

# Java project แบบข้อเสนอโครงการ

## MrRabbit's Adventure

### สมาชิก

นางสาวสุนันท์ พิงสันเทียะ 6730300639

นางสาวณัฏฐ์อาภา เลิศศุภจิรโชติ 6730300728

โครงการนี้เป็นเกม MrRabbit's Adventure ซึ่งผู้เล่นจะรับบทเป็นกระต่ายให้กระโดดข้ามสิ่งกีดขวาง หลบศัตรู และเก็บเหรียญเพื่อผ่านด่าน

รายละเอียดโดยย่อ

### คุณลักษณะและขอบเขต

#### คุณลักษณะขั้นต่ำ

- ตัวละครหลักเป็นกระต่ายซึ่งสามารถควบคุมให้เคลื่อนที่ได้ 4 ทิศทาง
- มีดินไม้ ซึ่งเป็นอุปสรรคเมื่อชนแล้วจะเสียพลังชีวิต
- เก็บเหรียญให้ครบเพื่อผ่านด่าน
- มีด่านที่สามารถเล่นได้ 2 ด่าน
- มีระบบ Save/Load เพื่อให้บันทึกความคืบหน้าของเกมและให้กลับมาเล่นใหม่ได้

#### คุณลักษณะเพิ่มเติม

- มีด่านให้เล่นเพิ่มเติม
- มีระบบร้านค้าเพื่ออัปเกรดความสามารถของตัวละคร เช่น ความเร็วการเคลื่อนที่
- พื้นหลังมีการขยับในรูปแบบอนิเมชัน

### แผนการดำเนินโครงการ

ระยะ	ช่วงเวลา	ผลลัพธ์ที่คาดหวัง
เริ่มต้นโครงการ	สัปดาห์ที่ 1-2	- ส่งข้อเสนอโครงการ - สร้างGitHub repository - สร้างโครงการด้วย Maven และเขียนโค้ดตั้งต้น
สร้างต้นแบบ	สัปดาห์ที่ 3-4	- โปรแกรม/เกมทำงานได้ในเบื้องต้น- มีฟีเจอร์หลัก (core feature) ที่ใช้งานได้
ทดลองใช้งาน	สัปดาห์ที่ 5-6	- ฟีเจอร์ส่วนใหญ่พร้อมใช้งาน- โปรแกรมสามารถใช้งานได้ แต่อาจต้องปรับแก้หรือเพิ่มเติม
พร้อมส่งมอบงาน	สัปดาห์ที่ 7-8	- โปรแกรม/เกมมีฟีเจอร์ครบถ้วน

		ตามที่วางแผน- เอกสารประกอบ สมบูรณ์ พร้อมสำหรับการนำเสนอ
--	--	--

## การแบ่งงาน



1. นางสาวสุนันท์ พิงสันเทียะ 6730300639
  - จัดทำเอกสาร
  - ทดสอบระบบ
  - เขียนโค้ดส่วนจัดการข้อมูล
2. นางสาวณัฏฐ์อาภา เลิศศุภจิรโชติ 6730300728
  - ออกแบบ UX UI
  - จัดการ Git repository
  - เขียนโค้ดส่วนติดต่อผู้ใช้

## ความท้าทายและความเสี่ยง

1. ความท้าทายด้านเทคนิค
  - เนื่องจากไม่คุ้นเคยกับการใช้งาน javafx ในการทำ GUI
  - แนวทางแก้ไข: ศึกษาบทเรียนและขั้นตอนต่างๆจากลิงก์ Tutorialspoint JavaFx  
<https://docs.oracle.com/en/java/javase/21/>
2. ความท้าทายด้านการพัฒนาเกม FXGL
  - เนื่องจากการพัฒนาระบบการเคลื่อนที่ที่ตอบสนองต่อผู้เล่นทั้ง 4 ทิศทางอย่างลื่นไหลใน FXGL
  - แนวทางแก้ไข: ศึกษาวิธีการใช้ Animation และ Physics ใน FXGL เพื่อให้การเคลื่อนที่ของกระด่ายลื่นไหลและมีการตอบสนองที่ดี  
[FXGL: Improved Physics and Animations - YouTube](#)
3. ความเสี่ยงด้านเวลา
  - เนื่องจากต้องการเรียนรู้การใช้ FXGL และ JavaFX สำหรับการสร้างเกม ซึ่งอาจใช้เวลาในการศึกษาและทดลองพัฒนา
  - แนวทางแก้ไข: วางแผนเวลาในการเรียนรู้เทคโนโลยีใหม่ๆ โดยการแบ่งเวลาศึกษาเทคนิคต่างๆ และพัฒนาโค้ดในส่วนที่ง่ายก่อน ติดตามความคืบหน้าเป็นประจำและปรับแผนพัฒนาให้เหมาะสม

## ต้นแบบและเอกสารอ้างอิง

- Jumping game code : <https://github.com/alexmadd4/JumpNow>
- Jumping and Gravity code tutorial :  
<https://stackoverflow.com/questions/35478428/how-should-i-implement-jumping-gravity-falling-in-this-2d-java-game>
- Save and Load code :  
<https://gist.github.com/EudyContreras/a3ad63067e7e16ed7cf580eb992b4a60>

- Basics Javafx game tutorials :  
<https://jvm-gaming.org/t/getting-started-with-javafx-game-programming-for-java-programmers/56581/2>
- Multiple screen for Javafx :  
<https://stackoverflow.com/questions/37886059/javafx-game-with-multiple-screens>
- Basic Javafx : <https://www.javatpoint.com/first-javafx-application>
- Basic java  Java Full Course for free 

## แนวทางการแบ่งงาน

### 1.การแบ่งงานตามฟีเจอร์ (Feature-Based Division)

แนวทาง: แบ่งงานตามฟีเจอร์ของแอปพลิเคชันหรือกลไกของเกม โดยอาจค่อย ๆ แบ่งและทำทีละฟีเจอร์ตามลำดับความสำคัญ

ข้อดี: ง่ายต่อการแบ่งงาน

ข้อเสีย: การทำงานร่วมกันอาจทำให้เกิดปัญหาบ่อยกว่าวิธีอื่น

### 2.การแบ่งงานตามเลเยอร์ (Layer-Based Division)

แนวทาง: แบ่งงานตามเลเยอร์ของซอฟต์แวร์ (เช่น Front-end และ Back-end)

- สมาชิก A: พัฒนากลไกของเกมหรือการประมวลผลฝั่งเซิร์ฟเวอร์
  - สมาชิก B: สร้าง UI และจัดการการโต้ตอบของผู้ใช้
- ข้อดี: แบ่งส่วนงานชัดเจน สามารถพัฒนาไปพร้อมกันได้ง่าย
- ข้อเสีย: ภาระงานอาจไม่สมดุล

### 3.การจับคู่ทำงานแบบวนซ้ำ (Iterative Pairing)

แนวทาง: สมาชิกทั้งสองทำงานร่วมกันในแต่ละส่วนของโค้ด สลับบทบาทกันระหว่าง ผู้เขียนโค้ด (Driver) และ ผู้ช่วยตรวจสอบ (Navigator)

ข้อดี: ส่งเสริมการเรียนรู้และการทำงานร่วมกัน

ข้อเสีย: อาจทำให้ความเร็วในการพัฒนาลดลง