

**โครงงาน**

Mini Project

**จัดทำโดย**

6504062620078 นาย พงศ์ภรณ์ แย้มประดิษฐ์

**เสนอ**

ผู้ช่วยศาสตราจารย์ สถิต ประสมพันธ์

**วิชา Object Oriented Programming**

ภาคเรียนที่ 1 / 2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**เกี่ยวกับโครงงาน**

ชื่อโปรเจค : Mole’s Smash Game

นำเสนอโดย : นาย พงศ์ภรณ์ แย้มประดิษฐ์

อาจารย์ผู้สอน : ผู้ช่วยศาสตราจารย์ สถิต ประสมพันธ์

**บทที่ 1 ที่มาและความสำคัญของโครงงาน**

โครงงานนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างชิ้นงานนรูปแบบเว็บไซต์การคำนวณดเชิงคณิตศาสตร์ ผู้จัดทำจึงได้สร้างชิ้นงานนี้ขึ้นมา

**ประเภทโครงงาน :**

* โครงการทางวิทยาศาสตร์ และ เทคโนโลยี
* Project game

**ประโยชน์ :**

1. เพื่อความสนุกสนาน
2. ได้เรียนรู้เกี่ยวกับการสร้าง OOP
3. ช่วยในการวางแผนการทำงาน และ การคิดแบบเป็นลำดับขั้นตอน

**ตารางแผนการทำงานในเดือนกันยายน-ตุลาคม**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ลำดับ** | **รายการ** | **25 ก.ย.-30 ก.ย.** | **2 ต.ค.- 10 ต.ค.** | **11 ต.ค.-14 ต.ค.** | **> 15 ต.ค.** |
| 1. | หาจัดทำรูปแบบตัวละครและกราฟิกต่างๆ |  |  |  |  |
| 2. | ศึกษาการเขียนโปรแกรมและค้นหาข้อมูลที่เกี่ยวข้อง |  |  |  |  |
| 3. | ลงมือเขียนโปรแกรม |  |  |  |  |
| 4. | จัดทำเอกสาร |  |  |  |  |
| 5. | ตรวจสอบและปรับปรุงแก้ไขข้อผิดพลาด |  |  |  |  |

**บทที่ 2 ส่วนการพัฒนา**

**เนื้อเรื่องย่อ**

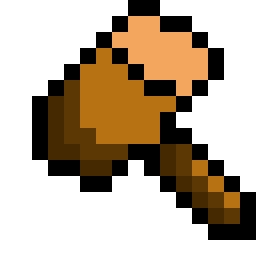
เกมแนว Action โดยให้เรารับบทเป็นคนสวน ซึ่งได้พบกับหลุมปริศนาและได้มีตัวตุ๋นออกมาจำนวนมาก ดังนั้นคนสวนจึงต้องใช้ค้อนสำหรับการตีหรือไล่เจ้าตัวตุ๋นออกไปให้มากที่สุด

