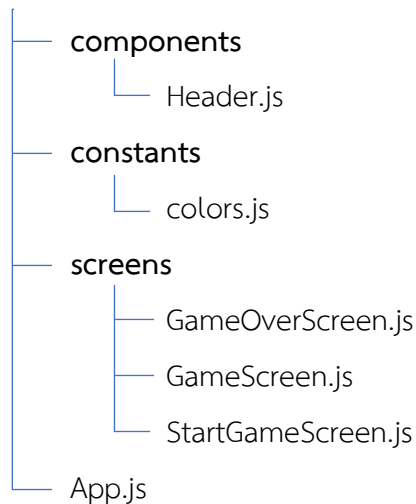


Lab 4 : Components Communication

จงเขียนโปรแกรม Guess a Number โดยเมื่อเริ่มเกม โปรแกรมจะทำการสุ่มตัวเลข 1-99 ขึ้นมา และให้ผู้เล่นเดาเลขที่โปรแกรมสุ่มขึ้นมา

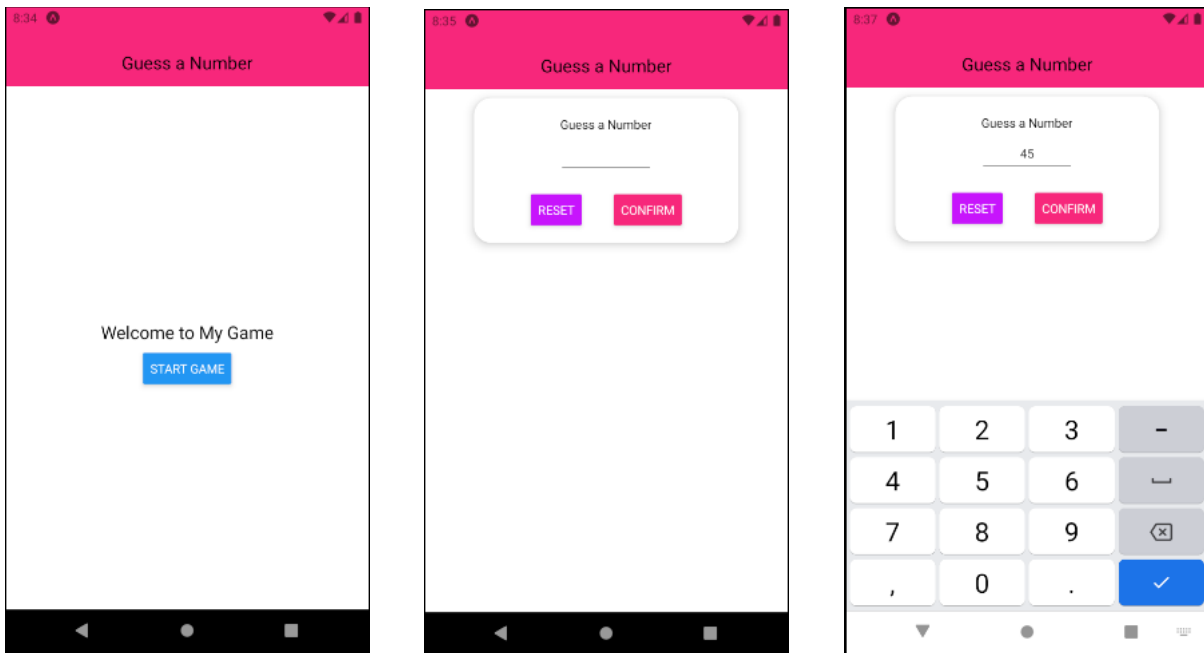
- กรณีที่ผู้เล่น เดาเลขที่มีค่าน้อยกว่าเลขโจทย์ของโปรแกรม จะแสดงข้อความ The answer is greater. และบอกจำนวนรอบที่ผู้เล่นเดา
- กรณีที่ผู้เล่น เดาเลขที่มีค่ามากกว่าเลขโจทย์ของโปรแกรม จะแสดงข้อความ The answer is lower. และบอกจำนวนรอบที่ผู้เล่นเดา
- กรณีที่ผู้เล่นเดาเลขได้ตรงกับเลขโจทย์ของโปรแกรม โปรแกรมจะแสดงหน้า GameOver ที่สรุปผลการเล่นของผู้เล่น

