MINI PROJECT : เกม Tetris

เสนอ

อาจารย์ สถิตย์ ประสมพันธ์

โดย นาย รุ่งอรุณ อุ่นแก้ว

รหัสนักศึกษา 6004062636335

Section : 1

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา OBJECT-ORIENTED PROGRAMMING

ภาคเรียนที่ 1 ปีการศึกษา 2563

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1 บทนำ

ชื่อ Project

- Tetris

ที่มาและความสำคัญของโปรเจค

- ในยุคปัจจุบันนี้ถือได้ว่าเกมมีบทบาทสำคัญสำหรับเยาวชนเป็นส่วนมาก เนื่องจากเข้าถึงง่าย ดังนั้นหากเล่นเกมที่ไม่ได้มีส่วนช่วยในการพัฒนาสมองก็อาจเป็นการเล่นเกมที่ส่งผลเสียได้ ทั้งนี้ที่กล่าวมาถือเป็นแรงบันดาลใจให้ผู้จัดทำออกแบบและพัฒนาเกมนี้ขึ้นมา เนื่องจากผู้จัดทำมีความสนใจในการเล่นเกมเป็นทุนเดิมอยู่แล้วและได้มีการศึกษาการใช้โปรแกรม NetBeans IDE ในการสร้างเกม จึงมีจุดมุ่งหมายในการสร้างเกม ที่สามารถฝึกฝนสมอง สร้างระบบความคิดมากกว่าเกมที่มุ่งเน้นความรุนแรง ได้อย่างเหมาะสม จากเหตุผลข้างต้นจึงเป็นที่มาและความสำคัญให้ผู้จัดทำมีความต้องการสร้างเกมส์นี้ขึ้นมา

ประภทของโครงการ

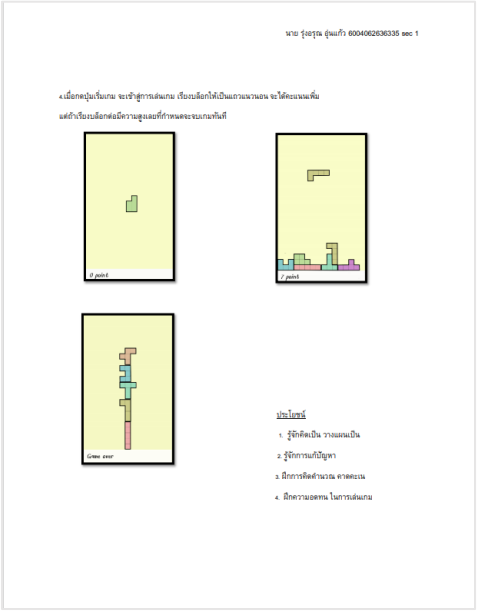
- โครงงานออกแบบและพัฒนาเกม

ประโยชน์

- รู้จักคิดเป็น วางแผนเป็น
- รู้จักการแก้ปัญหา
- ฝึกการคิดคำนวณ คาดคะเน
- ฝึกความอดทน ในการเล่นเกม

ขอบเขตของโครงการ(แนบ Proposal)