**วิธีการเล่นใช้**

บังคับด้วยการใช้เคอเซอร์เมาส์ในการคลิกซ้ายตีตัวตุ๋นที่โพล่ออกมาจากหลุม

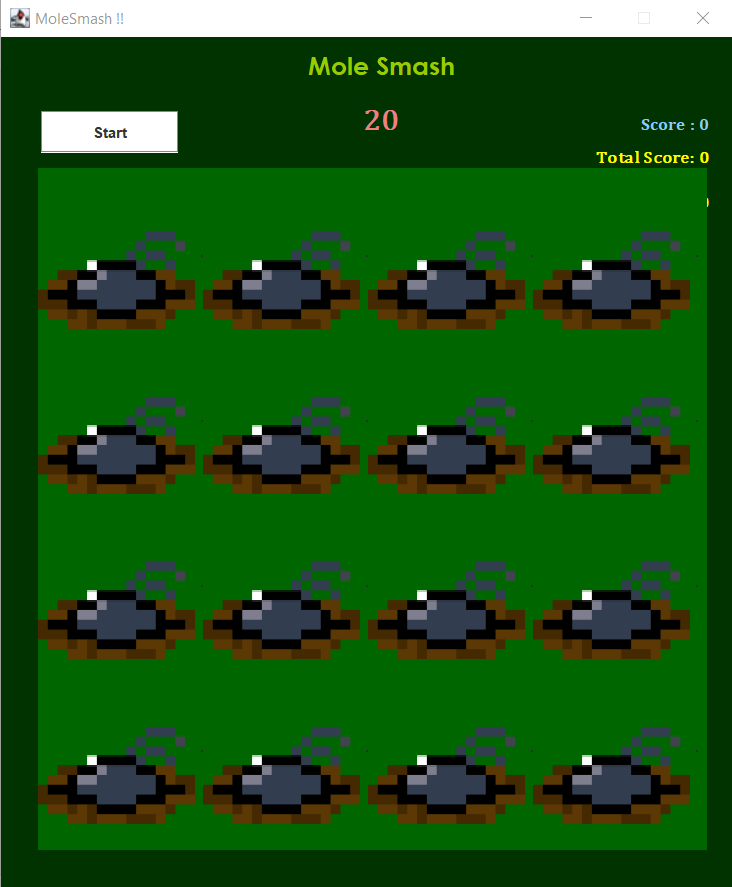
**Story Borad**

ตัวละคร****

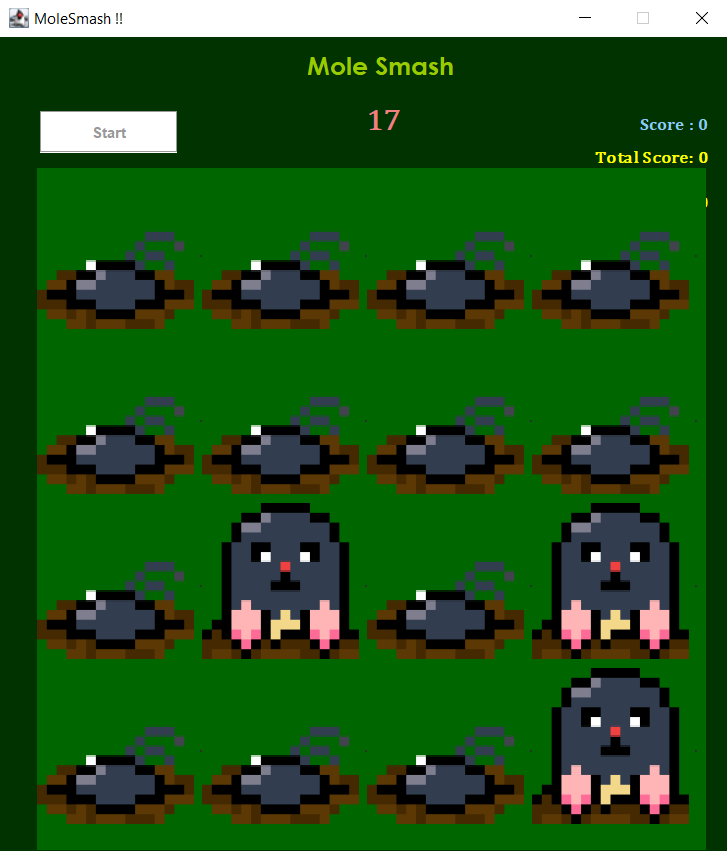
****

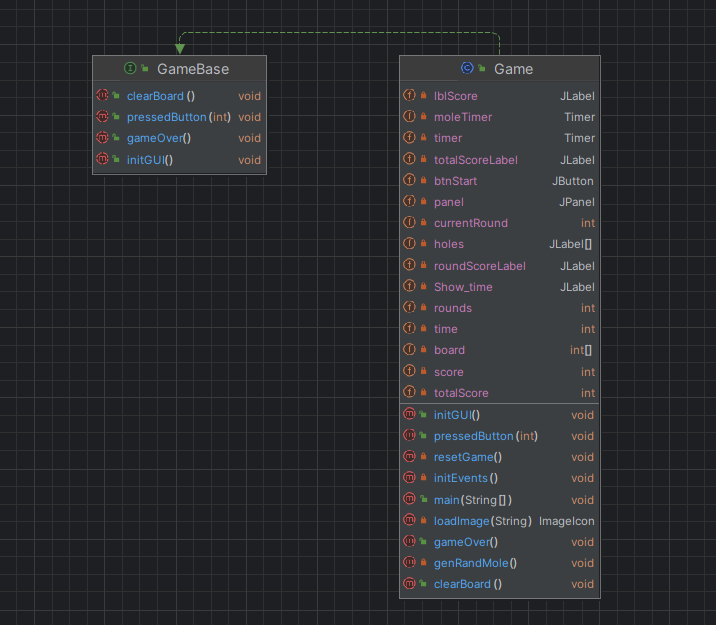
ค้อน

เจ้าตัวตุ่น

ฉาก – แสดงหน้าเกมเริ่มต้นก่อนกด Start ( การเกิดตัวตุ่นตอนโพล่ + การลดเวลา + SCORE การนับคะแนน )

* เลข 20 ก่อนกดเริ่มคือ ใช้เวลาเล่นในแต่ละรอบ 20 วินาที
* เมื่อกดปุ่ม Start จะเกิดเหตุการณ์ขึ้นคือ ตัวตุ่นจะโพล่ออกมา เวลาจะเริ่มนับถอยหลัง



**Class Diagram**

**คำอธิบาย Class Diagram**

* Game เป็นคลาสที่สืบทอดจาก JFrame และ implements GameBase interface.