โครงสร้างของโปรแกรม

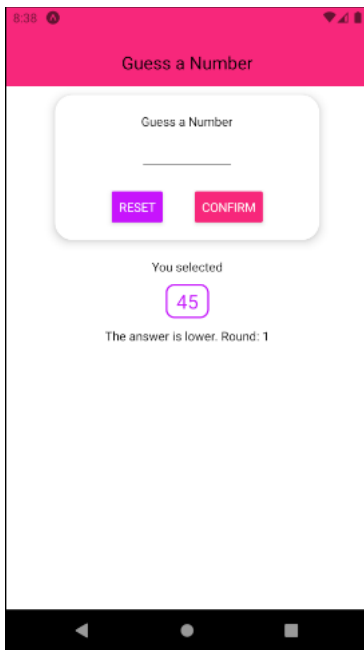


หมายเหตุ นักศึกษาสามารถแก้ไขโค้ดโปรแกรมที่ได้เตรียมไว้ให้ (OnLearn) ได้ตามความเหมาะสม

ตัวอย่างการทำงานของโปรแกรม

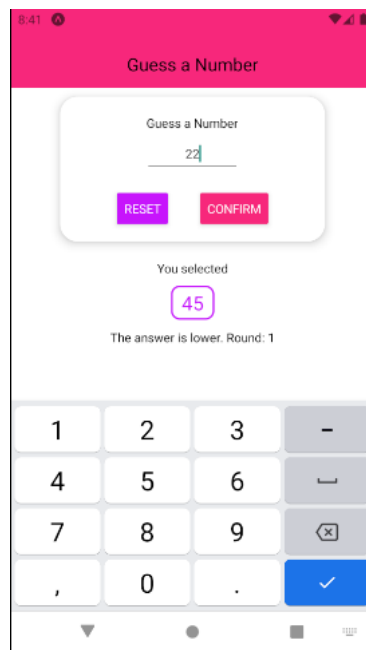


หน้า StartGame



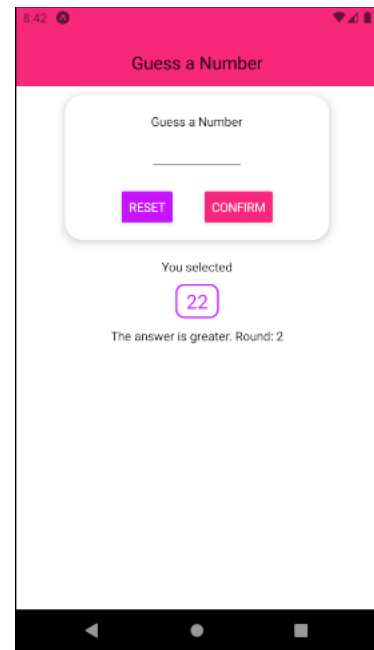
ผลลัพธ์เมื่อกด Confirm

หน้า GameScreen (คำตอบคือ 24)

