**โจทย์**

เราต้องการสร้างเกม กดรัวแข่งกับเครื่อง โดยเกมได้ถูกออกแบบให้มีส่วนประกอบดังนี้

Progressbar ที่มีค่าตั้งแต่ 0 ถึง 100 โดยเริ่มต้นค่าจะอยู่ที่ 50



จำนวนเกมที่เครื่องชนะมา

จำนวนเกมที่คนชนะมา

เลือกความยากง่าย สามารถเปลี่ยนระหว่างการเล่นตอนไหนก็ได้ ตอนเริ่มเกมให้มีค่าเท่ากับ 1

ปุ่มเพื่อกดเริ่มเล่น หรือหยุดการเล่นไว้ชั่วคราว ถ้าหยุดแล้ว จะไม่มีใครสามารถเลื่อน progressbar ได้ ส่วนถ้ากดเล่นแล้ว progressbar จะเลื่อนขึ้นเป็นจำนวนหน่วยเท่ากับความยาก ต่อหนึ่งวินาที

ปุ่มไว้ให้กดรัว กดหนึ่งทีค่า progressbar จะเลื่อนลง 1หน่วย

การทำงานของโปรแกรมได้ถูกออกแบบไว้แล้ว โดยแบ่งออกเป็นสี่คลาสดังนี้ (ดูคำอธิบายให้ครบทุกคลาส)

1. คลาส UserInterface

เป็นตัวสร้างส่วนประกอบของหน้าจอทั้งหมด สร้างวัตถุที่ใช้เป็นข้อมูลร่วมกันระหว่าง thread และสร้าง thread ต่างๆสำหรับโปรแกรมนี้

**สิ่งที่นักเรียนต้องทำในคลาสนี้ (10 คะแนน)**

* เขียนโค้ดสำหรับสร้าง buttonPanel ซึ่งจะต้องมีปุ่ม “Press” ปุ่ม “play/pause” และ radio button ทั้งหมด โดย radio button จะเลือกได้แค่อันเดียวในเวลาหนึ่งเท่านั้น ทำให้รูปร่างออกมาดังรูปข้างบน
* สร้าง object ที่เป็นชนิด Score ขึ้นมา (ดูคำอธิบายคลาส Score ได้ในส่วนต่อไป) เพื่อไว้ใช้เก็บดาต้าที่นับคะแนนและดาต้าอื่นๆที่ใช้ร่วมกันระหว่าง thread
* ปุ่มแต่ละปุ่ม เมื่อถูกกด จะไปเปลี่ยนสถานะของ object ของคลาส Score ที่เราใช้ ตามความเหมาะสม
* สร้าง thread ที่เป็นฝ่ายตรงข้ามของผู้เล่นขึ้นมา แล้วสั่งรัน
* สร้าง thread สำหรับอัพเดท progressbar และคะแนนส่วนอื่นขึ้นมา แล้วสั่งรัน
1. คลาส Score

เป็นตัวจัดเก็บค่าคะแนนและสถานะในการเล่นเกมนี้ ทุก thread จะทำงานกับข้อมูลออบเจ็กต์ของคลาสนี้เป็นหลัก โดยออบเจ็กต์ของคลาสนี้มีตัวแปรสามตัวคือ

* counter เป็นค่า 0 – 100 ซึ่งจะใช้เป็นค่าของ progressbar นั่นเอง
* level มีค่า 0 – 3 เป็นค่าระดับความยากง่ายของเกม
* playing เป็น boolean ที่บอกว่าขณะนี้ เล่นอยู่หรือหยุดเกมอยู่ (play หรือ pause อยู่)

