแกรมเกม และการใช้งานตัวแปร การใช้งานเงื่อนไข การใช้งานลูป การใช้งานฟังก์ชัน และการใช้งานคำสั่งพื้นฐานอื่นๆ ซึ่งเป็นสิ่งที่สำคัญในการพัฒนาซอฟต์แวร์และการเขียนโปรแกรมทั่วไป

นอกจากนี้ การทำโปรเจค **Tic-tac-toe** ยังสามารถสร้างความสนุกสนานและความบันเทิงให้กับผู้เล่นได้อีกด้วย ซึ่งจะช่วยส่งเสริมทักษะการวางแผนและการคิดเชิงตรรกะในผู้เล่นอีกด้วย

Class Diagramme



```
public TicTacToeFrame()
{
    createQuitButton();
    createTitlePanel();
    createBoardPanel();
    createWholePanel();
    findHorzWins();
    findVertWins();
    findDiagWins();
    showResults();

    //รับความกว้างของหน้าจอ
    Toolkit kit = Toolkit.getDefaultToolkit();
    Dimension screenSize = kit.getScreenSize();
    int screenHeight = screenSize.height;
    int screenWidth = screenSize.width;

    //กำหนดขนาดหน้าจอเป็นครึ่งหนึ่งของความสูงและความกว้างของหน้าจอ และ จัดกึ่งกลาง
    setSize(WIDTH, HEIGHT);
    setLocation(screenWidth / 4, screenHeight / 4);
}
```

การออกแบบ UI

ใน **TicTacToeFrame()**

จะมีการเรียกใช้ทุก **methods** ใน **Class TicTacToeFrame**

ใช้ **Toolkit** เพื่อผูกคอมโพเนนต์

ใช้ **Dimension** เพื่อความแม่นยำในการรับความกว้างและความสูง

การออกแบบ UI

```
private void createQuitButton() //สร้างปุ่มกดออกจากหน้าต่าง UI
{
    quitBtn = new JButton(text:"-- Quit --");
    quitBtn.setFont(new Font(Font.SERIF, style:0, size:24));

    class QuitListener implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent ae) //เมื่อกดแล้วให้ exit จากหน้าต่าง UI
        {
            System.exit(status:0);
        }
    }

    ActionListener quitListener = new QuitListener();//add quitListener ไปใน quitBtn เพื่อไปใช้ต่อใน createWholePanel();
    quitBtn.addActionListener(quitListener);
}
```

ผมอยากมีปุ่มไว้กดออกจากหน้าต่าง UI เมื่อไม่อยากจะเล่นกลางคัน
ฉะนั้นใน **methods** นี้ ผมจะสร้างปุ่มที่มี **Text** ชื่อว่า **--Quit--**
โดย สร้าง **class QuitListener** ที่ **implements** กับ **ActionListener** เพื่อให้เมื่อกดปุ่มเกิด **ActionEvent**
โดยเมื่อกดปุ่มจะเกิด **Runtime Exit** และ สร้าง **Object**
เพื่อ **add quitListener** ไปใน **quitBtn** เพื่อไปใช้ต่อใน **createWholePanel()**;

การออกแบบ UI

```
private void createTitlePanel() //สร้าง คำว่า Welcome to my Tic Tac Toe Game! โดยที่ setFont style size
{
    title = new JLabel(text:"Welcome to my Tic Tac Toe Game!");
    titlePanel = new JPanel();
    title.setFont(new Font(Font.SERIF, style:0, size:30));
    titlePanel.add(title);
}
```

Methods นี้ผมสร้างขึ้นมาเพื่อกล่าวต้อนรับคนที่มาเล่นเกมผม
โดยสร้าง **Jlabel** ข้างใน **text** พิมพ์ว่าต้อนรับ
setFont อะไรให้เรียบร้อยแล้วก็เอา **title** ที่ได้ **add** เข้าไปใน
titlePanel เพื่อไปใช้ต่อใน **createWholePanel()**;