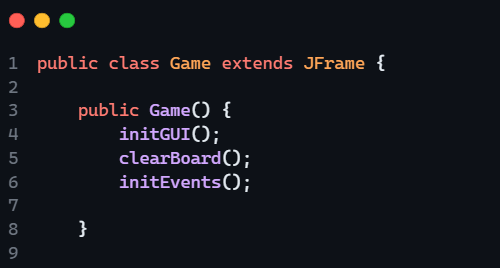
คลาส Game ในที่นี้ถูกใช้เพื่อจัดเก็บและจัดการข้อมูลที่เกี่ยวข้องกับเกม "Mole Smash"

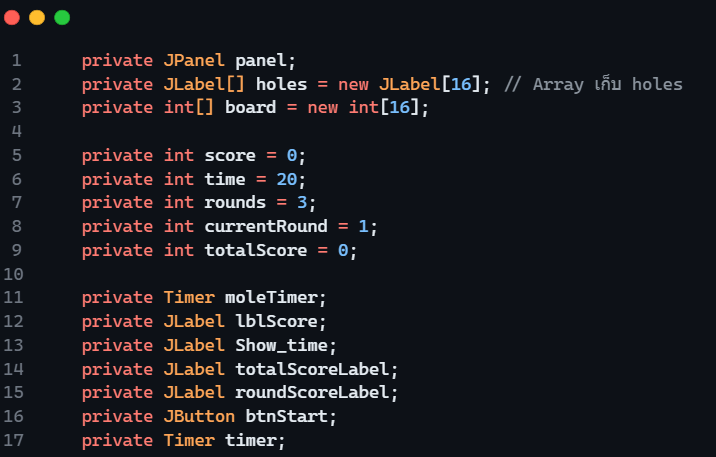
* GameBase เป็น interface ที่กำหนดเมธอดที่ Game ต้องมีการแสดงตัวแปรและเมธอดต่าง ๆ ของ Game บน Class diagram ด้วย

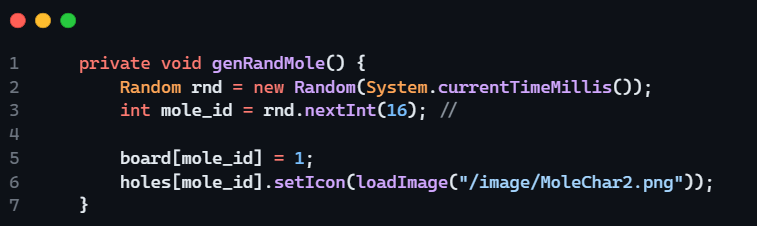
และ คลาส GameBase เป็นอินเทอร์เฟซที่ถูกนำมาใช้เพื่อกำหนดการทำงานของเกม "Mole Smash" และคลาส Game ได้ทำการ implement อินเทอร์เฟซนี้ ดังนั้น Game จึงต้องทำการปรับเปลี่ยนการทำงานตามที่กำหนดใน GameBase

**รูปแบบการพัฒนา > Application**

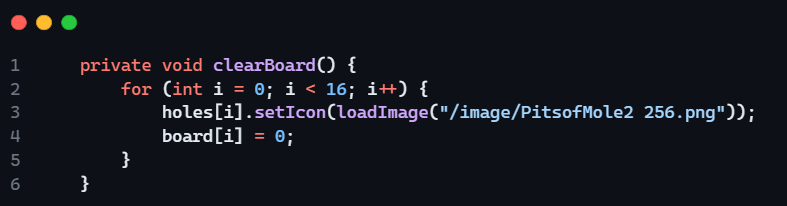
* ส่วนของโปรแกรมที่มีการใช้
* **Constructor**
* Constructor เป็นแบบไม่มี parameter ที่ใช้คือสำหรับการสร้าง Object ของ class Game ซึ่งมีหน้าที่เริ่มต้นสถานะเกม initGUI() กำหนด GUI ภายในเกม / Clearborad() รีเซตข้อมูล / initEvents() กำหนดเหตุการณ์ที่เกิดขึ้น



* **Encapsulation**
* ****ในโปรแกรมใช้ตัวแปรเป็น private เพื่อทำให้ไม่สามารถเข้าถึงได้จากภายนอกคลาส.
* มีการประกาศใช้ Private Method เพื่อควบคุมการเข้าถึงและการแก้ไขข้อมูลในคลาส Game ให้เกิดขึ้นเฉพาะภายในคลาสนั้นเท่านั้น