- แนบ Proposal

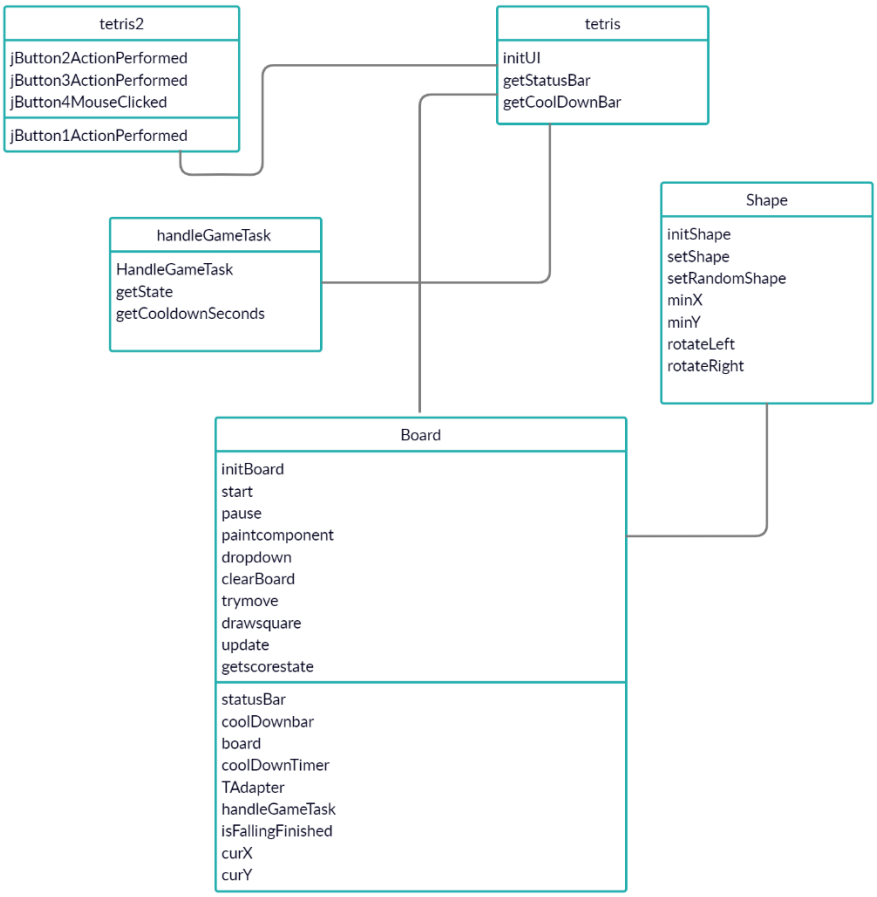


บทที่ 2 ส่วนการพัฒนา

เนื้อเรื่องย่อหรือวิธีการเล่น

- ผู้เล่นจะต้องเรียงบล็อกในแต่ละแถวให้เต็มแถว แถวนั้นทั้งแถวจะหายไป หากสามารถทำให้หาย หลาย ๆ แถว จะ ได้คะแนนมากขึ้นอีกด้วย ถ้าไม่สามารถต่อให้ครบแถวจนแถวหาย และบล็อกมีความสูงเรื่อย ๆ จนถึงบนสุดหรือไม่ได้คะแนนตามเวลาที่กำหนด ก็จะแพ้เกมทันที โดยใช้แป้นพิมพ์ ในการขยับบล็อกและการเคลื่อนทิศทางของบล็อก

คลาสไดอะแกรม พร้อมคำอธิบาย



อธิบายโค้ด

Class Tetris

```
    private void initUI() {

        statusbar = new JLabel("  0    Point - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -");
        add(statusbar, BorderLayout.SOUTH);

        coolDownbar = new JLabel();
        add(coolDownbar, BorderLayout.NORTH);

        Board board = new Board(this);
        add(board);
        board.start();

        setTitle("Tetris Game");
        setSize(300, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);




    }

    JLabel getStatusBar() {
        return statusbar;
    }

    JLabel getCoolDownBar() {
        return coolDownbar;
    }

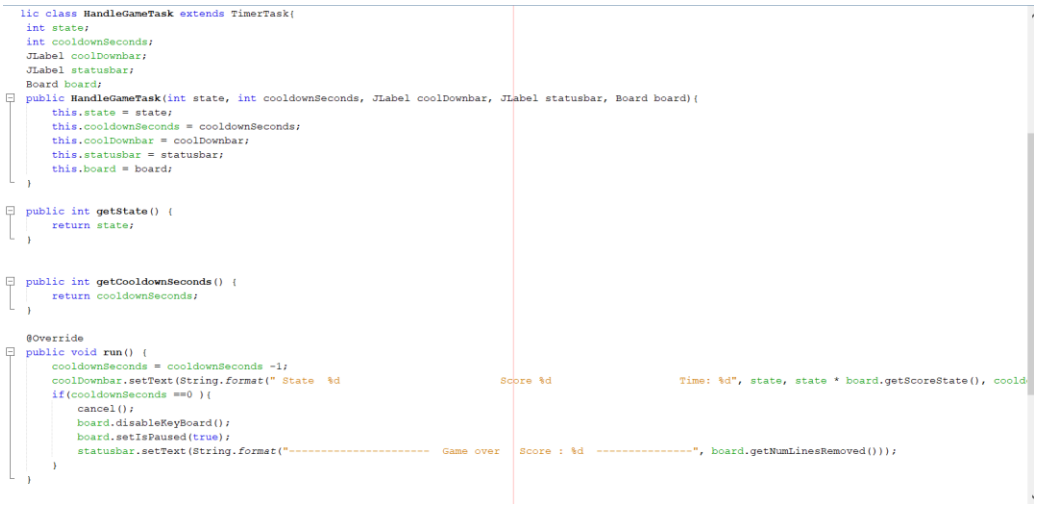
    public static void main(String[] args) {

        EventQueue.invokeLater(() -> {

            Tetris game = new Tetris();
            game.setVisible(true);
        ```.
```

