# SUDOKU

Make by Java

# SUDOKU

เป็นเกมแนว

PUZZLE LOGICAL

- จุดประสงค์ของเกม

  คือ เกมสำหรับผู้สูงอายุ เล่นง่าย แต่สามารถฝึกทักษะและพัฒนาทักษะได้

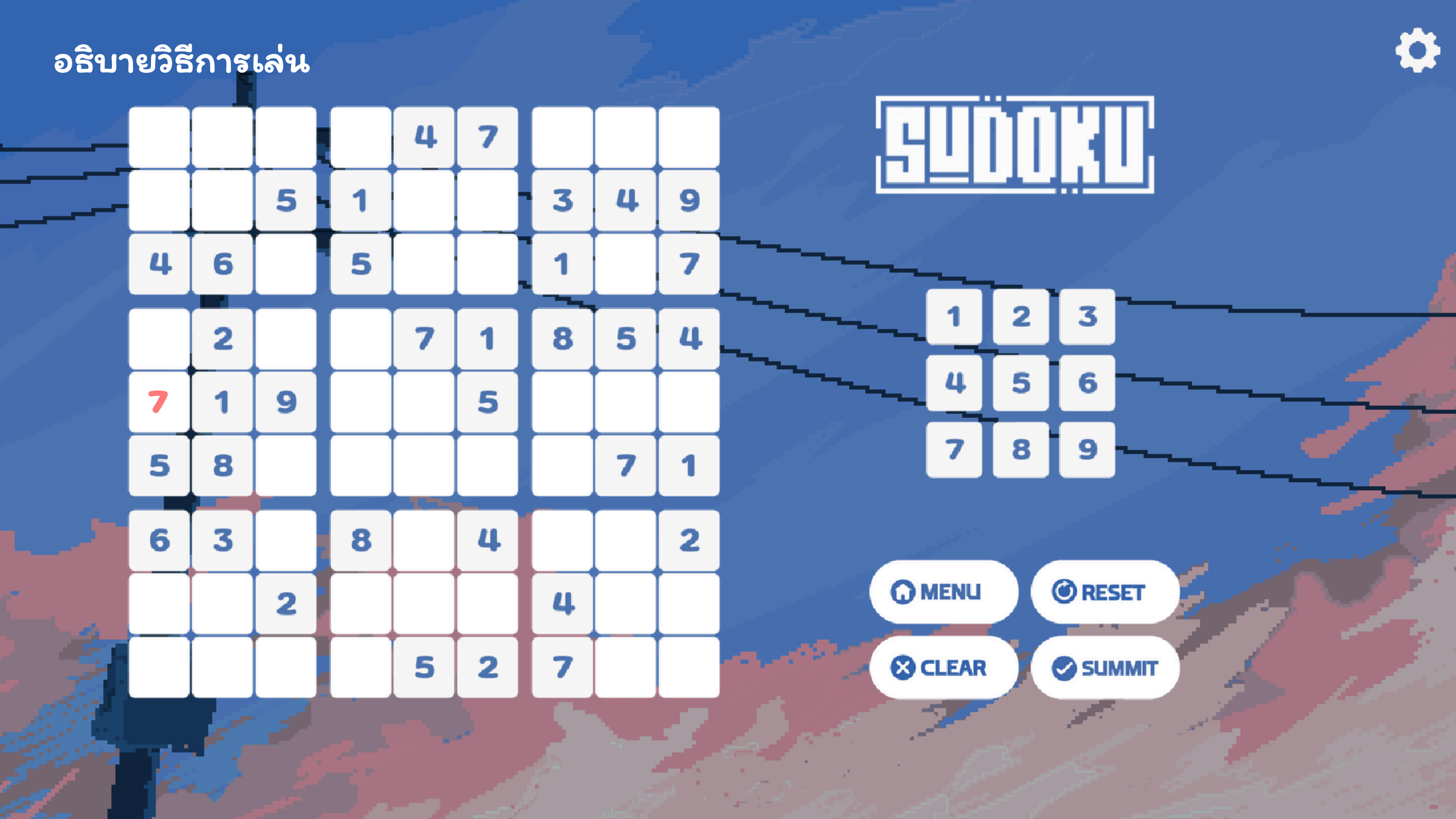
- สิ่งที่คาดว่าผู้เล่นจะได้จากการเล่น

  ปรับปรุงการใช้เหตุผลเชิงตรรกะ

  พัฒนาทักษะการตัดสินใจ

  ช่วยเพิ่มความสามารถในการจดจำข้อมูล

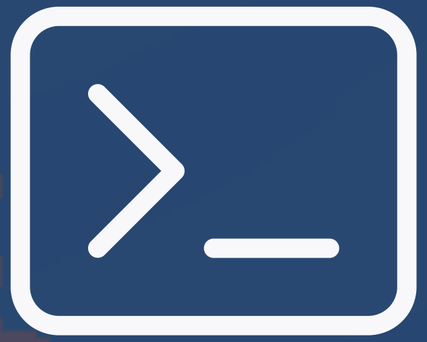