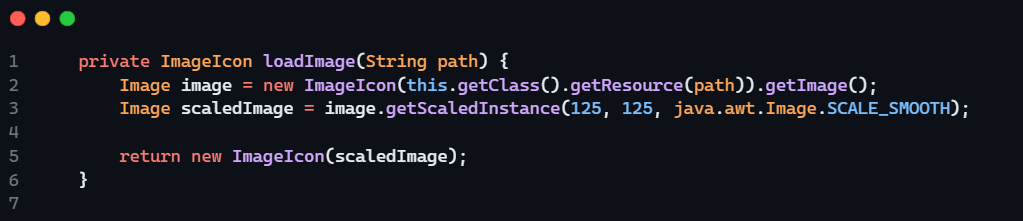


****

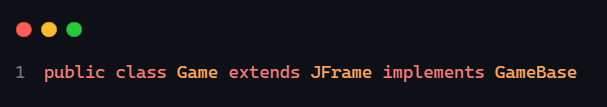
****

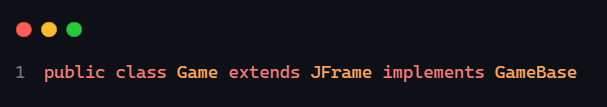
* **Composition**
* ใช้ในการสร้าง GUI components โดย Game class



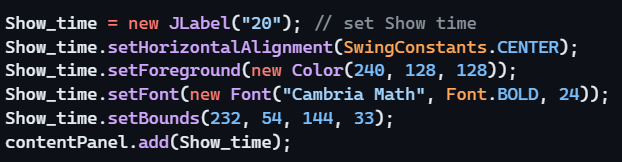
****

* การใช้ประกาศตัวแปร holes และ board ในคลาส Game เป็น JLabel [ ] และ board เป็น int [ ] คือการใช้ Composition เพื่อเก็บ Object ของคลาส JLabel และ int
* **Inheritance**



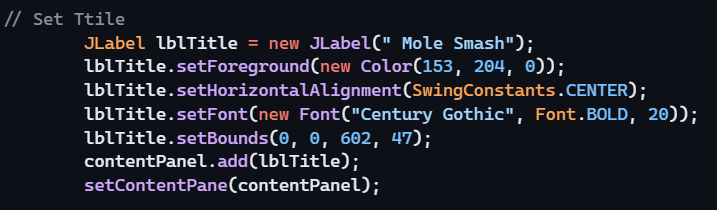
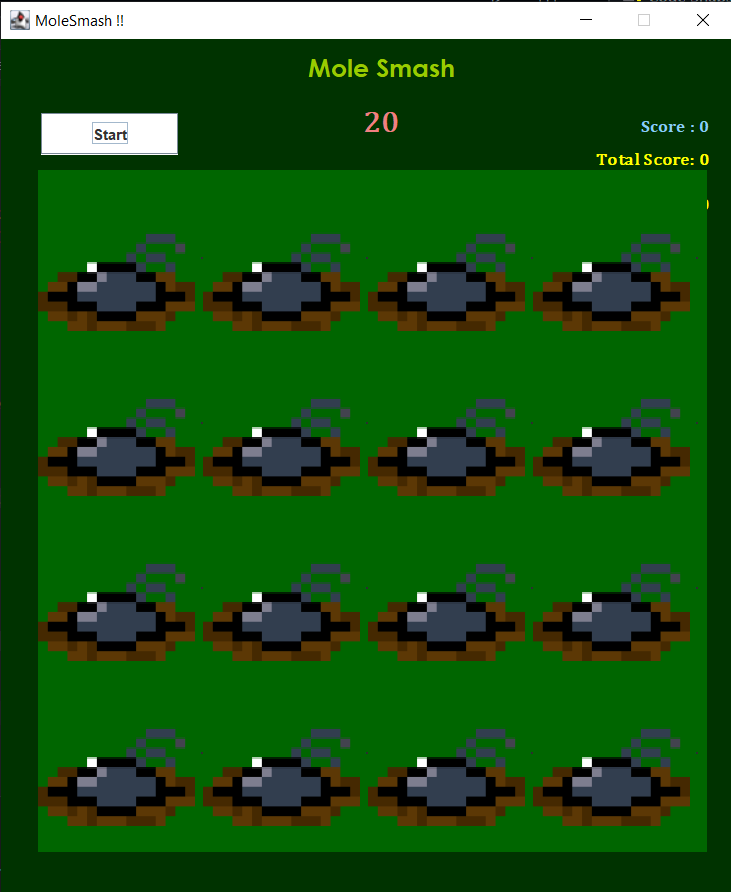
* คลาส Game สืบทอดจาก JFrame มันเป็นประเภทของ JFrame การสืบทอดเป็นวิธีที่จะใช้โค้ดซ้ำและสร้างความสัมพันธ์ระหว่างคลาส
* **Polymorphism**
* โดยการประกาศตัวแปร game โดยให้มีประเภท GameBase ซึ่งเป็น interface ที่มีเมทอดทั้ง 4 ตัวที่เป็น abstract และได้มีการนิยามทั้งหมดในคลาส Game ที่ implement
* **Abstract**