ในคลาส Tetris ที่กำหนดเรียกใช้ main เพื่อเป็นตัวสร้าง JFrame และมีการเรียกใช้ getStatusBar คือ แถบแสดงสถานะของเกม และ getCoolDownBar คือ การแสดงการจับเวลา

## Class HandleGameTask



```
lic class HandleGameTask extends TimerTask{
 int state;
 int cooldownSeconds;
 JLabel coolDownbar;
 JLabel statusbar;
 Board board;
public HandleGameTask(int state, int cooldownSeconds, JLabel coolDownbar, JLabel statusbar, Board board){
    this.state = state;
    this.cooldownSeconds = cooldownSeconds;
    this.coolDownbar = coolDownbar;
    this.statusbar = statusbar;
    this.board = board;
}

public int getState() {
    return state;
}


public int getCooldownSeconds() {
    return cooldownSeconds;
}

@Override
public void run() {
    cooldownSeconds = cooldownSeconds -1;
    coolDownbar.setText(String.format(" State  %d                    Score %d                  Time: %d", state, state * board.getScoreState(), coold
    if(cooldownSeconds ==0 ){
        cancel();
        board.disableKeyBoard();
        board.setIsPaused(true);
        statusbar.setText(String.format("----------------------  Game over   Score : %d  ---------------", board.getNumLinesRemoved()));
    }
}
```

เป็น Class ที่ใช้ @Overide แสดงผลในหน้า JFrame และการเรียกใช้คลาสอื่น ๆ
```

Class Shape

```
public Shape() {

    initShape();
}

private void initShape() {

    coords = new int[4][2];
    //refer object
    coordsTable = new int[][][] {
          //  { { 0, 0 },   { 0, 0 },   { 0, 0 },   { 0, 0 } },
            { { 0, -1 },  { 0, 0 },   { -1, 0 },  { -1, 1 } },
            { { 0, -1 },  { 0, 0 },   { 1, 0 },   { 1, 1 } },
            { { 0, -1 },  { 0, 0 },   { 0, 1 },   { 0, 2 } },
            { { -1, 0 },  { 0, 0 },   { 1, 0 },   { 0, 1 } },
            { { 0, 0 },   { 1, 0 },   { 0, 1 },   { 1, 1 } },
            { { -1, -1 }, { 0, -1 },  { 0, 0 },   { 0, 1 } },
            { { 1, -1 },  { 0, -1 },  { 0, 0 },   { 0, 1 } }
    };

    setShape(Tetrominoe.NoShape);
}

protected void setShape(Tetrominoe shape) {

    for (int i = 0; i < 4 ; i++) {

        for (int j = 0; j < 2; ++j) {

            coords[i][j] = coordsTable[shape.ordinal()][i][j];
        }
    }
```

Method initShape เป็นการระบุตัวแปร coords และ coordsTable คือการอ้างอิงตำแหน่ง object ในหน้าเกม

