



경성대학교
KYUNGSUNG UNIVERSITY

결과보고서

이름 : 전상훈
학번 : 2019775054

목차

- 001 게임 기획 및 설명
- 002 클래스 다이어그램
- 003 소스 코드 설명
- 004 출처

Part 1,

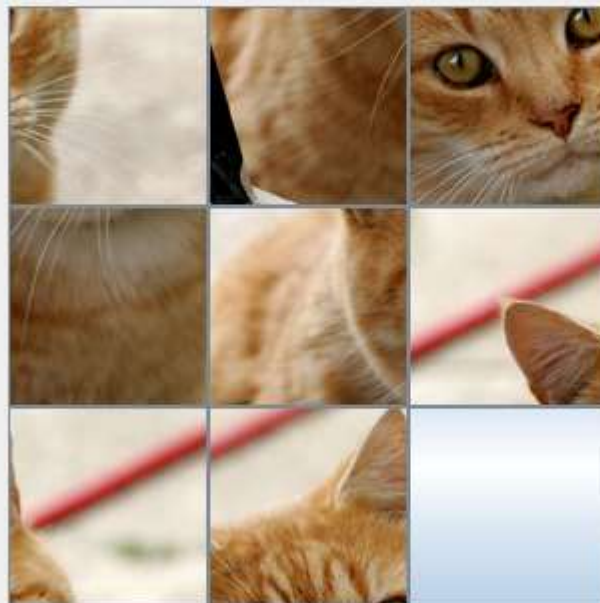
게임 기획 설명

슬라이딩 퍼즐 게임



시작 화면

그만 하기

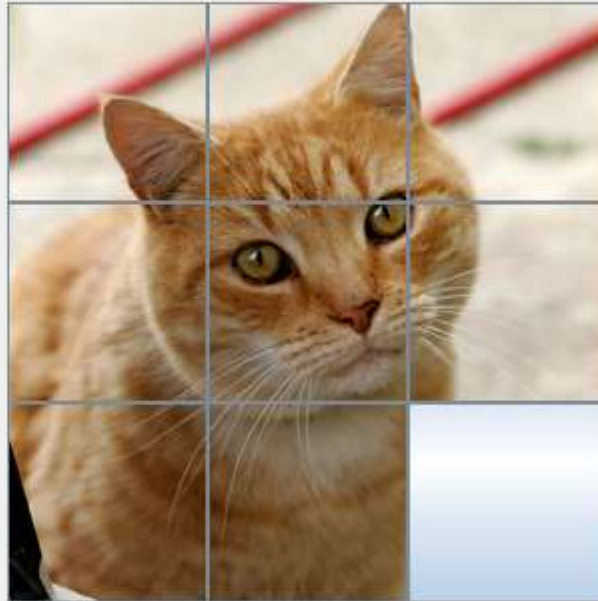


0:2 0

도전중입니다

완료 화면

그만 하기

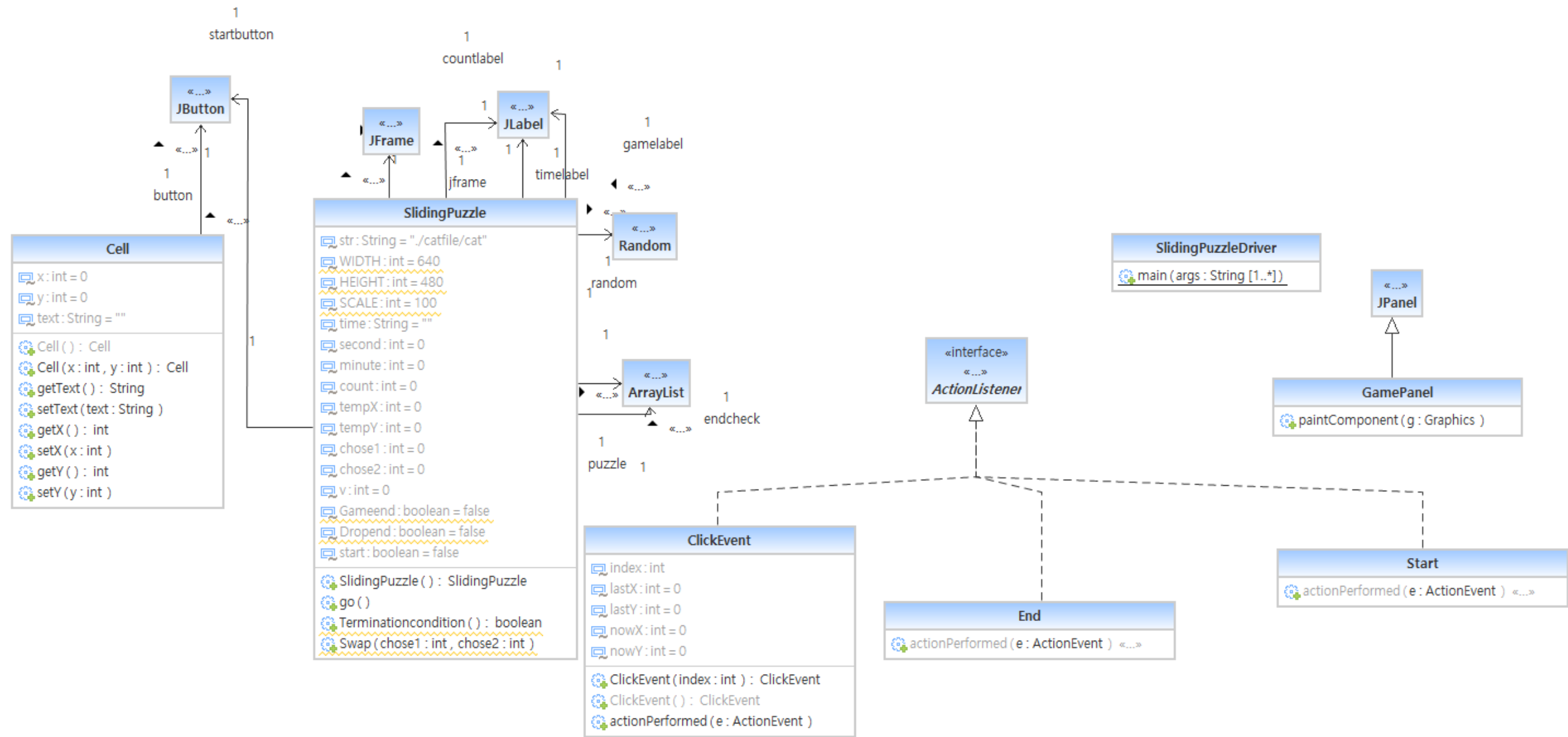


0:32 87

완성했습니다

Part 2,

클래스 다이어그램



Part 3,

소스 코드 설명

시작, 종료 버튼

퍼즐 생성, 랜덤 위치 이동

```
public SlidingPuzzle() {
    startbutton.setBounds(0,0,623,440);
    startbutton.addActionListener(new Start());
    startbutton.setIcon(new ImageIcon("./catfile/cat10.png"));
    jframe.add(startbutton);

    JButton endbutton = new JButton("그만하기");
    endbutton.setBounds(20, 20, 100, 20);
    endbutton.addActionListener(new End());
    jframe.add(endbutton);

    timelabel.setBounds(460, 0, SCALE, SCALE);
    timelabel.setFont(new Font("맑은 고딕", Font.BOLD, 25));

    countlabel.setBounds(550, 0, SCALE, SCALE);
    countlabel.setFont(new Font("맑은 고딕", Font.BOLD, 25));

    gamelabel.setBounds(460, 44, SCALE + 100, SCALE);
    gamelabel.setFont(new Font("맑은 고딕", Font.BOLD, 19));

    jframe.add(timelabel);
    jframe.add(countlabel);
    jframe.add(gamelabel);
    jframe.setSize(WIDTH, HEIGHT);
    jframe.getContentPane().add(new GamePanel());
    jframe.setVisible(true);
    jframe.setLayout(null);