การออกแบบ UI

Methods นี้ผมสร้างขึ้นมาเพื่อสร้าง **Button** ในการเล่นเกม **TicTacToe** โดยเริ่มแรกผม **setLayout** ไปใน **boardPanel** ที่มีขนาด **3x3** และสร้าง **class ButtonListener** ที่ **implements** กับ **ActionListener** เพื่อให้เมื่อกดปุ่มเกิด **ActionEvent** เมื่อกดปุ่มให้ **setText** ไปใน **letter** โดยที่ **letter** รับค่าเป็น **String** ทุก **turns** ที่หาร **2** ลงตัว **letter** จะได้ค่า **O** ถ้าไม่ได้หาร **2** ลงตัว ก็จะเป็น **X** ใช้ **for loop** เพื่อสร้างปุ่มจำนวน **9** ปุ่ม **add buttonListener** ลงใน **button[i]** และ **add button[i]** ลงใน **boardPanel** เพื่อไปใช้ต่อไปใน **createWholePanel()**;

```
private void createBoardPanel() //สร้าง board TicTacToe
{
    boardPanel = new JPanel();
    boardPanel.setLayout(LAYOUT);

    class ButtonListener implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent ae)
        {
            showResults();
            turns++;
            if(turns % 2 == 0)
            {
                letter = "O";
            }
            else {
                letter = "X";
            }

            JButton pressedButton = (JButton)ae.getSource();
            pressedButton.setText(letter);           //ให้ปุ่มรับค่า letter ได้จากการกดปุ่ม ทุก turns ที่หาร 2 ลงตัวจะเป็น O ถ้าไม่ใช่ก็ X
            pressedButton.setEnabled(!pressedButton.isEnabled());
        }
    }

    ActionListener buttonListener = new ButtonListener(); //สร้าง Object new ButtonListener(); ใช้ for loop เพื่อสร้างปุ่มจำนวน 9 ปุ่ม
    for(int i=1; i<=9; i++) //add buttonListener ลงใน button[i] และ add button[i] ลงใน boardPanel เพื่อไปใช้ต่อไปใน createWholePanel();
    {
        buttons[i] = new JButton(text:"");
        buttons[i].setFont(new Font(Font.SERIF, style:0, size:24));
        buttons[i].addActionListener(buttonListener);
        boardPanel.add(buttons[i]);
    }
}
```

การออกแบบ UI

Methods นี้ผมสร้างขึ้นมาเพื่อจัดวางตำแหน่ง **methods** ต่างๆ ลงในหน้า **UI** สร้างโครงร่างเส้นขอบใหม่โดยไม่มีช่องว่างระหว่างส่วนประกอบ ตง **new BorderLayout** และ **add titlePanel** ไว้บนสุด **add bordPanel** ไว้ตรงกลาง **add quitBtn** ไว้ล่างสุด

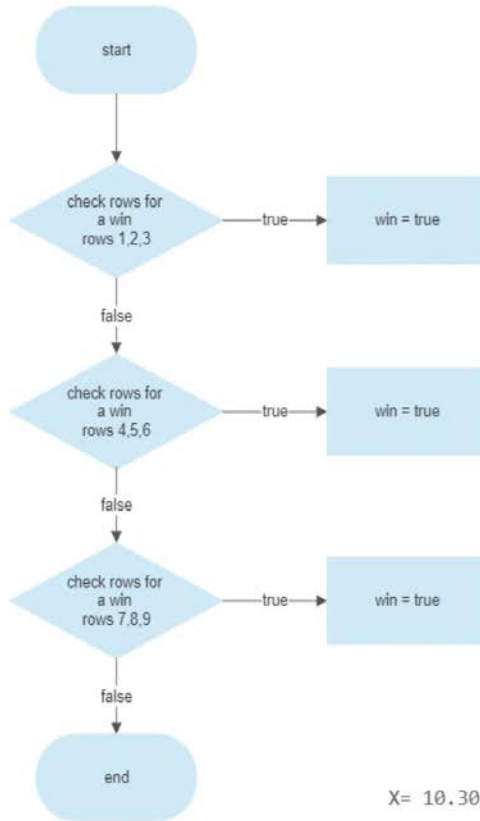
```
private void createWholePanel() //จัดเรียงหน้า UI setLayout สร้าง Object new BorderLayout
{
    wholePanel = new JPanel();
    wholePanel.setLayout(new BorderLayout());
    wholePanel.add(titlePanel, BorderLayout.NORTH);
    wholePanel.add(boardPanel, BorderLayout.CENTER);
    wholePanel.add(quitBtn, BorderLayout.SOUTH);
    add(wholePanel);
}
```