# อธิบายวิธีการเล่น



# ตัวเกมจะมีส่วนการทำงานใหญ่ๆอยู่ 3 ส่วนหลักๆ

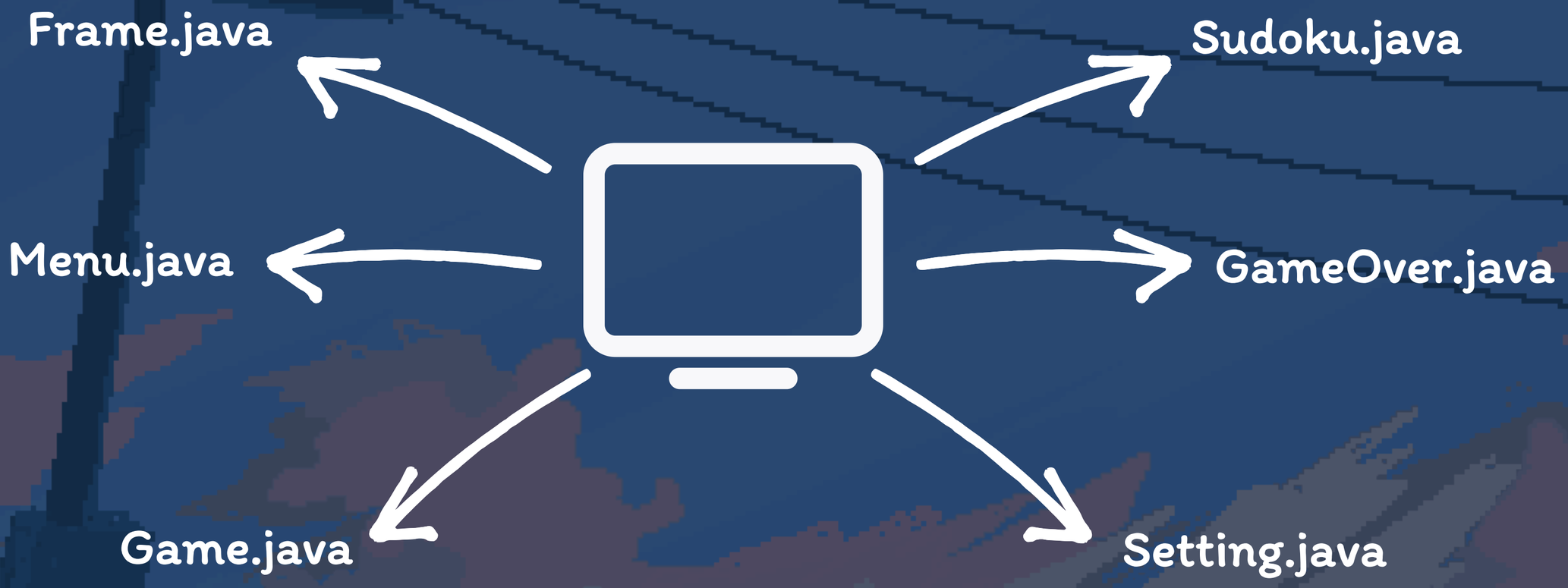
ส่วนที่ 1 Display

ส่วนที่ 2 Controller



ส่วนที่ 3 Generator

# ส่วนที่ 1 Display



# Sudoku.java

- สร้าง class Sudoku เพื่อเป็น class main ในการrun โปรแกรมเกม

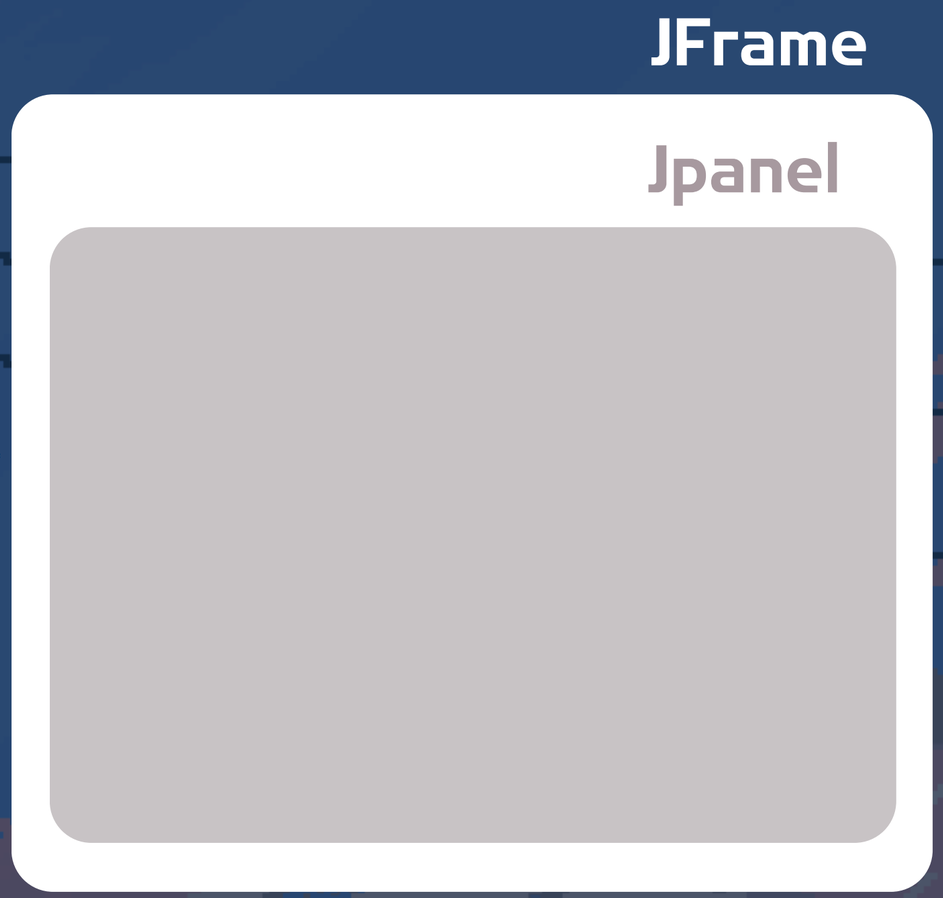
```java
public class Sudoku {

    Run | Debug
    public static void main(String[] args) {

        Frame game = new Frame();
    }

}
```

