**JAVA Project**

**แบบข้อเสนอโครงงาน**

ชื่อโครงงาน: [Tournament-Of-5-Hearts](https://github.com/Baconkinh/Tournament-Of-5-Hearts)

**สมาชิก**

1. อัจฉรา ดังดี 6730300655
2. ปภากร จันทร์ดี 6730300809

**รายละเอียดโดยย่อ**

โครงงานนี้เป็นเกม 2 มิติ(2D Adventure Game) จำนวนผู้เล่น 1 คน แนว Open-world ผสม เทิร์นเบส(Turn Based) ซึ่งผู้เล่นจะต้องตั้งชื่อตัวเองและเลือกตัวละครในแต่ละสายอาชีพ ได้แก่ นักธนู นักดาบ แทงค์ เมจ และมือปืน เพื่อนำมาต่อสู้กับมอนสเตอร์ในแมพ ถ้าหากฝ่ายไหนเลือดมีค่า <=0 จะถือว่าแพ้ เมื่อผู้เล่นชนะมอนสเตอร์จะได้รับแต้มมา เพื่อปลดล็อกการสู้กับมอนสเตอร์ตัวถัดไป

**คุณลักษณะและขอบเขต**

* สามารถเลือกตัวละครเล่นได้ 5 ตัวละคร และสามารถเลือกตัวละครซ้ำกันได้ทั้งสองฝ่าย
* มีระบบ HP เมื่อผู้เล่น HP อีกฝ่ายก็จะเป็นผู้ชนะ
* มีระบบ Mana ผู้เล่นสามารถวางแผน Mana เอาไว้เล่นในเทิร์นต่อๆไปได้
* มีระบบ Skill ผู้เล่นสามารถเลือกสกิลได้หลายอย่างในการเล่น
* สามารถตั้งชื่อตัวเองในเกมได้

**คุณลักษณะเพิ่มเติม**

* มีไอเท็มบัฟอื่นๆ

**แผนการดำเนินโครงงาน**

|  |  |  |
| --- | --- | --- |
| **ระยะ** | **ช่วงเวลา** | **ผลที่คาดหวัง** |
| เริ่มต้น | สัปดาห์ที่ 1-2 | * ส่งข้อเสนอโครงงาน * สร้าง GitHub Repository * สร้างโครงงานด้วย Maven และเขียนโค้ดตั้งต้น |
| สร้างทดลอง | สัปดาห์ที่ 3-4 | * เกมทำงานได้ในเบื้องต้น * มีฟีเจอร์หลัก (core feature) ที่ใช้งานได้ |
| ทดลองใช้ | สัปดาห์ที่ 5-6 | * ฟีเจอร์ส่วนใหญ่พร้อมใช้งาน * เกมสามารถเล่นได้ แต่อาจต้องปรับแก้หรือเพิ่มเติม |
| ส่งมอบงาน | สัปดาห์ที่ 7-8 | * เกมมีฟีเจอร์ครบถ้วยตามที่วางแผน * เอกสารประกอบสมบูรณ์ พร้อมสำหรับการนำเสนอ |

**การแบ่งงาน**

1. อัจฉรา ดังดี 6730300655
2. ตกแต่งหน้า UI
3. ทำรูปภาพประกอบเกม
4. ออกแบบสกิลตัวละคร
5. ปภากร จันทร์ดี 6730300809
6. จัดหน้า UI
7. เพิ่มฟังก์ชันต่างๆ
8. ออกแบบรูปแบบการเล่น

**ความท้าทายและความเสี่ยง**

1. **ความท้าทายด้านเทคนิค:**  
   ยังไม่คุ้นเคยกับ JavaFX ในการทำ GUI  
   **แนวทางแก้ไข:** ศึกษาตัวอย่างและบทเรียนจากลิงก์ Pragmatic Coding  
   (<https://www.pragmaticcoding.ca/beginners/intro>)
2. **ความท้าทายด้านการพัฒนาเกมด้วย FXGL:**  
   ต้องเรียนรู้ FXGL เพิ่มเติม เพราะเคยใช้แต่ JavaFX พื้นฐาน  
   **แนวทางแก้ไข:** ทำตาม tutorial ใน FXGL Wiki  
   (<https://github.com/AlmasB/FXGL/wiki/FXGL-11>) เพื่อพัฒนาความเข้าใจ
3. **ความเสี่ยงด้านเวลา:**  
   หากใช้เวลาเรียนรู้ส่วนประกอบใหม่มากเกินไป อาจทำให้พัฒนาโค้ดไม่ทัน  
   **แนวทางแก้ไข:** วางแผนฝึกทักษะควบคู่กับการพัฒนาโครงงาน และติดตามความคืบหน้า (progress) ของตนเองทุกสัปดาห์

**ต้นแบบและเอกสารอ้างอิง**  
FXGL Tutorial: <https://github.com/AlmasB/FXGL/wiki/FXGL-11>  
JavaFX Documentation: <https://openjfx.io/>

