

The background is a dark blue gradient. It features several vertical white lines of varying lengths. Scattered throughout are small squares in teal, orange, and pink. Some squares are solid, while others are outlined.

राशि RASRITAIRE

CP241 DATA STRUCTURE

PROJECT - TEAM



61102010158

นาย.ภูริปกรณ์ ศรียอด



61102010419

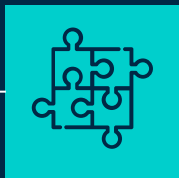
นาย.กานต์ชนิต โพธิ์สุวรรณ

นาย.วรวิทย์ นาคนาวา

61102010160



ขอบเขตการนำเสนอ



01

บทนำ



02

วิธีการดำเนินงาน







03

ผลการทำงานและสรุปผล

บทนำ

01

ที่มาและความสำคัญ

- แอ็พพลิเคชันที่ได้ทำ เป็นเกมเรียงไพ่คล้ายๆกับ solitaire โดยจะเปลี่ยนเป็นการนำเสนอในรูปแบบจักรราศีแทนซึ่งจะเป็นการแบ่งหมวดไพ่ในสำหรับทั้ง 52 ใบ เป็น 4 หมวด คือ โพดำ() โพแดง() ข้าวหลามตัด() และดอกจิก()
- โดยจะให้เรียงจาก ราศีมังกร ราศีกุมภ์ ราศีมีน ราศีเมษ ราศีพฤษภ ราศีเมถุน ราศีกรกฎ ราศีสิงห์ ราศีกันย์ ราศีตุล ราศีพิจิก ราศีธนู ราศีคนแบกงู
- ซึ่งเกมของเราจะเป็นเกมที่ช่วยในเรื่องการฝึกสมองด้านความจำ ไหวพริบ การวางแผนต่างๆ และยังเป็นการทดสอบความรู้รอบตัวเรื่องราศีกับผู้เล่นอีกด้วย

ปัญหาก่อนที่จะมีโปรแกรมนี้

1. ก่อนที่จะมีโปรแกรม สมาชิกภายในกลุ่มไม่ค่อยเห็นแนวทางในการทำโปรเจกต์ที่หลากหลายและง่ายต่อการจัดทำ
2. การเปลี่ยนหัวข้อกระทันหันทำให้เกิดความไม่เข้าใจในการทำงานของเรา



02

วิธีการดำเนินงาน

โครงสร้างข้อมูล (Data Structure)

ในส่วนของ Data Structure ที่ใช้คือ Stack ซึ่งใช้หลักการ Last In First Out (LIFO) โดยการใช้ Stack เก็บข้อมูลของการ์ดแต่ละใบโดยให้เก็บไว้ในตัวแปร deck และใช้ Stack ในการเก็บไฟส่วนของ Waste ส่วนที่เปิดไฟบนสุดของกองออกมาจะถูกเก็บไว้ที่ตัวแปรนี้ด้วยคำสั่ง `deck.pop()`

For more information about Stack : <https://www.geeksforgeeks.org/stack-data-structure/>

การทำงานของโครงสร้างข้อมูล

โดยโปรแกรมนี้ได้ใช้ Method ต่างๆดังนี้

Stack

- ทำงานโดยใช้หลักการ LastInFirstOut(LIFO) คือ เมื่อใส่ข้อมูลเข้าไปใน stack ตัวสุดท้ายจะออกมาตัวแรก