- โดยสร้าง Object ใหม่จาก class Frame ให้ถูกเก็บในตัวแปร game

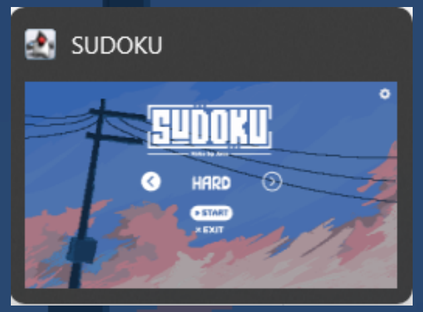


# Frame.java

- สร้าง class ที่ชื่อว่า Frame แล้วให้ extend JFrame เพื่อให้เป็นส่วนที่แสดงผลต่างๆ บนหน้าเกม จาก class Menu โดยใช้ .getContentPane เพื่อดึงค่าลงบน Frame

JFrame

- ทำการ .setTitle เพื่อตั้งชื่อให้กับหน้าต่าง JFrame




```
this.getContentPane().add(menu);
```

- สร้าง Icon ให้กับหน้าต่าง JFrame โดยใช้ .setIconImage และดึงรูปที่ต้องการ ด้วย new ImageIcon จาก Folder ที่เก็บไว้

```
    this.setTitle(title:"SUDOKU");
    this.setIconImage(new ImageIcon(filename:"src\\Gameicon.png").getImage());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setUndecorated(undecorated:true);
    this.setResizable(resizable:false);
    this.pack();
    this.setVisible(b:true);
}
```

- กำหนดให้โค้ด “หยุดทำงาน” เมื่อเราปิด เกม/JFrame

# Frame.java

JFrame

- ตั้งค่าให้ Frame ไม่มี จึงใช้ .setUndecorated เพื่อต้องการเพื่อทำให้ ตัว Frame เต็มหน้าจอ

- .setResizable เป็น false เพื่อไม่ให้ User สามารลดขนาดหน้าต่างของFrameได้

```
this.getContentPane().add(menu);
```

```
this.setTitle(title:"SUDOKU");
this.setIconImage(new ImageIcon(filename:"src\\Gameicon.png").getImage());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setUndecorated(undecorated:true);
this.setResizable(resizable:false);
this.pack();
this.setVisible(b:true);
```

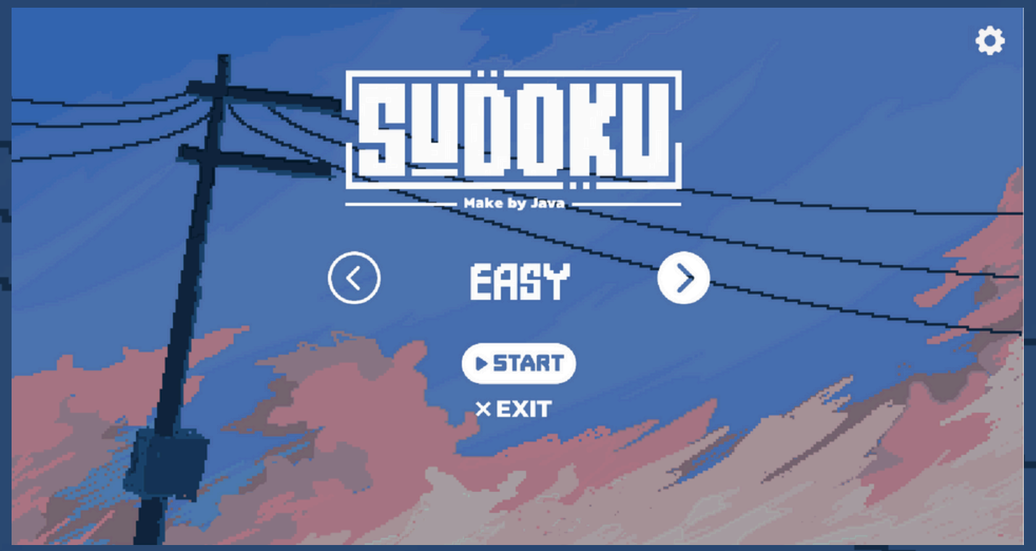
- .pack

- set .setVisible เป็น true เพื่อให้Frame แสดงออกมาทางหน้าจอ

# Menu.java

JPanel

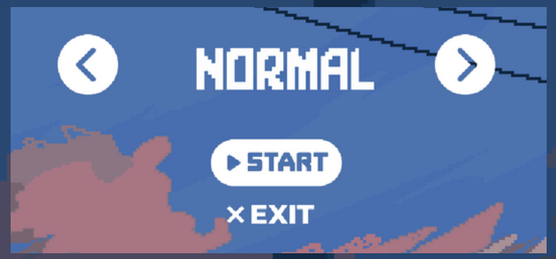
- สร้าง class ที่มีชื่อว่า Menu ขึ้นมา แล้วทำการ extend Jpanel เพื่อให้ง่ายต่อการอ่านและนำไปใช้บนFrame




height

width

- ทำการ implements ActionListener กับ class Menu เพื่อให้เกิด Action กับปุ่มที่มี



- กำหนดเลขไว้เป็น attribute เพื่อมาใช้กำหนดขนาดของ Panel ใน constructor

```
final int SCREEN_WIDTH = 1535;
final int SCREEN_HEIGHT = 850;
```

```
Menu(Frame t) {
    this.setPreferredSize(new Dimension(SCREEN_WIDTH, SCREEN_HEIGHT));
    this.setFocusable(focusable:true);
```