```
public Tetrominoe getShape()  { return pieceShape; }

public void setRandomShape() {

    Random r = new Random();
    int x = Math.abs(r.nextInt()) % 7 + 1;

    Tetrominoe[] values = Tetrominoe.values();
    setShape(values[x]);
}

public int minX() {

    int m = coords[0][0];

    for (int i=0; i < 4; i++) {

        m = Math.min(m, coords[i][0]);
    }

    return m;
}


public int minY() {

    int m = coords[0][1];

    for (int i=0; i < 4; i++) {

        m = Math.min(m, coords[i][1]);
    }

    return m;
}

public Shape rotateLeft() {
```

Methods setRandomShape เป็นการ Random ตำแหน่ง ส่วน constructor minX และ minY เป็นการระบุตำแหน่งในการเคลื่อนที่ X และ Y

```
    }

    public Shape rotateLeft() {

        if (pieceShape == Tetrominoe.SquareShape) {

            return this;
        }

        Shape result = new Shape();
        result.pieceShape = pieceShape;

        for (int i = 0; i < 4; ++i) {

            result.setX(i, y(i));
            result.setY(i, -x(i));
        }

        return result;
    }

    public Shape rotateRight() {

        if (pieceShape == Tetrominoe.SquareShape) {

            return this;
        }

        Shape result = new Shape();
        result.pieceShape = pieceShape;

        for (int i = 0; i < 4; ++i) {

            result.setX(i, -y(i));
            result.setY(i, x(i));
        }
```

Constructor rotateLeft และ rotateRight คือการหมุนบล็อกในตำแหน่ง และคืนค่าตำแหน่งเป็น X และ Y

Class Board

```
void start() {

    curPiece = new Shape();
    board = new Tetrominoe[BOARD_WIDTH * BOARD_HEIGHT];

    clearBoard();
    newPiece();

    timer = new Timer(PERIOD_INTERVAL, new GameCycle());
    timer.start();

    coolDownTimer = new java.util.Timer();

    handlGameTask = new HandleGameTask(1, timeState, coolDownbar, statusbar, this);
    coolDownTimer.schedule(handlGameTask, 0, 1000);
}
```

Method start เป็นการกำหนดการเริ่มเกม ทั้ง CoolDownTimer คือการจับเวลา handGameTask คือแถบแสดงสถานะ

```
private void pause() {

    isPaused = !isPaused;

    if (isPaused) {

        statusbar.setText("paused");

        handlGameTask.cancel();
        //disableKeyBoard();
        removeKeyListener(tAdapter);
        addKeyListener(tAdapterPBtn);
    } else {
        removeKeyListener(tAdapterPBtn);
        enableKeyBoard();
        statusbar.setText(String.valueOf(numLinesRemoved));

        coolDownTimer = new java.util.Timer();
        handlGameTask.cancel();
        handlGameTask = new HandleGameTask(handlGameTask.getState(), handlGameTask.getCooldownSeconds(), coolDownbar, statusbar, this);
        coolDownTimer.schedule(handlGameTask, 0, 1000);
    }

    repaint();
}
```

Method pause คือการกำหนดเมื่อเล่นเกม และเกิดการกดหยุดขณะเล่นเกม ให้แสดงข้อความ pause และหยุดการทำงานของ handleGameTask หรือแถบแสดงสถานะ

```
private void dropDown() {

    int newY = curY;

    while (newY > 0) {

        if (!tryMove(curPiece, curX, newY - 1)) {

            break;
        }

        newY--;
    }

    pieceDropped();
}

private void oneLineDown() {

    if (!tryMove(curPiece, curX, curY - 1)) {

        pieceDropped();
    }
}
```

Method dropdown และ oneLineDown คือการกำหนดการเคลื่อนที่ของบล็อกในการ drop ลงมา