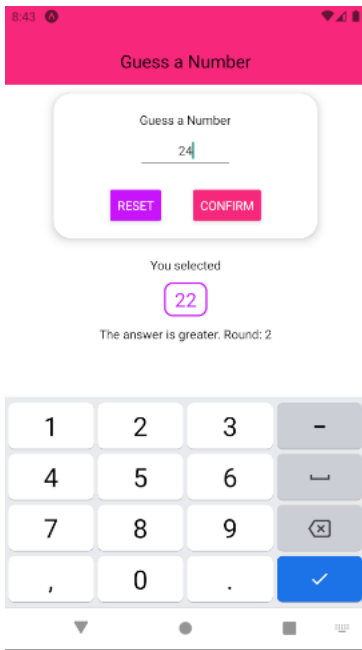


ผู้เล่นเดาเลข 22

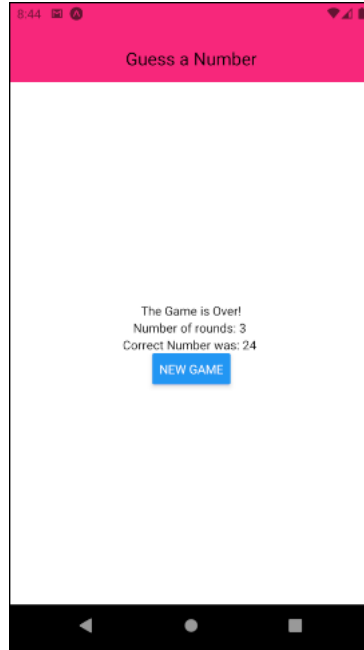
ผู้เล่นเดาเลข 45



ผลลัพธ์เมื่อกด Confirm

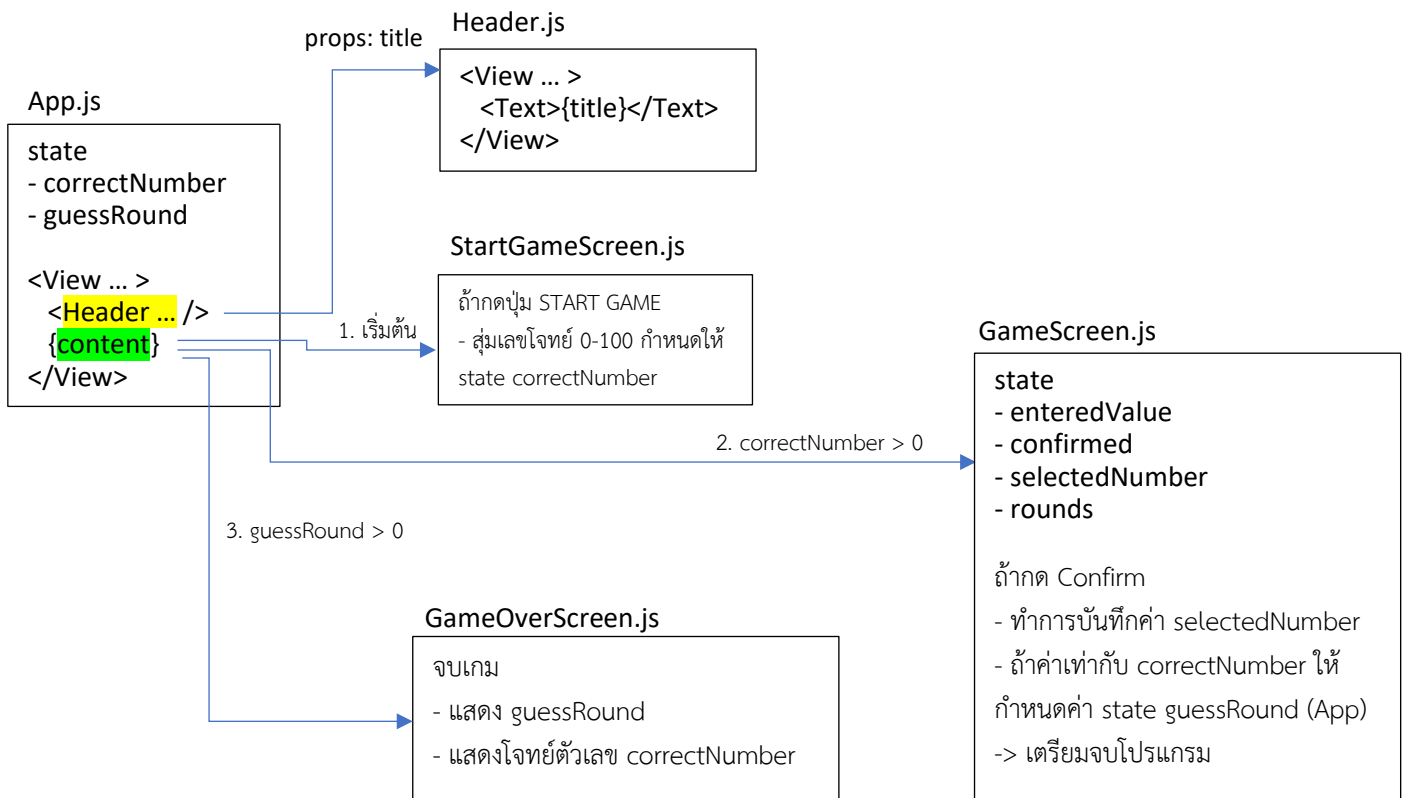


ผู้เล่นเดาเลข 24 (ตรงกับคำตอบ)



หน้า GameOver

ภาพรวมการทำงาน



ขั้นตอนการทำงานอย่างละเอียด

1. ทำการสร้างโปรเจกใหม่ และคัดลอกไฟล์โปรแกรมตามโครงสร้างข้างต้น (OnLearn)
2. ไฟล์ต่างๆ มีรายละเอียดดังนี้
 - Header.js เป็นคอมโพเนนต์แสดงส่วนหัวของโปรแกรม พร้อมแสดงชื่อโปรแกรม
 - color.js เป็นไฟล์ที่ใช้เก็บค่าสีที่จะใช้ในโปรแกรม (สามารถประยุกต์ในการทำ theme สีของโปรแกรมได้)
 - StartGameScreen.js เป็นคอมโพเนนต์แสดงหน้าแรกของโปรแกรม ซึ่งผู้เล่นสามารถกดปุ่มเริ่มเกมได้จากหน้านี้
 - GameScreen.js เป็นคอมโพเนนต์ที่ผู้เล่นสามารถเดาเลข และโปรแกรมแสดงผลการเดาตัวเลขของผู้เล่นไปเรื่อยๆ จนกว่าผู้เล่นจะเดาถูก
 - GameOverScreen.js เป็นคอมโพเนนต์ที่แสดงหน้าจบเกม พร้อมสรุปผลการเล่น
 - App.js เป็นส่วนคอมโพเนนต์หลักที่ทำการเลือก Screen ต่างๆ (StartGameScreen, GameScreen, GameOverScreen) มาแสดง