- ตั้งค่า Layout ใน Panel เป็น null เพื่อง่ายต่อการจัดวางองค์ประกอบต่างๆ ด้วยพิกัด

```
this.setLayout(mgr:null);
```

- เพื่อให้ Panel สามารถรับข้อมูลจากการคลิ๊กได้ โดย setFocusable เป็น true

SUDOKU
Make by Java
NORMAL
START
EXIT

**MENU** ประกอบไปด้วย 3 ตัวหลักๆ

1. JLabel ( สำหรับใส่รูป )
2. JPanel
3. JButton ( สำหรับใส่ปุ่ม )

JLabel ( LOGO)

JButton ( SettingBtn)

JLabel ( Mode)

JButton ( Minus_ModeBtn)

JButton ( Plus_ModeBtn)

JButton ( StartBtn)

JButton ( ExitBtn)

# SUDOKU

Make by Java

```java
    private JButton StartBtn;
    private JButton modeBtn_plus;
    private JButton modeBtn_minus;
    private JButton ExitBtn;
    private JLabel Logo;
    private JLabel Background;
    private JLabel setMode;

```

- เราสร้างตัว JLabel ขึ้น มาโดย ให้ชื่อมันคือ “ Logo “

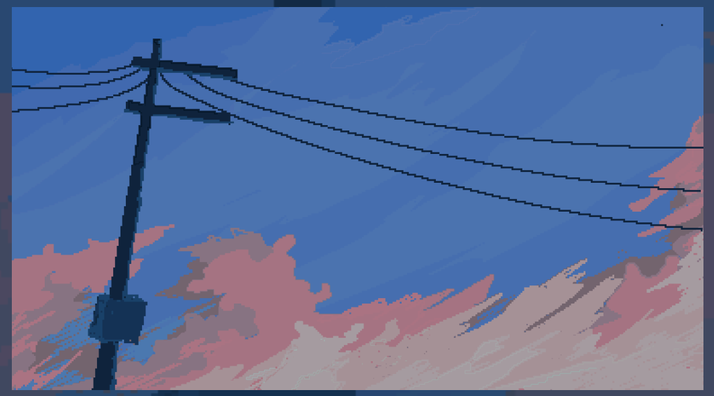
```java
Logo = new JLabel(new ImageIcon(filename:"src\\GameLogo2.png"));

final int LOGO_X = (460);
final int LOGO_Y = (-100);

Logo.setBounds(LOGO_X, LOGO_Y, width:600, height:600);
this.add(Logo);
```

- เรียกตัว JLabel “ setMode” มาใส่ภาพเข้าไปด้วย คำสั่ง new ImageIcon () โดยดึงรูปจาก Folder “ src “
- setBound ตั้งตำแหน่ง x y ตามที่ต้องการ และ ตั้งค่า width กับ height
- add ตัว JLabel ตัวนี่ลงไปใน JPanel

height



width

```java
    private JLabel Logo;
    private JLabel Background;
    private JLabel setMode;

```

- ตัว Background ทำคลายกับตัว Logo แต่เราสร้างตัวของ method “ BackgroundMode “ มาเพื่อเก็บข้อมูลแทนที่จะ ใส่ไว้ในตัว Start เหมือน Logo

```java
// Background
private void Backgroundmode() {

    Background = new JLabel(new ImageIcon(filename:"src\\BackgroundMenu.png"), JLabel.CENTER);
    Background.setBounds(x:0, y:0, SCREEN_WIDTH, SCREEN_HEIGHT);
    this.add(Background);
}
```

```java
    private JButton StartBtn;
    private JButton modeBtn_plus;
    private JButton modeBtn_minus;
    private JButton ExitBtn;
    private JLabel Logo;
    private JLabel Background;
    private JLabel setMode;

```

- เราสร้างตัว JLabel ขึ้น มาโดยให้ชื่อมันคือ “ setMode “

```java
protected int mode = 1;
boolean MusicPlay = false;
boolean MusicPath;
```

- สร้างตัว mode ขึ้นมาเพื่อให้เก็บค่า mode ของเกมไว้โดยค่าเริ่มต้นให้เริ่มที่ 1 ( Easy )

```java
    // Mode
    private void Gamemode() {
        setMode = new JLabel(new ImageIcon(filename:"src\\ModeEasy.png"));
        setMode.setBounds(x:570, y:320, width:400, height:200);
        this.add(setMode);
    }
```

- สร้างตัว method Gamemode()
  - เรียกตัว JLabel “ setMode” มาใส่ภาพเข้าไปด้วยคำสั่ง new ImageIcon () โดยดึงรูปจาก Folder “ src “
  - setBound ตั้งตำแหน่ง x y ตามที่ต้องการ และ ตั้งค่า width กับ height
  - โดยตอนนี้ เรา extends Class นี่ให้เป็น JPanel อยู่เราจึงใช้ this. ไปเลย แล้ว add ตัว JLabel ตัวนี่ลงไปใน JPanel

```java
// Change Mode setMode
if (mode == 1) {
    setMode.setIcon(new ImageIcon(getClass().getResource(name:"src\\ModeEasy.png")));
    setMode.setBounds(x:570, y:320, width:400, height:200);
} else if (mode == 2) {
    setMode.setIcon(new ImageIcon(getClass().getResource(name:"src\\ModeNormal.png")));
    setMode.setBounds(x:580, y:320, width:400, height:200);
} else {
    setMode.setIcon(new ImageIcon(getClass().getResource(name:"src\\ModeHard.png")));
    setMode.setBounds(x:570, y:320, width:400, height:200);
}
```