```
private void pieceDropped() {

    for (int i = 0; i < 4; i++) {

        int x = curX + curPiece.x(i);
        int y = curY - curPiece.y(i);
        board[(y * BOARD_WIDTH) + x] = curPiece.getShape();
    }

    removeFullLines();

    if (!isFallingFinished) {

        newPiece();
    }

    //Check win game
    if(numLinesRemoved >= handlGameTask.getState() * scoreState && handlGameTask.getState() == maxState){
    //if(numLinesRemoved > 0 && handlGameTask.getState() == 1){
        handlGameTask.cancel();
        isPaused = true;
        // Render text winner
        // stop game
        //remove addKeyListener
        disableKeyBoard();
        statusbar.setText("--------------------------You win!--------------------------");
    } //Check win current state
    else if(numLinesRemoved >= handlGameTask.getState() * scoreState){
        // render new state
        coolDownTimer = new java.util.Timer();
        handlGameTask.cancel();
        handlGameTask = new HandleGameTask(handlGameTask.getState() + 1, timeState, coolDownbar, statusbar, this);
        coolDownTimer.schedule(handlGameTask, 0, 1000);
    }
}
```

คือการใช้ if else ในการตรวจสอบการเล่น โดยเช็คจาก cooldown แสดงผลในการเล่น

```
private void drawSquare(Graphics g, int x, int y, Tetrominoe shape) {

    Color colors[] = {new Color(0, 0, 0), new Color(204, 102, 102),
            new Color(102, 204, 102), new Color(102, 102, 204),
            new Color(204, 204, 102), new Color(204, 102, 204),
            new Color(102, 204, 204), new Color(218, 170, 0)
    };

    Color color = colors[shape.ordinal()];

    g.setColor(color);
    g.fillRect(x + 1, y + 1, squareWidth() - 2, squareHeight() - 2);

    g.setColor(color.brighter());
    g.drawLine(x, y + squareHeight() - 1, x, y);
    g.drawLine(x, y, x + squareWidth() - 1, y);

    g.setColor(color.darker());
    g.drawLine(x + 1, y + squareHeight() - 1,
            x + squareWidth() - 1, y + squareHeight() - 1);
    g.drawLine(x + squareWidth() - 1, y + squareHeight() - 1,
            x + squareWidth() - 1, y + 1);


}
```

Method drew คือการเรียกใช้ method ในการวาดรูปบล็อก โดยมีการระบุสีที่ใช้ในการวาด การกำหนดรูปร่างของบล็อกสี่เหลี่ยม

```
class TAdapter extends KeyAdapter {

    @Override
    public void keyPressed(KeyEvent e) {

        if (curPiece.getShape() == Tetrominoe.NoShape) {

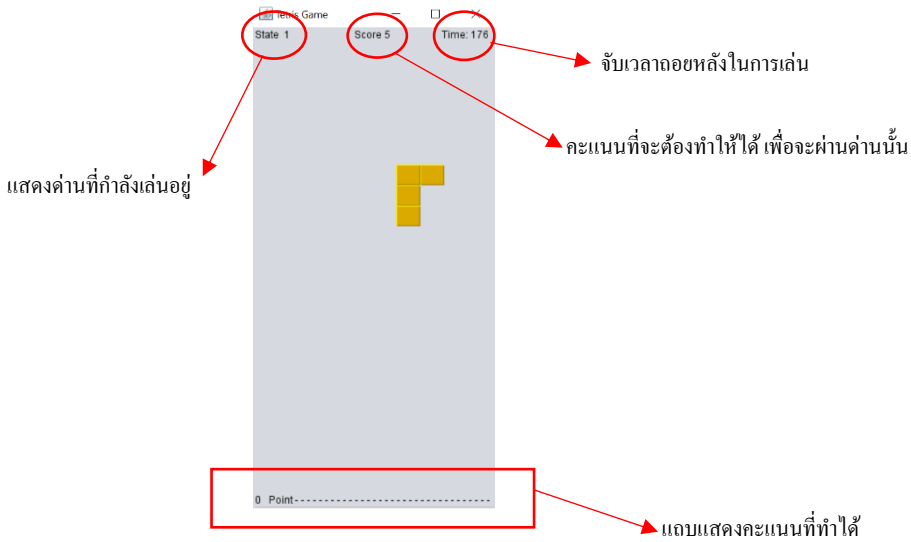
            return;
        }

        int keycode = e.getKeyCode();