3. หลักการทำงานของ App.js

กำหนดให้ state ของ App ประกอบด้วย

- correctNumber ใช้เก็บตัวเลขที่เป็นโจทย์ของโปรแกรม โดยกำหนดค่าเริ่มต้นเป็น 0 และอัปเดตค่า state ด้วยฟังก์ชัน setCorrectNumber
- guessRound ใช้เก็บจำนวนรอบที่ผู้เล่นเดาตัวเลข โปรแกรม โดยกำหนดค่าเริ่มต้นเป็น 0 และอัปเดตค่า state ด้วยฟังก์ชัน setGuessRounds

หากพิจารณาที่ return()

```
return (  
  <View style={styles.screen}>  
    <Header title="Guess a Number" />  
    {content}  
  </View>  
);
```

จะทำการแสดงส่วนคอมโพเนนต์ Header ก่อน แล้วตามด้วยโค้ดที่เก็บในตัวแปร content ซึ่งจะเป็นการเรียก Screen ต่างๆ พร้อมกับกำหนด props ของแต่ละ Screen

- ค่าเริ่มต้นของ content คือการเรียกคอมโพเนนต์ StartGameScreen
- ถ้า correctNumber > 0 และ guessRounds <= 0 นั่นคือ เมื่อมีการกำหนดตัวเลขโจทย์แล้ว content จะเป็นการเรียกคอมโพเนนต์ GameScreen เพื่อให้ผู้เล่นได้เล่นเกม

- ถ้า `guessRounds > 0` ซึ่งค่า `guessRound` จะถูกกำหนดค่าเมื่อผู้เล่นเดาเลขได้ตรงกับที่โปรแกรมกำหนด `content` จะเป็นการเรียกคอมโพเนนต์ `GameOverScreen`