- Action ที่จะเกิด

ถ้าค่า mode == 1 ให้ Image ใน setMode new รูปใหม่ตามค่า ที่ set ไว้ โดยดึงรูปจาก Folder “ src “

# Setting side

## JButton ( SettingBtn)

```
// Setting
SettingBtn = new JButton(new ImageIcon(filename:"btn\\SettingBtn.png"));
SettingBtn.setBorderPainted(b:false);
SettingBtn.setContentAreaFilled(b:false);
SettingBtn.setBounds(this.gameContext.getScreenWidth() - 80, y:20, width:60, height:60);
SettingBtn.addActionListener(this);
this.add(SettingBtn);
```

- สร้างตัวปุ่มขึ้นมา และดึงรูปมาใส่จาก Folder “ btn ” (SettingBtn.png)
- ตั้ง .setBorderPainted ให้เป็น False เพื่อ ซ่อนตัวกรอบของปุ่ม JButton
- ตั้ง .setContentAreaFilled ให้เป็น False เพื่อ ซ่อนตัวพื้นหลังของปุ่ม JButton

```
@Override
public void actionPerformed(ActionEvent e) {
```

- ใส่ .addActionListener เพื่อเพิ่ม Action ให้กับตัวของปุ่ม โดยเราจะไปเขียน Action ในตัวของ @Override actionPerfromed ( ActionEvent e )

```
if (e.getSource() == SettingBtn) {
    gameContext.toSetting();
}
```

- Action ที่จะเกิด
  ถ้ากด ปุ่น SettingBtn ตัว ActionEvent e ถ้าเกิดการรับค่า SettingBtn ให้เปิดตัว Setting page ขึ้นมา

# ⚙ SETTING

- ประกอบไปด้วย ตัว ปุ่ม ปิดเพลง - เปิดเพลง
  ปุ่ม Close setting

## AUDIO

| MUSIC | ON |
|---|---|

- On กับ Off เพื่อ เปิด / ปิด เพลง

- ปุ่ม “ CLOSE SETTING ” เพื่อปิดหน้า Setting page

✕ CLOSE SETTING

# Setting side

JButton ( SettingBtn)

✕ CLOSE SETTING

```
this.setLayout(mgr:null);
CloseSettingBtn = new JButton(new ImageIcon(filename:"btn\\CloseSettingBtn.png"));
CloseSettingBtn.setBounds(x:620, y:680, width:300, height:100);
CloseSettingBtn.setBorderPainted(b:false);
CloseSettingBtn.setContentAreaFilled(b:false);
CloseSettingBtn.addActionListener(this);
this.add(CloseSettingBtn);
```

```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == CloseSettingBtn) {

        gameContext.toPreventPage();
        System.out.println(x:"CloseSetting");
```

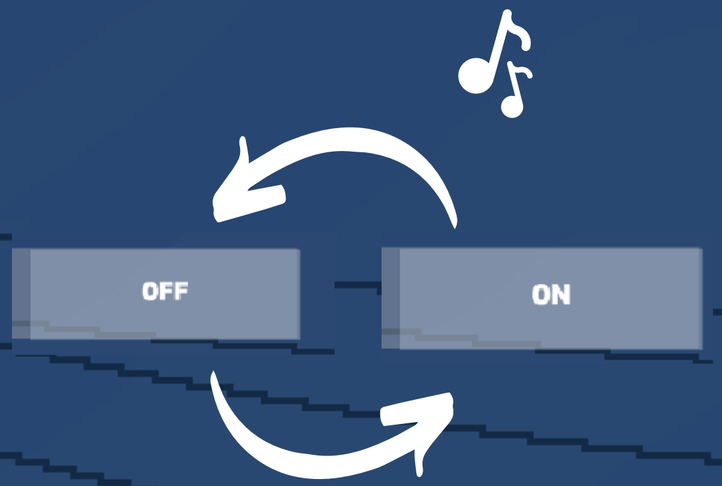
- ปุ่ม “ CLOSE SETTING ” เมื่อกดจะเรียกตัว toPreventPage (เราเก็บค่าไว้ในตัวนี้) จากตัวของ Class “ gameContext ” เพื่อเอาค่าที่ Save ไว้มา และ run ตัว menu ด้วยค่าที่เอามา เพื่อให้ ตัวที่เลือกไว้ยังเหมือนเดิม

Ex. สมมุติถ้าเราเลือก Mode Hard ไว้แล้วกดหน้า setting ค่าจะเอาไปเก็บไว้ใน gameContext และหลังจากที่กดปิดหน้า setting ตัวของ gameContext จะส่งค่ากลับคืนไปที่ menu เพื่อให้ยังคงอยู่ที่โหมด Hard เหมือนเดิม

# Music Button JButton ( toggleMusicBtn)

```
toggleMusic = new JButton(new ImageIcon(filename:"btn\\Open_Off.png"));
toggleMusic.setBounds(x:1047, y:218, width:320, height:100);
toggleMusic.setBorderPainted(b:false);
toggleMusic.setContentAreaFilled(b:false);
toggleMusic.addActionListener(this);

this.add(toggleMusic);
```