****

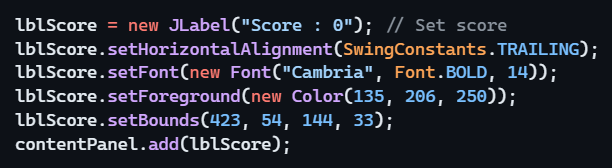
* การใช้งาน abstract ใน interface มีทั้งการประกาศหรือเปลี่ยนแปลง interface และการสร้างคลาสที่ implement interface
* **ส่วนประกอบของโครงสร้างของ GUI ประกอบด้วย Component**

สร้างข้อความ JLabel แสดงเวลา

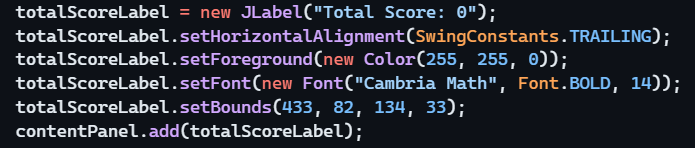
สร้างหัวข้อ JLabel ของเกมที่มีชื่อว่า Mole Smash

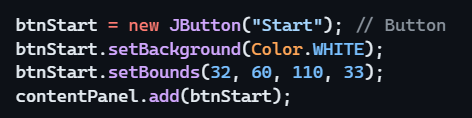
****

สร้างข้อความ JLabel แสดง score

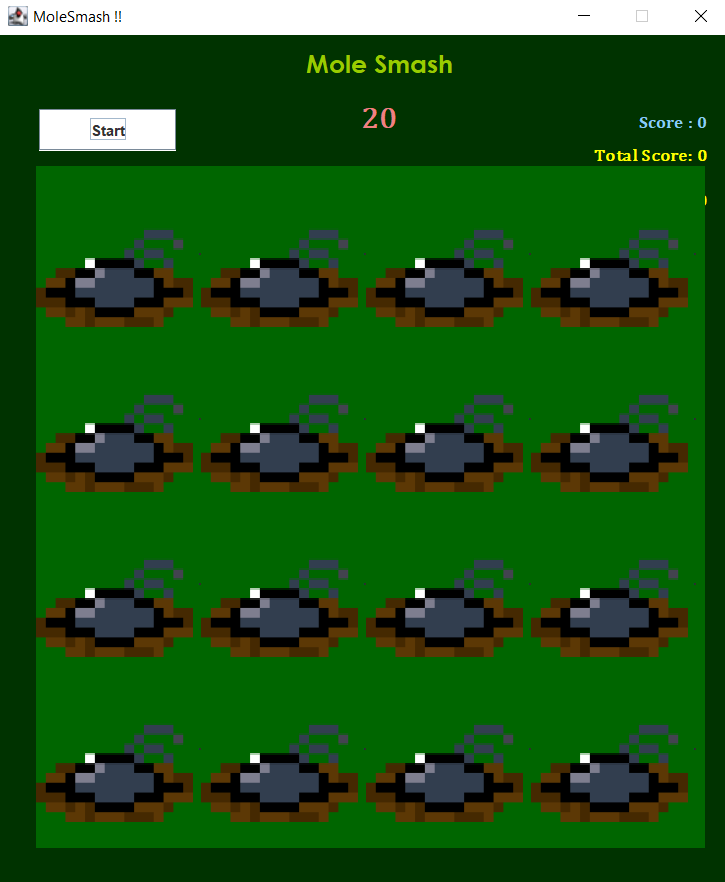
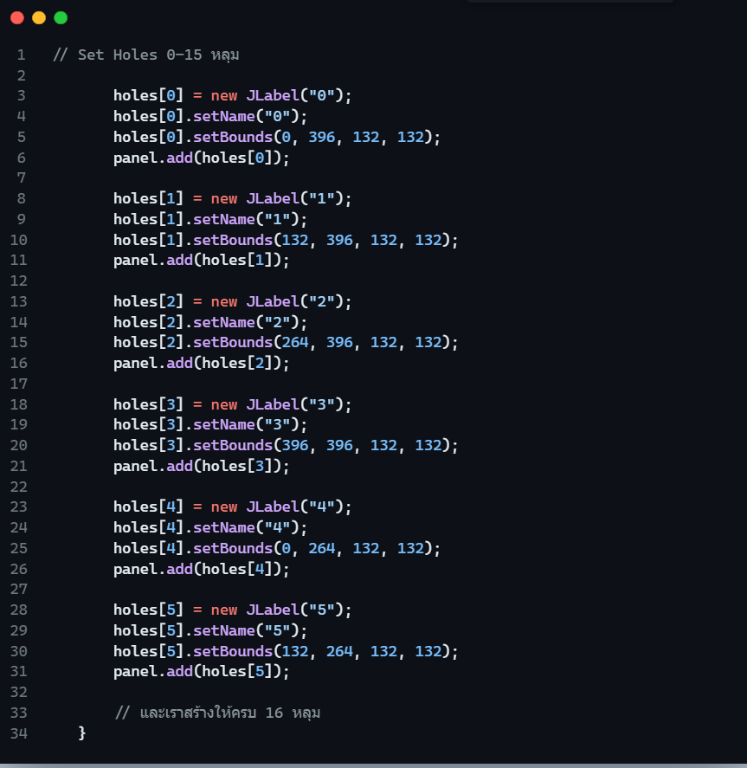
****