        // Java 12 switch expressions
        switch (keycode) {
            case KeyEvent.VK_P : pause(); break;
            case KeyEvent.VK_LEFT : tryMove(curPiece, curX - 1, curY); break;
            case KeyEvent.VK_RIGHT : tryMove(curPiece, curX + 1, curY); break;
            case KeyEvent.VK_DOWN : tryMove(curPiece.rotateRight(), curX, curY); break;
            case KeyEvent.VK_UP : tryMove(curPiece.rotateLeft(), curX, curY); break;
            case KeyEvent.VK_SPACE : dropDown(); break;
            case KeyEvent.VK_D : oneLineDown(); break;
        }
    }
}
```

Class TAdapter คือการเรียกใช้ method keypressed ในการกำหนด case ต่าง ๆ ในการกดเพื่อดำเนินการเล่นเกม

หน้าจอ GUI อธิบายส่วนประกอบของ GUI โครงสร้างของ GUI ประกอบด้วย Component อะไรบ้าง




```
private void initUI() {

    statusbar = new JLabel(" 0   Point - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -");
    add(statusbar, BorderLayout.SOUTH);

    coolDownbar = new JLabel();
    add(coolDownbar, BorderLayout.NORTH);

    Board board = new Board(this);
    add(board);
    board.start();

    setTitle("Tetris Game");
    setSize(300, 600);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
```

```
@Override
public void run() {
    //System.out.println("Hello World");
    cooldownSeconds = cooldownSeconds -1;
    coolDownbar.setText(String.format(" State  %d                    Score %d                    Time: %d", state, state * board.getScore
    if(cooldownSeconds ==0 ){
        cancel();
        board.disableKeyBoard();
        board.setIsPaused(true);
        statusbar.setText(String.format("----------------------  Game over   Score : %d  ---------------", board.getNumLinesRemoved()));
    }
```

โค้ดการสร้าง GUI

อธิบาย Event handling ที่มีในหน้าจอ

กดคลิกปุ่มเริ่มเกม เพื่อเริ่มเกม เรียกใช้ JButton1ActionPerformed

กดคลิกปุ่มวิธีเล่น เพื่ออ่านวิธีเล่น เรียกใช้ JButton2ActionPerformed

กดคลิกปุ่มเกี่ยวกับ เพื่อดูข้อมูลเกี่ยวกับผู้จัดทำ เรียกใช้ JButton3ActionPerformed

กดคลิกปุ่มออกจากเกม เพื่อปิดหน้าจอ เรียกใช้ jButton4MouseClicked

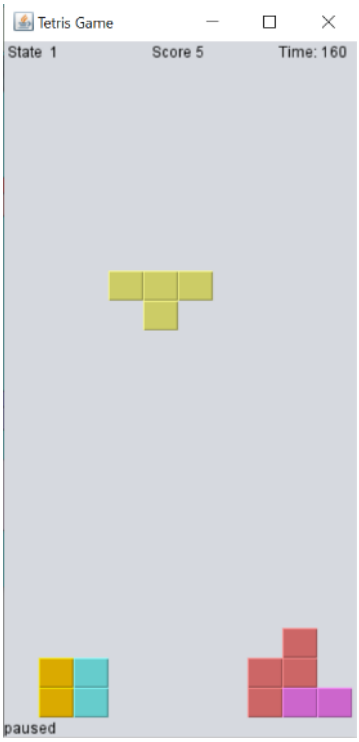
กดแป้นพิมพ์ P เพื่อ paused / start

กดแป้นพิมพ์ ⬆ เพื่อเปลี่ยนรูปร่างบล็อก

กดแป้นพิมพ์ ⬅ เพื่อเคลื่อนมาทางซ้าย

กดแป้นพิมพ์ ➡ เพื่อคลื่อนมาทางขวา

กดแป้นพิมพ์ spacebar เพื่อดึงบล็อกลงอย่างรวดเร็ว