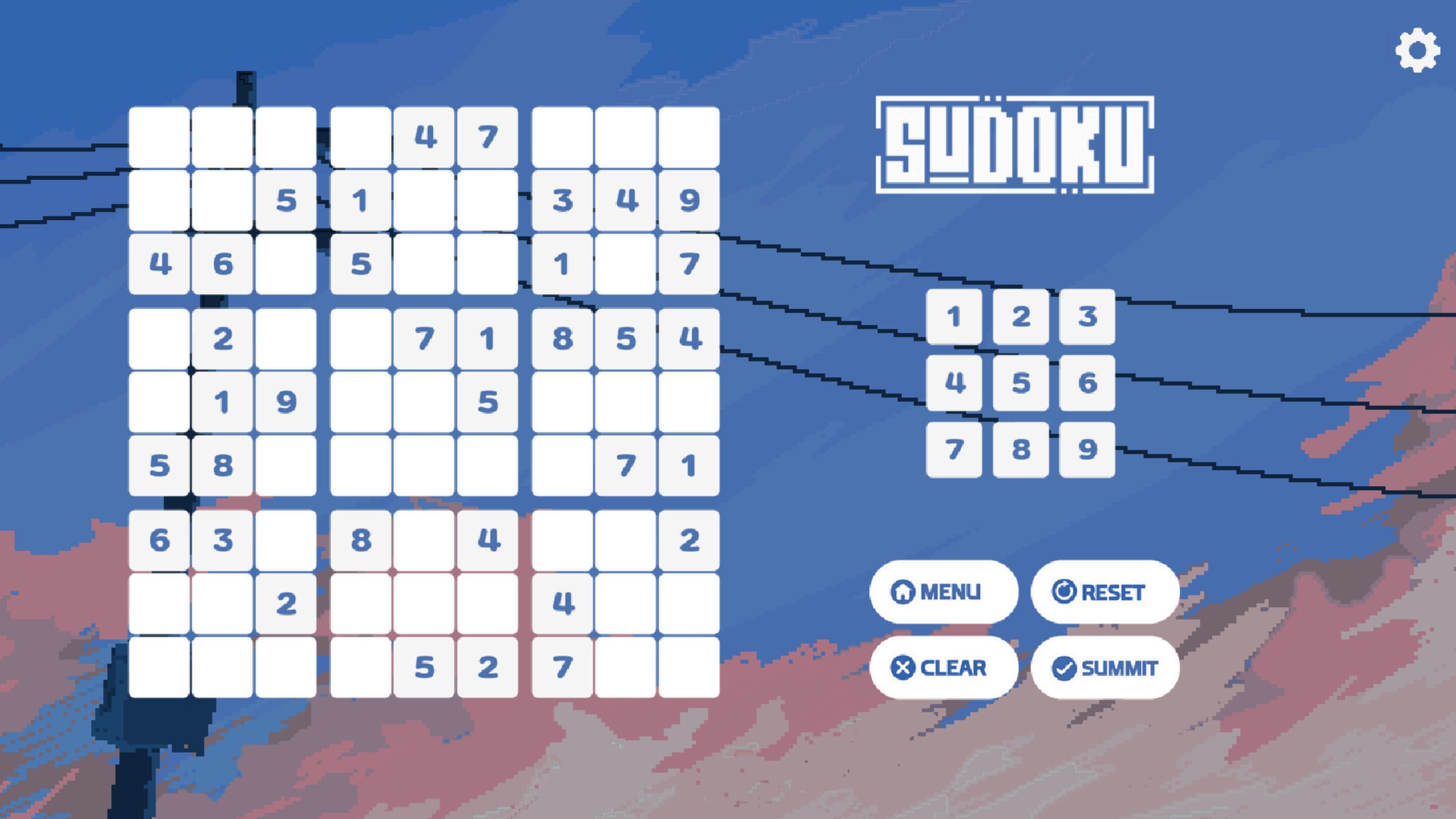

```
if (!gameContext.isSoundOpen()) {
    toggleMusic.setIcon(new ImageIcon(getClass().getResource(name:"btn\\Open_On.png")));
}
```

- ถ้า“ soundOpen ” ( เริ่มเกมค่าจะเป็น false ) ที่ส่งข้อมูลจากตัว gameContext.isSoundOpen มาถ้ามันเป็น false ตัวรูปปุ่ม toggleMusic จะถูกเปลี่ยนใหม่ เป็นรูป Open

```
} else if (e.getSource() == toggleMusic) {
    gameContext.toggleSound();
    if (!gameContext.isSoundOpen()) {
        toggleMusic.setIcon(new ImageIcon(getClass().getResource(name:"btn\\Open_Off.png")));
    } else {
        toggleMusic.setIcon(new ImageIcon(getClass().getResource(name:"btn\\Open_On.png")));
    }
}
```

- ถ้ากดปุ่ม toggleMusic ถ้าค่า isSoundOpen ที่ส่งมาจากตัว gameContext

  ถ้ามันเป็น false ให้ new Image เป็นตัวของ ภาพ Open_off.png แต่ถ้ามันเป็น true ให้ new Image เป็นตัวของ ภาพ Open_On.png


SUDOKU
MENU
RESET
CLEAR
SUMMIT

# ส่วนที่ 2 Controller

# Controller

```java
// state of the game
private boolean isGameStarted = false;
private int difficulty = -1;
```

- ค่าความยากของเกม difficulty mode ที่เก็บค่ามาจากตัว menu

```java
// game board
private boolean soundOpen = false;
private String soundPath = "sound\\Song1.wav";
private int screenWidth = 1535;
private int screenHeight = 850;
```

- ค่า default ของ เพลง ( ถ้าเป็น true ตัวเพลงจะไม่เริ่มตอนต้นเกม)
- Path ของเพลงที่เราใช้
- ค่า ScreenWidth ของเกม
- ค่า ScreenHeight ของเกม

```java
public Game getGame() {
    return game;
}

public int getScreenWidth() {
    return screenWidth;
}

public int getScreenHeight() {
    return screenHeight;
}
```