**\**

สร้างข้อความ JLabel แสดง Total score****

****

สร้าง JButton กดปุ่ม start

******- การสร้างหลุมทั้ง 16 หลุม** ตั้งแต่ holes[0] ถึง holes[15]

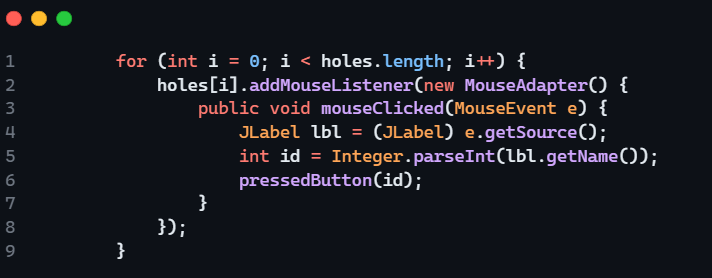
* **รวมถึงการเรียกใช้ clearBoard();**

คือการ set รูปภาพเริ่มต้นของหลุมใน holes ทุกตัว ดยกำหนดให้แสดงรูปภาพของหลุมเป็นไฟล์ png บ่งบอกถึงหลุมที่ไม่มีตัวตุ่นโพล่มา และทำการกำหนดค่า board ให้ทุก index เป็น 0



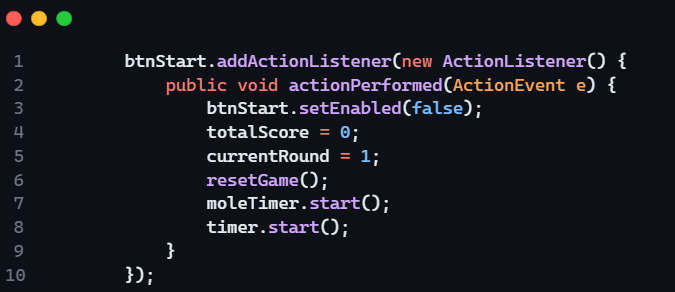
* **อธิบาย Event handling ที่มีในหน้าจอ**
* โดย Event ที่เกิดขึ้นทั้งหมดจะอยู่ใน function ของ initEvents(); ซึ่งจะมีองค์ประกอบต่างๆก็คือ

**Mouse Click Event สำหรับ holes (หลุม)**

****

* การทำ Loop โดยผมได้กำหนด Mouse Click Event Listener สำหรับทุก JLabel ที่เป็น holes (หลุม) ใน array เมื่อเรามีการคลิกที่หลุม จะเรียก pressedButton(); โดยส่งหลุมที่ถูกคลิกไป (id) เพื่อไปคิดคะแนนต่อ

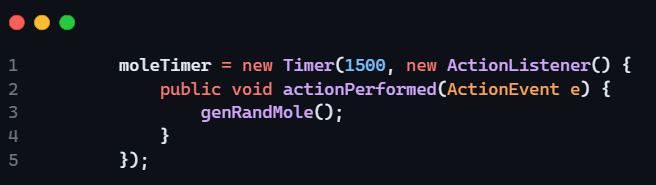
**Start Button Click Event**

****

* กำหนด ActionListener สำหรับปุ่ม "Start" ก็คือ btnStart เมื่อถูก Click จะทำการปิดการใช้งานปุ่ม "Start", รีเซ็ตค่าเกม, เริ่มการทำงานของ moleTimer และ timer เพื่อเริ่มเกม.

**Timer Tick Event**

* กำหนด ActionListener สำหรับ timer, ทำงานทุก 1 second = 1000 millisecond แล้วก็มีการจะทำการลดเวลาและตรวจสอบว่าถึงเวลาหรือไมถ้าถึงเวลาจะทำการหยุด timer และ moleTimer, และตรวจสอบว่าเกมเล่นได้ต่อไปจนถึงรอบสุดท้ายของเกมส์ คือถ้ายังไม่ถึงรอบสุดท้าย เราจะทำการรวมคะแนนในแต่ละรอบมาบวกเข้ากัน แล้วก็ยังมีการลดเวลา, ปรับความเร็วของ moleTimer, และเริ่มรอบใหม่. ถ้าเป็นรอบสุดท้าย, จะเรียก gameOver(); มาใช้งาน

**Mole Timer Tick Event**

* กำหนด ActionListener สำหรับ moleTimer, ทำงานทุก 1.5 second = 1500 millisecond เมื่อ moleTimer ทำงาน, จะเรียก genRandMole เพื่อแสดงมอลในตำแหน่งสุ่มใน GUI.