    jframe.setDefaultCloseOperation(jframe.EXIT_ON_CLOSE);
}
```

```
for (int i = 0; i < 9; i++) { /// 그림 타일 설정 부분
    String str = "./catfile/cat";
    if (i < 3) {
        puzzle.add(new Cell(((i % 3) * 100) + 100, 100));
        endcheck.add(new Cell(((i % 3) * 100) + 100, 100));
    } else if (i < 6) {
        puzzle.add(new Cell(((i % 3) * 100) + 100, 200));
        endcheck.add(new Cell(((i % 3) * 100) + 100, 200));
    } else {
        puzzle.add(new Cell(((i % 3) * 100) + 100, 300));
        endcheck.add(new Cell(((i % 3) * 100) + 100, 300));
    }

    str = str + (i + 1) + ".png";

    ImageIcon img = new ImageIcon(str); // 이미지 설정
    puzzle.get(i).button.setIcon(img);
    puzzle.get(i).button.addActionListener(new ClickEvent(i)); // 버튼 클릭 이벤트 추가
}

while (puzzle.get(8).getX() != 300 || puzzle.get(8).getY() != 300 || v < 500) { // 그림타일 랜덤위치 지정
    chose1 = random.nextInt(8);
    chose2 = 8;
    v++;
    int num = Math.abs(puzzle.get(chose1).getX() - puzzle.get(chose2).getX())
        + Math.abs(puzzle.get(chose1).getY() - puzzle.get(chose2).getY());
    if (num == 100) {
        Swap(chose1, chose2);
    }
}

for (int i = 0; i < 9; i++) {
    puzzle.get(i).button.setBounds(puzzle.get(i).getX(), puzzle.get(i).getY(), SCALE, SCALE); // 버튼위치설정
    jframe.add(puzzle.get(i).button); // 프레임에 붙이기
}
```