หน้าต่าง UI
TicTacToe game

Flowchart : findHorzWins();

1	2	3
4	5	6
7	8	9



X= 10.30

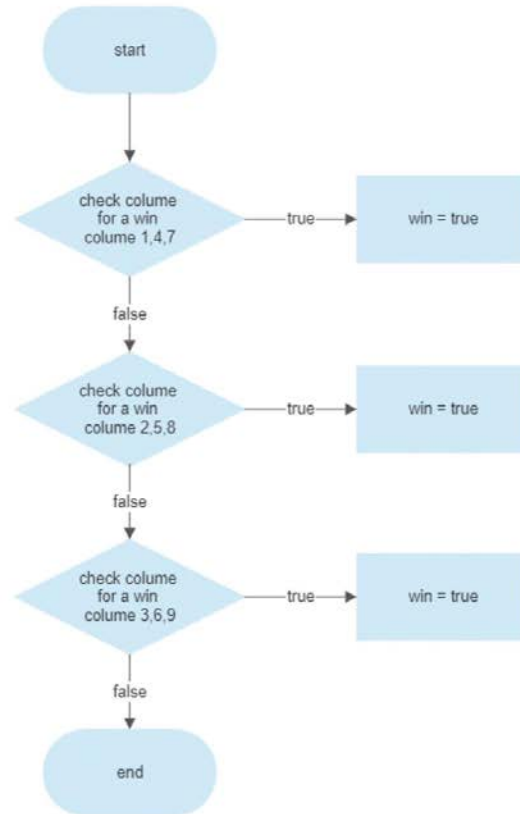
```
public void findHorzWins() //checkwin ตามแถว
{
    if (buttons[1].getText().equals(buttons[2].getText()) && buttons[2].getText().equals(buttons[3].getText()) && buttons[1].getText().equals(anObject:"")!=false)
    {
        win=true; //check ตามแถวที่ 1,2,3 ตามระดับ buttons.getText() เปรียบเทียบ buttons.getText()
        //equals มีparameter anObject วัตถุจะเปรียบเทียบกัน String(getText) ,!= false return เป็นอย่างอื่น win=true
    }

    else if (buttons[4].getText().equals(buttons[5].getText()) && buttons[5].getText().equals(buttons[6].getText()) && buttons[4].getText().equals(anObject:"")!=false)
    {
        win=true;
    }

    else if (buttons[7].getText().equals(buttons[8].getText()) && buttons[8].getText().equals(buttons[9].getText()) && buttons[7].getText().equals(anObject:"")!=false)
    {
        win=true;
    }
}
```

Flowchart : findVertWins();