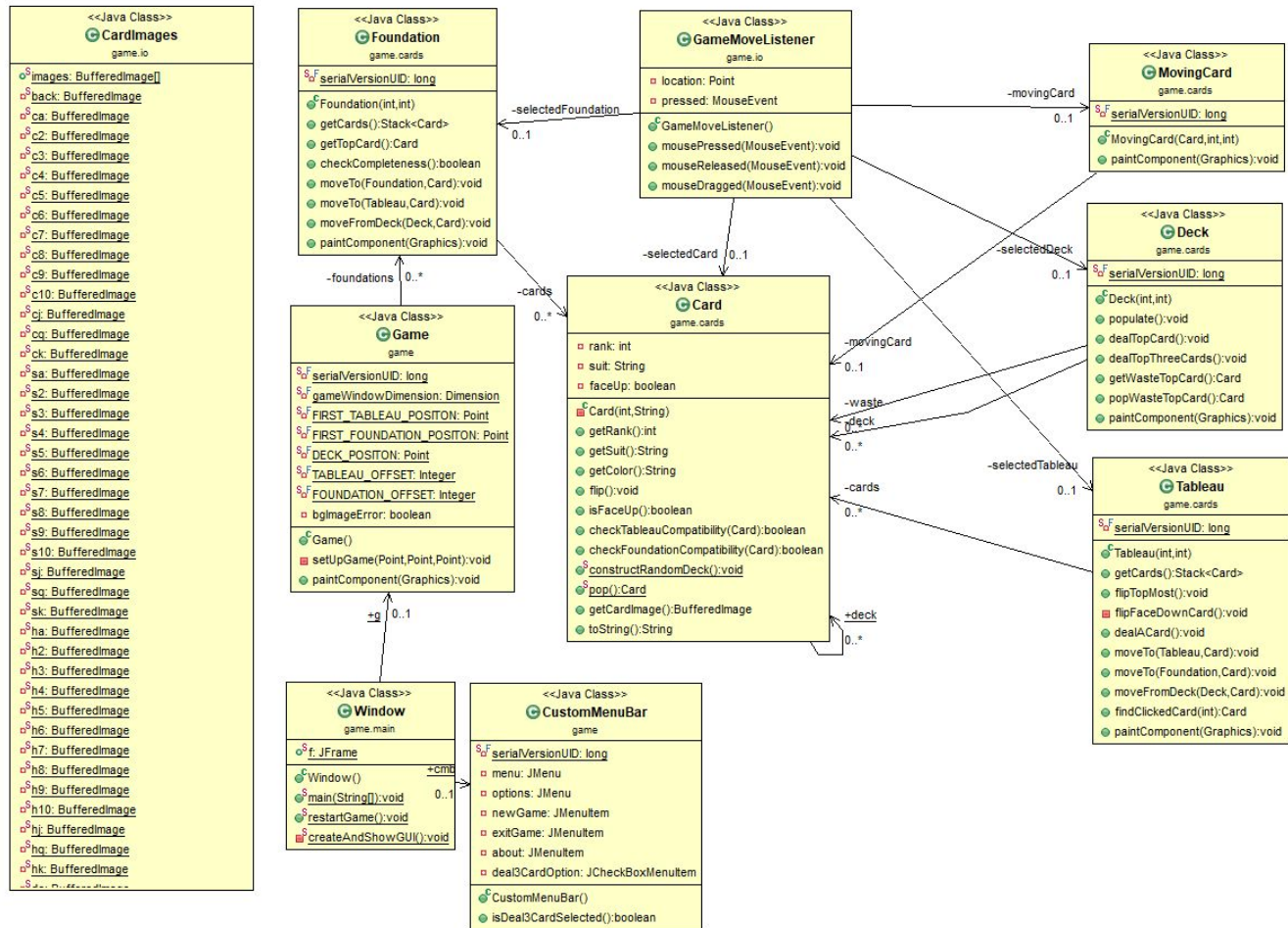
- ใช้ push() ในการนำการ์ดที่เหลือจากการแจกไปยังแถวต่างๆเข้าไปในสำหรับ
- ใช้ pop() เมื่อมีการคลิกที่กองการ์ด จะทำการ pop() ไฟล์บนสุดของสำหรับออกมาข้างๆกองการ์ด

ขั้นตอนวิธี (Algorithm)