Go 메소드

```
public void go() {
    while(!start) {
        jframe.repaint();
    }

    gamelabel.setText("도전중입니다");

    while (!Gameend && !Dropend) {
        timelabel.setText(time);
        countlabel.setText(Integer.toString(count));

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        jframe.repaint();
        second++;
        if (second / 10 > 60) {
            minute++;
            second = 0;
        }
        time = minute + ":" + (second / 10);
    }
    if (Gameend)
        gamelabel.setText("완성했습니다");
    else
        gamelabel.setText("포기했습니다");
    jframe.repaint();
}
```

퍼즐을 구성하는 Cell 클래스

```
class Cell {
    JButton button = new JButton();
    int x = 0;
    int y = 0;
    String text = "";
}

class Start implements ActionListener{

    @Override
    public void actionPerformed(ActionEvent e) {
        start = !start;
        jframe.remove(startbutton);
        jframe.repaint();
    }
}

class End implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        for (int i = 0; i < puzzle.size(); i++) {
            puzzle.get(i).setX(endcheck.get(i).getX());
            puzzle.get(i).setY(endcheck.get(i).getY());
            puzzle.get(i).button.setBounds(puzzle.get(i).getX(), puzzle.get(i).getY(), SCALE, SCALE);
            Dropend = true;
        }
    }
}
```

Start, End 버튼의
ActionListener

게임 종료 조건을
판단하는 메소드

```
public boolean Terminationcondition() {  
    for (int i = 8; i >= 0; i--) {  
        if (puzzle.get(i).getX() != endcheck.get(i).getX()) {  
            return false;  
        }  
        if (puzzle.get(i).getY() != endcheck.get(i).getY()) {  
            return false;  
        }  
    }  
    return true;  
}
```

```
class GamePanel extends JPanel {  
    public void paintComponent(Graphics g) {  
        g.setColor(Color.LIGHT_GRAY);  
        g.fillRect(450, 24, 150, 100);  
    }  
}
```

```
public void Swap(int chose1, int chose2) {  
    tempX = puzzle.get(chose1).getX();  
    tempY = puzzle.get(chose1).getY();  
  
    puzzle.get(chose1).setX(puzzle.get(chose2).getX());  
    puzzle.get(chose1).setY(puzzle.get(chose2).getY());  
  
    puzzle.get(chose2).setX(tempX);  
    puzzle.get(chose2).setY(tempY);  
}
```

시간, 카운트, 진행상황이있는
패널

두 퍼즐의 위치를 변경하는 S
wap 메소드

퍼즐 클릭시 위치를 변경하는 ActionListener

```
class ClickEvent implements ActionListener {
    int index;
    int lastX = 0;
    int lastY = 0;
    int nowX = 0;
    int nowY = 0;

    public ClickEvent(int index) {}

    public ClickEvent() {}

    public void actionPerformed(ActionEvent e) {
        this.nowX = puzzle.get(index).getX();
        this.nowY = puzzle.get(index).getY();
        this.lastX = puzzle.get(8).getX();
        this.lastY = puzzle.get(8).getY();
        int check = Math.abs(nowX - lastX) + Math.abs(nowY - lastY);
        if (check == 100) {

            Swap(index,8);

            puzzle.get(index).button.setBounds(puzzle.get(index).getX(), puzzle.get(index).getY(), SCALE, SCALE);
            puzzle.get(8).button.setBounds(puzzle.get(8).getX(), puzzle.get(8).getY(), SCALE, SCALE);
            count++;
            if (Terminationcondition()) {
                Gameend = true;
            }
        }
    }
}
```

Part 4,

출처

<https://www.freeimages.com/kr/photo/cat-s-face-1553769>

https://yusaeyeol.blogspot.com/2020/07/powerpoint-template-free-download-ppt_20.html

감사합니다