เขียนโปรแกรมเพื่อแสดงหน้า `StartGameScreen`

1. ในไฟล์ `App.js` กำหนดค่าให้ `content` ดังนี้

```
let content = <StartGameScreen onStartGame={startGameHandler} />;
```

กำหนดให้มี property ชื่อ `onStartGame` ซึ่งจะส่งฟังก์ชัน `startGameHandler` ซึ่งทำหน้าที่กำหนดค่าเลขโจทย์ของโปรแกรมให้กับสแตท `correctNumber`

2. ในไฟล์ `StartGameScreen.js` ให้เขียนโค้ดเพิ่มเติม กรณีที่มีการกดปุ่ม `START GAME` (

- ทำการสุ่มเลขโจทย์ ด้วยคำสั่ง `Math.floor(Math.random() * 100)` เก็บไว้ในตัวแปรหนึ่ง
- ทำการแสดงค่าเลขโจทย์ที่ `console` ด้วยคำสั่ง `console.log()`
- ทำการกำหนดค่าเลขโจทย์ให้กับ `correctNumber` ใน `App` ด้วยการเรียกผ่าน props ดังนี้
`props.onStartGame(ค่าเลขโจทย์);`

3. ทดลองรันโปรแกรม โปรแกรมจะแสดงหน้า `StartGame` และเมื่อกดปุ่ม `START GAME` จะมีการแสดงเลขโจทย์ที่ terminal ซึ่ง ณ ตอนนี ค่าสแตท `correctNumber` จะทำการเก็บเลขโจทย์เรียบร้อยแล้ว

เขียนโปรแกรมเพื่อแสดงหน้า `GameScreen`

จากการทำงานก่อนหน้านี้ ค่า `correctNumber` ได้ถูกอัปเดตแล้ว ซึ่งจะช่วยให้เงื่อนไข (`correctNumber > 0 && guessRounds <= 0`) ใน `App.js` เป็นจริง ขั้นตอนต่อไป จะทำการแสดงคอมโพเนนต์ `GameScreen` เพื่อเล่นเกม ดังนี้ (แก้ไขโปรแกรมใน `App.js`)

1. ให้เขียนโปรแกรมเพิ่มเติมเพื่อทำการเรียกคอมโพเนนต์ `GameScreen` โดยกำหนดให้มี property ดังนี้
 - `answer` เป็น property ที่ส่งค่าเลขโจทย์ของโปรแกรม
 - `onGameOver` เป็น property ที่ส่งตำแหน่งของฟังก์ชัน `gameOverHandler` ซึ่งจะถูกเรียกเมื่อจบเกม
2. ให้เขียนโปรแกรมเพิ่มเติมในฟังก์ชัน `gameOverHandler` โดย
 - ฟังก์ชันนี้จะรับ `numOfRounds` ซึ่งเป็นจำนวนรอบที่ผู้เล่นใช้ในการเดาจนกว่าจะตรงกับเลขโจทย์
 - ให้กำหนดค่าสแตท `guessRounds` ด้วยค่า `numOfRounds`
3. ทดลองรันโปรแกรม เมื่อผู้เล่นกดปุ่ม `START GAME` โปรแกรมจะเรียกคอมโพเนนต์ `GameScreen` ขึ้นมา โดยจะแสดงส่วนให้กรอกตัวเลขที่ต้องการเดา แต่ปุ่ม `RESET` และ `CONFIRM` ยังทำงานไม่สมบูรณ์

เขียนโปรแกรมปรับปรุงในไฟล์ GameScreen.js

ณ ตอนนี้ โปรแกรมสามารถแสดงหน้า GameScreen ได้แล้ว แต่ยังทำงานได้ไม่สมบูรณ์ ให้นักศึกษาปรับปรุงโปรแกรมใน **GameScreen.js** ดังนี้