- เริ่มจากการเพิ่มการ์ดทั้งหมดเก็บเข้าไว้ใน deck ก่อนจากนั้น เรียกใช้ Method shuffle เพื่อทำการสลับไพ่ในกองการ์ด
- จากนั้นจะทำการแจกไพ่ในสำรับลงมาในแต่ละแถว และการ์ดที่เหลือจะเก็บไว้ในที่ deck อย่างเดิม
- ในแต่ละแถวที่แจกออกมาจะมีการตรวจสอบว่า ถ้าการ์ดในแถวยังไม่หมดและการ์ดใบล่างสุดไม่ได้หงายอยู่ ให้ทำการหงายไพ่ใบล่างสุดนั้น
- ทำการเรียงไพ่แต่ละใบตามที่ระบุไว้ข้างต้นโดยที่ไพ่สีดำต้องเรียงสลับกับไพ่อสีแดง
- ไพ่ราศีมังกร สามารถนำมาอยู่เหนือไพ่ทั้ง 7 กองได้
- สี่กองข้างบนต้องเริ่มจากไพ่ราศีมังกร ถ้าเจอไพ่ราศีมังกรและแยกเอาไปวางไว้ใน 4 ช่องข้างบนแล้ว ให้หยิบไพ่ที่ดอกตรงกันกับไพ่ราศีมังกรข้างบนซ้อนทับต่อไปเรื่อยๆ

ขั้นตอนวิธี (Algorithm)

- หากจำนวนไฟที่หยายอยู่ไม่สามารถนำมาเรียงต่อกันได้ ก็สามารถคลิกที่กองการ์ด เพื่อ pop ใบบนสุดออกจากกองการ์ด เพื่อนำการ์ดใบนั้นมาเล่นกับไฟในแต่ละแถวได้
- ถ้ามีไฟใบไหนซ้อนอยู่ข้างหลัง ให้โยกย้ายไฟใบข้างหน้าไปมาจนกว่าจะหยิบใบที่ต้องการออกมาได้ แล้วเอาไปเรียงตามตำแหน่งต่อไป
- ถ้าหยิบไฟจากกองใดกองหนึ่งใน 7 กองขึ้นมาจนครบ ก็จะเกิดช่องว่างใช้วางไฟราศีคนแบกุงได้ (และเฉพาะไฟราศีคนแบกุงเดียวเท่านั้น)
- เกมจะจบก็ต่อเมื่อจำนวนไฟ 4 กองด้านบนเรียงจากไฟราศีมังกร(ล่างสุด) จนถึง ไฟราศีคนแบกุง(บนสุด)



03

ผลการดำเนินงาน

ภาพหน้าจอ (Screenshot)

หน้าของตัวเกม

เมื่อเล่นเกมจบ

วิเคราะห์ความเร็วหรือความซับซ้อนของโปรแกรม (Big O)

```
private void setUpGame(final Point tableauPos, final Point foundationPos, final Point deckPos) {  
    Card.constructRandomDeck();  
  
    final Tableau[] tableau = new Tableau[7];  
    for (int tableauIndex = 1; tableauIndex <= tableau.length; tableauIndex++) {  
        tableau[tableauIndex - 1] = new Tableau((int) tableauPos.getX() + Game.TABLEAU_OFFSET * (tableauIndex - 1), (int) tableauPos.getY());  
        for (int numberOfCards = 0; numberOfCards < tableauIndex; numberOfCards++) {  
            tableau[tableauIndex - 1].dealACard();  
        }  
        tableau[tableauIndex - 1].flipTopMost();  
        super.add(tableau[tableauIndex - 1]);  
    }  
  
    foundations = new Foundation[4];  
    for (int i = 0; i < foundations.length; i++) {  
        foundations[i] = new Foundation((int) foundationPos.getX() + Game.FOUNDATION_OFFSET * i, (int) foundationPos.getY());  
        super.add(foundations[i]);  
    }  
}
```

ได้ Big-O เท่ากับ $O(n \log n)$

วิเคราะห์ความเร็วหรือความซับซ้อนของโปรแกรม (Big O)

เมธอด setupGame ของคลาส Game.java

```
for (int i = 0; i < foundations.length; i++) {  
    if (!foundations[i].checkCompleteness()) {  
        isGameOver = false;  
    }  
}
```

ได้ Big-O เท่ากับ $O(n \log n)$

คลาส Game.java

```
public void populate() {  
    while (!Card.deck.isEmpty()) {  
        deck.push(Card.pop());  
    }  
}
```