1	2	3
4	5	6
7	8	9



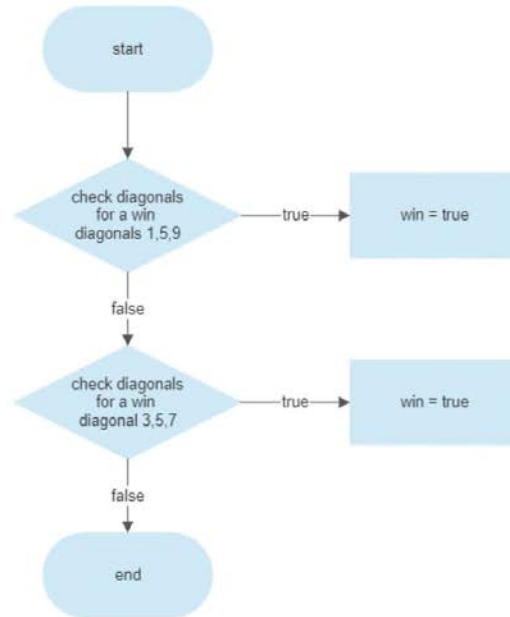
```
public void findVertWins()//checkwin ตามคอลัมน์
{
    if (buttons[1].getText().equals(buttons[4].getText()) && buttons[4].getText().equals(buttons[7].getText()) && buttons[1].getText().equals(anObject:"")--false)
    {
        win=true;//check ตามคอลัมน์ที่ 1,4,7 ตามระดับ buttons.getText() เปรียบเทียบ buttons.getText()
        //equals มีparameter anObject วัดค่าจะเปรียบเทียบกับ String(getText) ,== false return เป็นอย่างอื่น win=true
    }

    else if (buttons[2].getText().equals(buttons[5].getText()) && buttons[5].getText().equals(buttons[8].getText()) && buttons[2].getText().equals(anObject:"")--false)
    {
        win=true;
    }

    else if (buttons[3].getText().equals(buttons[6].getText()) && buttons[6].getText().equals(buttons[9].getText()) && buttons[3].getText().equals(anObject:"")--false)
    {
        win=true;
    }
}
```

Flowchart : findDiagWins();

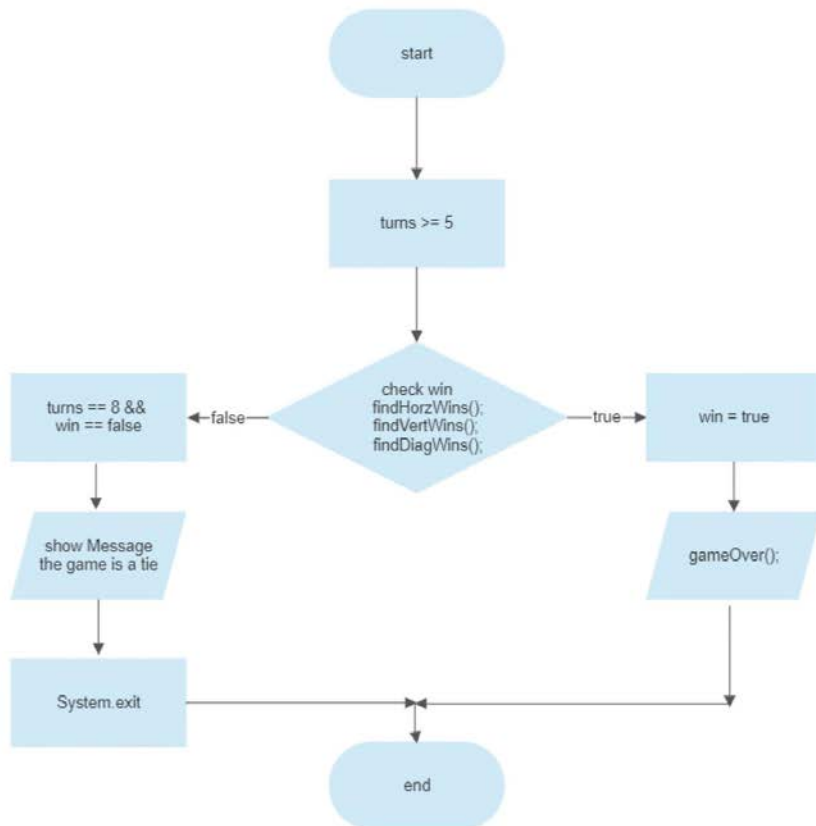
1	2	3
4	5	6
7	8	9



```
public void findDiagWins() //checkwin ตามแนวแฉงมุม
{
    if (buttons[1].getText().equals(buttons[5].getText()) && buttons[5].getText().equals(buttons[9].getText()) && buttons[1].getText().equals(anObject:"")!=false)
    {
        win=true;//check ตามแนวแฉงมุมที่ 1,5,9 ตามระดับ buttons.getText() เปรียบเทียบ buttons.getText()
        //equals มีparameter anObject วัดผลจะเปรียบเทียบกัน String(getText) ,== false return เป็นอย่างอื่น win=true
    }

    else if (buttons[3].getText().equals(buttons[5].getText()) && buttons[5].getText().equals(buttons[7].getText()) && buttons[3].getText().equals(anObject:"")!=false)
    {
        win=true;
    }
}
```