- ค่าที่เก็บไว้ในตัว GameContext ทั้งหมดก็จะถูกส่งไปยัง Class ต่างๆเพื่อ

```java
public boolean toGame() {
    return toGame(this.difficulty);
}
```

## ตัวอย่างการใช้

### enum Number

```java
enum Number {
    EMPTY(path:"number\\0.png"), ONE(path:"number\\1.png"), TWO(path:"number\\2.png"), THREE(path:"number\\3.png"), FOUR(path:"number\\4.png"),
    FIVE(path:"number\\5.png"),
    SIX(path:"number\\6.png"), SEVEN(path:"number\\7.png"), EIGHT(path:"number\\8.png"), NINE(path:"number\\9.png");

    private String path;
    private ImageIcon icon;

    Number(String path) {
        this.path = path;
        this.icon = new ImageIcon(this.path);
    }

    public ImageIcon getIcon() {
        return this.icon;
    }
}
```

- เมื่อเรากด Start ตัวของ Class “ Game “ จะมาดึงค่าของ enum Number เพื่อสร้างปุ่มแต่ละปุ่มขึ้นมา

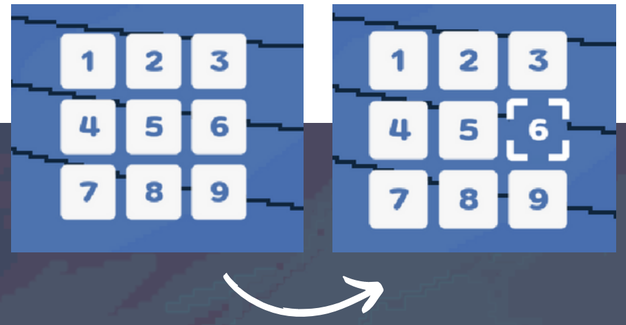
### enum NumberPicked

```java
enum NumberPicked {
    EMPTY(path:"number\\0.png"), ONE(path:"number\\Pick1.png"), TWO(path:"number\\Pick2.png"), THREE(path:"number\\Pick3.png"),
    FOUR(path:"number\\Pick4.png"),
    FIVE(path:"number\\Pick5.png"),
    SIX(path:"number\\Pick6.png"), SEVEN(path:"number\\Pick7.png"), EIGHT(path:"number\\Pick8.png"), NINE(path:"number\\Pick9.png");

    private String path;
    private ImageIcon icon;

    NumberPicked(String path) {
        this.path = path;
        this.icon = new ImageIcon(this.path);
    }

    public ImageIcon getIcon() {
        return this.icon;
    }
}
```




- แต่เมื่อ เกิด Action จากปุ่มตัวเลข ใน Game ตัว Class ก็จะกลับมาดึงค่าของตัว enum NumberPicked และทำให้เกิดการเปลี่ยนรูปจาก ในปุ่มที่ส่ง Action

# ส่วนที่ 3 Generator

references

**Program for Sudoku Generator**

Your All-in-One Learning Portal: GeeksforGeeks is a comprehensive educational platform that empowers learners across domains-spanning computer science and...

GeeksforGeeks / Jul 23, 2025

# Generator

แบ่งออกเป็น 2 ส่วนหลักๆ

GENERATE

CHECK

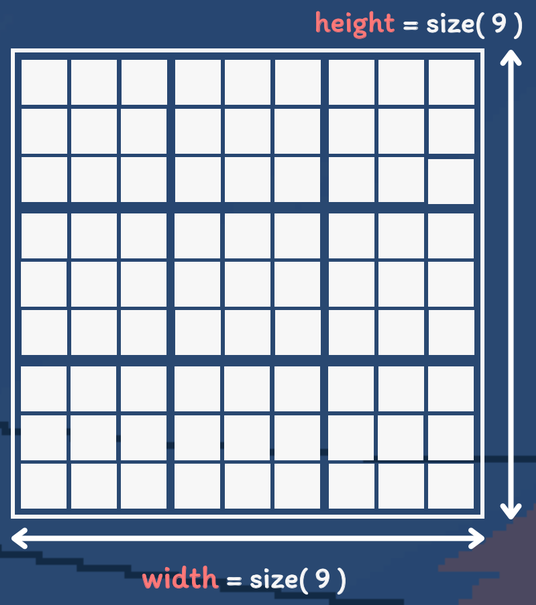
# GENERATE

```java
table = new SudoTable(this.gameContext, this, size:9, difficulty.Random());
```

```java
public void generate() {
    do {
        mat = new int[size][size];
        fillDiagonal();
        fillRemaining(i:0, sizePerBox);
    } while (!puzzleCheck());

    removeRandom();
}
```

- method generate สร้าง กระดานเปล่า โดยให้สร้างตาราง Arrays ขนาด size * size ( 9 * 9 ) โดย รับค่า Size จากตัว table




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | | | | | | |
| 6 | 9 | 7 | | | | | | |
| 1 | 2 | 8 | | | | | | |
| | | | 2 | 4 | 1 | | | |
| | | | 7 | 5 | 3 | | | |
| | | | 8 | 6 | 9 | | | |
| | | | | | | 9 | 1 | 4 |
| | | | | | | 2 | 6 | 3 |
| | | | | | | 8 | 7 | 5 |