1. จงสร้าง state ในคอมโพเนนต์ GameScreen ดังนี้
 - enteredValue ใช้เก็บข้อมูลที่พิมพ์ลงใน TextInput ก่อนที่จะกด CONFIRM โดยกำหนดค่าเริ่มต้นเป็น “ ” และอัปเดตค่า state ด้วยฟังก์ชัน setEnteredValue
 - selectedNumber ใช้เก็บข้อมูลที่ผู้เล่นเดา และทำการกด CONFIRM ยืนยันคำตอบ โดยไม่กำหนดค่าเริ่มต้น และอัปเดตค่า state ด้วยฟังก์ชัน setSelectedNumber
 - confirmed ใช้เก็บค่าบูลีนว่าผู้เล่นได้ทำการกดปุ่ม CONFIRM แล้วหรือไม่ โดยกำหนดค่าเริ่มต้นเป็น false และอัปเดตค่า state ด้วยฟังก์ชัน setConfirmed
 - rounds ใช้เก็บจำนวนครั้งที่ผู้เล่นเดาตัวเลข โดยกำหนดค่าเริ่มต้นเป็น 0 และอัปเดตค่า state ด้วยฟังก์ชัน setRounds
2. เขียนโปรแกรมเพิ่มในฟังก์ชัน return() ดังนี้
 - ให้เพิ่ม property ชื่อ value และ onChangeText เข้าไปใน TextInput
 - กำหนดให้ value = ค่าสแตต enteredValue
 - กำหนดให้ onChangeText = ฟังก์ชัน numberInputHandler
 - ให้เพิ่ม property ชื่อ onPress เข้าไปใน Button – Reset
 - กำหนดให้ onPress = ฟังก์ชัน resetInputHandler
 - ให้เพิ่ม property ชื่อ onPress เข้าไปใน Button – Confirm
 - กำหนดให้ onPress = ฟังก์ชัน confirmInputHandler
3. เขียนฟังก์ชัน numberInputHandler เพื่อเก็บค่าที่ผู้เล่นกรอก (inputText) ลงในสแตต enteredValue เมื่อการพิมพ์ค่าลงใน TextInput
4. เขียนฟังก์ชัน resetInputHandler เพื่อเคลียร์ค่าสแตต enteredValue ให้เป็น “ ” เมื่อมีการกดปุ่ม Reset
5. เขียนฟังก์ชัน confirmInputHandler เพื่ออัปเดตค่าสแตตต่างๆ เมื่อผู้เล่นกดปุ่ม Confirm ดังนี้
 - อัปเดตค่าสแตต selectedNumber ด้วยค่าที่ผู้เล่นเดา ซึ่งเกิดจากการแปลงค่าในสแตต enteredValue ให้เป็นตัวเลข ด้วยคำสั่ง parseInt(enteredValue)
 - อัปเดตค่าสแตต confirmed ให้เป็น true เพื่อบอกว่ามีการกด Confirm แล้ว
 - อัปเดตค่าสแตต enteredValue ให้เป็น “ ” เพื่อใช้รับค่าตัวเลขใหม่ต่อไป
 - อัปเดตค่าสแตต rounds ให้มีค่าเพิ่มขึ้น 1 รอบ

- นอกจากนี้ สามารถทดลองใช้คำสั่ง `Keyboard.dismiss()`; เพื่อทำการซ่อนแป้นพิมพ์หลังจากกดปุ่ม Confirm ได้ ซึ่งต้อง import Keyboard เข้ามาใช้ด้วย
6. หากพิจารณาฟังก์ชัน `return()` จะพบว่าส่วนล่างของฟังก์ชันนี้ จะมีการอ้างอิงถึงตัวแปร `confirmedOutput` ซึ่งจะใช้แสดงตัวเลขที่ผู้เล่นได้หายไป และแสดงผลการทายของผู้เล่น ว่าถูกต้องหรือไม่ กำหนดให้นักศึกษาเขียนโปรแกรมเพิ่มเติม หลังจากการประกาศตัวแปร `confirmedOutput` ดังนี้
- กรณีที่ผู้เล่นมีการกดปุ่ม Confirm (ค่าสแตต `confirmed` เป็นจริง)
 - ให้เพิ่มคำสั่งที่ทำการแสดงค่าตัวเลขที่ผู้เล่นได้เลือกไว้
 - ให้เขียนโปรแกรมเพิ่มเติม และแสดงข้อความผลลัพธ์การทายตัวเลข และจำนวนรอบในการทาย