ได้ Big-O เท่ากับ $O(n)$

วิเคราะห์ความเร็วหรือความซับซ้อนของ โปรแกรม (Big O)

มี 1 เมธอดที่มี Big-O เท่ากับ $O(n)$

มี 4 เมธอดที่มี Big-O เท่ากับ $O(n \log n)$

มี 2 เมธอดที่มี Big-O เท่ากับ $O(\log n)$

ส่วนเมธอดอื่นๆส่วนมากก็จะมี Big-O เท่ากับ $O(1)$

จึงสรุปได้ว่าตัวโปรแกรมของเรานั้นมีความเร็วอยู่ในระดับที่ “ปานกลางค่อนข้างเร็ว”

เนื่องจากความซับซ้อนของลูปนั้นมีอยู่แค่ไม่กี่เมธอดนั่นเอง

สรุปผล

- อธิบายความรู้ความเข้าใจใหม่ ๆ จากการทำงานในโครงการนี้
สิ่งใหม่ที่ได้จาก การทำงานในครั้งนี้ก็คือ การใช้ Data Structure Stack มาใช้ทำโปรเจค และ การใช้ BufferedImage ในการนำภาพมาใส่ในโปรแกรมนั่นเอง



อุปสรรค / ปัญหาระหว่างโครงการและวิธีแก้ปัญหา

- เนื่องจากสถานการณ์ปัจจุบันของการระบาดโควิด-19 ทำให้การพบกันของสมาชิกในกลุ่มลดลงและขาดความเข้าใจในการทำงานร่วมกัน

ทางแก้ไขปัญหา : ปรับตัวมาคุยการออนไลน์ผ่านโปรแกรม Discord และ Google Meet

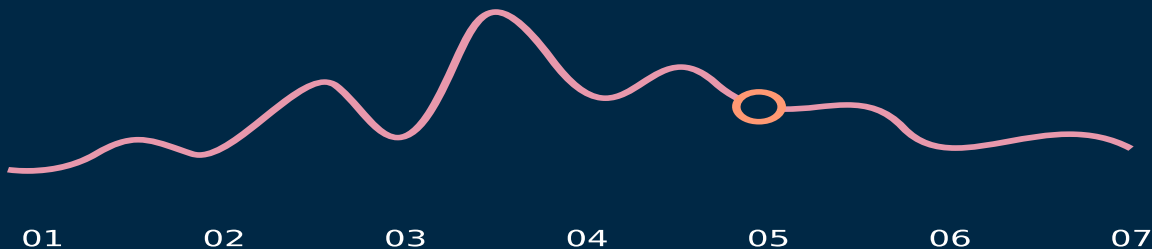
- เนื่องจากตัวโปรเจกต์นั้นทางเราไม่ได้เขียนขึ้นมาเองทำให้ไม่รู้ว่าโค้ดส่วนไหนใช้ทำอะไรนั่นเอง

ทางแก้ไขปัญหา : ทำการศึกษาตัวโปรแกรมควบคู่กับหาข้อมูลในอินเทอร์เน็ตเพื่อทำความเข้าใจในตัวโค้ด



การปรับปรุงโปรแกรมประยุกต์นี้เป็นไปได้ในอนาคต

เนื่องจากโปรแกรมนี้ฟังก์ชันที่มีอยู่น้อย เข้าใจง่าย ไม่ซับซ้อน พวกเราคิดว่าในอนาคตสามารถสร้างฟังก์ชันอื่นๆ ขึ้นมาเพิ่มได้ อาทิเช่น มีการจับเวลา มีปุ่ม Hint มีการรับข้อมูลผู้เล่น เป็นต้น



The background is a dark blue field decorated with a pattern of small squares and thin vertical lines. The squares are in three colors: light blue, pink, and orange. Some squares are solid, while others are hollow. The vertical lines are thin and white, extending from the top or bottom of the frame. The overall effect is a modern, minimalist aesthetic.

THANK YOU

Special Thanks

—AERIS107

For inspired and reference In This Project