GitHub: https://github.com/Monraz/Javatales.git

**แนวทางการแบ่งงาน**

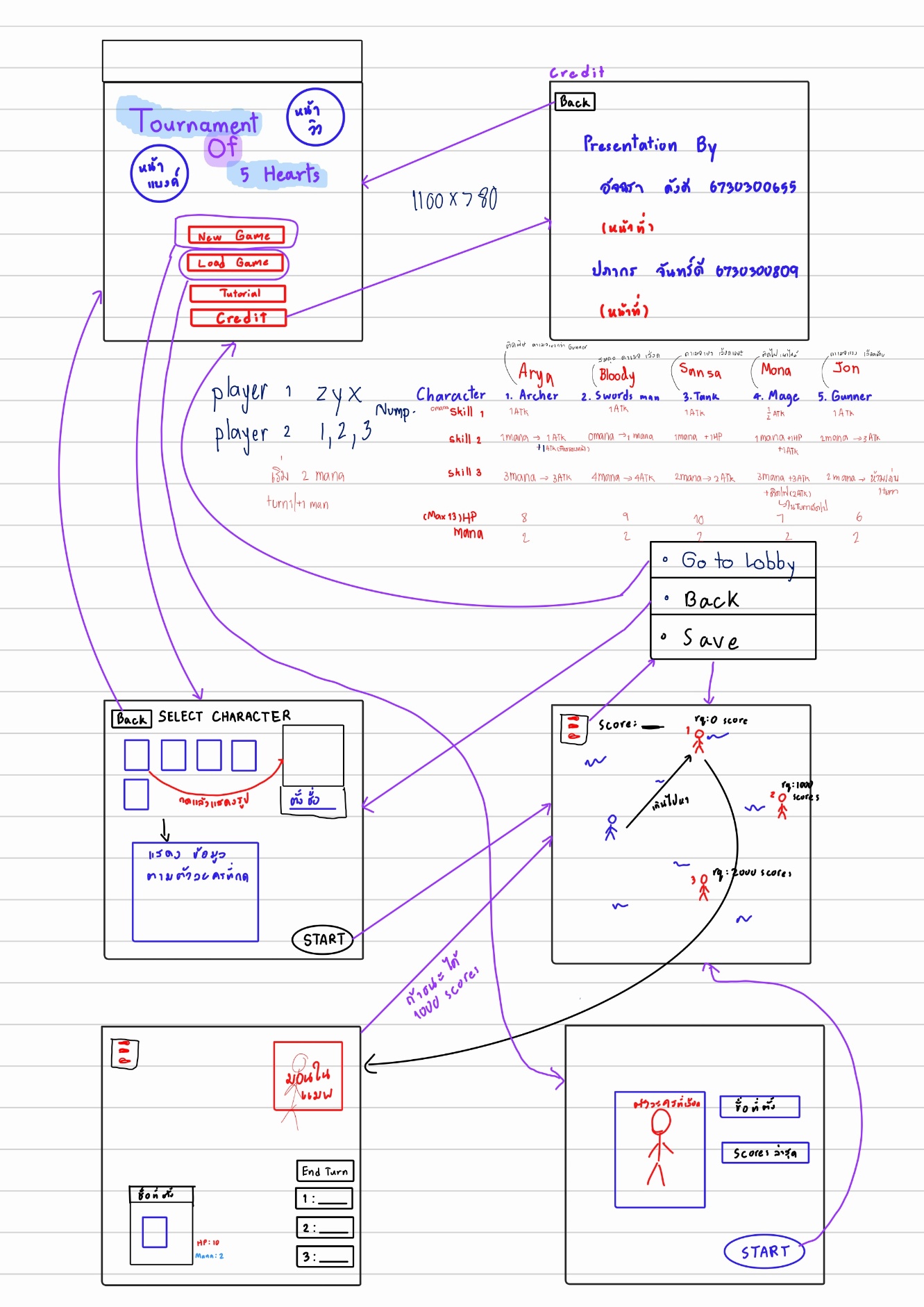
1. **การแบ่งงานตามฟีเจอร์ (Feature-Based Division):**

อัจฉรา ดังดี : พัฒนากลไกสกิลในเกมหรือการประมวลผลด้านหลังของโปรแกรม  
ปภากร จันทร์ดี : สร้าง UI และกลไกความสมดุลของเกม  
  
**แนวทาง:** แบ่งงานตามฟีเจอร์ของแอปพลิเคชันหรือกลไกของเกม โดยอาจจะค่อย ๆ แบ่งและทำไปทีละฟีเจอร์ตามลำดับความสำคัญ  
**ข้อดี:** ง่ายในการแบ่งงาน แต่การรวมงานกันอาจทำให้เกิดปัญหาได้บ่อยกว่าแนวทางอื่น

1. **การแบ่งงานตามเลเยอร์ (Layer-Based Division):**  
   **แนวทาง:** แบ่งงานตามเลเยอร์ของซอฟต์แวร์ (เช่น front-end และ back-end)  
   **ตัวอย่าง:**  
   อัจฉรา ดังดี : ทำภาพประกอบและตกแต่งภาพรวมโดยรอบของเกม  
   ปภากร จันทร์ดี : ทำหน้า Ui ให้สมบูรณ์และปุ่มแต่ละปุ่มประกอบให้สมบูรณ์

**ข้อดี:** การแบ่งส่วนชัดเจน พัฒนาคู่ขนานกันไปได้ง่าย แต่ภาระงานอาจไม่สมดุล

1. **การจับคู่ทำงานแบบวนซ้ำ (Iterative Pairing):**  
   **แนวทาง:** สมาชิกทั้งสองทำงานร่วมกันในแต่ละส่วนของโค้ด สลับบทบาทกันระหว่างผู้เขียนโค้ด (driver) และผู้ช่วยตรวจสอบ (navigator)  
   **ข้อดี:** ส่งเสริมการเรียนรู้และความร่วมมือ แต่มีโอกาสทำให้ความเร็วในการพัฒนาโดยรวมลดลง

 A screenshot of a video game

Description automatically generated A screenshot of a cartoon

Description automatically generated