**สิ่งที่นักเรียนต้องทำในคลาสนี้ (5 คะแนน)**

* เขียนคอนสตรัคเตอร์ ให้เซ็ตค่าต่างๆเป็นค่าที่ทำให้เกมสามารถเริ่มได้ดี
* เขียนเมธอด get/set ตามที่เห็นว่าจำเป็น
* เขียนเมธอด decrement ซึ่งจะเอาไว้ใช้ตอนปุ่ม “Press” ถูกกด การทำงานคือ จะลดค่า counter ลงหนึ่ง (ทำงานเมื่อ playing เป็นจริงเท่านั้น เพราะถ้าไม่ได้เล่นอยู่ จะไปเปลี่ยนค่า counter ทำไม)
* เขียนเมธอด aiIncrement ซึ่งจะเอาไว้ใช้เพิ่มค่า counter (จะถูกเรียกโดย thread ที่เป็นฝ่ายตรงข้ามของผู้เล่นไง) โดยจะทำงานเมื่อ playing เป็นจริงเท่านั้น และเมื่อทำงาน จะทำการเพิ่มค่า counter ไปเท่ากับค่าระดับความยากของเกม
* อย่าลืมว่า ตัวแปรทุกตัวของออบเจ็กต์ของคลาสนี้สามารถถูกอ่านและเขียนได้จากหลายๆ thread พร้อมกัน ดังนั้น จึงควรทำการเขียนโค้ดเพื่อให้แน่ใจว่า thread 2 thread ไม่สามารถเข้าถึงข้อมูลในเวลาเดียวกันได้
1. คลาส Enemy

เป็นคลาสที่นิยาม thread ที่ทำตัวเป็นฝ่ายตรงข้ามของผู้เล่นนั่นเอง มีตัวแปรคือ score ซึ่งเป็นข้อมูลคะแนนที่ตัวฝ่ายตรงข้ามนี้จะเปลี่ยนแปลงระหว่างการเล่นเกม โดยในการใช้งานนั้น score จะรับพอยต์เตอร์มาจากออบเจ็กต์ของคลาส Score ที่สร้างในคลาส UserInterface

**สิ่งที่นักเรียนต้องทำในคลาสนี้ (5 คะแนน)**

* เขียนคอนสตรัคเตอร์ให้ถูกต้อง
* เขียนเมธอด get/set ตามที่จำเป็น
* เขียนเมธอดสำหรับรัน thread โดย ให้เรียกเพิ่ม score ทุกๆหนึ่งวินาทีให้ถูกต้อง
1. คลาส UpdateInterface

เป็น thread ที่เอาไว้เขียน progressbar และเขียนคะแนนว่าคนกับคอม ชนะมาคนละกี่นัดแล้ว มีตัวแปรสี่ตัวคือ

* mybar คือตัว progress bar ที่ต้องอัพเดทเรื่อยๆ
* myscore คือ ออบเจ็กต์ของคลาส Score ที่ต้องใช้ในการอัพเดทค่า mybar
* human คือ text field ที่ใช้บอกว่า คน ชนะมากี่เกมแล้ว
* ai คือ text field ที่บอกว่า ตัวฝ่ายตรงข้ามชนะมากี่เกมแล้ว

**สิ่งที่นักเรียนต้องทำในคลาสนี้ (10 คะแนน)**

* เขียนคอนสตรัคเตอร์ให้ถูกต้องใช้งานได้จริง
* เขียนโค้ดให้ thread รัน โดยทุกๆ 0.1 วินาที ต้องเซ็ตค่าของ progress bar ตามค่าที่ได้จาก score แล้วตรวจด้วยว่ามีคนชนะหรือยัง (progress bar ถึง 100 หรือ ถึง 0) ถ้ามีต้องทำดังนี้:
	+ ถ้าเครื่องชนะ ให้ pop dialog บอกว่า “You lose.” แล้วพอกดปิด dialog ก็ไปเพิ่มจำนวนเกมที่เครื่องชนะที่หน้าจอ จากนั้นก็เซ็ตให้โปรแกรมอยู่ในสถานะเหมือนการเริ่มเกมใหม่(เกมต้องหยุด)
	+ ถ้าคนชนะ ให้ pop dialog บอกว่า “You win.” แล้วพอกดปิด dialog ก็ไปเพิ่มจำนวนเกมที่คนชนะที่หน้าจอ จากนั้นก็เซ็ตให้โปรแกรมอยู่ในสถานะเหมือนการเริ่มเกมใหม่(เกมต้องหยุด)

สภาพหน้าจอในระหว่างเล่น แสดงในรูปข้างล่าง



สภาพเมื่อมีการชนะกัน แสดงในรูปด้านล่าง





จำนวนการชนะเปลี่ยนแล้ว

วิธีการส่ง ให้ export jar ส่งที่ progmethcp@gmail.com ภายในวันอังคารที่ 20 กันยายนนี้ โดยตั้งชื่อ subject mail ว่า <id>\_rapidClick และไฟล์ jar ก็ตั้งชื่อเช่นเดียวกัน