```
class TAdapter extends KeyAdapter {

    @Override
    public void keyPressed(KeyEvent e) {

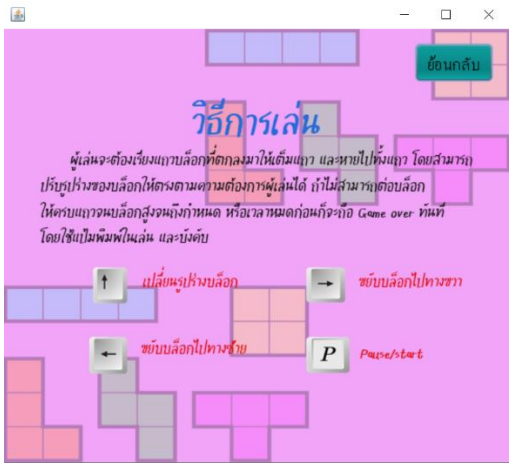
        if (curPiece.getShape() == Tetrominoe.NoShape) {

            return;
        }

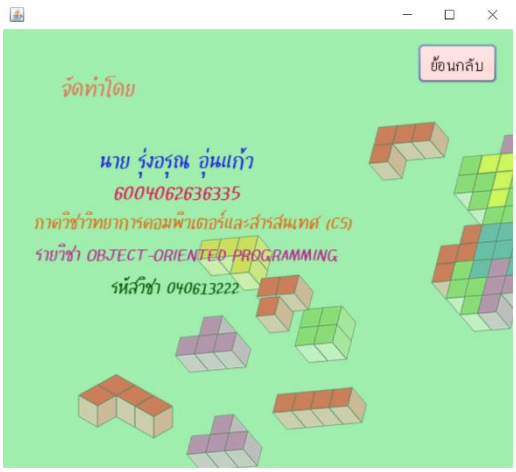
        int keycode = e.getKeyCode();

        // Java 12 switch expressions
        switch (keycode) {
            case KeyEvent.VK_P : pause(); break;
            case KeyEvent.VK_LEFT : tryMove(curPiece, curX - 1, curY); break;
            case KeyEvent.VK_RIGHT : tryMove(curPiece, curX + 1, curY); break;
            case KeyEvent.VK_DOWN : tryMove(curPiece.rotateRight(), curX, curY); break;
            case KeyEvent.VK_UP : tryMove(curPiece.rotateLeft(), curX, curY); break;
            case KeyEvent.VK_SPACE : dropDown(); break;
            case KeyEvent.VK_D : oneLineDown(); break;
        }
    }
}
```

ตัวอย่างโค้ด Key Event



หน้าวิธีการเล่น



หน้าเกี่ยวกับ

บทที่ 3 สรุป

ปัญหาที่พบระหว่างการพัฒนา

- รันโปรแกรมแล้วเกิดปัญหา error
- ในการสร้างเกมต้องมีการศึกษา cording และรู้จักประยุกต์ใช้ให้เข้ากับเกมที่สร้าง ทำให้ต้องใช้เวลาในการศึกษาและการเขียนโปรแกรม
- ทำตามขั้นตอนแล้ว แต่ผลที่แสดงออกมาไม่ตรงตามความต้องการ
- Code มีความซับซ้อน เขียนแล้วเกิดความสับสนมึนงง

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

เป็นเกมที่ทดสอบการวางแผน การใช้ความคิด เล่นแล้วได้ฝึกฝนการใช้สมอง ไม่ใช่เกมที่เล่นแล้วได้แต่ความสนุก จึงทำให้ได้ออกแบบที่ใช้งานง่าย เข้าใจง่าย แต่ได้ฝึกการใช้ความคิด

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้อง ๆ รุ่นต่อไป

ผมไม่ค่อยเข้าใจเรื่อง react ครับ ผมลองไปดูสอนตามในยูทูปแล้วครับ แต่พอมาลองทำเองก็ทำไม่ได้ครับ อาจจะเพราะว่าการเขียนที่ค่อนข้างแตกต่างเลยงงๆ ครับ