ภาพรวมของ Event handling ที่มีในหน้าจอ

**Start Button Click Event:** จะทำการปิดการใช้งานปุ่ม "Start", รีเซ็ตค่าเกม, เริ่มการทำงานของ moleTimer และ timer เพื่อเริ่มเกม.

**Timer Tick Event :** ก็มีการจะทำการลดเวลา สุดท้าย เราจะทำการรวมคะแนนในแต่ละรอบมาบวกเข้ากัน แล้วก็ยังมีการลดเวลา, ปรับความเร็วของ moleTimer, และเริ่มรอบใหม่

****

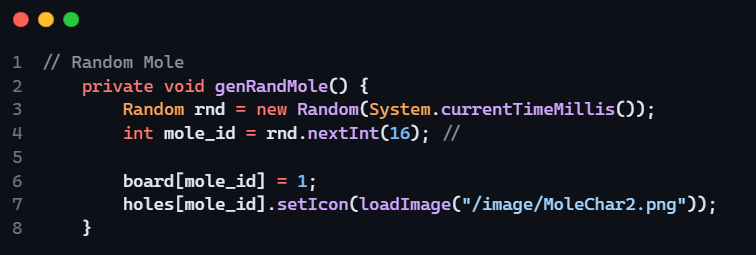
****

**Mole Timer Tick Event** : ทำงานทุก 1.5 second = 1500 millisecond และจะเรียก genRandMole(); เพื่อแสดงมอลในตำแหน่งสุ่มใน GUI

**Mouse Click Event :** เราสามารถ Click ได้ ทุก JLabel ที่เป็น holes (หลุม) ใน array เมื่อเรามีการคลิกที่หลุมเกิดขึ้น

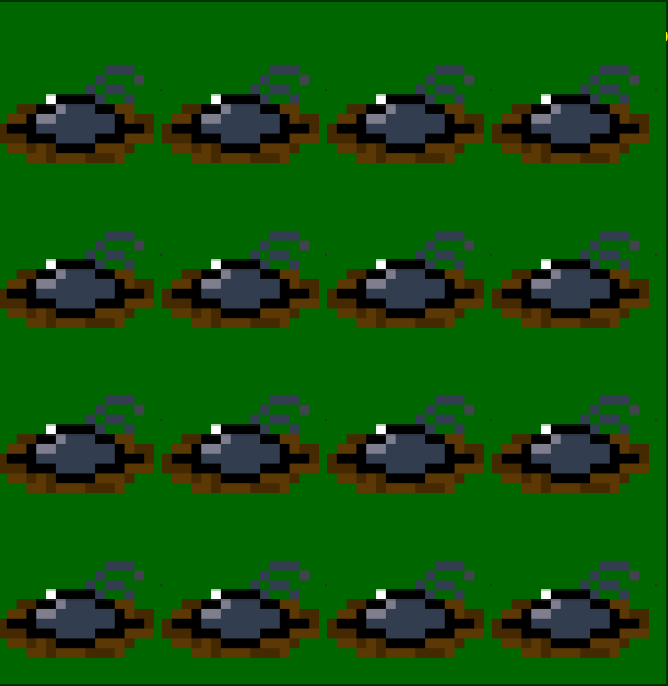
* **อธิบายอัลกอริทึมที่สำคัญในโปรแกรม**

**genRandMole() :** สุ่มตำแหน่งของมอล

****

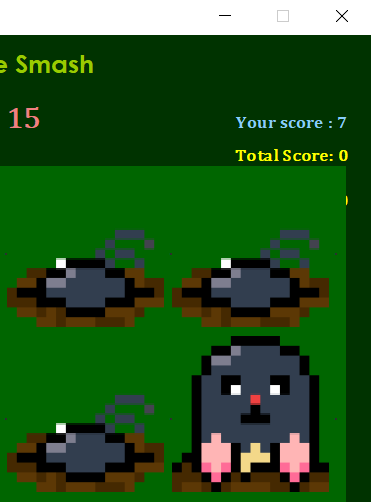
* ฟังก์ชันนี้ใช้สุ่มตำแหน่งของมอลใน GUI และตั้งค่าใน board เพื่อบอกว่ามีมอลอยู่ในตำแหน่งนั้น คือ
  + สร้าง Object แล้วให้ seed เป็นค่าเวลาปัจจุบันเพื่อให้การสุ่มเป็นไปตามเวลาที่เปลี่ยนไป
  + สุ่มตำแหน่งของมอลโดยใช้ nextInt(16) ซึ่งจะได้ค่าตั้งแต่ 0 ถึง 15.
  + กำหนดค่าใน board ที่ตำแหน่งที่สุ่มได้ให้เป็น 1 เพื่อบอกว่ามี และแสดงรูปตัวตุ่น