หมายเหตุ ในส่วนนี้ นักศึกษาต้องเขียนโปรแกรมที่ตรวจสอบคำตอบของผู้เล่นเพิ่มเติม อีกทั้ง กรณีที่ผู้เล่นตอบถูกต้องจะต้องมีการเรียกฟังก์ชัน `gameOverHandler` ใน `App.js` เพื่ออัปเดตสแตต `guessRounds` เพื่อใช้ในการเปลี่ยนไปแสดงหน้า `GameOverScreen` ต่อไป

7. ทดลองรันโปรแกรม ผู้เล่นจะสามารถทายตัวเลขได้ และโปรแกรมสามารถตอบได้ว่าตัวเลขที่ทายมาถูกต้องหรือไม่ และคำตอบจริงๆ ของโปรแกรม มีค่ามากกว่า หรือน้อยกว่าตัวเลขที่ผู้เล่นทายไป แต่ในกรณีที่ผู้เล่นทายถูก โปรแกรมยังทำงานไม่สมบูรณ์ เนื่องจากยังไม่ได้กำหนดการเรียกคอมโพเนนต์ `GameOverScreen` ซึ่งจะกล่าวถึงในลำดับต่อไป

เขียนโปรแกรมเพื่อแสดงหน้า `GameOverScreen`

จากการทำงานก่อนหน้า เมื่อผู้เล่นทายเลขถูกจะทำการอัปเดตสแตต `guessRounds` ซึ่งจะทำให้เงื่อนไข (`guessRounds > 0`) ใน `App.js` เป็นจริง ขั้นตอนต่อไป จะทำการแสดงคอมโพเนนต์ `GameOverScreen` เพื่อสรุปผลลัพธ์การเล่น เกม ดังนี้ (แก้ไขโปรแกรมใน `App.js`)

1. ให้เขียนโปรแกรมเพิ่มเติมเพื่อทำการเรียกคอมโพเนนต์ `GameOverScreen` โดยกำหนดให้มี property ดังนี้
 - `rounds` เป็น property ที่ส่งค่าจำนวนรอบที่ผู้เล่นทายตัวเลข
 - `answer` เป็น property ที่ส่งค่าเลขโจทย์ของโปรแกรม
 - `onRestart` เป็น property ที่ส่งตำแหน่งของฟังก์ชัน `configureNewGameHandler` เมื่อผู้เล่นต้องการเล่นเกมใหม่
2. ให้เขียนโปรแกรมเพิ่มเติมในฟังก์ชัน `configureNewGameHandler` โดย
 - อัปเดตค่าสแตต `guessRounds` ให้เป็น 0 เพื่อใช้ตั้งต้นในการเริ่มเกมใหม่
 - อัปเดตค่าสแตต `correctNumber` ให้เป็น 0 เพื่อใช้ตั้งต้นในการเริ่มเกมใหม่

3. ทดลองรันโปรแกรม เมื่อผู้เล่นทายตัวเลขตรงกับตัวเลขที่โปรแกรมกำหนดไว้ จะมีการแสดงหน้า GameOverScreen แต่ยังไม่แสดงผลสมบูรณ์

เขียนโปรแกรมปรับปรุงในไฟล์ GameOverScreen.js

ให้นักศึกษาเขียนโปรแกรมเพิ่มเติมในไฟล์ GameOverScreen.js เพื่อแสดง

- จำนวนรอบที่ผู้เล่นใช้ในการทายตัวเลข (แสดงข้อความ Number of rounds: ...)
- เลขโจทย์ที่โปรแกรมกำหนดไว้ (แสดงข้อความ Correct Number was: ...)
- เพิ่มปุ่ม NEW GAME หากกดปุ่มนี้โปรแกรมจะทำการเริ่มเกมใหม่