- generate เลขสุ่มใน “แนวทแยงมุม”

```java
private boolean fillDiagonal() {
    for (int i = 0; i < this.sizePerBox; i++) {
        if (!fillBox(i)) {
            return false;
        }
    }
    return true;
}

private boolean fillBox(int i) {
    for (int row = i * sizePerBox; row < (i * sizePerBox) + sizePerBox; row++) {
        for (int col = i * sizePerBox; col < (i * sizePerBox) + sizePerBox; col++) {
            do {
                try {
                    this.mat[row][col] = randomNumber(start:1, size);
                } catch (Exception e) {
                    return false;
                }
            } while (!checkInBox(mat, row, col));
        }
    }
    return true;
}
```

# GENERATE

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 1 | 7 | 2 | 6 | 8 | 9 |
| 6 | 9 | 7 | 3 | 8 | 5 | 4 | 2 | 1 |
| 1 | 2 | 8 | 4 | 9 | 6 | 3 | 5 | 7 |
| 5 | 3 | 6 | 2 | 4 | 1 | 7 | 9 | 8 |
| 2 | 8 | 9 | 7 | 5 | 3 | 1 | 4 | 6 |
| 4 | 7 | 1 | 8 | 6 | 9 | 5 | 3 | 2 |
| 7 | 6 | 3 | 5 | 2 | 8 | 9 | 1 | 4 |
| 8 | 5 | 4 | 9 | 1 | 7 | 2 | 6 | 3 |
| 9 | 1 | 2 | 6 | 3 | 4 | 8 | 7 | 5 |

- Generate เลขสุ่มใน cell ที่เหลือ

```java
private boolean fillRemaining(int i, int j) {
    if (j >= size && i < size - 1) {
        i = i + 1;
        j = 0;
    }
    if (i >= size && j >= size)
        return true;

    if (i < sizePerBox) {
        if (j < sizePerBox)
            j = sizePerBox;
    } else if (i < size - sizePerBox) {
        if (j == (int) (i / sizePerBox) * sizePerBox)
            j = j + sizePerBox;
    } else {
        if (j == size - sizePerBox) {
            i = i + 1;
            j = 0;
            if (i >= size)
                return true;
        }
    }

    for (int num = 1; num <= size; num++) {
        if (this.allCheck(this.mat, i, j, num)) {
            mat[i][j] = num;
            if (this.fillRemaining(i, j + 1))
                return true;
            mat[i][j] = 0;
        }
    }
    return false;
}
```

# GENERATE

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 1 | 7 | 2 | 6 | 8 | 9 |
| 6 | 9 | 7 | 3 | 8 | 5 | 4 | 2 | 1 |
| 1 | 2 | 8 | 4 | 9 | 6 | 3 | 5 | 7 |
| 5 | 3 | 6 | 2 | 4 | 1 | 7 | 9 | 8 |
| 2 | 8 | 9 | 7 | 5 | 3 | 1 | 4 | 6 |
| 4 | 7 | 1 | 8 | 6 | 9 | 5 | 3 | 2 |
| 7 | 6 | 3 | 5 | 2 | 8 | 9 | 1 | 4 |
| 8 | 5 | 4 | 9 | 1 | 7 | 2 | 6 | 3 |
| 9 | 1 | 2 | 6 | 3 | 4 | 8 | 7 | 5 |

- สุ่มลบเลขออกจาก cell

```java
private void removeRandom() {
    int i = 0;
    do {
        int row = randomNumber(start:0, size - 1);
        int col = randomNumber(start:0, size - 1);
        if (mat[row][col] != 0) {
            mat[row][col] = 0;
            i++;
        }
    } while (i < missingDigits);
}
```

# CHECK

1. check ใน Box

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 1 | 7 | 2 | 6 | 8 | 9 |
| 6 | 9 | 7 | 3 | 8 | 5 | 4 | 2 | 1 |
| 1 | 2 | 8 | 4 | 9 | 6 | 3 | 5 | 7 |
| 5 | 3 | 6 | 2 | 4 | 1 | 7 | 9 | 8 |
| 2 | 8 | 9 | 7 | 5 | 3 | 1 | 4 | 6 |
| 4 | 7 | 1 | 8 | 6 | 9 | 5 | 3 | 2 |
| 7 | 6 | 3 | 5 | 2 | 8 | 9 | 1 | 4 |
| 8 | 5 | 4 | 9 | 1 | 7 | 2 | 6 | 3 |
| 9 | 1 | 2 | 6 | 3 | 4 | 8 | 7 | 5 |

2. check ใน Row

```java
private boolean checkInBox(int mat[][], int row, int col, int num) {
    int myNum = num <= -1 ? mat[row][col] : num;
    int startRow = (int) (this.sizePerBox * (row / this.sizePerBox));
    int startCol = (int) (this.sizePerBox * (col / this.sizePerBox));
    for (int i = startRow; i < startRow + this.sizePerBox; i++) {
        for (int j = startCol; j < startCol + this.sizePerBox; j++) {
            if (i == row && j == col) {
                continue;
            }

            if (mat[i][j] == myNum) {
                return false;
            }
        }
    }
    return true;
}
```

```java
private boolean checkInRow(int mat[][], int row, int col, int num) {
    int myNum = num <= -1 ? mat[row][col] : num;
    for (int i = 0; i < size; i++) {
        if (i == col) {
            continue;
        }
        if (mat[row][i] == myNum) {
            return false;
        }
    }
    return true;
}
```

```java
private boolean checkInCol(int mat[][], int row, int col, int n
    int myNum = num <= -1 ? mat[row][col] : num;
    for (int j = 0; j < size; j++) {
        if (j == row) {
            continue;
        }
        if (mat[j][col] == myNum) {
            return false;
        }
    }
    return true;
}
```

3. check ใน Column