Flowchart : showResults();

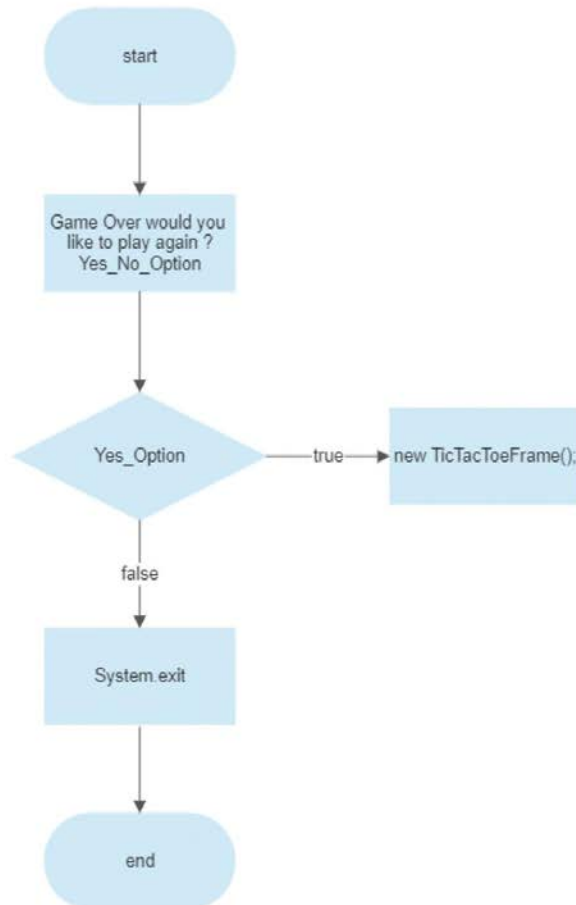


```
public void showResults() //methods นี้ใช้เช็ค turns
{
    if(turns>=5) //เมื่อผ่านไประยะ 5 turns ให้ check methodsตามด้านล่าง
    {
        findHorzWins();
        findVertWins();
        findDiagWins();

        if (win==true) //ถ้า win จริงให้โชว์ การทำงาน gameOver();
        {
            gameOver();
        }

        else if (turns==8 && win==false) //ถ้า turns = 8 หรือ ไม่มี win เลย ให้ showMessage
        {
            JOptionPane.showMessageDialog(parentComponent:null, message:"The game is a tie."); //parentComponent กำหนดเฟรมที่แสดงไดอะล็อก
            System.exit(status:0); //ถ้า null หรือถ้า parentComponent ไม่มี Frame จะใช้ Frame เริ่มต้น
        }
    }
}
```

Flowchart : gameOver();



```
public void gameOver() { //methods นี้ให้ ShowOption YES_NO_OPTION
    JOptionPane pane = new JOptionPane();
    int dialogResult = JOptionPane.showConfirmDialog(pane, "Game Over. " + letter + " wins. Would you like to play again?", "Game over.",
        JOptionPane.YES_NO_OPTION);

    if(dialogResult == JOptionPane.YES_OPTION) { //ถ้าเลือก Yes Option ให้ทำงาน new TicTacToeFrame();
        this.dispose(); // close the current frame
        TicTacToeFrame newGame = new TicTacToeFrame();
        newGame.setVisible(b:true);
    } else { //ถ้าเลือก No Option ให้ Runtime exit
        System.exit(status:0);
    }
}
```

Yotube

GitHub

Thank
You