**clearBoard() :** ล้างสถานะทั้งหมดของบอร์ด

****

* ฟังก์ชันนี้เราใช้การ LOOP FOR เพื่อไปผ่านทุกตำแหน่งใน holes และ board ของ i จาก 0 ถึง 15
  + แล้วเรียกใช้ setIcon() เพื่อกำหนดรูปไอคอนให้กับ JLabel ใน holes โดยใช้รูปตัวตุ่น
  + ค่าใน board จะถูกกำหนดเป็น 0 คือ ไม่มีมอลปรากฎในหลุมที่นั้นในตอนนั้น

**pressedButton(int id)** : ตรวจสอบการกดปุ่ม

****

**Click โดนตัวตุ่นที่โพล่ออกมา จะแสดงว่าได้ 1 คะแนน**

****

****

**Click หลุมตัวตุ่นที่ยังไม่โพล่ออกมา จะเสีย -1 คะแนน**

* ฟังก์ชันการกดปุ่มเพื่อได้คะแนน และมี set ค่าหลุม กับการ random หลุม
  + id ของ board จะถูกเก็บไว้ในตัวแปร val. ค่านี้เป็นสถานะของหลุมที่ถูกกด
  + ค่าที่ได้จาก board[id] เราได้กำหนดเป็น 1 คะแนน เมื่อกดโดนที่ตัวตุ่นตอนโพล่ออกมาจะเพิ่มขึ้น 1 ถ้าไปคลิกที่ไม่มี จะถูกลดลง 1 คะแนนไปเรื่อยๆ
  + คะแนนจะถูกแสดงผลที่ JLabel ที่ชื่อ lblScore โดยใช้เมทอด setText()
  + เรียก clearBoard() เพื่อทำการรีเซ็ตหลุมทั้ง 16 หลุมใหม่
  + เรียก genRandMole() เสร้างตัวตุ่นในหลุมที่สุ่มได้ใหม่

**gameOver():** การจัดการเมื่อเกมจบ

**เมื่อเล่นครบจนจบเกมจะ reset ทุกอย่างและทำอะไรไม่ได้ แล้วจะแสดงคะแนนรวมทั้งหมด**



* ฟังก์ชันนี้จะเกิดขึ้นเมื่อเกมส์จบลงไปแล้วทัน ซึ่งที่จะเกิดขึ้นคือ
  + btnStart.setEnabled(true) ปุ่ม Start สามารถกลับมาใช้งานได้อีกครั้ง
  + คะแนนจาก 3 รอบที่ผ่าน จะถูกเพิ่มเข้า totalScore
  + คะแนนรอบปัจจุบัน จะถูกรีเซ็ตเป็น 0 รวมทั้งเวลาจะกลับไปเป็น 20 วินาที
  + จะมีการตรวจสอบรอบที่เล่นในปัจจุบัน ถ้าน้อยกว่ารอบที่ตั้งไว้จะเล่นไปเรื่อยๆจนครบรอบและในแต่ละรอบใหม่ จะมีการปรับค่า moleTimer ให้เร็วขึ้น 200 millisecond พอเล่นครบรอบทำการรีเซ็ตการเล่นเกมทั้งหมดด้วย clearBoard()

**บทที่ 3 สรุป**

**ปัญที่เกิดขึ้นระหว่างการพัฒนาเว็บไซต์**

1. การดำเนินการ หรือ การจัดสรรเวลาไม่เป็นตามที่วางแผนไว้ จึงอาจจะต้องมีการเปลี่ยนแปลงบางส่วนของตัวเกม
2. ชิ้นงานออกมาไม่เป็นไปตามที่คาดหวังไว้
3. เสียดายที่ทำออกมาไม่ดีเท่าที่ควร

**จุดเด่นของโปรแกรมที่ไม่เหมือนใคร**

* การใช้ Timer เพื่อนับเวลาถอยหลังและจัดการเหตุการณ์ที่เกิดขึ้น และ การใช้ Random เพื่อสุ่มตำแหน่งของมอลทำให้เกมมีความสุ่มแปรปรวน และไม่ซ้ำกันทุกรอบ.

**คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับรุ่นน้องต่อไป**

* ไม่มีคำติให้กับอาจารย์ครับ อาจารย์สอนเข้าใจช่วยนักศึกษาทุกคนที่มีปัญหาหรือติดตรงไหนตลอด แต่บางครั้งตามที่อาจารย์สอนไม่ทันด้วย แต่ผมจะไม่ค่อยเข้าใจโจทย์การคิดแบบอัลกอริทึม
* อาจารย์สอนดี แต่ผมพยายามแล้วครับ ช่วยให้ผมรอดทีครับ